Hybrid Human-AI Decision Support for Enhanced Human Empowerment in Dynamic Situations

D4.1 Advanced AI Platforms for human-AI collaboration v1

Project Title	**Hybrid Human-AI Decision Support for Enhanced Human Empowerment in Dynamic Situations**
Project Acronym	**HumAIne**
Project Number	**101120218**
Deliverable Identifier	**D4.1 Advanced AI Platforms for human-AI collaboration v1**
Deliverable Due Date	**31/12/2024**
Deliverable Submission Date	**23/12/2024**
Deliverable Version	**v.2.0**
Author (s) and Organization	**Klemen Kenda, Rok Klančič** *(JSI)*, **Katharina Hengel** *(DFKI),* **Spyros Theodoropoulos** *(UPRC),* **Titos Georgoulakis** (UBI), **Michele Canteri** (GFT)
Work Package	**WP4 Advanced Learning Paradigms for Human-AI Collaboration**
Task	**T4.1 – T4.2 – T4.3 – T4.4 – T4.5**
Dissemination Level	**Public** *(PU)*
Type	**Report**

Funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Document Control Page

Deliverable Number	**D4.1**
Deliverable Title	**Advanced AI Platforms for human-AI collaboration v1**
Deliverable Version	**v.2.0**
Work Package Number	**WP4**
Work Package Title	**Advanced Learning Paradigms for Human-AI Collaboration**
Submission Date	**23/12/2024**
Lead Beneficiary	**JSI**
Dissemination Level	**Public**
Status	**Release**
Author	**Klemen Kenda, Rok Klančič** *(JSI)*, **Katharina Hengel** *(DFKI)*, **Spyros Theodoropoulos** *(UPRC)*, **Titos Georgoulakis** (UBI), **Michele Canteri** (GFT)
Contributors	**Mateja Škraba** *(JSI)*, **Fotis Tlantzis** *(NOVO)*, **Segev Shlomov** *(IBM)*, **Emiel Miedema** (RuG), **George Fatouros** (INNOV)
Peer Reviewer (s)	**Kostas Mylonas** *(UBI)* | **Matteo Frattini** *(GFT)*
Approved by	**Fabrizio Di Peppo** *(Project Coordinator)*
Granting Authority	**European Commission**
Type of Action	**HORIZON Research and Innovation Actions**
Topic	**HORIZON-CL4-2022-HUMAN-02-01**
Rights	**HumAIne consortium**

Disclaimer

©Copyright Message

Revision History

Version	Date	Edited by	Description
v.0.1	17/09/2024	Mateja Škraba, Klemen Kenda *(JSI)*	**Initial draft with preliminary TOC**
v.0.2	15/11/2024	Rok Klančič *(JSI)*, Katharina Hengel *(DFKI),* Spyros Theodoropoulos *(UPRC)*	**Initial input from learning paradigms.**
v.0.3	01/12/2024	Klemen Kenda *(JSI)*, Katharina Hengel *(DFKI),* Spyros Theodoropoulos *(UPRC)*	**Refinement of the learning paradigms.**
v.0.4	09/12/2024	Fotis Tlantzis *(NOVO)*, Segev Shlomov *(IBM),* Titos Georgoulakis (UBI), Michele Canteri (GFT), Emiel Miedema (RuG)	**Input from other partners regarding learning paradigms, input for T4.1 an T4.5.**
v0.5	09/12/2024	Klemen Kenda *(JSI)*	**Writing introduction, conclusion, summary, missing parts of active learning, reformatting.**
v0.6	11/12/2024	Katharina Hengel *(DFKI),* Titos Georgoulakis *(UBI)*, Mateja Škraba, Klemen Kenda *(JSI)*	**Additional input for Swarm Learning. Additional input for Secure data collection and management. Reformatting.**
v0.7	11/12/2024	George Fatouros *(INNOV)*	**Additional input for Swarm Learning.**
v0.8	15/12/2024	Titos Georgoulakis *(UBI)*	**SOTA, Swarm Learning Metadata.**
v0.9	15/12/2024	Mateja Škraba, Klemen Kenda *(JSI)*	**Reformatting.**
v.1.0	20/12/2024	Klemen Kenda *(JSI)*	**Considering internal review comments.**
v.2.0	23/12/2024	Maurizio Megliola (GFT)	**Final Version ready for submission**

Funded by the European Union

List of Abbreviations and Acronyms

Abbreviation	Meanining
AL	Active Learning
DL	Deep Learning
DNN	Deep Neural Network
EC	European Commission
EU	European Union
GA	Grant Agreement
HAIRS	Human-AI Resolution System
HumAIne	Hybrid Human-AI Decision Support for Enhanced Human Empowerment in Dynamic Situations
NSAI	Neuro-Symbolic AI
SL	Swarm Learning
WP	Work Package

Funded by the European Union

HumAIne consortium

The HumAIne consortium consists of the following organizations:

#	Role	Short name	Participant organisation name	Country
1	COO	GFT	GFT ITALIA SRL	IT
2	BEN	IBM	IBM ISRAEL - SCIENCE AND TECHNOLOGY LTD	IL
3	BEN	Ebit	EBIT S.R.L.	IT
4	BEN	UPRC	UNIVERSITY OF PIRAEUS RESEARCH CENTER	EL
5	BEN	JSI	INSTITUT JOZEF STEFAN	SI
5.1	AE	QLE	QLECTOR, RAZVOJ CELOVITIH RESITEV ZA PAMETNE TOVARNE DOO	SI
6	BEN	DFKI	DEUTSCHES FORSCHUNGSZENTRUM FUR KUNSTLICHE INTELLIGENZ GMBH	DE
7	BEN	RUG	RIJKSUNIVERSITEIT GRONINGEN	NL
8	BEN	HUA	CHAROKOPEIO PANEPISTIMIO	EL
9	BEN	OKYS	OKYS LTD	BG
10	BEN	LXS	LEANXCALE SL	ES
11	BEN	UBI	UBITECH ENERGY	BE
12	BEN	ISP	INNOVATION SPRINT	BE
13	BEN	INNOV	INNOV-ACTS LIMITED	CY
14	BEN	UNP	UNPARALLEL INNOVATION LDA	PT
15	BEN	IML	IMMERSIVELIVES, LDA	PT
16	BEN	INSO	INNEUROPE INITIATIVE S.L.	ES
17	AP	NOVO	NOVOVILLE LIMITED	UK

Funded by the European Union

Table of Contents

Funded by the European Union

Funded by the European Union

Funded by the European Union

Executive Summary

This document details the development of AI platforms for human-AI collaboration within the HumAIne project, funded by the European Union. It focuses on three core platforms: Active Learning, Swarm Learning, and Neuro-Symbolic AI, explaining their methodologies and showcasing initial development and implementations across various use cases (Smart Cities, Smart Healthcare, Smart Finance).

The initial version of the Active Learning platform has been implemented and tested in multiple pilot programs, achieving an acceleration of knowledge acquisition of over 130% in the Smart Finance scenario. An initial stand-alone Swarm Learning platform has been developed and tested in the Manufacturing pilot, achieving a 10% increase in accuracy compared to decentralized training without Swarm Learning. The Swarm Learning component also demonstrated superior scalability, reducing training time by over 25% compared to the centralized training algorithm. The Neuro-Symbolic AI platform, focusing on diabetes care, is in the early stages of development, leveraging synthetic patient data and expert knowledge to provide AI-generated advice for diabetes management.

The document also covers secure data management and platform integration, emphasizing the project's commitment to trust, transparency, and explainability in AI. Finally, it presents progress towards integrating these platforms into a unified system, namely the HumAIne platform.

Funded by the European Union

1 Introduction

This document is a key output of WP4 within the HumAIne project. The purpose of this document is to detail the development and implementation of the core AI platforms that enable human-AI collaboration for enhanced decision-making in dynamic environments.

D4.1 focuses on three specific AI platforms: Active Learning, Swarm Learning, and Neuro-Symbolic AI. These platforms represent a significant advancement over the current state-of-the-art, which often struggles to combine multiple AI paradigms within integrated applications and lacks sufficient integration with trust-enhancing technologies like Explainable AI (XAI). The document will showcase the development and initial software implementations of these platforms, including the models developed and the data they utilize. It will also partially address the adaptation of these platforms to different use case requirements, drawing on the work done in tasks T4.1 through T4.5.

1.1 The HumAIne Learning Paradigms

The three learning paradigms depicted in Figure 1. focus on enhancing human-AI collaboration for optimized outcomes. Active Learning aims to create platforms where AI systems can actively select the most informative data points, optimizing models and improving human-AI communication. Swarm Learning emphasizes research and development to enable collaboration between humans and AI systems, allowing for collective decision-making that supports optimized decision processes. Neuro-Symbolic Learning combines machine learning and deep learning techniques with symbolic knowledge and reasoning, creating AI platforms that integrate data-driven insights with logical reasoning to support human-centered systems.

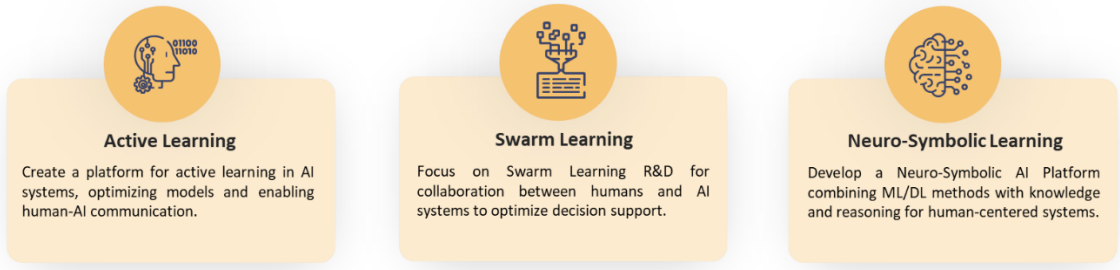

Figure 1. *The three HumAIne learning paradigms.*

These paradigms have been tested across six pilot cases as shown in Table 1. . Active Learning has been applied in Smart Cities, Health (Oncology), Finance, and Energy, demonstrating its flexibility in diverse domains. Swarm Learning has been used in Health (Oncology) and Manufacturing to enhance decision support through collaborative learning. Neuro-Symbolic Learning has been applied in the Health (Diabetes) sector, leveraging a combination of knowledge and reasoning to improve outcomes.

Table 1. *Distribution of pilots across learning paradigms in M15 of the project.*

Pilot	Smart Cities	Health (onc.)	Health (diab.)	Finance	Manufact.	Energy
Active Learning	✓	✓		✓		✓
Swarm Learning		✓			✓	
NS Learning			✓			

Funded by the European Union

The learning paradigms are modular and can be used beyond the selected pilot cases, offering versatility for various applications. Initial prototypes have already been developed, and early use cases have shown promising results. By promoting human-AI collaboration, these platforms can achieve better results than non-hybrid approaches, combining human insight with AI capabilities for superior performance.

Figure 2. depicts human in the center of AI development and deployment process, which is typical for HumAIne. Each learning paradigm receives the data from the HumAIne storage as well as feedback from the user, either in the form of knowledge graphs for neuro-symbolic learning, workflow graphs to facilitate active learning or swarm learning setup, or labeling feedback for efficient training of the models in active learning. Final models, workflows and corresponding metadata, ensuring traceability, are stored in the HumAIne platform.

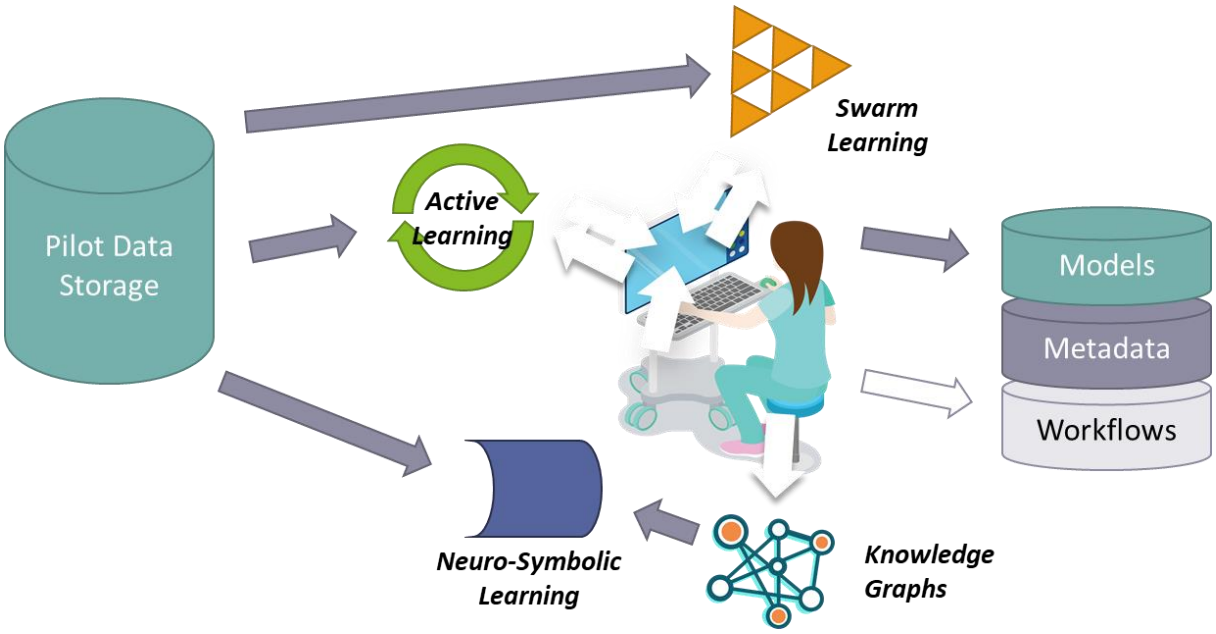

Figure 2. *Human in the centre of AI development and deployment.*

1.2 Generalization of Learning Paradigm Platforms

The generalization of the learning paradigms—**active learning**, **swarm learning**, and **neuro-symbolic learning**—focuses on ensuring that these approaches can be applied flexibly across diverse use cases and platforms. While the **HumAIne pilots** are used as test environments for developing these components, the goal is to avoid limiting development to these pilots alone. Instead, each paradigm is designed to function as an independent module, adaptable to various domains through **customization** using **knowledge graphs** and **Node-RED workflows**. This modular and pilot-agnostic design ensures that the components can serve a broader range of applications, enhancing the robustness and utility of these learning paradigms.

> **Examples:** ***Active learning*** *demonstrates versatility by supporting diverse tasks such as* ***named entity recognition (NER)*** *in medical texts,* ***ticketing system automation****, and* ***image classification*** *for medical diagnostics or quality inspection in manufacturing. Its adaptability allows for generalization to fields like* ***smart cities****,* ***news classification****,* ***legal tech****, and* ***predictive modeling*** *in sectors such as energy, water, or physics. Modular customization through* ***knowledge graphs*** *and* ***workflows*** *ensures it remains domain-agnostic.* ***Neuro-symbolic learning*** *combines neural networks with symbolic reasoning to create interpretable*

Funded by the European Union

*models. It can enhance tasks like **traffic prediction**, **environmental modeling**, or **financial forecasting** by incorporating physical or regulatory constraints, making it suitable for any scenario requiring hybrid reasoning. **Swarm learning** offers decentralized, collaborative capabilities for tasks like **cybersecurity threat detection**, **predictive maintenance** in IoT, and **collaborative robotics**. This approach enables local learning while benefiting from collective intelligence, enhancing adaptability across various distributed systems.*

The end result of these efforts will be the development of **independent paradigm platforms** for active learning, swarm learning, and neuro-symbolic learning. These platforms will be versatile, and able to be integrated seamlessly with different pilots by leveraging customization options through knowledge graphs and Node-RED workflows. The flexibility built into these components ensures that they can address a wide variety of real-world problems, extending beyond the initial HumAIne pilots. This approach, driven by the work in WP5, ensures that the learning paradigms remain adaptable, scalable, and ready for integration into emerging domains and technologies.

1.3 Intended Readership

The intended readership for Deliverable D4.1 encompasses several key stakeholders within the HumAIne project. **WP5 partners** will benefit from D4.1 for integration purposes, ensuring seamless incorporation of the core AI platforms into the broader project infrastructure. **WP2 and WP6 partners**, as well as **pilot leaders**, are crucial readers who will leverage D4.1 to guide the evolution of WP4, aligning platform development with real-world pilot implementations and specific needs identified through pilot activities. Additionally, **AI/ML experts** might benefit, as D4.1 provides insights into the project's advancements and innovations in human-AI learning paradigms.

1.4 Relation with other HumAIne deliverables

Deliverable D4.1 in the HumAIne project is closely related to several other deliverables and tasks that provide essential tools, frameworks, and specifications for its development. Notably, ***D3.1 – XAI Methods and Benchmark Suite for Human-AI Systems v1*** (due in month 15, led by UPRC) offers software implementations for Glass-box and Black-box model interpretability, along with a Benchmark Suite, which D4.1 is likely to utilize for data analysis and platform evaluation. Similarly, ***D3.3 – Next Level Human Machine Interfaces v1*** (due in month 15, led by IML) delivers Human Machine Interface implementations and Knowledge Exchange Techniques that inform D4.1's platform analysis. Furthermore, D4.1 is significantly influenced by tasks in ***WP2: Vision and Specifications for Human-AI Collaboration***. Specifically, ***T2.1 – Vision, Driving Requirements and Reference Scenarios*** defines functional and non-functional requirements for use cases, ensuring D4.1's platforms align with these needs. ***T2.2 – Task and Workflows Modelling*** outlines data management, orchestration, and communication workflows, which D4.1's implementations must integrate. Additionally, ***T2.5 – OS Reference Architecture and Technical Specifications*** provides a reference architecture for HumAIne OS, guiding the platform design for D4.1.

1.5 Structure of the Deliverable

Deliverable D4.1 is structured to provide a comprehensive analysis and development of core AI platforms within the HumAIne project. Following the *Executive Summary* and *Introduction*, which outline the learning paradigms, purpose, scope, intended readership, and interrelations with other deliverables, the document dives into key areas of AI development.

Funded by the European Union

Chapter 2 focuses on *Active Learning*, covering methodologies, related work, experimental results across various use cases (SmartCities, SmartHealthcare, Smart Energy, and SmartFinance), early prototypes, and integration plans. Chapter 3 addresses *Swarm Learning*, detailing use cases, client integration, and synchronization, alongside experimental results. Chapter 4 explores *Neuro-Symbolic AI*, including system architecture, expert knowledge, and initial findings.

Chapter 5 discusses *Secure Data Collection and Management*, highlighting state-of-the-art solutions, components explanation, and integration mechanisms. Chapter 6 covers *Platform Customization and Bootstrapping*, analyzing platform requirements and configurations. The deliverable concludes with a summary of findings in Chapter 7.

Funded by the European Union

2 Active Learning

This section reports on the work accomplished within **HumAIne task T4.2 - Active Learning Models and Platform Development**.

2.1 Introduction

Even when dealing with the active learning approach within the HumAIne project consortium we have encountered several pre-conceptions of **what** active learning approach represents. Mostly, active learning is understood as a general human-in-the-loop approach in machine learning. While this is close to the truth, active learning is mostly applied during the training phase of the data mining cycle and not in the inference phase. Therefore, the following definition is essential to bring all the stakeholders in agreement.

> **Definition of Active Learning in HumAIne:** Active learning is a machine learning paradigm where **an algorithm is able to choose the data it learns from**, rather than relying on passively observing all available labelled data. In active learning, **the algorithm interacts with a user** or an oracle **to selectively query and obtain labels for instances that it finds most informative** or uncertain. **The goal of active learning is to reduce the labelling cost** by focusing on the most useful data points, improving model performance with fewer labelled examples compared to traditional supervised learning approaches.

In the inference phase, active learning principles can be used for involving human feedback. For example, when the algorithm is uncertain about the correct label of the data instance, it can give feedback to the user/oracle in order to assign the correct label. These labels can furthermore be used for additional model training.

After answering the what question, it is time to address the **why**! The project philosophy of HumAIne revolves around leveraging cutting-edge technology innovations and groundbreaking scientific discoveries to facilitate human-AI collaboration. The emphasis is on developing reliable AI systems based on HumAIne learning paradigms that prioritize trust and explainability. HumAIne aims to lead in the development of **trustworthy AI systems that support human-AI cooperation**, paving the way for innovative applications across various sectors and fostering a digital environment grounded in trust and transparency.

The remaining of this section is structured as follows. The subsections of the introduction describe related work and the scientific contribution of HumAIne in the field of active learning, evaluate project's KPIs, explore relation of HumAIne T4.2 dedicated to active learning with other HumAIne work packages and tasks and finally present results on the active learning research presented in previous HumAIne deliverables. The next sections provide insights into methodology, experiments and early prototypes. We conclude the active learning section by outlining the future work within the HumAIne project.

2.2 Current Progress

The implementation of the Active Learning paradigm within the HumAIne project shows varied progress across different pilot cases. In Smart Finance, Active Learning has advanced significantly, with scenarios defined, data available, and a Proof of Concept (PoC) successfully developed. For Smart Cities, scenarios are defined, and synthetic datasets are available, which indicates progress toward PoC development. The Smart Energy pilot has defined scenarios but lacks available models (to be learned), a PoC, or further development stages such as standalone or integrated implementations. In Smart Healthcare (oncology), scenarios have been outlined, but data availability is hindered by privacy issues that are currently being resolved. As a result, no PoC or further implementation phases have been achieved so far. Several experiments have been done, however, in the field of named entity recognition (using active learning) in medical domain.

Funded by the European Union

Overall, the Active Learning paradigm has made solid progress in the initial phase of the project, defining scenarios for all the involved pilots. Furthermore, Active Learning has been included in Smart Finance use case, which was not envisioned by the project proposal.

Table 2. *Current progress of the implementation of Active Learning (M15).*

Pilot	Scenarios	Data Available	PoC	Standalone	Integrated
Smart **Finance**	✓	✓	✓		
Smart **Cities**	✓	✓ Synthetic dataset	✓		
Smart **Energy**	✓				
Smart **Healthcare**	✓	Privacy issues (being resolved)			

2.3 Related Work

This section covers key techniques and tools in Active Learning (AL) and natural language processing (NLP) to improve efficiency and reduce computational costs.

The first part is dedicated to NLP, which is the basic machine learning approach in all the currently addressed pilots and is used for creating named entity recognition models or ticket clustering algorithms. DistilBERT addresses the challenge of using large language models under budget constraints by employing knowledge distillation, reducing BERT's size by 40% while retaining 97% of its performance. This leads to faster inference and lower operational costs. Sentence Transformers, based on Sentence-BERT (SBERT), enhance sentence similarity tasks, reducing computation time from 65 hours to 5 seconds. BERTopic uncovers latent topics by clustering document embeddings and using a class-based TF-IDF approach for interpretability. These tools enable efficient NLP applications without sacrificing accuracy or performance. This was tested and used in Smart Finance pilot.

The section also studies Active Learning, which minimizes labeling costs by selecting the most informative data points for model improvement. The section thoroughly describes various selection strategies. Challenges in applying Active Learning to deep learning (including language models and large language models) arise due to neural networks' overconfident predictions and computational demands. We analyze various active learning libraries.

2.3.1 DistilBERT

Using large models in natural language processing (NLP) can be challenging when operating under a constrained budget. To address this, smaller general-purpose language representation models like DistilBERT (Sanh, 2019) were developed. These models maintain most of the performance while being more resource-efficient, making them a practical choice for budget-conscious applications.

Through a process called knowledge distillation, BERT's (Devlin, 2018) size is reduced by 40%, while still retaining 97% of its language understanding capabilities. This reduction makes DistilBERT approximately 60% faster than the original BERT, enabling quicker inference and reducing operational costs (Sanh, 2019).

Funded by the European Union

2.3.1.1 Sentence Transformers

The Sentence Transformers library is a popular tool for creating sentence embeddings and is built on Sentence-BERT (SBERT) (Reimers, 2019), a modified version of BERT. SBERT uses siamese and triplet network architectures to produce high-quality sentence embeddings that can be easily compared using cosine similarity.

By using this structure, Sentence Transformers dramatically reduces the time required to find the most similar sentence pairs, from 65 hours with BERT down to about 5 seconds, while preserving BERT's accuracy. This makes it an ideal solution for efficient similarity tasks in natural language processing (Reimers, 2019).

2.3.2 BERTopic

It is a tool designed to uncover latent topics within a collection of documents, approaching topic modeling as a clustering task. It leverages pre-trained language models to generate document embeddings, which are then clustered to identify potential topics within the data.

To represent these topics effectively, BERTopic (Grootendorst, 2022) uses a class-based variation of TF-IDF, which enhances the interpretability of each cluster by highlighting words most indicative of the underlying themes. This combination of embedding-based clustering and class-based TF-IDF enables BERTopic to provide insightful and coherent topic representations from large text datasets.

2.3.3 Approaches for Query Strategies

The main objective in active learning is to design strategies that select the most informative data points for querying. By choosing instances strategically, we aim to separate classes with the fewest examples possible, maximizing model performance while minimizing labeling costs.

Query strategies are generally divided into three categories:

- **Heterogeneity-Based Models**: These models focus on selecting instances from regions of the data space where the model is most uncertain. Examples include uncertainty sampling, where the learner queries instances closest to the decision boundary, and query-by-committee, where multiple models "vote" on instances, selecting those with the most disagreement. This approach helps define the decision boundaries between classes more accurately.

- **Performance-Based Models**: These strategies aim to directly improve the model's performance, such as by reducing classification error or variance. For instance, expected error reduction models prioritize instances that will reduce prediction error on the remaining unlabeled data, while variance reduction models choose instances expected to lower the model's output variance. These methods focus on optimizing the model's generalization by targeting performance metrics.

- **Representativeness-Based Models**: These models prioritize data points that best represent the underlying distribution of the data, often using density-based measures. By selecting instances in dense regions, these models ensure that the queried instances are not outliers and provide a broader picture of the data distribution.

Each strategy offers different trade-offs based on the goals of the model and the nature of the data. Heterogeneity-based models tend to focus on refining the decision boundary, performance-based models aim to improve specific model metrics, and representativeness-based models provide comprehensive data representation (Aggarwa, 2014).

Funded by the European Union

Applying Active Learning to deep learning is challenging due to the overconfidence of neural networks' soft-max outputs, making uncertainty-based selection difficult. Additionally, the high computational cost of training deep neural networks means selecting batches of instances, rather than individual ones, is more efficient. Research focuses on developing strategies to address these challenges and efficiently choose or batch instances for querying in deep learning settings (Gildenblat, 2020).

2.3.4 Comparison of Active Learning Querying Strategies

Active learning employs three main groups of querying strategies to select the most informative data points: heterogeneity-based, performance-based, and representativeness-based models, as mentioned in the previous subsection. Each approach offers unique advantages for improving model performance with minimal labeling costs. Below, concrete strategies belonging to these three groups are analysed.

Uncertainty Sampling

Uncertainty sampling is an active learning strategy that focuses on selecting instances for which the model is least confident in its predictions. These are usually the samples most likely to enhance the model's performance.

In the simplest case, with binary classification, a Bayesian classifier can be used to make predictions, and the instance with a best label probability closest to 0.5 is selected. For a balanced comparison, probabilities are normalized so that they sum up to 1. The criterion for selection is often the difference between the predicted probabilities of the two classes—the smaller the difference, the higher the uncertainty.

When working with multi-class classification, different criteria, such as maximizing entropy or the Gini index, are used to identify instances with the highest uncertainty. Higher entropy or Gini index values reflect a more even distribution across classes, suggesting greater uncertainty about the instance's true label.

However, one limitation of uncertainty sampling is that it can perform poorly on imbalanced datasets, as the strategy may over-sample instances from less frequent classes. Despite this, numerous techniques have been developed to address various challenges within uncertainty sampling, making it a flexible and widely used approach in active learning (Aggarwa, 2014).

Query by Committee

In the Query by Committee (QBC) active learning strategy, the selection of samples is driven by heterogeneity among classifiers. This strategy relies on a group, or "committee," of different classifiers that are trained on a partially labeled dataset. The core idea is to leverage disagreement among the classifiers to identify the most informative samples for labeling.

Each classifier in the committee is used to predict the labels of instances. The instance selected for labeling is the one with the highest disagreement across the classifiers, meaning that it has a high likelihood of being classified differently by various members of the committee. This approach is similar to uncertainty sampling, as it also seeks out instances for which the classification confidence is low. However, QBC adds an additional layer by comparing predictions across multiple models instead of relying on a single model's confidence score.

Disagreement can be quantified using various metrics, such as entropy or the Gini index, to capture the diversity in predictions. Additionally, measures like Kullback-Leibler (KL) divergence may also be applied to assess the variability in class probabilities across the models.

There are two common approaches for forming the committee: one can either use a single model architecture with different parameter settings or combine a group of diverse models. Both methods

Funded by the European Union

encourage a broader range of opinions in the committee, making the QBC approach an effective strategy for selecting the most informative samples in active learning tasks (Aggarwa, 2014).

Expected Model Change

The expected model change strategy in active learning focuses on selecting instances that are anticipated to have the most significant impact on updating the model. This approach aligns with the overarching goal of active learning, which is to prioritize instances that will drive the greatest improvement in model performance.

Specifically, the strategy identifies instances expected to cause the largest gradient change when added to the training set. By selecting these instances, the model is effectively "pushed" in directions that enhance its accuracy and adaptability. This makes it a heterogeneity-based strategy, as it seeks instances that are most distinct from the model's current understanding.

The expected model change strategy is particularly well-suited to models that rely on gradient-based training methods, such as neural networks, where gradient calculations guide the optimization process. This strategy leverages these gradients to anticipate which instances will have the most influence on model parameters, thereby accelerating learning and improving model robustness (Aggarwa, 2014).

Expected Error Reduction

The expected error reduction strategy is a performance-based approach in active learning. In this strategy, expected error reduction is calculated on the unlabeled portion of the data. The aim is to select instances for querying that will minimize the expected label uncertainty across the remaining unlabeled dataset. By doing so, we attempt to ensure that the predicted labels for the unlabeled data are as far from 0.5 as possible, indicating high confidence in the label predictions. This strategy operates on the assumption that a greater certainty in the labels of the unlabeled data correlates with a lower error rate, ultimately enhancing the model's performance (Aggarwa, 2014).

Clustering Uncertainty-weighted Embeddings (CLUE)

The CLUE strategy falls under the category of representativeness-based models in active learning. This approach employs uncertainty-weighted clustering to identify the most informative and representative instances for labeling. It selects instances that are both uncertain under the current model and diverse within the feature space, aiming to capture a broad range of the data distribution.

By focusing on dense regions of the feature space, CLUE avoids selecting outliers, which may not represent the general patterns within the data. The strategy clusters deep embeddings of the target instances, where each instance is weighted by its associated uncertainty under the model. From the resulting clusters, the centroid of the most relevant cluster is then chosen as the instance for labeling, ensuring that the selected sample is both informative and representative of the data distribution (Prabhu, Chandrasekaran, Saenko, & Hoffman, 2021).

2.4 KPIs

The following KPIs are identified in DOA to assess the work on HumAIne task T4.2 regarding active learning:

KPI 3.1 Active Learning Platform to be implemented

- Goal M18: Initial version implemented (**reached**)
- Goal M36: Final version integrated with HumAIne OS

Initial version of active learning platform is implemented as described in the next sections of this deliverable. Experiments have been conducted in several pilots.

Funded by the European Union

KPI 3.2 Average rate of knowledge acquisition

- Goal M36: Acceleration >= 50% compared to systems without human intervention (**reached in the Smart Finance scenario; >=130%**)

Acceleration of knowledge acquisition (example from the Smart Finance pilot is depicted in Figure 3.) is the improvement in learning efficiency, where fewer labelled examples are needed to reach a desired level of performance. The key is that active learning algorithms strategically select data points that yield the highest information gain, thus accelerating the learning process. The acceleration can be expressed as the ratio of performance improvement in active learning over passive learning, typically with respect to a measure of accuracy or loss reduction. We can define acceleration of knowledge acquisition as:

$$Acceleration\ (p) = \frac{n_{\text{passive}}(p)}{n_{\text{active}}(p)},$$

where $n_{\text{active}}(p)$ is the number of annotated instances using active learning after reaching accuracy p of the model and $n_{\text{passive}}(p)$ is the number of annotated instances with passive learning (i. e., random sampling) after reaching the same model accuracy.

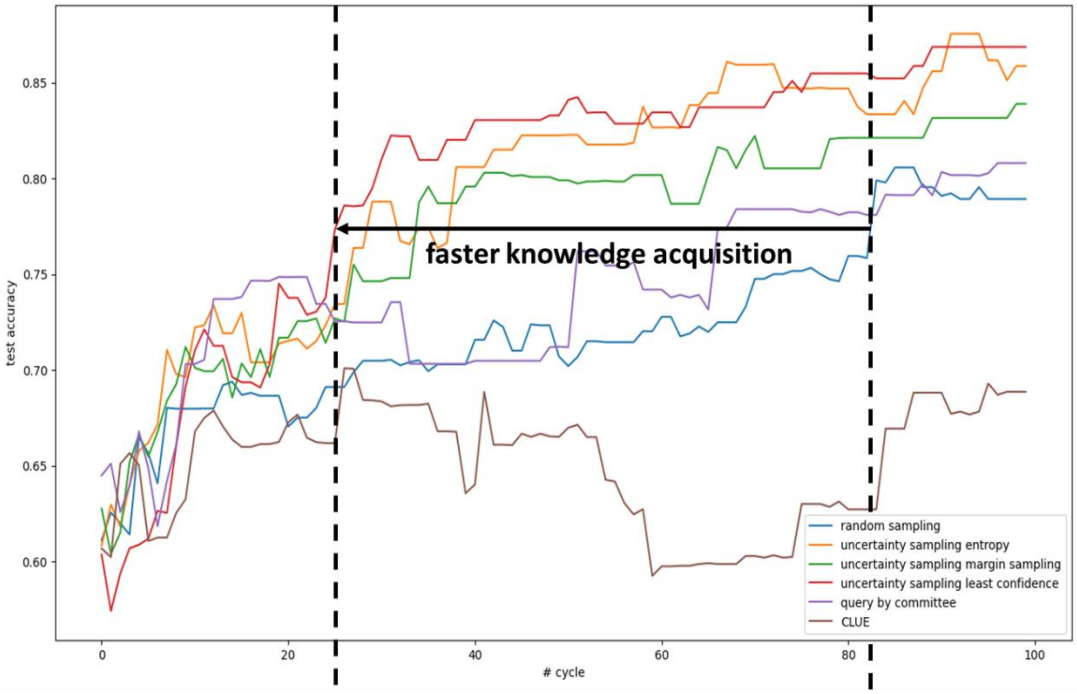

Figure 3. *Figure depicts improvement of a classification model based on the selection of instances for labelling. With random sampling, more than 80 iterations were needed to achieve model accuracy of 0.76, while with uncertainty sampling (using least confidence), only 25 iterations were needed. The difference represents knowledge acquisition acceleration.*

2.5 Methodology

2.5.1 General Active Learning Methodology

In active learning, the process begins by randomly splitting the initial unlabeled dataset into training and test sets. The test set is fully labelled (in order to test the accuracy of the active learning model), while the training set remains unlabeled to allow for iterative learning.

This unlabeled training data is then fed into an active learning loop, which iteratively selects the most informative instance (or group of instances) based on a chosen strategy—those instances expected to have the greatest impact on improving the model's performance. Once selected, these instances are

Funded by the European Union

sent to an oracle (e.g., a human expert) to obtain their labels, and the model is then trained on this partially labeled training set.

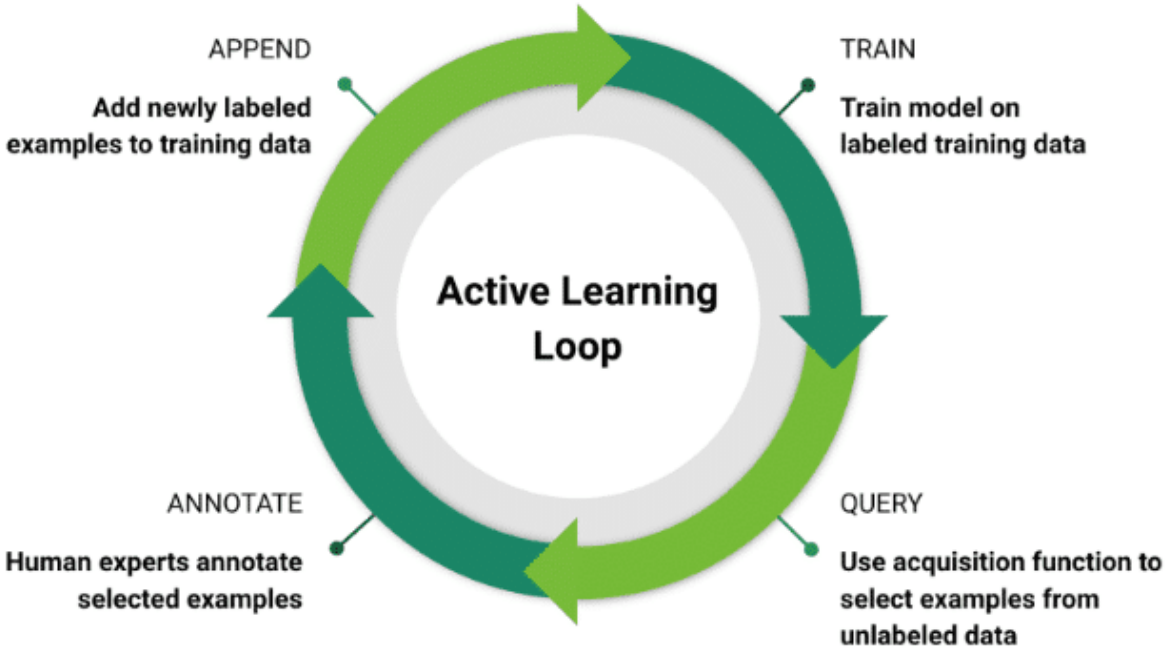

Figure 4. *Active learning loop*[1].

After each iteration, the model's performance is evaluated on the held-out labeled test set. If the accuracy meets a satisfactory threshold, the training loop ends; if not, the process continues with further querying. This cycle repeats until the desired model performance is achieved, resulting in a trained model that has been developed efficiently through targeted instance selection (Aggarwa, 2014).

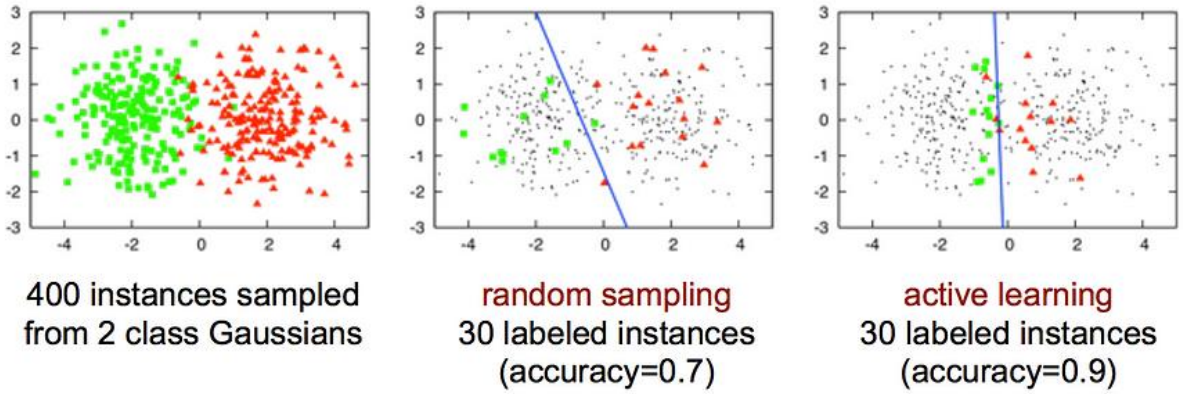

Figure 5. *Illustration of efficiency of active learning against random sampling*[2].

Figure 5. depicts an example of how active learning is more efficient than random sampling. Active learning selects the most informative instances to be labelled in order to fit the model as good to the data as possible with limited labelling effort. Results of active learning are usually depicted in active

[1] https://blogs.nvidia.com/blog/what-is-active-learning/
[2] https://towardsdatascience.com/introduction-to-active-learning-117e0740d7cc

Funded by the European Union

learning performance charts as depicted in Figure 6. Graph represents amount of annotated data on the *x* axis and accuracy of the trained model on the *y* axis. Random sampling models learn steady; however, they need more data in order to achieve the same accuracy. Active learning models provide better results faster.

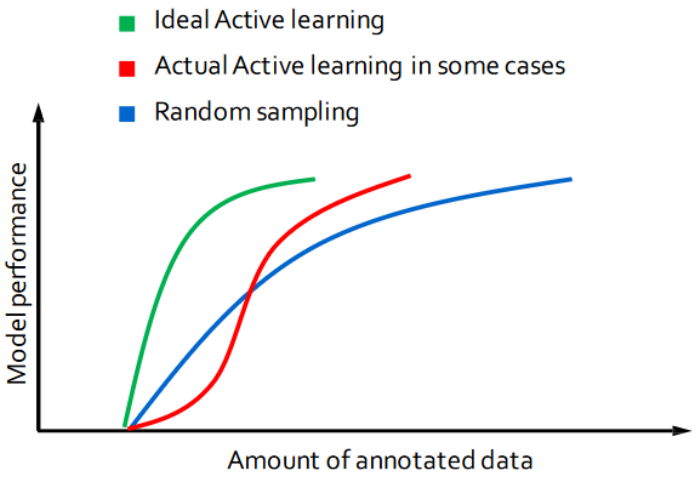

Figure 6. *Typical active learning performance*[3]*.*

2.5.2 Using Knowledge Graphs to Define AL Workflows

The preparation of the **HumAIne Active Learning (AL) platform** leverages a knowledge graph (based on ontologies, produced in T2.3) to represent the entire AL pipeline, making it intuitive and flexible for various use cases. The platform supports the rapid setup of essential components, including instances for data management, model training, prediction, and user interaction. It defines relationships between datasets, selection strategies, and algorithms, streamlining the configuration of AL experiments. The platform also incorporates Graphical User Interfaces (GUIs) for ease of use, enabling users to visually manage different parts of the pipeline, from data handling to model retraining and benchmarking with oracles. Currently, we envision the usage of LabelStud.io as a GUI.

The goal is to enable quick deployment of AL scenarios for both experimentation and production environments. In production, the platform can be seamlessly integrated with Node-RED, a low-code tool for workflow automation, facilitating real-time decision-making and process automation (used in WP5). This modular and relational structure ensures that users can efficiently design, test, and implement AL strategies, reducing setup time and accelerating the transition from prototype to production-ready solutions.

[3] https://towardsdatascience.com/introduction-to-active-learning-117e0740d7cc

Funded by the European Union

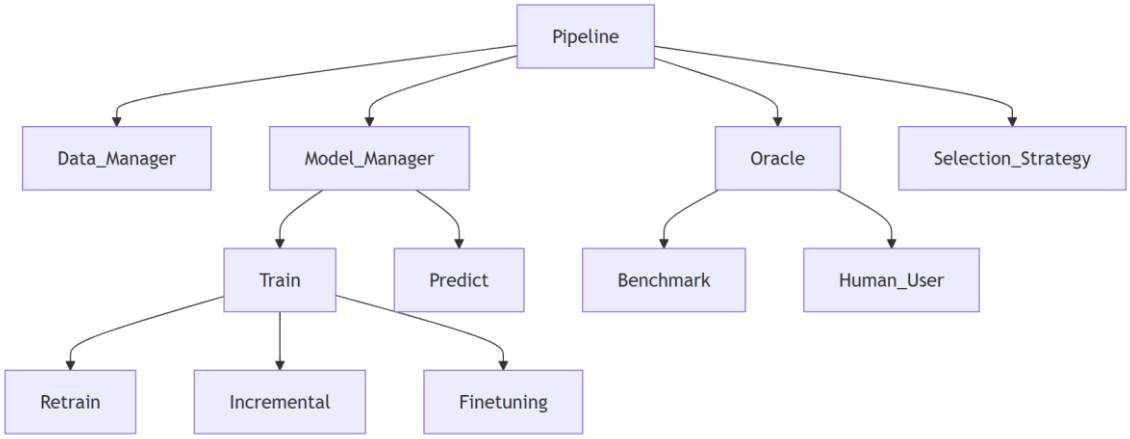

Figure 7. *Relations between several entities in the knowledge graph, representing AL process.*

2.5.3 Active Learning for Complex Unstructured Data Extraction

In many real-world applications, such as processing documents or handling unstructured data, traditional active learning strategies face significant challenges due to the inherent complexity and variability of the data. This subsection outlines a methodology designed to enhance active learning workflows when dealing with unstructured data that includes multiple formats, such as large documents with diverse tables, nested sub-tables, multi-headers, and irregular layouts. The proposed methodology integrates active learning with advanced data extraction techniques, focusing on leveraging human-in-the-loop feedback to train models to navigate and parse complex structures. The key steps in this methodology include:

1. Defining Regions of Interest (ROI): Active learning is used to involve human annotators in marking ROIs within complex documents. For example, in the case of tables with multi-row or multi-column layouts, human operators can annotate bounding boxes that highlight areas containing relevant data. These annotations guide the system to focus on specific sections of the document, reducing the noise introduced by irrelevant information. Regions of interest can be implemented using an annotation tool to allow users to draw bounding boxes, saving these annotations in structured formats like JSON. During processing, these bounding boxes guide extraction by isolating relevant areas of the document. Many can then extract text or data from these specific regions.
2. Iterative Feedback for Ambiguous Cases: When the system encounters ambiguous or uncertain data extraction scenarios—such as overlapping headers or inconsistent formatting—it queries human experts for clarification. This targeted feedback loop allows the model to adapt incrementally, learning to resolve ambiguities and improving its overall robustness. Ambiguous cases can be flagged by using uncertainty estimation methods and sent to annotators through a feedback interface. Human corrections are stored and used to retrain the model, improving its handling of similar cases.
3. Integrating Semantic Understanding: To complement ROI-based extraction, the methodology incorporates semantic understanding to interpret unstructured data. For instance, context-aware embeddings are used to identify relationships between data fields, enabling the system to correctly map extracted values to their respective categories. Semantic understanding can be implemented using contextual embeddings generated by pre-trained models like BERT. These embeddings allow the system to relate fields to their surrounding context. Similarity measures or learned rules help map extracted values to their corresponding categories.

Funded by the European Union

4. Active Learning for Rule Discovery: Beyond data extraction, active learning is applied to discover implicit rules in the data. For example, a model might learn that a particular field always appears within a certain table or follows a specific format. These cases can then be used to infer rules by analyzing the relationships between fields, positions, or formats, allowing the system to generalize its understanding. The discovered rules can be incorporated back into the pipeline to refine extraction logic and ensure more reliable data processing.

This methodology can be applied to any domain involving complex unstructured data, such as finance, healthcare, or energy document processing. The integration of human feedback ensures that the system remains adaptable to new formats, while the semantic understanding capabilities enable scalability to broader datasets.

2.5.4 Comparison of Existing Open-source Active Learning Frameworks

Name	WWW/Git	Experiment / Live	Last update	Capabilites
pyrelational	https://github.com/RelationRx/pyrelational	exp / live	Sep 2024	DL, ML
scikit-activeml	https://github.com/scikit-activeml/scikit-activeml	exp / live	Sep 2024	DL, ML
ALaaS	https://github.com/HuaizhengZhang/Active-Learning-as-a-Service	no / live	2022	DL
baal	https://github.com/baal-org/baal	exp / live	Jun 2024	DL / ML
ALiPy	https://github.com/NUAA-AL/alipy	exp / no	2022	DL / ML
modAL	https://github.com/modAL-python/modAL	exp / no	2023	DL / ML

The **PyRelationAL** library is designed to support any PyTorch model, providing flexibility for a wide range of neural network architectures. It includes several standard active learning strategies, such as uncertainty sampling, and can be extended to incorporate custom strategies. This makes PyRelationAL particularly suitable for deep learning projects where a high degree of customizability is required. However, it has limited support for non-PyTorch models, making it less ideal for simpler tasks or those not involving neural networks.

Scikit-activeml, built on top of scikit-learn, offers an array of query strategies like uncertainty, expected error reduction, and density-based sampling, making it well-suited for traditional machine learning models. The library integrates seamlessly with scikit-learn models and provides a wide range of compatible base models, fitting well into most ML workflows. It is well-documented and user-friendly, especially for newcomers. However, it is less optimized for deep learning tasks and lacks compatibility with frameworks like PyTorch or TensorFlow.

AlaaS (Active Learning as a Service) is a cloud-based solution that offloads infrastructure requirements to the cloud, making it convenient for users who need scalability without the burden of setting up infrastructure. Active learning strategies are typically accessible via APIs, simplifying the implementation of workflows. However, it is less flexible for custom query strategies and model configurations, and the cost can vary significantly depending on usage and the service provider.

Baal focuses on tasks involving high-dimensional input data and supports Bayesian deep learning through MC Dropout. This makes it particularly effective for uncertainty-based strategies in deep learning models. The library is compatible with both TensorFlow and PyTorch, offering flexibility in model choice. Despite its strengths, Baal primarily focuses on uncertainty sampling, which may require users to implement additional strategies if a broader variety of query options is needed.

Funded by the European Union

AliPy offers a diverse set of built-in query strategies, including uncertainty, density, and diversity-based sampling. The library provides fine-grained control over various stages of the active learning process, allowing for detailed experimentation. It also facilitates easy testing and comparison of strategies across different datasets and models. However, AliPy may require more setup effort compared to simpler libraries, which could be a drawback for users seeking rapid deployment.

ModAL is a lightweight library built on scikit-learn and is known for its user-friendly design and clear documentation. It allows for easy customization of query strategies and includes a wide selection of predefined strategies. ModAL works seamlessly with scikit-learn models, making it an excellent choice for active learning in simpler machine learning tasks. However, it offers limited support for deep learning and may struggle with large-scale data compared to PyTorch- or TensorFlow-based libraries.

Due to our analysis, based on simplicity of use, coverage of active learning strategies and reliability of implementation, we have opted for scikit-activeml platform and are considering PyRelationAL to be used in production in healthcare use cases.

2.6 Experimental Results - SmartCities Use Case

Novoville has digitised and managed a number of mobility-related services for local councils. One of them is a solution for managing resident-permit applications and issuing them upon approval. A resident fills up a number of different fields on a web-based form while also attaching PDF documents that prove the validity of some of the form fields. Examples include their permanent address, licence-plate of the vehicle that will be parked etc. An operator uses the PDF files to validate the data entered into the web-form prior to approving the application, accepting payment and finally issuing the actual permit. The process is repeated annually. The repetitive nature of the task and the tedious cross-checking of data makes the whole process often prone to errors.

IBM with Novoville are deploying a Robotic Process Automation system that will allow users to use an intuitive interface to automate the repetitive task of finding and cross-checking submitted data. The first step is to create automation scripts that train a bot to detect data on the PDF files and automatically compare them to the data submitted by citizens. The script will be created by leveraging upon a simple UI where council operators process sample PDF files and draw rectangles around areas of interest ie areas where the data we want to cross-check exist e.g. part of a tax return with the details of the vehicle.

This process in essence tarins a LLM that the bot will use to automate the process. Thus, after a submission of an application the bot knows where to find the citizen details e.g. their postal address on their tax return PDF and compare it to the one submitted on the application. The application itself will be fetched from the novoville APIs. The same APIs are used from the bot to confirm the validity of the data and move the application to a new stage e.g. pending payment.

2.6.1 Data

Parking permit applications typically involve submission of sensitive personal data. To facilitate development while waiting for corresponding data-agreement approvals with willing councils we have opted for creation of synthetic data.

For this reason, we have created a registry that will be used to test the system without sacrificing accuracy or quality of experience:

A list of synthetic data tuples that would allow any user to submit an application, namely random values for: name, surname, email, mobile phone, ID number, VAT number, address and postcode, power supply number, vehicle licence plate. Each tuple is completely synthetic and does not correspond to a real individual. Nevertheless, the format of each data point follows specific rules to resemble reality e.g. Greek ID numbers that start with two characters and are followed by six digits.

Funded by the European Union

Tax return PDF files that are always included in the applications. Each tuple of form data comes with its own version of a tax return. All data are placed into an actual/real template of a tax return. These PDF files are then used by the algorithms to extract the data and compare to those submitted in the form.

We have initially created 500 tuples of training and testing synthetic data.

2.6.2 Active Learning

The HumAIne project leverages active learning principles to develop and refine an RPA agent in collaboration with Novoville. The agent is designed to automate the validation of parking permit applications by cross-checking data provided in web forms with details extracted from corresponding PDF documents. This approach can incorporates active learning at various stages to improve the robustness and accuracy of data extraction, validation, and comparison.

We are mixing humans with RPA to fully automate the process and build confidence around the quality of the automation. Eventually, council operators will use a set of provided training PDFs to draw rectangles around regions where data detection will occur. This will be facilitated through a new module on the Novoville council dashboard, which is already used to process parking permit applications. The module will present a list of PDFs along with annotation tools for marking these regions. For development purposes, we are currently conducting this process using off-the-shelf marking tools and a custom codebase to annotate areas of interest on the PDFs. These early steps are crucial for training the RPA system to extract key data fields with precision and accuracy. Below, we outline the contributions of active learning to this use case.

Iterative Data Extraction and Validation: The current algorithm iterates through all submitted applications, extracting user-provided data from the UI, opening the associated PDF files, and parsing them for key-value pairs located in predefined regions. Active learning can play a critical role by enabling the identification and annotation of ambiguous or challenging data extraction scenarios. Council operators interact with the system to mark specific regions of interest within PDF documents, such as addresses or vehicle details, using bounding boxes. These annotations can provide the foundation for training the model, allowing it to generalize to new and unseen formats with increased accuracy.

Adaptive Learning Through Human Feedback: Active learning ensures that the system continuously improves by incorporating feedback from human operators. For instance, if the RPA agent encounters a discrepancy or uncertainty in extracting key-value pairs—such as distinguishing between similar fields like "Owner's Address" and "Registered Address"—it can flag these cases for human review. By querying humans only on ambiguous cases, the model reduces labeling costs while focusing on the most informative data points. Over time, this feedback loop can enhances the agent's ability to handle variations in PDF formats, such as differences in table structures, merged cells, or non-standard column titles.

Integration with Future Algorithm Enhancements: While the current system focuses on extracting data from predefined regions, future iterations of the algorithm will incorporate more advanced features. These include context-aware extraction, OCR capabilities, and the use of semantic understanding to identify key-value pairs beyond bounding box detection. Active learning will remain pivotal in these advancements by prioritizing human intervention for complex scenarios, such as poorly scanned documents or PDFs with overlapping data fields.

Validation of Extracted Data Against Web Forms: A core feature of the system is comparing extracted data with user-provided information on web forms. Active learning can also helps the system learn from edge cases where discrepancies arise due to formatting differences, such as variations in date formats or unit conventions. For example, if a mismatch occurs between a "License Plate Number" extracted from a PDF and its representation on the web form, active learning facilitates human review

Funded by the European Union

to confirm whether this is an error or a formatting issue. The system can uses this feedback to adjust its comparison logic, improving its ability to handle similar discrepancies autonomously in the future.

Early Results and Future Directions

Initial results show that without marking the bounding boxes where the data appears, the system makes many mistakes in extracting key-value pairs. This is due to the inherent complexity of the PDFs, which are often very large, contain multiple tables and sub-tables, and feature diverse headers, titles, multi-rows, and multi-columns. Extracting such unformatted data proves to be a significant challenge, even for state-of-the-art frontier models. The lack of clear structure and consistency in document formatting results in ambiguities that make accurate extraction difficult. The integration of human annotations through active learning is the first crucial step in addressing these challenges. By leveraging bounding box annotations, the system gains precise guidance on where to focus its extraction efforts, significantly improving its accuracy and reliability. Moving forward, scaling the system to handle a broader range of document formats and refining its ability to learn from ambiguous cases will be central to achieving high-quality automation. Active learning will continue to play a central role, ensuring that the RPA agent delivers high-quality automation while maintaining trust and transparency in its operations.

2.7 Experimental Results - SmartHealthcare Use Case – Oncology

In the context of the HumAIne project, we explored the application of Active Learning (AL) for Named Entity Recognition (NER) tasks, specifically focusing on the fine-tuning of the GLiNER model. The goal was to efficiently extract key information from medical records, such as patient names, illnesses, diagnoses, and treatments. This work was conducted in collaboration with the EU project PREPARE and used a dataset generated from our previous research on synthetic generation of medical datasets (awarded paper at SIKDD conference, Ljubljana, Slovenia)[4]. GLiNER, known for its ability to recognize a wide range of entities, performed well on a synthetic Slovenian dataset of 200 examples, achieving an F1 score of 0.81, with precision at 0.84 and recall at 0.78. Labels such as "name" and "person" often identified overlapping entities, demonstrating GLiNER's adaptability.

How gliner operates?

- as an inputs, we give GLiNER:
  - text that we want it to analyse,
  - labels we want it to find inside text,
- as an output we get:
  - entities that GLiNER thinks belong to the said labels,
  - his confidence in each of them.

Code is depicted in the figure below.

[4] https://aile3.ijs.si/dunja/SiKDD2024/Papers/IS2024_-_SIKDD_2024_paper_4.pdf

Funded by the European Union


```
text = """
            Our patient Alexis Rose was born on 12.12.1984 in Madagaskar.
            Due to bad circumstations in her home she developed a lung cancer.
            We are still trying to see whether or not she will be perceptive to chemotherapy.
            For two days now, she suffered from severe coughing.
        """

labels = ["Person", "location", "disease", "date of birth", "symptoms"]

entities = model.predict_entities(text, labels, threshold=0.5)
for entity in entities:
    print(entity["label"], "=>", entity["text"])
```

```
Person => Alexis Rose
date of birth => 12.12.1984
location => Madagaskar
disease => lung cancer
symptoms => severe coughing
```

Figure 8. *Usage of GLiNER.*

```
['ime', 'ulica', 'poštna številka', 'mesto', 'poklic', 'e-poštni naslov', 'osebno ime', 'zdravstvena ustanova', 'oseba', 'simpt
om', 'številka potnega lista', 'diagnoza', 'zdravljenje', 'priimek', 'ime podjetja', 'lokacija', 'pravni dokument', 'datum rojs
tva', 'državljanstvo', 'država', 'ime stranke', 'bančni termin', 'kraj rojstva', 'naslov prebivališča', 'pacient', 'zdravstveno
stanje', 'zdravnik', 'organizacija', 'podjetje', 'email', 'naslov', 'elektronska pošta', 'datum', 'čas', 'plačilno sredstvo',
'številka kreditne kartice', 'medicinski postopek', 'kreditna kartica', 'banka', 'finančna transakcija', 'ime banke', 'časovni
okvir', 'finančni izraz', 'stik z banko', 'trgovina', 'časovno obdobje', 'anatomska lokacija', 'vsota', 'številka zavarovalne p
olice', 'številka zdravstvene kartice', 'dokument', 'ime na kartici', 'številka kartice', 'starost', 'bolezen', 'leto', 'telefo
nska številka', 'mednarodna klicna koda', 'elektronski naslov', 'pravni termin', 'številka osebnega dokumenta', 'kraj', 'kraj z
aposlitve', 'ustanove ime', 'mobilni telefon', 'družinski član', 'bančni produkt', 'bančni račun', 'zakon', 'dan v tednu', 'zdr
avilo', 'znesek', 'finančni produkt', 'ime zdravila', 'ime storitve', 'vrsta računa', 'vrsta aplikacije', 'izraz za zdravilo',
'cpf', 'bančna institucija', 'osebni dokument', 'pravno besedilo', 'e-naslov', 'identifikacijska številka', 'kraj prebivališč
a', 'delovno mesto', 'zdravstvena storitev', 'zdravstveni parameter', 'finančna ustanova', 'obrestna mera', 'časovna enota', 'v
ozniško dovoljenje', 'kategorija', 'vozniska dovoljenje številka', 'osebje', 'trajanje', 'področje', 'davčna številka', 'ime us
tanove', 'medicinski izraz', 'zdravstveno osebje', 'medicinsko stanje', 'odstotek', 'stroški', 'pravno področje', 'bančna ustan
ova', 'zavarovalnica', 'finančni stroški', 'institucija', 'številka osebne izkaznice', 'zdravila', 'bančni izdelek', 'bančni po
stopek', 'komunikacija', 'mesto rojstva', 'nacionalna identifikacijska številka', 'oddelek', 'emšo', 'znesek varčevanja', 'mobi
lna aplikacija', 'ip naslov', 'zdravstvena zavarovalnica']
```

Figure 9. *A list of entities (in Slovenian), that are currently extracted with our version of GLiNER (including personal data, symptoms, diagnosis, treatment, anatomic location and name of the treatment.*

To enhance GLiNER's performance and ensure scalability, we designed an Active Learning workflow to fine-tune the model for specific use cases and improve labeling efficiency. We plan to test transfer learning capabilities of GLiNER to determine how well knowledge from one domain can be adapted to another with minimal labeled data. For AL strategies, we evaluated libraries like Scikit-activeml and PyRelationAL, which offer robust support for model retraining, incremental learning, and selection strategies.

Additionally, our pilot is working on training the mT5 model for question-answering tasks, which will also be tested using Active Learning to assess improvements in efficiency and accuracy.

2.8 Experimental Results – Smart Finance

2.8.1 Problem Description

The dataset utilized in this study contains information from helpdesk tickets, which are generated by users needing assistance with various IT issues. Each ticket logs essential details, including the description of the problem reported by the user, the department or IT team to which it was assigned, the responses provided by these teams, and others. The objective of our analysis is to predict the content of the IT team's first reply to each ticket. By focusing on the initial response, we aim to model how IT teams typically address specific issues, providing a foundation for automation and potentially improving response times and lowering the amount of work for the IT team.

Funded by the European Union

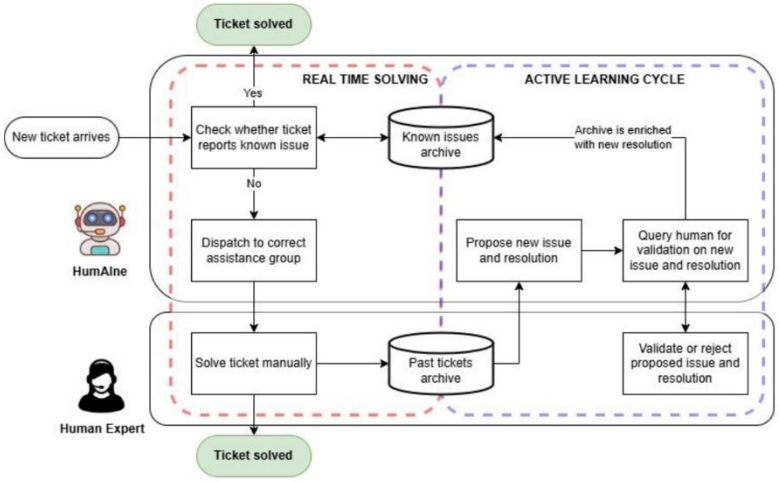

Figure 10. *Architecture of active learning solution for smart ticketing system.*

2.8.2 Clustering

Our initial approach involved exploring the possibility of identifying natural groups within the helpdesk tickets. Clustering was selected as a suitable method for this task. To begin, we applied a standard approach using TF-IDF vectorization of the ticket texts combined with k-means clustering. However, this approach yielded unsatisfactory results, with clusters lacking meaningful distinctions.

We then shifted to using embeddings generated with BERT, combined again with k-means. Despite this adjustment, the resulting clusters did not adequately capture distinct topics. To further explore other clustering techniques, we applied hierarchical clustering using TF-IDF features, yet once more, the results failed to provide clear clusters. Additionally, experimenting with DBSCAN (Density-Based Spatial Clustering of Applications with Noise) using TF-IDF also produced clusters that were too inconsistent.

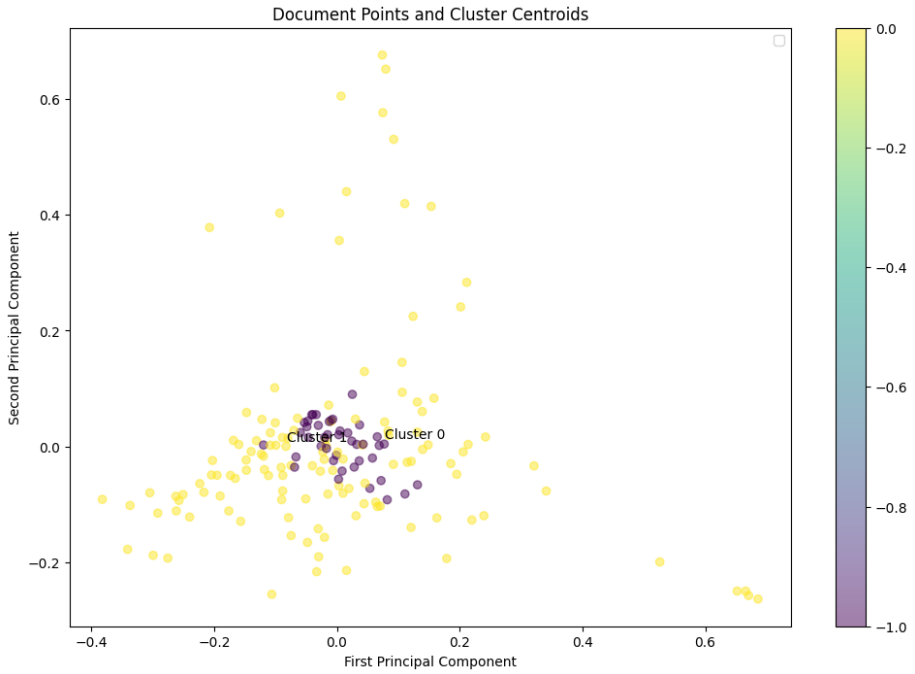

Figure 11. *Clusters Produced by DBSCAN.*

The results improved, when we employed BERTopic, a framework specifically designed for topic modeling. BERTopic allowed us to use sentence embeddings, which were generated with Sentence Transformers, and apply HDBSCAN, a density-based clustering method, to group the tickets. This

Funded by the European Union

combination offered improved performance, with BERTopic's flexibility allowing us to experiment with different configurations, such as predefining topics and adding topic seeds to guide the clustering process.

In the first Figure 12. , the BERTopic results reveal nicely separated groups, with similar tickets clustered together. The second figure offers a visual comparison between our defined topics and the generated clusters. While there were generally more topics identified than clusters, the overlap was favorable, indicating a strong alignment between the clusters and the thematic topics we anticipated. Although the results from BERTopic were markedly better than previous approaches, some inconsistencies remained, suggesting that further refinement might be required before these clusters could be used for automated categorization.

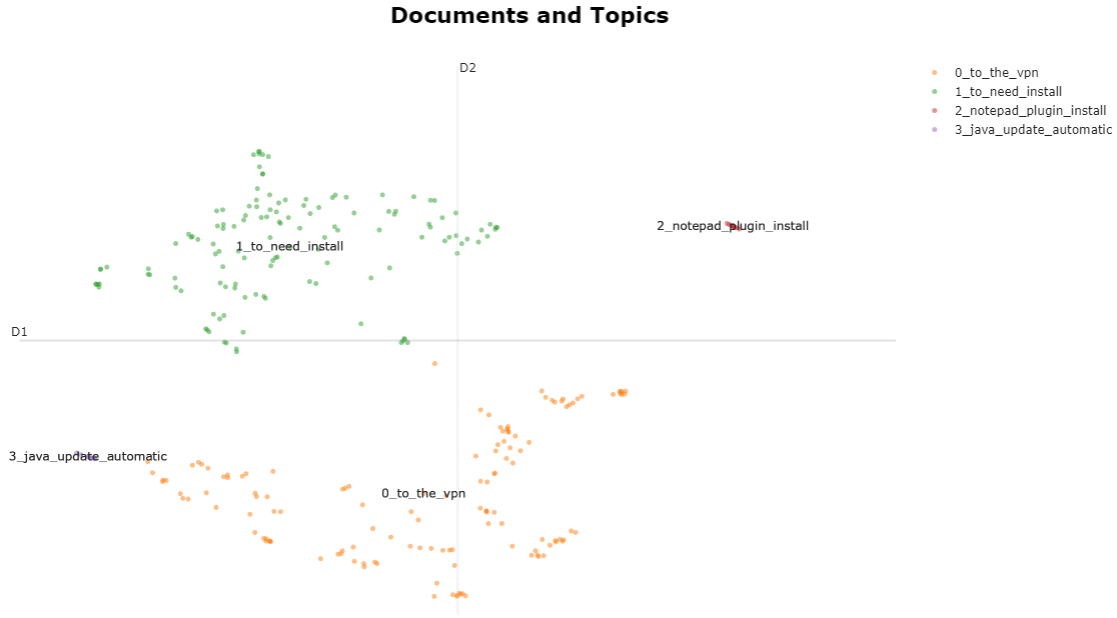

Figure 12. *Clusters Produced by BERTopic.*

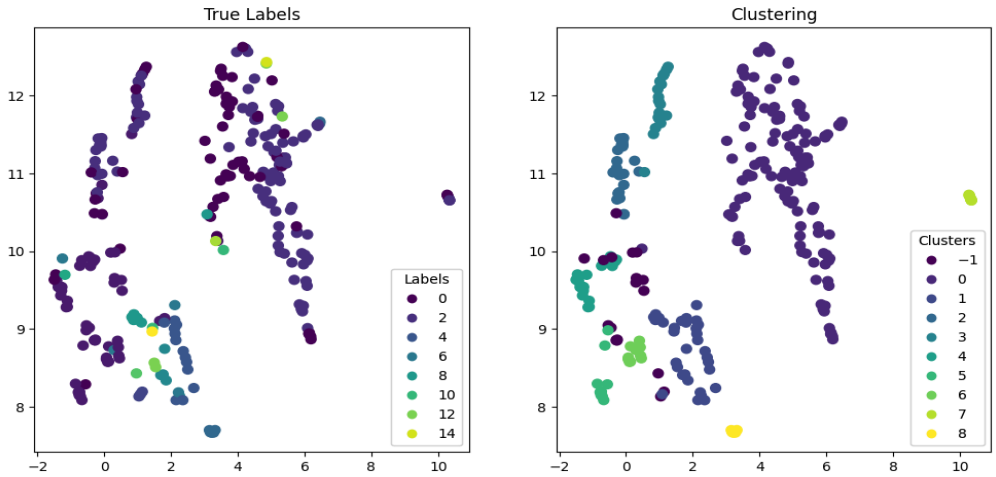

Figure 13. *Comparison Between Clusters and Predefined Labels.*

2.8.3 Active Learning

Following the clustering analysis, we aimed to apply active learning techniques to the helpdesk ticket data to determine if we could accurately predict the IT team responses with minimal labeled data. Specifically, we were interested in comparing various active learning query strategies to identify the most effective approach for this dataset.

Funded by the European Union

The strategies tested are:

- uncertainty sampling with entropy
- uncertainty sampling with margin
- uncertainty sampling with least confidence
- query by committee
- clustering uncertainty-weighted embeddings (CLUE)

We compared these active learning strategies against a baseline approach of random selection, which allowed us to measure the acceleration of each targeted strategy. A detailed comparison of these strategies is provided in the related work section. Additionally, our active learning pipeline can be seen in the figure below.

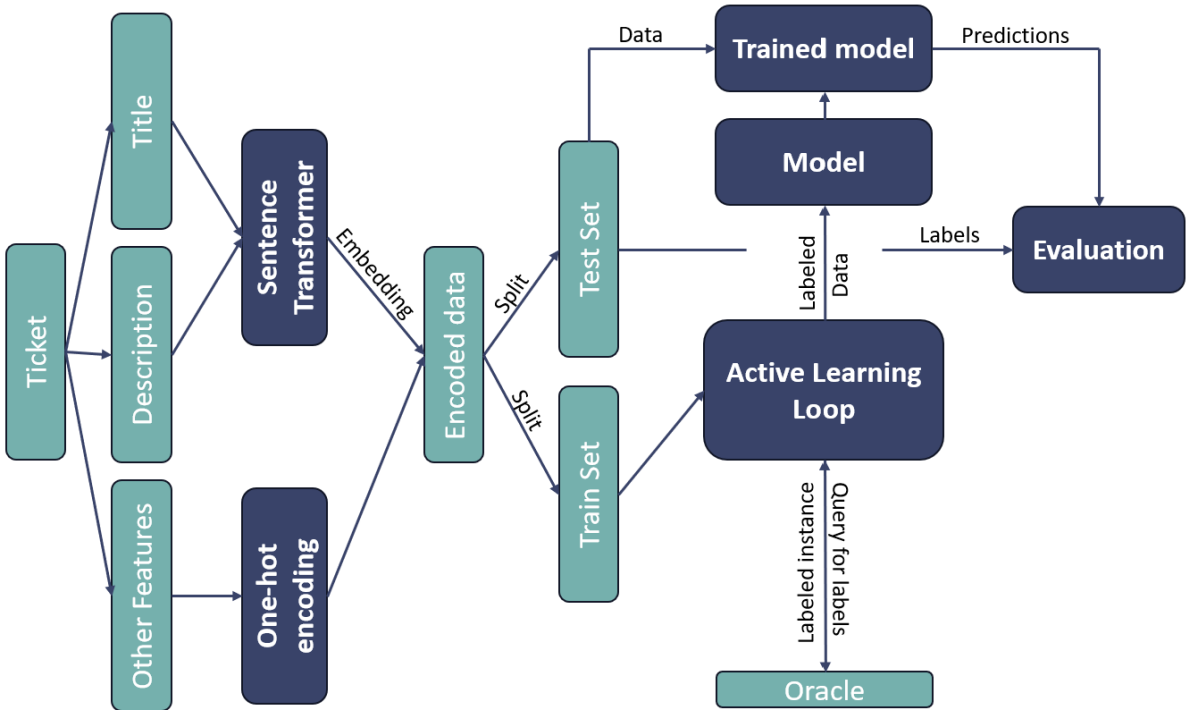

Figure 14. *Active Learning Pipeline.*

2.8.4 Active Learning for IT Team Prediction

Our first active learning task involved predicting the assigned IT team for each ticket to assess the model's performance under a limited label scenario. As a baseline, we trained a model on the full dataset to predict the IT team. This dataset contains 21 different IT teams, with each ticket represented by embeddings from Sentence Transformers alongside additional features encoded using one-hot encoding. With Support Vector Classifier (SVC), we achieved an **F1 score** of **0.83** on this fully labeled dataset, setting our performance target for the active learning experiments.

We began with an initial setup of one labeled instance per class, ensuring a minimal but diverse starting point. In each following iteration, we added two more labeled instances to simulate the active learning process gradually. The graph of results demonstrates the performance across different query strategies.

Funded by the European Union

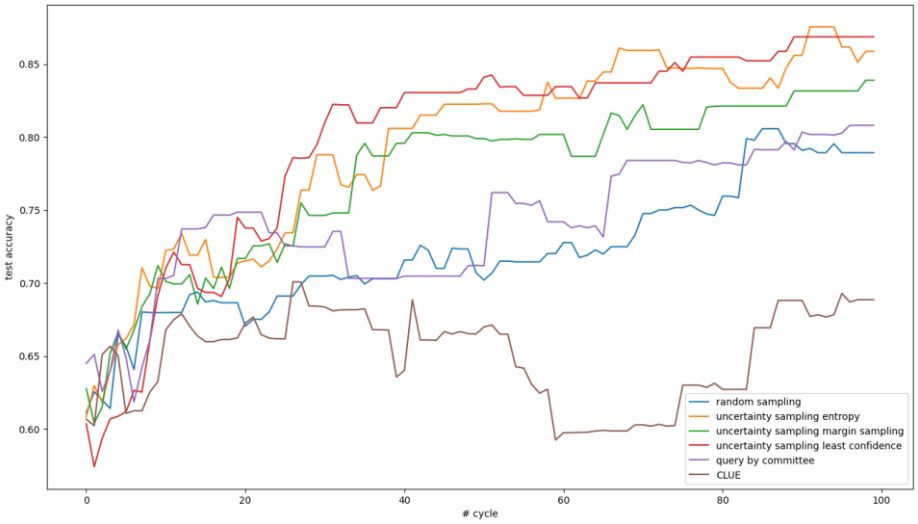

Figure 15. *The Graph of F1 Scores for Various Query Strategies (IT Team Prediction)*

Among the active learning strategies, the uncertainty-based approaches performed best. These strategies consistently outperformed random selection by selecting instances where the model's predictions were least certain. On the other hand, CLUE, which combines clustering with uncertainty weighting, performed poorly, possibly due to the dense nature of the IT team classes, which reduced the effectiveness of its clustering approach. As we continued adding labeled instances, the F1 score achieved by the active learning models eventually reached the baseline score of 0.83.

In addition to F1 scores, we also calculated the acceleration metrics for different F1 values. Here, we compared the best performing active learning strategy to random selection. As shown in the figure, when approaching the baseline accuracy, the acceleration factor reached approximately 2.3. This indicates that the active learning process achieved the same predictive accuracy as random selection while using roughly half the labeled instances, highlighting the efficiency of active learning in reducing labeling costs.

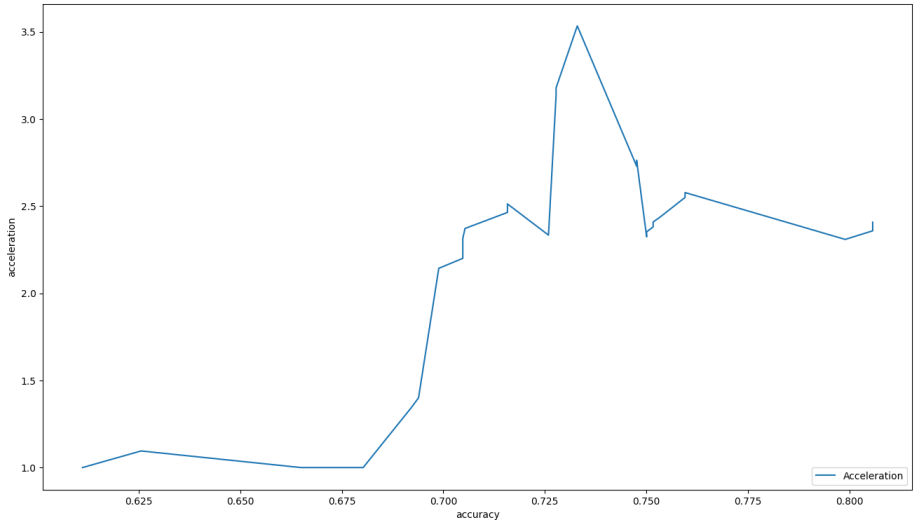

Figure 16. *The Graph of Accelerations Across Different Accuracies (IT Team Prediction)*

2.8.5 First Reply Extraction

Following the IT team predictions, we shifted focus to directly predicting the IT team's first reply to each ticket. This required us to extract the initial replies, aiming to create classes based on their common patterns and content. However, the first replies in the dataset were only partially structured, which presented some challenges for direct classification.

Funded by the European Union

To address this, we first cleaned the beginning of each reply, removing any inconsistent elements that could interfere with grouping similar responses. Once cleaned, we analyzed the initial segments of each reply, identifying distinctive patterns that helped us separate the replies into more cohesive groups.

Next, we created classes based on specific recurring phrases or structures in the replies, allowing us to cluster similar responses together. Each class contained replies that were largely uniform, with only minor variations. For cases that did not fit into any established class, we created a catch-all category labeled 'other,' ensuring that every ticket had a designated reply class.

This classification step enabled us to group similar replies effectively, simplifying the prediction task by reducing the variety in reply types to a manageable set of classes.

2.8.6 First Reply Prediction

With the first replies now arranged between the classes, we proceeded to predict them based on the ticket content. To ensure meaningful predictions, we filtered out any reply classes with fewer than 10 instances, focusing on the more frequently occurring classes to provide adequate data for training.

Our initial approach, similarly as before, involved embedding the ticket text with Sentence Transformers and encoding additional features using one-hot encoding. We then applied a SVC and linear regression model to predict the reply classes. SVC outperformed linear regression and achieved an F1 score of 0.93, indicating that the combination of embeddings and one-hot encoded features could effectively capture the patterns necessary for accurate reply prediction.

To explore further, we also experimented with fine-tuning DistilBERT, using the ticket content and additional features as text inputs. This approach was designed to capture deeper semantic patterns in both the ticket descriptions and structured features. The results were comparable, but didn't beat those from the Sentence Transformer and SVC method.

2.8.7 Active Learning for First Reply Prediction

Given that many of the IT team's initial replies lack consistent structure and may require manual labeling to ensure accurate predictions, we wanted to evaluate the effectiveness of active learning for predicting the first replies. In this task, there were 22 possible reply classes.

The active learning initialization mirrored that of our previous IT team prediction case: we began with a minimal labeled set and added two labeled instances in each iteration. We applied the same query strategies as before, uncertainty sampling (with entropy, margin, and least confidence variants), query by committee, CLUE, and a random selection baseline.

The results, shown in the figure, reveal that uncertainty sampling consistently outperformed random selection, reaffirming its effectiveness in scenarios with limited labeled data. CLUE once again achieved worse results than random selection. Similarly, query by committee did not perform well with this dataset, showing performance close to that of random selection and indicating that this approach may be less suited to predicting reply classes.

Funded by the European Union

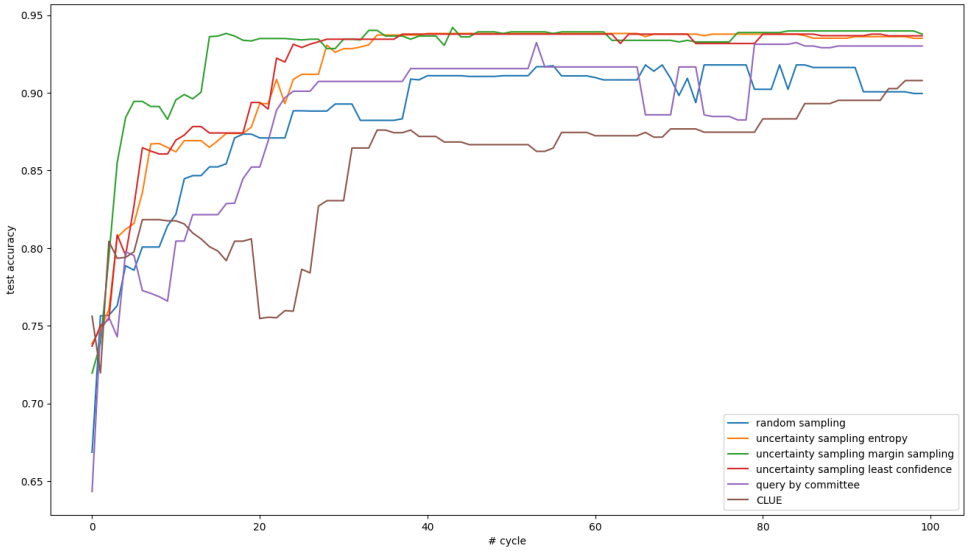

Figure 17. *The Graph of F1 Scores for Various Query Strategies (First Reply Prediction)*

As the active learning iterations progressed, the model's F1 score reached the baseline set by the fully labeled dataset. For different F1 scores, we also calculated the acceleration metrics, depicted in the figure. When the model reached the baseline accuracy, the acceleration was approximately 2.5, meaning that active learning achieved similar predictive performance with less than half the labeled instances needed for the same accuracy with random selection strategy.

These results indicate that active learning is an effective method for predicting the IT team's first replies, providing an efficient way to reduce manual labeling requirements while maintaining predictive accuracy.

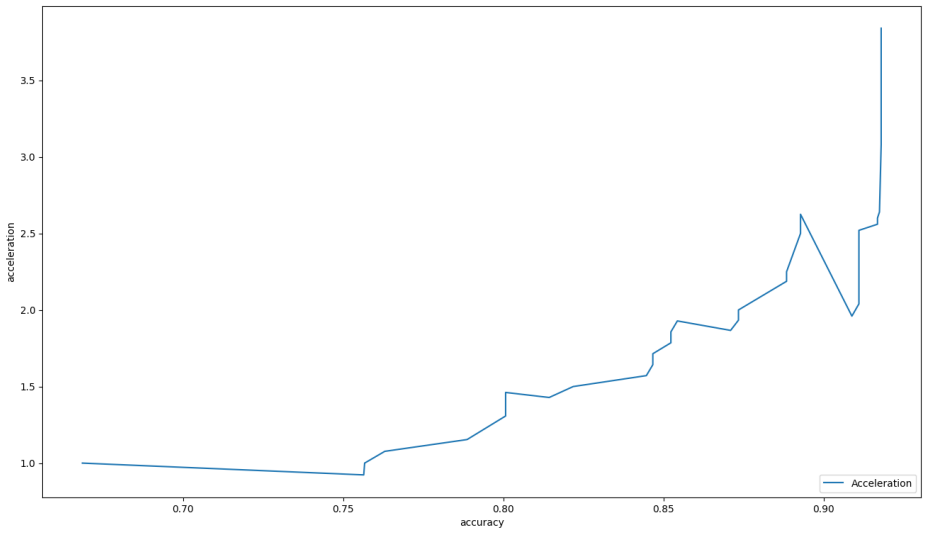

Figure 18. *The Graph of Accelerations Across Different Accuracies (First Reply Prediction)*

2.9 Early Prototypes

2.9.1 SmartCities Prototype

We have currently created an early prototype that performs the following parts of what later will be a full user story:

Submission of an application

Funded by the European Union

We have created a real parking-permit application that we will be using to demonstrate the overall process on the novoville web app. This application can be seen in the screenshot that follows and consists of several personal information data points (meaningfully grouped) and the requirement to attach at least one PDF with the user's tax return.

Submission triggers a number of events including storage of the PDF file, storage of the form data and eventually pushing this same information to the Humaine servers. This triggers a further event, namely the launch of the RPA algorithm that will detect whether the form data matches the information detected on the PDF file. The result is conveyed back to the novoville servers and depending on the business process of each council either approve the issuing of the permit automatically or request a payment prior to issue.

Currently this automation of events is still work in progress. We submit the form and the algorithm is manually launched to fetch the data and the corresponding PDF file.

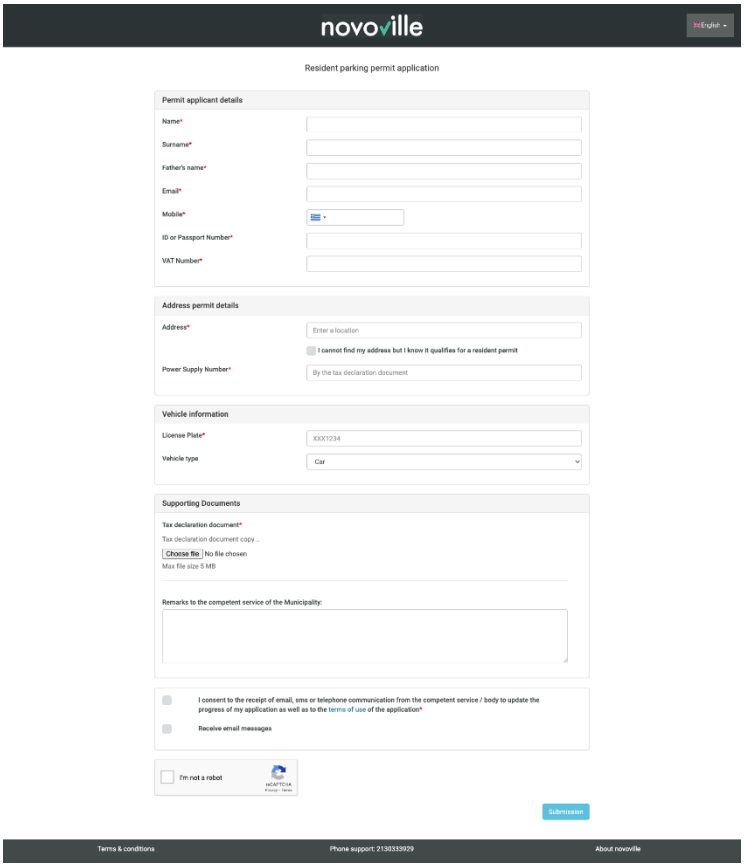

Figure 19. *Screenshot from the GUI for resident parking permit application.*

RPA Algorithm Overview in Early Prototype

The early prototype of the SmartCities solution includes an RPA algorithm that automates data validation by cross-checking user-provided form data against information extracted from attached PDF documents. This prototype forms the backbone of the automation pipeline, and its functionality is described in detail below:

Iteration Through Applications: The algorithm begins by iterating through all submitted parking-permit applications. Each application contains user-submitted form data and an attached PDF document. This step ensures that every submission is processed systematically.

Reading User Data from the UI: For each application, the algorithm extracts relevant data fields from the UI form. This includes personal information (e.g., name, address, and license plate) and other

Funded by the European Union

required details entered during the submission process. This extracted data forms the basis for comparison against the data found in the PDF.

Opening and Parsing the Attached PDF: The algorithm opens the corresponding PDF file attached to the application and processes it to extract textual information. Given the unstructured and complex nature of the PDFs, this step involves parsing the document into key-value pairs, identifying tables, and isolating relevant sections of the document.

Extracting Key-Value Pairs Using Active Learning: To address the challenges of unstructured data in large PDFs, the algorithm relies on bounding boxes identified through active learning. These bounding boxes, marked during the training process, guide the system to focus on specific areas of the PDF where relevant data is expected to appear (e.g., address fields, tax information). This targeted extraction helps to improve accuracy and reduce ambiguity.

Data Comparison and Validation: Once the key-value pairs are extracted, the algorithm compares each data point against the corresponding fields from the UI form. This step involves checking for exact matches or identifying discrepancies in information such as names, addresses, or numerical values.

Handling Discrepancies: Any mismatches or uncertainties identified during validation are flagged for further review. In future iterations, the system will incorporate additional logic to handle discrepancies caused by formatting differences or OCR inaccuracies, enhancing its robustness.

Current Limitations and Future Enhancements

Initial results highlight significant challenges in extracting accurate key-value pairs without human annotations to guide the algorithm. The PDFs attached to applications often contain multiple tables, sub-tables, headers, and diverse layouts, which make automated extraction difficult even for state-of-the-art models. To address this, the integration of bounding box annotations through active learning is planned. This will provide the system with precise regions of interest, enabling more reliable data extraction. Future improvements to the algorithm will include advanced OCR techniques, semantic understanding for context-aware data extraction, and the ability to generalize across varying document formats. These enhancements will ensure that the RPA system achieves higher accuracy and efficiency in automating the validation process.

Currently, the entire process—from form submission to data validation—is not fully automated. After the form submission, the algorithm is manually triggered to fetch the data and corresponding PDF files for validation. Future iterations of the prototype will aim to fully automate this workflow, ensuring that all events, including data extraction and validation, are seamlessly integrated into the overall business process. This RPA algorithm serves as a foundational step toward a fully automated, reliable, and efficient solution for validating parking-permit applications. Through iterative refinement and active learning integration, the prototype will evolve into a robust tool for SmartCities applications.

2.9.2 SmartFinance Prototype

The prototype was developed using a Python-based backend, specifically using the scikit-activeml library for active learning . Initially, data is fetched from the database and processed for active learning. For the frontend, Streamlit was used, which allowed us to visualize the real-time results of the active learning process.

Funded by the European Union

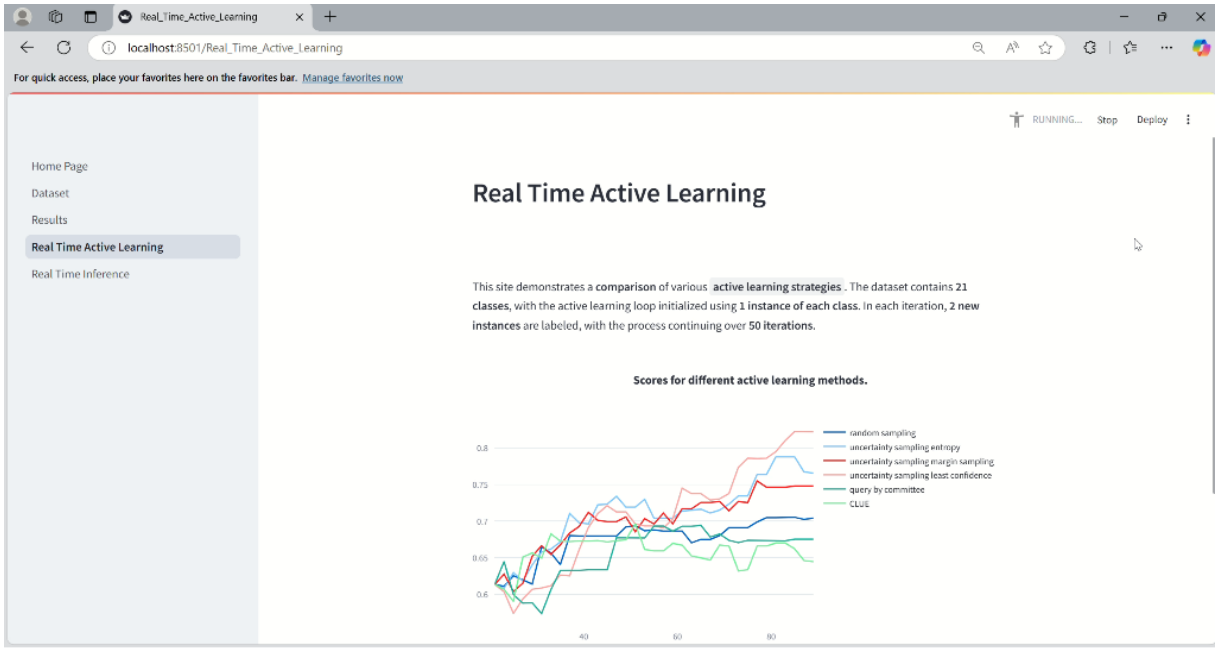

Figure 20. *Screenshot from the initial active learning prototype.*

Once trained, the best-performing model is utilized for inference. Through the Streamlit interface, users can select specific instances they wish to predict, and the model then returns the predictions. Additionally, tests were conducted using Label Studio to manually annotate the data, which can be used during the active learning process.

In the figure below, we can see the pipeline of the prototype.

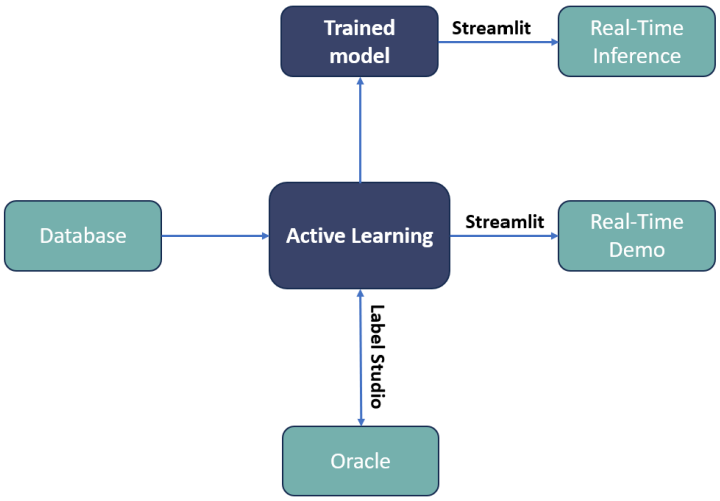

Figure 21. *Prototype Pipeline.*

In the figures below, screenshots from Label Studio are presented. Label studio was used for experiments with labeling of the helpdesk tickets.

Funded by the European Union

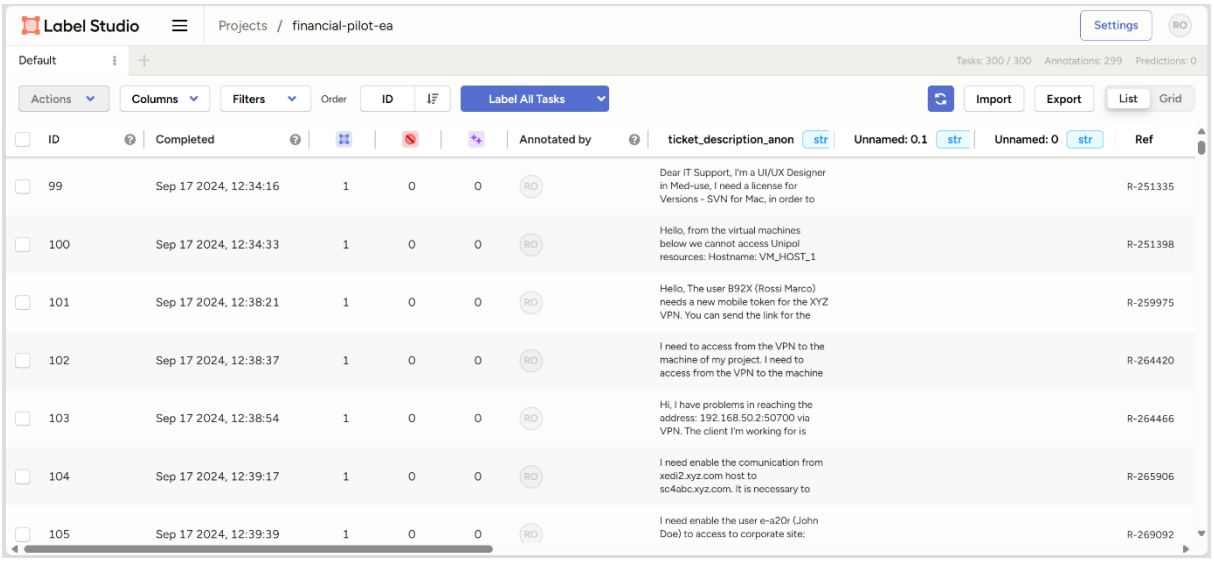

Figure 22. *Overview of the labeling project in Label Studio.*

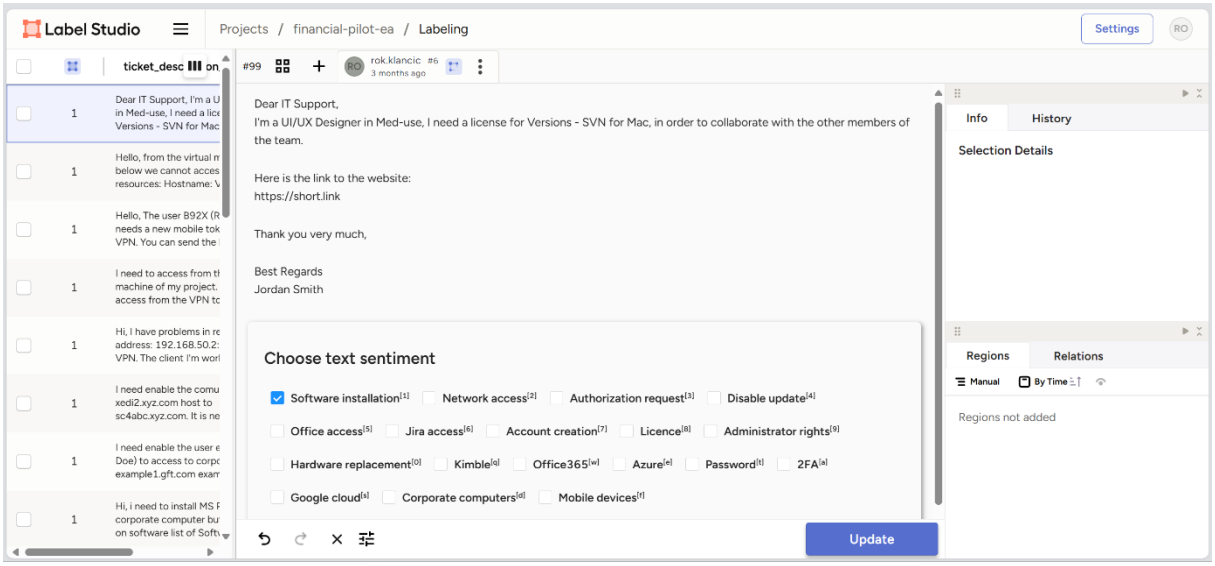

Figure 23. *Labeling a particular instance (for smart finance pilot) in Label Studio.*

2.10 Future Work

In the upcoming period, our efforts in Task 4.2 will focus on advancing the active learning (AL) platform development to ensure robust and versatile functionality across multiple domains. As a priority, we aim to complete proof-of-concept (POC) implementations for the smart energy and smart healthcare pilot cases. This will be complemented by obtaining comprehensive datasets from all defined use cases, enabling the development of standalone demonstrators that showcase the platform's applicability and potential. Furthermore, significant attention will be dedicated to aligning the active learning platform with the overarching HumAIne project architecture, ensuring seamless integration and interoperability.

From a technical perspective, we will enhance the user interface components by extending LabelStud.io to support diverse tasks such as text classification, named entity recognition (NER), and question answering. On the backend, we will finalize the HumAIne AL platform to simplify deployment and facilitate workflows driven by knowledge graphs (KG). This comprehensive approach will ensure that the platform is accessible, scalable, and adaptable to various use cases, making it a cornerstone of the HumAIne ecosystem.

Funded by the European Union

To extend the state of the art, we will explore innovative batch selection strategies, focusing on multi-objective approaches for multiple NER labeling. Additionally, we will research the application of active learning for fine-tuning large language models (LLMs), using models like Gliner for NER tasks and mT5 for question answering. Another line of inquiry will investigate the human-in-the-loop methodology, where human expertise guides the active learning process, particularly for training data-driven models that rely on complex numerical models. These research efforts will not only strengthen the theoretical foundation of active learning but also provide practical insights to enhance its application in real-world scenarios.

2.11 References (Active Learning)

- Aggarwal, Charu C., et al. "Active learning: A survey." Data classification. Chapman and Hall/CRC, 2014. 572-634.
- "Baifanxxx/Awesome-Active-Learning." GitHub, **https://github.com/baifanxxx/awesome-active-learning**, Accessed: 2024-11-15.
- Devlin, J., “Bert: Pre-training of deep bidirectional transformers for language understanding,” arXiv preprint arXiv:1810.04805, 2018.
- Gildenblat, J. (2020). *Overview of Active Learning for Deep Learning*. **https://jacobgil.github.io/deeplearning/activelearning**
- Goel, Akshay, et al. "Llms accelerate annotation for medical information extraction." Machine Learning for Health (ML4H). PMLR, 2023.
- Grootendorst, M., “BERTopic: Neural topic modeling with a class-based TF-IDF procedure,” arXiv preprint arXiv:2203.05794, 2022.
- Keraghel, Imed, Stanislas Morbieu, and Mohamed Nadif. "A survey on recent advances in named entity recognition." arXiv preprint arXiv:2401.10825 (2024).
- Kottke, D., “scikit-activeml: A Library and Toolbox for Active Learning Algorithms,” Preprints, 2021.
- Liu, Mingyi, et al. "LTP: a new active learning strategy for CRF-based named entity recognition." Neural Processing Letters 54.3 (2022): 2433-2454.
- Prabhu, V., Chandrasekaran, A., Saenko, K., and Hoffman, J., “Active domain adaptation via clustering uncertainty-weighted embeddings,” in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 8505-8514.
- Reimers, N., “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” arXiv preprint arXiv:1908.10084, 2019.
- Remstam, Sophie. "A Novel Low Annotation-Cost Interactive Framework for Named Entity Recognition." (2020).
- Sanh, V., “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter,” arXiv preprint arXiv:1910.01108, 2019.
- Scherer, P., Gaudelet, T., Pouplin, A., del Vecchio, A., S, S. M., Bolton, O., Soman, J., Taylor-King, J. P., & Edwards, L. (2022). *PyRelationAL: a python library for active learning research and development*. **https://arxiv.org/abs/2205.11117v2**

Funded by the European Union

3 Swarm Learning

The goal of Task 4.3 is to develop a swarm learning platform which fosters the collaboration, information exchange and gain of distributed humans and AI systems for the efficient optimization of a common objective. The swarm learning platform will be utilized and evaluated in the manufacturing (T6.2) and the oncology pilot (T6.4).

3.1 Introduction

3.1.1 Definition

The term “swarm learning“ is a composition of the terms swarm intelligence and machine learning. (Luo, Shi, & Cai, 2020) defines swarm intelligence “as a collective behaviour of a decentralized or self-organized system. These systems consist of numerous individuals with limited intelligence interacting with each other based on simple principles. Although the individual is not so smart and there is no central leader dictating how individuals behave, only by the interaction between individuals can the whole system achieve a holistic intelligence. In general, no individual has an overall cognition but just carries out its simple actions”. On the other hand according to the Oxford English Dictionary is machine learning described as “the use and development of computer systems that are able to learn and adapt without following explicit instructions, by using algorithms and statistical models to analyse and draw inferences from patterns in data”.

Hence the combination of the terms swarm intelligence and machine learning results in a system which consists of multiple distributed and autonomous entities. By the communication of the different entities and the exchange of knowledge an overall intelligence can be built which exceeds the intelligence of a single entity. Thus, swarm learning defined by the HumAIne project is a way of distributed entities to learn, collaborate, improve and share knowledge for a common objective.

In literature there is no common agreement of the term swarm learning. Instead, it is used for different technological approaches. The term was introduced by Hewlett Packard Enterprise and the German Center for Neurodegenerative Diseases (DZNE). They define it as a “decentralized machine-learning approach that unites edge computing, blockchain-based peer-to-peer networking and coordination while maintaining confidentiality without the need for a central coordinator” (Warnat-Herresthal, et al., 2021). This approach was picked up. However, there is no common agreement on whether the blockchain is mandatory technology addition in swarm learning (cp. (Han, Ma, & Han, 2022), (Chester Chen & Flores, 2024)). Further, (Wang, et al., 2024) use the term distributed swarm learning for the combined technology of federated learning (FL) and particle swarm optimization. The described approach is hereby independent of the FL architecture.

In summary, the term swarm learning contains similarities with federated learning, in particular with a peer-to-peer decentralized architectural approach of federated learning. The combined optimization technology is not agreed upon.

3.1.2 Swarm Learning Use Cases in HumAIne

Swarm learning is applied in the manufacturing and oncology pilot. The manufacturing pilot focuses on scheduling in a machine-as-a-service setting. The multi agent reinforcement learning scheduling algorithm is enhanced by a swarm learning component to ensure confidentiality of different companies while efficiently optimizing the production schedule. In this vertically federated reinforcement learning system the RL-agents observe distinct parts of a common environment.

The oncology pilot addresses the problem of creating structured reports from an unstructured text using natural language processing (NLP). The goal is to increase the model accuracy, by sharing knowledge between the hospitals without sharing the raw data. This use case unveils a horizontal federated learning problem where the federated data shares the same feature space but a distinct

Funded by the European Union

sample space. A more elaborate description of the swarm learning use cases can be found in D2.2 Requirements Analysis, Reference Scenarios, Applicable Standards, and Regulations V2.

3.1.3 Development progress and KPIs

Table 3. *KPIs related to Swarm Learning.*

KPI	M18	M36	Current
4.1 SL Platform to be implemented	Initial version implemented	Final version integrated with HumAIne OS	Initial version implemented
4.2 Forecast accuracy based on Swarm Intelligence		Average increase >=50% compared to traditional ML baselines	~10% in the manufacturing pilot on small scale instance
4.3 Required time for model training/convergence		Average decrease >=25% compared to traditional ML baselines	>=25% in the manufacturing pilot

The above table gives an overview of the defined KPI concerning the swarm learning paradigm in the HumAIne project. A first stand-alone swarm learning platform has been developed. The swarm learning technology was tested in the manufacturing pilot. First results achieve an increase of the accuracy of ~10% compared to decentralized training without swarm learning. By increasing the problem instance, we expect to achieve the proposed accuracy gain of >=50% until the end of the project. The swarm learning component showed superior scalability properties in the manufacturing pilot. Compared to the humaine agnostic scheduling solution which used a centralized training algorithm the training time could be decrease by >=25%.

3.2 Implementation

The research on federated learning frameworks indicated NVIDIA FLARE[5] (NVFlare) as promising candidate for the foundation of the HumAIne swarm learning platform. An introduction of the federated learning frameworks and its discussion can be found in D2.3 Reference Architecture and Specifications V1. While NVFlare comes with a comprehensive set of federated learning algorithms as well as workflows, it is easily extendible. Furthermore, NVFlare can be integrated into existing real-world systems and is agnostic to the application domain as well as the already used machine learning framework.

3.2.1 Client Integration

In both pilot cases there are existing machine learning systems. Hence, the goal is to extend these systems by a swarm learning component which is in charge of the secure exchange and aggregation of the machine learning model, i.e. the knowledge of the distributed clients. The existing system can be divided into a learning algorithm and a data component. While in the oncology pilot this data component is a data base, in the manufacturing pilot the data used to train the scheduling algorithm is created by trial and error, i.e. the data is created within the factory or a factory simulation.

The swarm learning component is attached to the learning algorithm of the prevailing system. It uncouples the data from the swarm learning component. By only forwarding knowledge in form of a

[5] https://github.com/NVIDIA/NVFlare

Funded by the European Union

machine learning model and not raw data to the swarm learning component the privacy of the data is ensured. The swarm learning component is composed of one swarm learning server and one swarm learning client for each machine learning client. Each swarm learning client is connected to one machine learning client and handles all swarm learning related tasks for one entity. The swarm learning server connects all swarm learning clients and coordinates them. The architecture is illustrated in Figure 24. and in Figure 25.

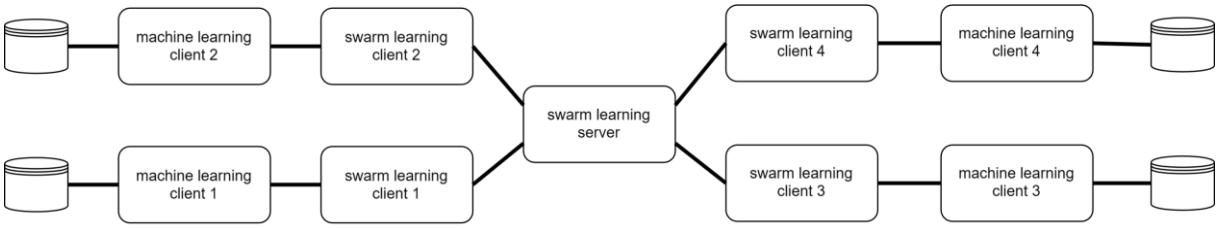

Figure 24. *Object diagram demonstrating the architecture of the system with four clients*

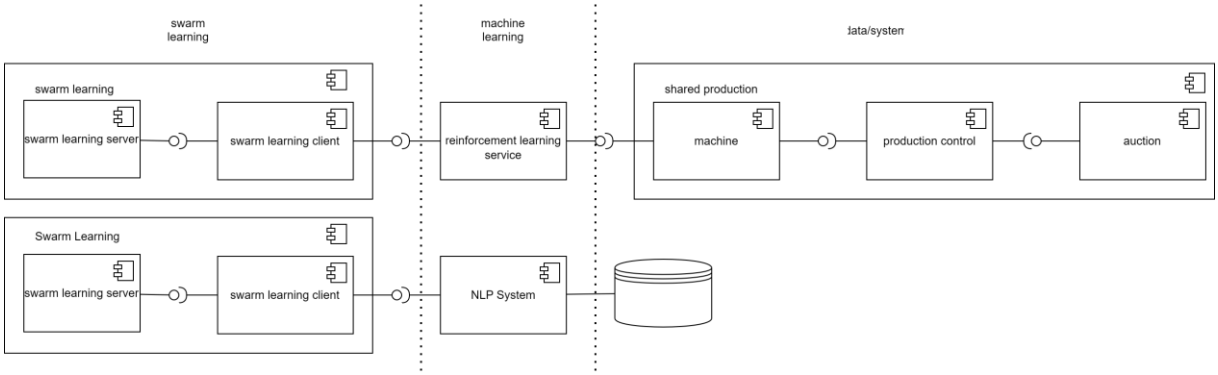

Figure 25. *Integrating the swarm learning component using a multi-tier architecture.*

A Flare Agent is integrated in the machine learning component. This is the attachment point to the swarm learning client. The Flare Agent is start at the beginning of the ML algorithm. After a specified amount of training iterations, the model weights are aggregated. In the manufacturing pilot this is done after 15 iterations.

3.2.2 Swarm Synchronization

Swarm learning is done in multiple rounds where the local training alternates with the synchronization of the local models. The swarm server handles connection to the client nodes and the overall organization of the swarm learning task, i.e. it monitors the number of synchronization rounds executed. The swarm clients are connected via a peer-to-peer connection. In each round one client is selected for the aggregation of the model weights. All clients send the model weights to this client. The client aggregates the weights using FedAvg (McMahan, Moore, Ramage, Hampson, & Agüera y Arcas, 2016) and distributes the updates back to the swarm clients.

3.2.3 Inference using Swarm Agents

To enhance accessibility and usability, HumAIne platform will include a modular and scalable service of swarm agents to integrate seamlessly with diverse applications and use cases. The architecture of this service is depicted in the diagram below and focuses on flexibility and human-centric interaction.

The platform comprises a **Swarm Manager** that orchestrates the interaction between multiple agents, external (trained) models, and users. It serves as the central orchestrator, managing incoming user queries and coordinating agent responses. The Swarm Manager relies on a System Prompt to

Funded by the European Union

standardize agent interactions, ensuring consistent response behaviour and maintaining contextual relevance across multiple use cases. It offers two primary access pathways:

1. **Direct API Access**: Supports programmatic integration for developers and HumAIne systems.
2. **Chat User Interface (Chat UI)**: Provides a conversational interface enabling natural language interaction with the platform.

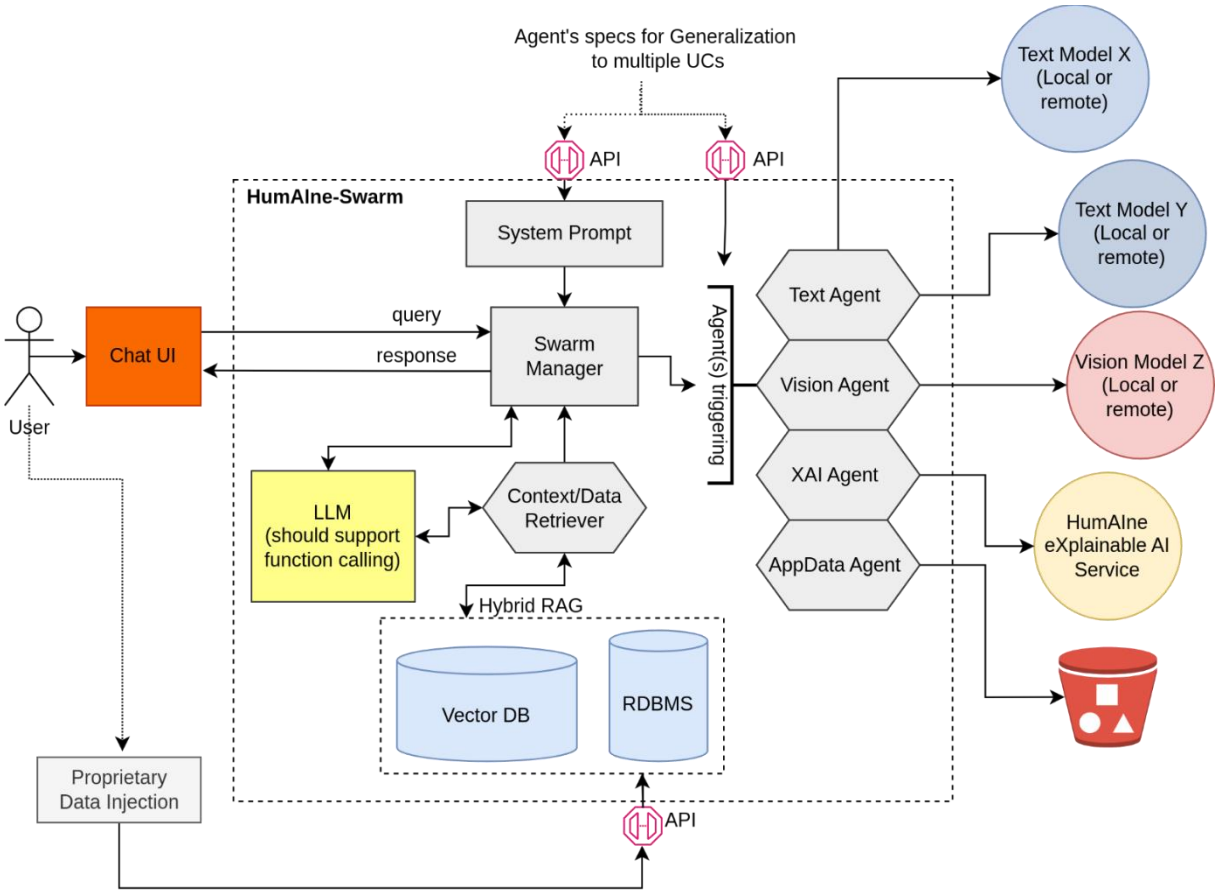

Figure 26. *Swarm Agents Orchestration*

The supported functionalities will be served by LLM-powered **Agents**. The architecture supports a modular agent design, allowing for seamless addition of functionality. Indicative agents include:

- **Text Agent**: Interfaces with local or remote text models for natural language processing tasks such as Named Entity Recognition (NER) and sentiment analysis.
- **Vision Agent**: Engages vision models for image-related tasks like classification and object detection.
- **XAI Agent**: Communicates with HumAIne's Explainable AI services to provide transparent and interpretable outputs.
- **AppData Agent**: Facilitates interaction with application-specific data models and services available at HumAIne's platform central repository (i.e., MinIO).

The **Context/Data Retriever** will implement Retrieval-Augmented Generation (RAG) to enhance response quality and context by incorporating proprietary domain knowledge. It retrieves data from a Vector Database and/or Relational Database to provide personalized, contextually aware outputs using data beyond the training of the LLM.

A **LLM** (e.g., OpenAI GPT-4) accessed via an API will interpret user queries, guide agent orchestration, and support advanced reasoning tasks by synthesizing the agents' responses. The LLM will support

Funded by the European Union

function calling for structured and precise agent interaction. The workflow for handling a user query is illustrated below.

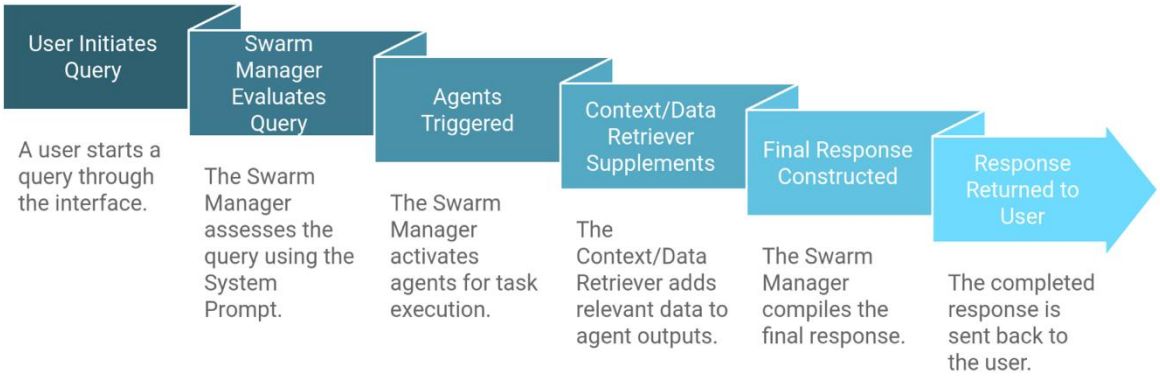

Figure 27. *Example Workflow of the LLM-powered Swarm Agents*

The implementation of this service will be based on the following technologies and frameworks:

- **FastAPI** for building RESTful APIs.
- **OpenAI GPT-4** for conversational and reasoning capabilities.
- **Chainlit** and **LangChain** for orchestrating agent interactions.
- **Qdrant Vector Database** for efficient data retrieval and management.
- **HumAIne Platform (WP5)** and **AI systems (WP3)** for the definition of the agents.

The implementation of this service will enhance the accessibility and usability of HumAIne's AI models and services while offering the flexibility required to support diverse integration scenarios. By harnessing the reasoning capabilities of LLMs, this approach ensures a human-centered design, enabling transparent, intuitive, and natural interactions with the project's AI systems.

3.3 Experimental Results

Swarm learning results have been conducted for two different setting in the manufacturing pilot. Both settings involve four machines. In the first setting two machines belong to one company (see Figure 28.). In the second setting each machine belongs to a different company (see Figure 29.).

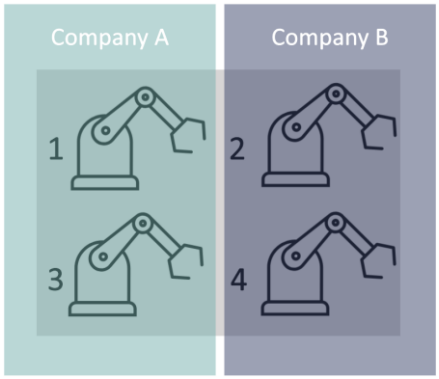

Figure 28. *First experimental setting with two companies and four machines.*

Funded by the European Union

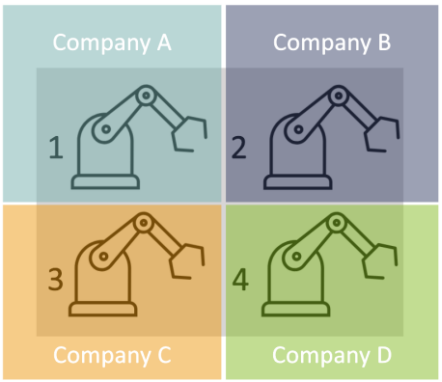

Figure 29. *Second experimental setting with four companies and four machines.*

Machines which belong to the same company share their data. Hence, they are connected to the same reinforcement learning component in the experiment. The interaction between the components is illustrated in Figure 30.

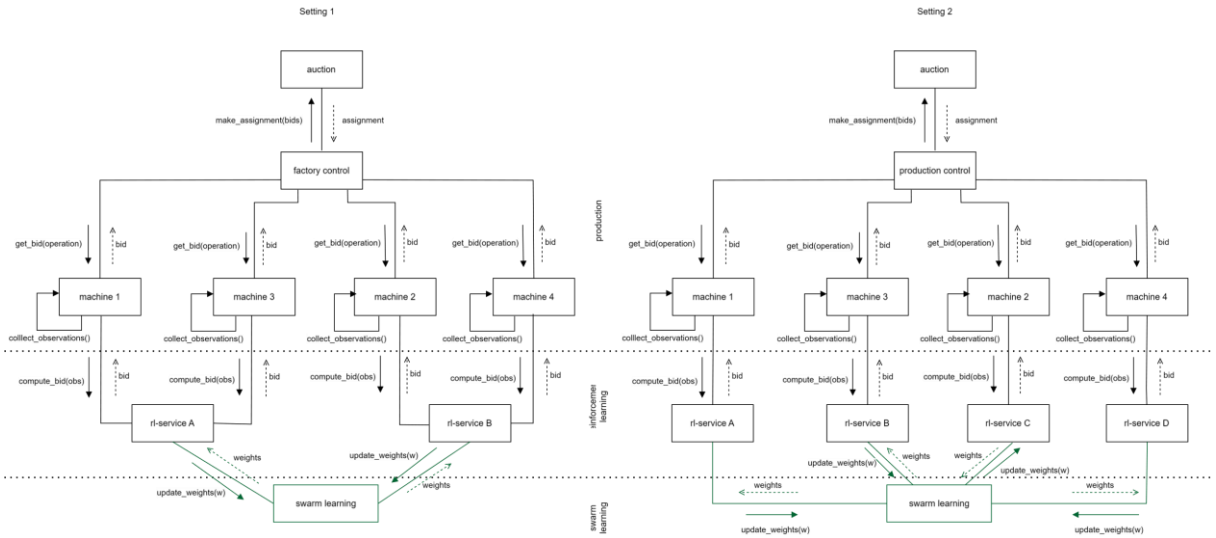

Figure 30. *Interaction of the diagrams in the different experimental settings.*

The data collected in the training episode is distributed to more rl servers in setting 2 than in setting 1. Hence, the amount of samples collected by one rl server per episode is reduced. To achieve the same information gain within the same amount of time, setting 2 uses a training batch size of 1000 compared to setting 1 where a training batch size of 2000 is used.

The problem includes 15 jobs each one with 5 operations which must be scheduled. The objective is to minimize the make span. Figure 31. and Figure 32. show the progress of the reward, resp. the make span over the training time for the experimental setting 1. Figure 33. and Figure 34. depict the reward resp. the make span for the second experimental setting. In all cases does the training converges. In setting 1 the training converges to a reward of 0.5 and make span of 1130 independently of whether swarm learning is used or not. However, if swarm learning is used during training, the training curve is smoother and converges faster.

Funded by the European Union

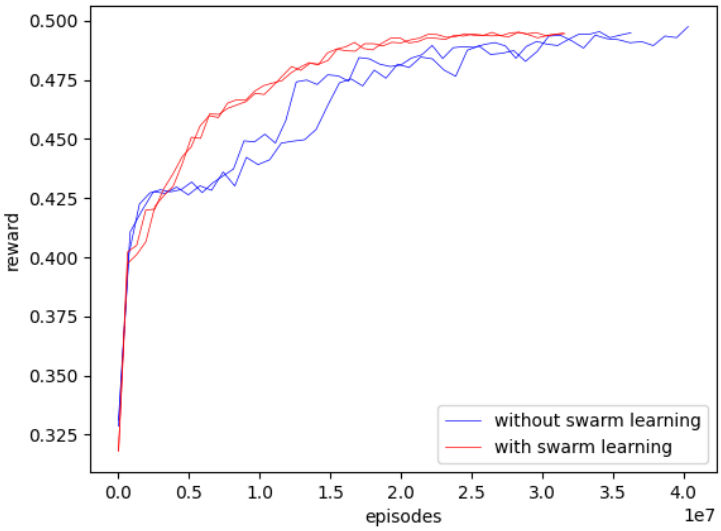

Figure 31. *Improvement of the reward over the training in the second experimental setup with two companies.*

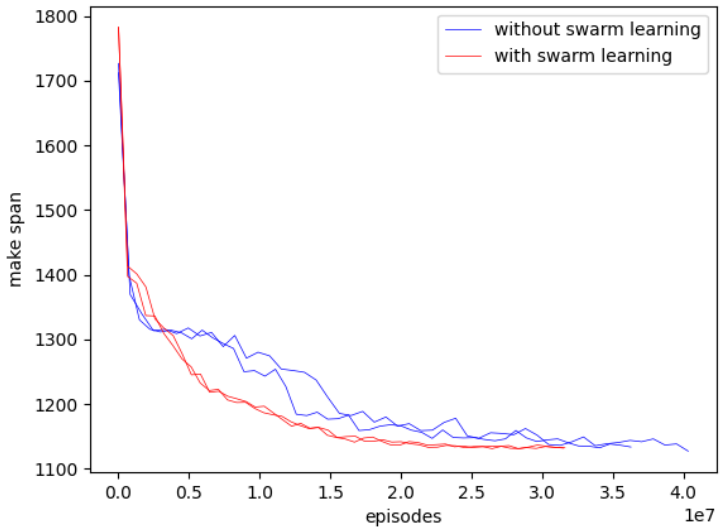

Figure 32. *Improvement of the make span over the training in the first experimental setup with two companies.*

In setting 2 swarm learning achieves a reward of 0.5 while only a reward of 0.46 is achieved without swarm learning. This results in an improvement of 0.04.

Funded by the European Union

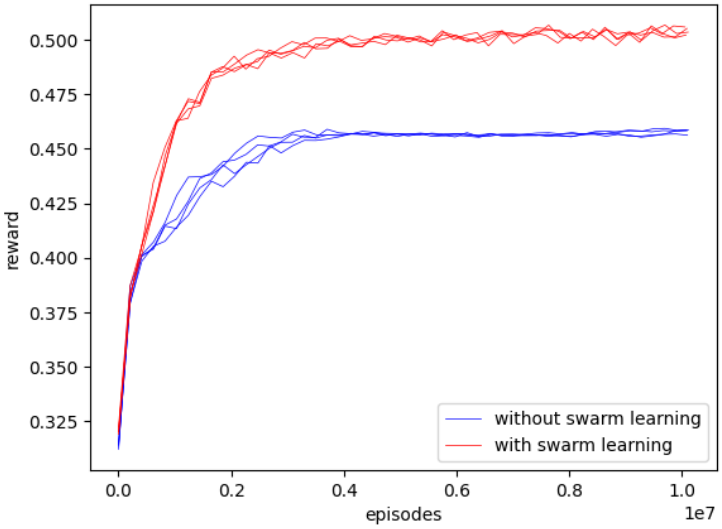

Figure 33. *Improvement of the reward over the training in the second experimental setup with four companies.*

The make span improves by 110 from 1220 to 1110 when swarm learning is used.

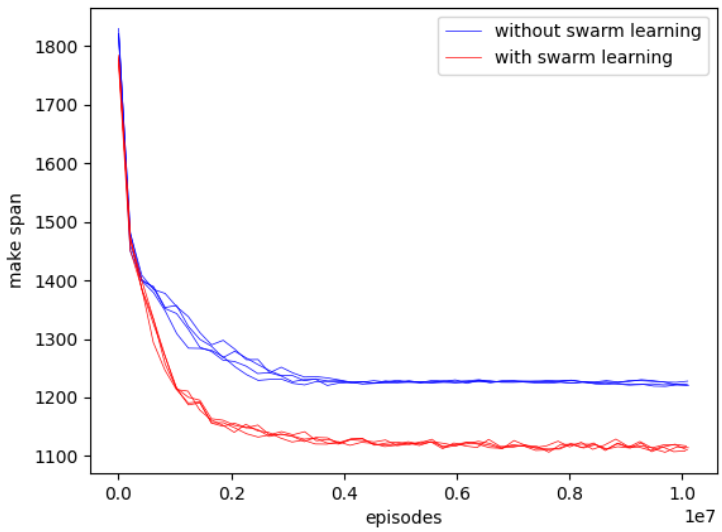

Figure 34. *Improvement of the make span over the training in the second experimental setup with four companies.*

Setting 1 converges at around 30 000 000 episodes while setting 2 converges at around 5 000 000 episodes. Hence, by increasing the number of companies the number of samples can be decreased by 83%.

3.4 Discussion

The experiments which have been conducted so far demonstrate the potential of swarm learning. Swarm learning preserves the privacy between distributed training data sources. To each data source one machine learning (ML) component is attached. The ML components exchange their machine learning models. In this way only knowledge and no raw data is shared between the different entities.

While there is no significant performance benefit of using swarm learning in setting 1, the benefit of swarm learning can be recognized by comparing the results of both settings. By increasing the problem instance from two swarm learning entities to four a gain in the objective as well as in the sample complexity can be observed. This demonstrates the scalability of the proposed approach. One can expect that the gain of swarm learning increases with the number of participating entities. Hence, by

Funded by the European Union

further increasing the problem instance, we expect to satisfy the KPIs of the project month 36 stated in Section 3.1.3.

The usage of the synthetic data in the manufacturing pilot allowed fast feedback in the development and examination of HumAInes swarm learning approach. In the next step this approach can be extended to the oncology use case and deployed to the testbed in the manufacturing use case.

3.5 References (Swarm Learning)

- Chester Chen, H. R., & Flores, M. (2024). Turning Machine Learning to Federated Learning in Minutes with NVIDIA FLARE 2.4. Turning Machine Learning to Federated Learning in Minutes with NVIDIA FLARE 2.4. Retrieved from https://developer.nvidia.com/blog/turning-machine-learning-to-federated-learning-in-minutes-with-nvidia-flare-2-4/
- Han, J., Ma, Y., & Han, Y. (2022). Demystifying Swarm Learning: A New Paradigm of Blockchain-based Decentralized Federated Learning. CoRR, abs/2201.05286. doi:10.48550/arxiv.2201.05286
- Luo, Y., Shi, Y., & Cai, N. (2020). Hybrid systems and multi-energy networks for the future energy Internet. Academic Press.
- McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2016, February). Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv e-prints, arXiv:1602.05629. doi:10.48550/arXiv.1602.05629
- Wang, Y., Tian, Z., Fan, X., Cai, Z., Nowzari, C., & Zeng, K. (2024, November). Distributed Swarm Learning for Edge Internet of Things. IEEE Communications Magazine, 62, 160–166. doi:10.1109/mcom.003.2300610
- Warnat-Herresthal, S., Schultze, H., Shastry, K. L., Manamohan, S., Mukherjee, S., Garg, V., . . . others. (2021). Swarm learning for decentralized and confidential clinical machine learning. Nature, 594, 265–270.

Funded by the European Union

4 NeuroSymbolic AI

4.1 Introduction

This section offers an overview of neuro-symbolic AI, identified as T4.4, highlighting its advancements within the corresponding research pillar. Neuro-symbolic AI represents a hybrid approach to artificial intelligence (AI), combining neural networks and symbolic reasoning systems. Neural networks, known for their data-driven learning capabilities, excel at identifying patterns and relationships within datasets. Symbolic representations, on the other hand, use knowledge through structured rules, logic, or ontologies, enabling reasoning and interpretability. By integrating these methodologies, neuro-symbolic AI aims to leverage the strengths of neural networks' adaptability and the precision of symbolic reasoning.

This learning paradigm addresses important challenges in AI, including trust, transparency, and scalability. Traditional data-driven AI models often face limitations in explainability and dependability, particularly in high-stakes applications like healthcare. Neuro-symbolic AI, by incorporating explicit reasoning processes and grounding decisions in structured knowledge, holds promise for bridging these gaps. It represents a step toward more robust and ethically responsible AI systems, where users can trace the rationale behind outcomes, fostering confidence and facilitating regulatory compliance. One of the advantages of neuro-symbolic AI lies in its potential to improve the trustworthiness and explainability of AI models. By incorporating expert knowledge into the learning process, these systems ground their reasoning in well-established, trusted domain expertise. This capability enhances the reliability of the AI's decisions. Furthermore, the explainability is augmented, as neuro-symbolic systems can offer insights into their decision-making processes based on domain-specific knowledge, enabling users to understand the rationale behind an AI's outputs. This dual focus on learning from data and leveraging structured knowledge distinguishes neuro-symbolic AI as a uniquely balanced approach in advancing AI methodologies.

Currently, neuro-symbolic AI is being applied to the healthcare case, specifically focusing on diabetes care. Diabetes is a chronic condition requiring continuous monitoring, treatment adjustments, and patient education to manage effectively. This application uses two data sources: expert knowledge from medical professionals and real-world data from diabetes patients. By integrating these diverse inputs, the neuro-symbolic AI model seeks to address the complexities of personalized diabetes care, enabling a nuanced understanding of patient needs while ensuring adherence to medical guidelines. The objective is to improve remote care for diabetes patients by providing AI-generated advice grounded in both patient data and expert insights. Such integration allows for tailored recommendations that adapt to individual patient profiles, potentially improving health outcomes and reducing the burden on healthcare providers. The model is designed with a human-in-the-loop mechanism. Any advice generated by the AI must be reviewed and approved by a medical doctor before it is communicated to the patient, ensuring that the final decision rests with the healthcare professional. This approach not only ensures safety and compliance with medical standards but also addresses ethical concerns about fully autonomous decision-making in sensitive domains like healthcare. The collaboration between AI systems and human expertise exemplifies how neuro-symbolic AI can augment human capabilities rather than replace them.

To provide an understanding of neuro-symbolic AI and its application, this section is organized as follows: Section 2 provides a theoretical foundation for neuro-symbolic AI, outlining relevant concepts and methodologies. Section 3 details the data, including the types of data and their relevance to the project. Section 4 explores the architecture and design of the neuro-symbolic AI models, explaining how the components integrate neural and symbolic elements. Section 5 demonstrates the application of these models in the context of diabetes care, illustrating how they address specific challenges in this

Funded by the European Union

domain. Section 6 presents preliminary results achieved over recent months, showcasing the progress and findings. Finally, Section 7 offers a summary of the main results obtained.

4.2 Background on Neuro-Symbolic AI

The aim of NeuroSymbolic AI [31] is to fuse two existing branches of AI, namely Symbolic AI (or symbolism) and Statistical Machine Learning (or connectionism), so as to combine the benefits of both approaches into the next generation of AI [17]. Symbolic AI relies on hand-crafted rules expressed through Logic Formulas and Ontologies, while Statistical Machine Learning is mainly characterized by neural networks that learn from data. As Symbolic AI was one of the earliest approaches used in building AI systems it is no surprise that it falls short on many aspects, as it requires significant effort from domain experts to gather and codify the symbolic knowledge consisting of entities, relationships and rules governing those relationships. While, after this effort making automated decisions is fast and straightforward, the resulting systems remain inflexible as they are intolerant of ambiguous or noisy data, which is very frequent in real world problems. On the other hand, bottom-up statistical approaches, such as (Deep) Neural Networks, deal with these problems quite well having found substantial real-world application, most notably in the domains of Computer Vision and Large Language Processing. Nevertheless, they come with their own set of issues, such as opaqueness to their inner workings and therefore lack of explainability and trustworthiness, lack of robustness to adversarial attacks and unknown inputs [19][27], as well as data inefficiency, which often becomes exacerbated in real-world problem that additionally suffer from data imbalances [28]. Finally another aspect in which Statistical Machine Learning methods are lacking is the ability of compositional generalization, i.e., the ability to solve a problem composed of smaller problems whose solutions already exist [9]. This aspect is particularly important in the progress toward Artificial General Intelligence (AGI), which is the ultimate goal of AI research.

4.2.1 Concepts and Definitions

The motivation behind NeuroSymbolic AI is to address the aforementioned limitations of current AI techniques and to combine the strengths of both approaches to achieve better performance, interpretability, and safety in AI systems. However, NeuroSymbolic AI is actually not that new as already from the 1990s researchers began exploring the integration of symbolic reasoning with connectionist learning. One of the early approaches was the compilation of hand-coded symbolic rules into a neural network, which can be further corrected by empirical learning [29]. Also recently, there has been a number of successful approaches that include the use of Monte Carlo tree search planning procedures aided by DNNs for discovering faster matrix multiplication algorithms [8], the development of the Neuro-Symbolic Concept Learner (NS-CL) [20] and Generative Neurosymbolic Machines (GNM) [13]. Neurosymbolic AI is also closely connected with the ”fast and slow thinking” theory proposed by D. Kahneman [14], which explains the machinery of human thought and motivated recent research interest in NeuroSymbolic AI. Kahneman argued that humans’ decisions are supported by the cooperation of two different kinds of capabilities, called system 1 (fast, effortless and intuitive thinking – trained through repetition and familiarity) and system 2 (slow thinking – deliberative and effort-consuming but better at tackling novel problems). NeuroSymbolic AI aims to provide a unifying view of these two systems, advance the modeling of cognition and further behavior, and build preferable computational methodologies for integrated machine learning and logical reasoning. While there exist many taxonomies of NS AI methods, the most notable and extensive one being [16], the two categories we consider most fundamental are the ones described in [36], namely Learning for Reasoning and Reasoning for Learning. The first group of methods are extensions of existing symbolic reasoning methods that utilize empirical machine learning either to make sense of unstructured data or to speed up their reasoning process. For instance, NeuroSymbolic Concept Learner (NS-CL) [20] uses a CNN-based visual perception module followed by a semantic parsing module, and a symbolic reasoning module to make sense of visual scenes. On the other hand, approaches such extending Markov Logic Networks such as [21] and [22] use statistical machine learning methods to automate the building of

Funded by the European Union

logical rules in a data-driven manner, while to accomplish the same task [35] used a differentiable version of Inductive Logic Programming. IBM toolkit's Neural Unification for Logic Reasoning over Natural Language [25] uses transformers to help detect logical contradictions between a natural language corpus and a natural language query. The other group, Reasoning for Learning, uses neural classifiers as the basis for learning, that are assisted through the incorporation of symbolic knowledge, either in the form of constraining/regularization or in the form of knowledge transfer. Knowledge transfer approaches usually use Knowledge Graphs out of which they extract vector embeddings using Graph Neural Networks, which are then combined with embeddings extracted from a problem-specific Neural Networks. This approach has found many applications in few-shot and zero-shot computer vision (e.g., Zero-shot Recognition via Semantic Embeddings (SEKB-ZSL)[32] and Dense Graph Propagation Module (DGP) [15]). However there are also Reasoning for Learning methods that focus on AI problems where data is abundant, but where symbolic constraints need to be adhered to for reasons of trustworthiness, fairness or safety, these are the methods based on regularization. There are four main such approaches namely Logic Tensor Networks (LTN) [3], Symbolic Probabilistic Layer (SPL) [1], Neural Attention for NeuroSymbolic Reasoning (NASR) [6], Signal Temporal Logic Network (STLNet) [18].

4.2.2 Existing Solutions and Technologies

Logic Tensor Networks (LTN) is one of the most established loss-based regularization methods. LTNs use grounding, a technique that maps first-order logic propositions to real-valued tensors and corresponding mathematical operations. These tensors have to be of different sizes depending on the input datatype and their elements are between 0 and 1 corresponding to their truth value (similar to fuzzy logic). The end result of this process is a real-valued equation of tensor variables (these depend on the algorithm inputs or on features of the inputs) whose result is the degree of truth of the initial logical proposition. This new equation is differential and can be used as a term in the loss function that will guide weight update in a Neural Network during back-propagation. LTNs have been used in a variety of real-life domains such as manufacturing [24] and maintain high accuracy also guaranteeing a high degree of satisfiability of the constraints (however complete satisfiability is not guaranteed due to the fuzzy logic process) as well as lower sample complexity. Semantic Probabilistic Layer (SPL) introduces a fully independent layer that can be added on top of an existing network architecture (e.g., Resnet50) that enforces the external logical constraints. In this layer simple logical formulas are encoded as Ordered Binary Decision Diagrams (OBDDs) which in turn are transformed to differentiable Probabilistic Circuits (PCs). It is important to note that although practically this transformation happens fast, it can in the worst case be exponential. The incorporation of this final layer leads to a readjustment of the conversion of logits to probabilities so that, for instance, prohibitive logical constraints output a pseudo-probability of 0, while the rest of the probability mass in readjusted. SPL guarantees consistency strictly and has low sample complexity, however it only works with simple logical propositions and does not incorporate first-order logic. Neural Attention for Symbolic Reasoning (NASR) is built upon the idea that neural networks are responsible for both perception and reasoning, but logical reasoning is used to fill in inconsistencies. More specifically it uses a Mask-Predictor module that identifies edges in a graph of symbolic relationships (e.g., indicating how objects in an image are spaced) that are most likely to be predicted wrongly and those are then recomputed using symbolic rules. While the approach is designed in a very modular manner, it consists of many different parts leading to complex interactions and is also not yet tested on a variety of datasets. Finally, STLNet (Signal Temporal Logic Network) constrains the underlying model using temporal logic properties using the Student-Teacher paradigm. Specifically, the teacher networks tries to find a constraint-adhering solution that is as close as possible to the prediction of the student network p, which is the given the feedback through an appropriate loss function. In fact both the teacher and student predictions can be used providing strict and fuzzy constraint satisfaction guarantees respectively.

Funded by the European Union

4.2.3 Applications of NSAI

In general, NeuroSymbolic AI methods and more specifically Reasoning for Learning Methods can have a large impact in the practical applications of Neural Networks as they enable their safer use making sure the network adheres to contstraints that have been pre-defined by experts. This also increases the trustworthiness of the model as well as human control over their behaviours which could manage to increase AI adoption in critical domains such healthcare, finance and manufacturing. Specifically in the domain of defect detection, neurosymbolic AI has been used to improve transparency and explainability in cantilever beam defect detection [24] and to drive diagnosis of automotive production faults [4]. In [10] convolutional neural networks perform localization and recognition on video inputs gathered from a real-life food product labelling production lines and their predictions are used by a knowledge-base-aided symbolic component to support decision making over the state of the production system. In [26] the authors create a knowledge base of rules to be satisfied through the training of a LTN for bearing fault diagnosis. The datasets used consisted of vibration signals collected from the bearings that included different sizes and positions of faults. To formulate the knowledge base of rules additional features were extracted such as the signal Kurtosis, as well as the occurrence of peaks at frequencies corresponding to faults occurring in different locations of the underlying machine. These were then compared with thresholds determined by experts to form the final rules. An interesting alteration of the original LTN framework is that different axiom groups were given different weights in the aggregate satisfaction (LTN loss) of the network. These weights were modulated throughout the training process with a pre-specified schedule. Finally, LTNs with different underlying architectures and parameters (e.g., weight update schedules) were compared along with their non-neurosymbolic equivalents across different sampling ratios of the original dataset, showing an advantage for NS models due to the addition of the expert rules. A similar application was developed in [5] for recommender systems, using expert knowledge to make up for the scarcity of data. The authors experimented on the MindReader dataset of movie ratings and built a knowledge base encoding common sense knowledge to assist the underlying gradient-decent-driven Matrix Factorization (MF) algorithm with training. For example the rule that ”if a user likes a certain genre and a movie is of that genre then the user will likely like the movie” was used. The final LTN/MF method was evaluated in different scenarios simulating data scarcity by the sampling 20-100% of the original dataset. The NS method achieved slightly higher performance that was more pronounced in the lower sampling ratios. Aspect-based sentiment analysis, i.e. the determination of people’s sentiment at a more fine-grained level as specifically related to a specific aspect of a topic or product, has seen another application of LTNs in [12]. The approach proposed was evaluated on product-review-related NLP datasets extracted from Twitter and from the SpABSA dataset (around 4k training samples). It was compared against nonBERT methods mainly based on LSTMs and RNNs and against BERT-based networks. The construction of the knowledge base of rules is based on the determination of the relevance every word in a sentence has to a specific ”aspect” word. This dependency can be determined by a variety of sequential NLP models and can be fused with prior knowledge about word dependencies e.g., from a pre-trained BERT model. Additionally rules extracted from an ”emotions lexicon” were used to determine whether a term has positive negative or neutral emotional connotation. The final combination of rules and data-driven learning outperformed both BERT-based and non-BERT-based baselines. Another example of NSAI background knowledge benefiting established neural methods comes from the field of computer vision and specifically Semantic Image Interpretation [7]. The dataset used for learning was the Pascal-Part dataset, a specially annotated dataset containing animals, vehicles and indoor objects together with bounding boxes identifying them as well as their sub-component parts in the image. To form the expert knowledge, part-of relations were extracted from WordNet related to the objects contained in the dataset and so-called mereological constraint were included (can be viewed as common-sense knowledge about is-part relationships) to guarantee properties such asymmetry of the parts-of relations, ensure that an object consists of a specific list of parts and no other etc. The features of the LTN classifier was information extracted by the Fast-RCNN object detector on top of which the labels of the different objects as well

Funded by the European Union

as the part-of relationships between were learned with the help of the prior symbolic knowledge. In both inference tasks the LTN outperformed the simple Fast-RCNN in terms of AUROC, with the mereological constraints proving especially useful. A couple of similar works have been produced relating to combining logical rules with vision DNNs for deriving meaningful actions by rendering images understandable. The first work is on Sudoku Puzzle classification [23] where the input dataset consists of combinations of symbols from different domains (e.g. MNIST, FashionMNIST etc.) that are organized in a 9x9 grid and the NSAI algorithm needs to decide whether the solution adheres to the rules of sudoku. In this case both the bottom-up aspect of the learning e.g., classifying the images into digits/symbols and the top-down logical part i.e., ensuring that no row, column, box contain the same symbol twice, are very straightforward. Additionally, an attempt has been made to include symbolic rules to the action selected process in vision-based Reinforcement Learning [2]. To that end, a simple dataset containing simplified games scenes was used. In each scene, an agent made decisions with the aim of collecting a series of tokens while avoiding enemies. The logical rules where used to enforce that the agent chooses action ”collect” when near a token and action ”avoid” when near an enemy. The resulting RL algorithm showed improved transferability properties across variable testing environments. Last but not least, aside from being used to improve accuracy performance, LTNs have also been employed in introducing fairness constraints and reducing bias [11]. The model is evaluated on the Adult Income, German Credit Risk and COMPAS (recidivism prediction based on criminal history) datasets from the UCI ML repository and attempts to used logical constraints to mitigate gender and race related biases found in these datasets. The majority of the constraints used aim at ensuring the probability mass of prediction for each sub-class of a sensitive feature is equally distributed between the sub-classes. In the evaluation LTN maintains comparable accuracy performance, while managing to largely debias the dataset.

4.2.4 Gap Analysis

In the context of HumAIne we are planning to expand the state of the art in NeuroSymbolic AI by focusing on the human-interaction aspect, whereby the user will be able to control the system through the addition of symbolic rules as they see fit. The current literature already includes some approaches in this domain. In [30] the authors introduce a two-way interaction framework between user and NeuroSymbolic model, where users are able to query the model to understand its behaviour and also provide feedback to alter or control it. The approach is based on Logic Tensor Networks and validated in two scenarios, one for detecting unfairness and introducing corresponding fairness constraints in datasets used for credit risk monitoring and one for correcting visual predictions based on introducing additional visual concepts and rules for the prediction. In the first step the user queries a trained model regarding a new sample to find out which constraints were satisfied for producing the model output. If the user disagrees with the explanation provided they can add new constraints to the knowledge base (based on the concepts used by the LTN model) to try and correct model behaviour, only having to run limited iterations for training. What is of interest in this approach is the process of extracting concepts. In tabular data used for credit risk detection concepts are decided based on input features such as gender or ethnicity. For visual data, concepts are learned for CNN activations collected when a subset of input images known beforehand to embody a particular concept is passed through the CNN. A similar approach for using symbolic rules to correct model behaviour after the training stage is employed in the metacognition approach [33]. Metacognition is referred to as the reasoning of an agent (or AI model) about its own internal processes. The framework proposed in the paper is based on Transparency (the AI model explaining it decisions based on its input data and internal parameters), Reasoning (self-aware reasoning, the process by which the explanations of decisions are used to influence future decisions), Adaptation (the ability of the model to adapt based on its own past errors) and Perception (simply the models ability to learn from sensory inputs (e.g., vision, text)). In a relevant work by the same authors [34] learning of error-correcting rules is applied to trajectory optimization. The key idea is the improvement of SOTA deep learning methods through the lightweight learning of rules that predict errors given model training and operation data. Additionally the approach involves

Funded by the European Union

the learning of correction rules once an error is detected that reclassify movement trajectories. The resulting NeuroSymbolic system managed to improve baselines significantly in testing conditions with out-of-distribution inputs. The above approaches point to a fertile direction for further research relevant to the HumAIne project. A variety of ways of querying/distilling information from a model can be used via XAI-related methods to inform the user of the inner workings and - most importantly - the shortcomings of a model on a particular task. Additionally, model errors can be handled in a variety of increasingly complex ways to determine their severity and ways of mitigation also combined with inputs from the user. Symbolic methods, as outlined in the metacognitive framework, can play an important part in this. Finally, LTNs, LNNs, SPL and other techniques that enable the embedding of rules into networks can help with the adaptation of the network to user feedback.

4.3 Data Collection and Description

This section outlines the data provided for the diabetes case, detailing its structure and purpose. It is divided into two subsections. First, an overview is given for the patient data. This subsection presents a summary of the patient-specific information contained in the dataset. Then an overview is given of the expert rules. This subsection explains the expert-driven rules applied in the neuro-symbolic AI model. Currently, the dataset comprises synthetic data supplied by Innovation Sprint, representing a single patient's information over a 14-week period. This dataset serves as a preliminary foundation, providing insights into data availability and enabling the initial exploration of neuro-symbolic AI models. As the study continues to include a greater number of diabetes patients, additional data will be made available by Innovation Sprint. This data is expected to maintain consistency with the structure and format of the current dataset, supporting the ongoing development and refinement of the neuro-symbolic AI models.

4.3.1 Patient data

4.3.1.1 Feature vectors

The current dataset includes synthetic patient data provided by Innovation Sprint. The data contains 14 weeks of data for one diabetes patient. For every day in the 14 weeks, the activity data in Table 4 are reported. The activity data includes data, such as the number of steps set per day. These data by the patients are tracked with a wearable device.

Table 4. *Description of feature vectors.*

Activity data	Description
Steps	The number of steps per day by the patient
Calories	The number of calories that are consumed
Floors	The number of floors the patient climbed per day
Intensity minutes	The number of intensity minutes per day
Sleep	The hours of sleep per day
Weight	The weight of the patient

The data can be processed in several ways. For instance, one approach is to transform the data into feature vectors using bi-weekly sliding windows. This method involves segmenting the data into overlapping two-week intervals, where each interval acts as a discrete window. Each window is then converted into a feature vector, which summarizes the characteristics or key metrics of the data within that specific period. This approach is particularly useful for capturing temporal patterns, trends, or anomalies in the data over time, enabling nuanced analysis and predictive modelling. By applying this technique, we can ensure that the time-dependent relationships and contextual insights are preserved and leveraged.

Funded by the European Union

Moreover, the data includes information about the demographics of the participants. Table 5 provides an overview of the demographic data in the dataset.

Table 5. *Overview of the demographic data.*

Demographics	**Description**
Residence	The participant's place of living,
Family history of DM	Indicates whether the participant has a family history of diabetes mellitus (DM).
DM duration	The length of time since the participant was diagnosed with diabetes.
Occupation	The type of work or employment status of the participant.
Education level	The highest level of education completed by the participant.
Marital status	The participant's marital status, e.g., single, married, divorced, or widowed.
Charlson index score	A score indicating the burden of comorbidities based on the Charlson Comorbidity Index.
Gender	The gender identity of the participant (e.g., male, female, or other).
Age	The participant's age in years.

Also, at the start of the study and end of the study, the participants fill in different questionnaires. Table 6 provides an overview of the questionnaires used. These results in additional data that could be leveraged in the training process.

Table 6. *Overview of the questionnaires used.*

Questionnaire	**Description**
EQ-5D	Measures health-related quality of life across five dimensions: mobility, self-care, usual activities, pain/discomfort, and anxiety/depression.
EQ-VAS	Visual analog scale where respondents rate their overall health from 0 (worst) to 100 (best).
DKQ-24	Assesses knowledge about diabetes management and lifestyle using a 24-item questionnaire.
DMSES	Evaluates self-efficacy in managing diabetes, including confidence in handling various care aspects.
PAID	Measures emotional distress related to diabetes and its management using the Problem Areas in Diabetes scale.
GAD-7	Screens for and assesses the severity of generalized anxiety disorder with seven items.
PHQ-9	A nine-item questionnaire used to screen, diagnose, and monitor the severity of depression.
MET-min/week	Quantifies physical activity in Metabolic Equivalent Task (MET) minutes per week.
IPAQ-SF Category	Categorizes physical activity levels (low, moderate, high) based on responses to the IPAQ-Short Form.
MedDietScore	Reflects adherence to the Mediterranean diet based on key dietary components.
Fagerstrom	Assesses nicotine dependence intensity through the Fagerström Test for Nicotine Dependence.

Funded by the European Union

4.3.2 Target variable

The SMBG value will initially serve as the ground truth for the neuro-symbolic AI models. The self-measured blood glucose (SMBG) measurements are reported regularly by the patients. Each day, the data contains at least one measurement of SMBG. On some days, two measurements have been taken. The different types of SMBG are presented in Table 7.

Table 7. *Different types of SMBG.*

SMBG type	Description
BB	Before breakfast
BL	Before lunch
BD	Before dinner
AB	After breakfast
BL	Before lunch
BD	Before dinner
BT	Bedtime

Figure 35 shows a boxplot of the SMBG value per SMBG type. Per SMBG type, e.g. BB (before breakfast), the range of SMBG values can be observed. As can be seen from the figure, the SMBG values differ per type. In general, the SMBG value is higher after meals than before meals.

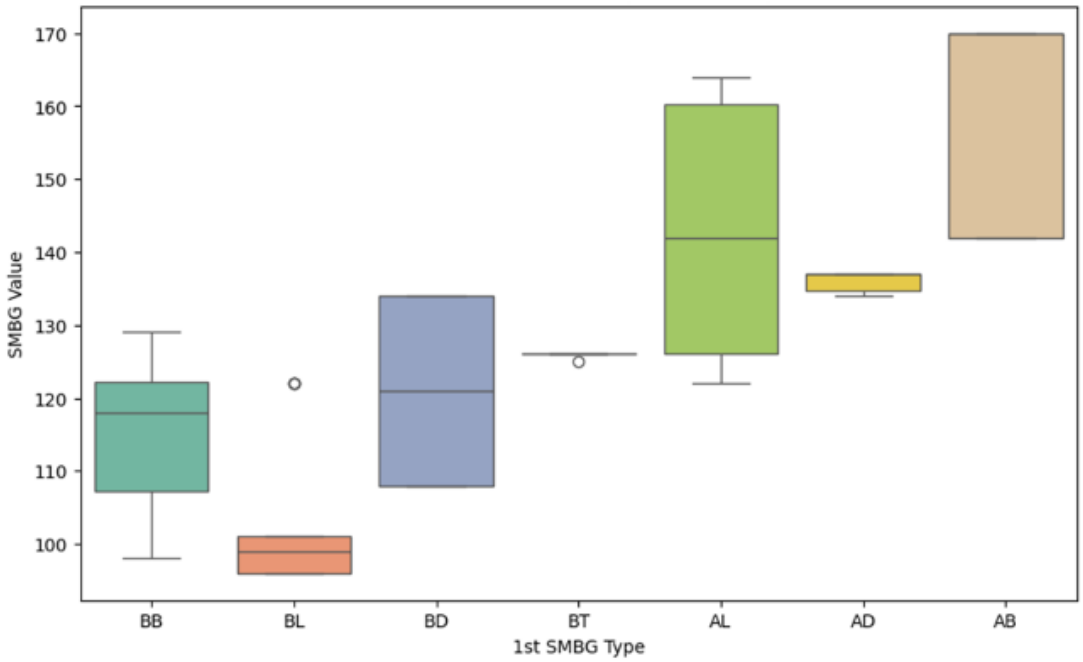

Figure 35. *Boxplot of the SMBG value per SMBG type.*

The minimum and maximum SMBG values are reported in Table 8.

Table 8. *Minimum and maximum SMBG values.*

SMBG type	Minimum value	Maximum value
AB	142	170

Funded by the European Union

AD	134	137
AL	122	164
BB	98	129
BL	96	122
BT	125	126

Multiple model definitions are considered at the current project stage. This includes using classification to predict whether unregulated blood glucose will occur in the next two weeks for the patient, based on the feature vector containing data about the previous two weeks. Regulated blood glucose means that any measurement in the biweekly interval should be at least 90 and at maximum 140 (for measurements before meal) or 180 (for measurements after meal). Unregulated blood glucose means that this is not the case. This can be posed as a binary classification problem. The ground truth can be "0" if unregulated blood glucose does not occur in the next two weeks. In contrast, the ground truth can be "1" if unregulated blood glucose occurs one or more than one time in the next two weeks.

4.4 Expert knowledge

Next to the patient data, expert knowledge is provided by Innovation Sprint. The expert knowledge is provided by the doctors and contains advice for the diabetes patient. Table 9 provides an overview of the expert knowledge. There are two types of expert knowledge. The first type is goals for the patient and relates to their behavior that has a direct effect on their health. The second type is actions and relates to actions that need to be taken by the patient, such as measuring their blood glucose regularly. The expert knowledge will be integrated with the data-driven AI models.

Table 9. *Goals and actions.*

Goals	Walk at least 500 steps more than the baseline established in weeks -2 and -1
	Exercise at least 150 moderately or intensive (counting x2) minutes per week
	Sleep at least 6 but no more than 8 hours every night
	Decrease your weight by at least 5% in the 12 weeks of the program if your BMI is between 25 and 35
	Mediterranean diet score should be as close as possible to 55
Actions	Weigh yourself at least once per week
	Spread as much as possible your SMBG before and after all meals within a week
	Measure your blood glucose at least 7 days per week

The rules relate to several aspects of the patient. The patient should be walking at least 500 steps more than their baseline. We compare this to the median of the steps of the biweekly interval. Every week, the total intensity minutes of the patient should be at least 150. We compare the sum of intensity minutes of the bi-weekly interval with 300. The patient should sleep between 6 and 8 hours. The sleep duration should not be averaged. The experts have informed us that catching up on lost sleep is not beneficial. Instead, the average of the absolute deviations from the respective limit (below 6 or above 8) should be close to zero. Furthermore, at the end of the intervention, a patient with a BMI higher than 25 should have decreased their weight by 5%. Assuming linear improvement, in the bi-weekly

Funded by the European Union

interval of weeks n-1 and n (of the total 12 of the intervention) the average patient weight should be less than baseline * [1 - (n-0.5)/12*0.05]. Also, the patient should eat well: their Mediterranean diet score should be as close as possible to 55. The possible values for this score are between 0-55. These rules serve as expert knowledge and are encoded in the neuro-symbolic AI model.

4.5 System Architecture and Design

4.5.1 Logic tensor networks

By using Neurosymbolic AI, and specifically LTNs, our ambition is to combine the benefits of unsupervised learning methods, such as logical rules, with the specificity of supervised methods. While the former can generalize to any anomalous output, the latter can learn very well how to recognize the particular defects that occur in the training dataset. For instance, in healthcare the problem of detection and classification of a pathology presents very particular challenges, which obstruct its full automation. Expert knowledge about what constitutes a pathology cannot always be fully encoded into clear-cut rules, which is another hindrance to symbolic and Neurosymbolic approaches. However, a Neurosymbolic approach can still benefit from clear-cut, but non-universal cases (e.g., when there are clear indications of a pathology but the expression of these indications through rules cannot be universally applicable due to its many edge-cases). These challenges led us to choose LTNs for this problem. LTNs do not strictly enforce their symbolic constraints, thus allowing their user to be more lax with formulating the symbolic rules. Moreover, the knowledge of clear-cut pathologies can still be leveraged to speed-up training compared to a classical supervised learning algorithm. An important aspect of Logic Tensor Networks is how constraints are transformed to be differentiable and part of the end-to-end training process. This is achieved through a technique called ”grounding” which is very close to fuzzy logics. More specifically each individual proposition or fact is encoded through a multidimensional tensor, which in our case corresponds to vector embeddings extracted from the input images. Predicates can be applied to these tensors in the form of differentiable mathematical functions which can also have learnable parameters such as Artificial Neural Networks. The application of these predicates should yield a real value between 0 and 1 which corresponds to the degree of truth of the predicate applied to one or multiple propositions. Building on top of that, logical operators can be used to combine different predicate results. For example, a logical $a \wedge b$ can now be calculated as $a*b$ and $a \Rightarrow b$ is calculated as b/a if $b < a$ or 1 if $b > a$. Of course there are many different mappings from first-order logic to real operators, many of which are described in detail in the LTN paper [3]. After making the logical propositional differentiable, their degree of satisfaction can be added as a loss function term to be optimized during training.

4.5.2 High-Level Architecture

As mentioned LTNs embed the symbolic rules as part of a neural network’s loss function. This leads to an architecture where the inputs are the empirical data e.g. text, images, tabular, time-series and the expert’s knowledge expressed in the Domain Specific Language of the LTN’s library. We should note here that symbolic knowledge is only part of the training pipeline and not of inference. Once the model is trained in a constrained way by the symbolic rules it can be used as any other traditional Machine Learning model. For other methods such as Semantic Probabilistic Layer, the rules are used to form a probabilistic circuit that then becomes a part of the neural architecture. Still during inference time the SPL does also not need any explicit knowledge of the symbolic rules to exist in its deployment environment since they are implicit in the model. Figure 36 presents a high-level schematic example of the LTN architecture.

Funded by the European Union

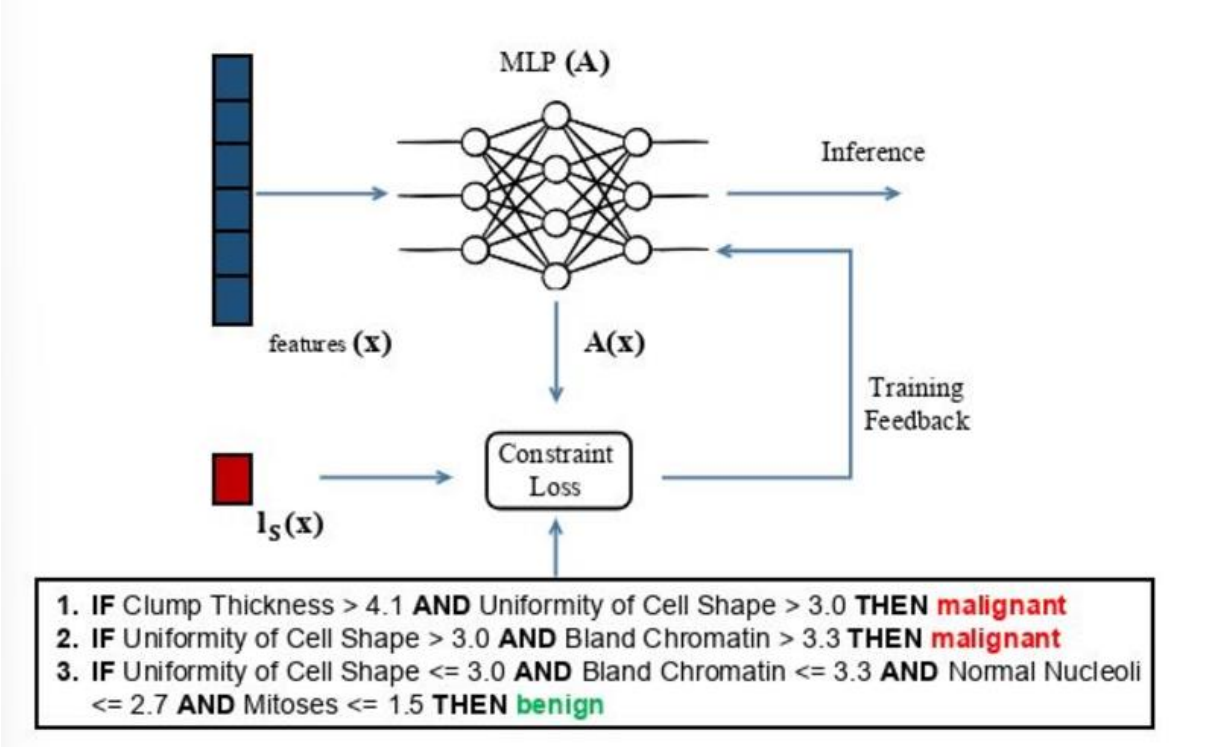

Figure 36. *Example of tabular data, where the pre-processed data vectors are directly input into the network.*

The above diagram shows an example for tabular data where the pre-processed data vectors are directly input into the network. The rules are also straightforward to define in terms that relate to the input feature vectors. It should be noted that during the rules definition, the thresholds need to be normalised in the same way as the feature vectors to have the desired effects. For more complex modalities such as images, we employed a slightly different architecture, where inputs to the LTN are extracted from images using pre-trained embeddings. See Figure 37 below:

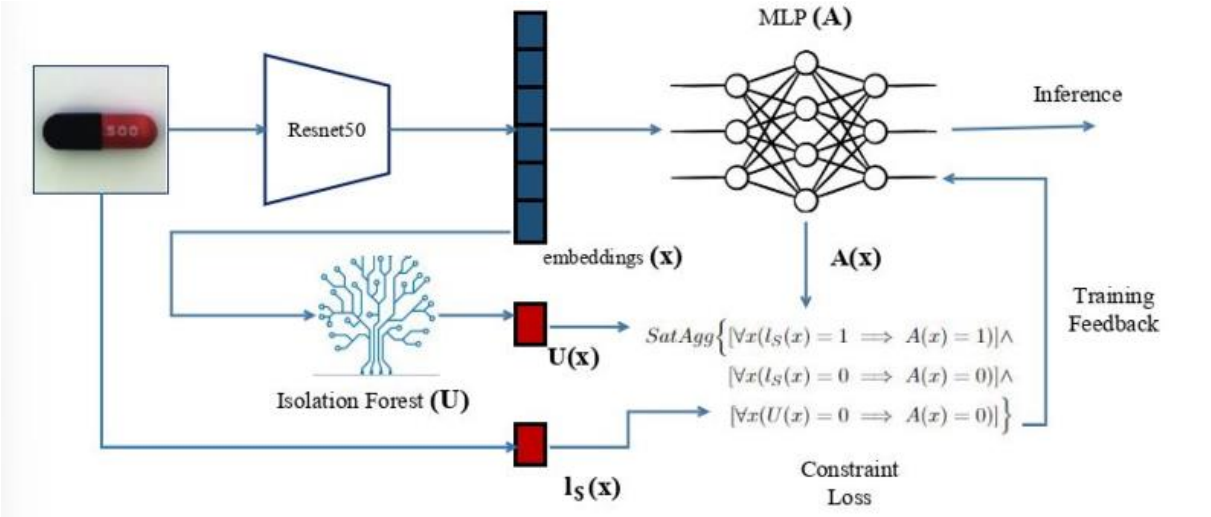

Figure 37. *Example of architecture for more complex modalities.*

Now defining rules that correspond to image features is not that straightforward. A promising approach is to use a concept extractor such as Concept Bottleneck Models and express the reasoning rules in terms of the extracted concepts. For this scenario we use the outputs of an unsupervised anomaly detector (Isolation Forest) which, defined as symbolic rules help the original MLP model be more robust with handling out-of-distribution inputs.

Funded by the European Union

In HumAIne we focus on the ISP pilot which deals with tabular/time-series data, so our overarching concept is similar to the one provided for tabular LTNs, but can use concepts from the images architecture to increase robustness against out-of-distribution inputs or learn better under conditions of reduced or noisy data. The architecture we envision for HumAIne will consist of an offline and online training pipelines.

The offline pipeline (shown below in Figure 38) will have as inputs the empirical data (e.g. feature vectors for the Healthcare Diabetes pilot) and the Expert Knowledge which is envisioned to be provided in text format out of which symbolic rules can be parsed. This parsing step along with the filtering and cleaning of the empirical data will form the pre-processing component. Preprocessed inputs will be passed on to the NeuroSymbolic AI algorithm (LTN) and the trained model will extracted in an interoperable format (e.g. ONNX) to be used both in the ISP infrastructure and the HumAIne platform for inference. Note that the extracted model no longer has any dependency on the expert knowledge and its method of encoding at this stage (to enable easier testing of the model on the ISP infrastructure).

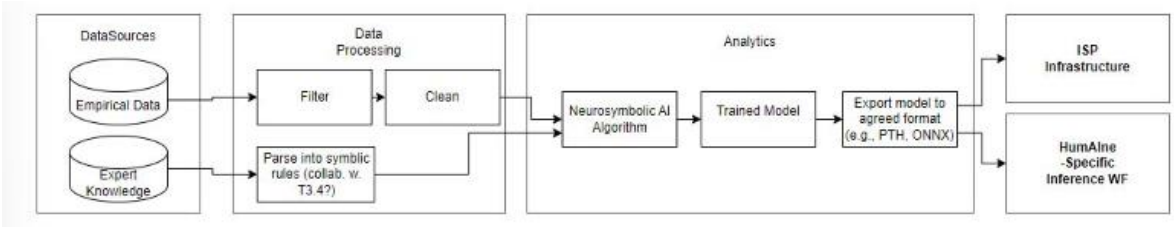

Figure 38. *The offline pipeline.*

The online pipeline (Figure 39) will be hosted in the HumAIne platform and will start by loading the trained model and serving predictions behind a user interface. Through the interface the model will additionally provide explanations on its decisions related both to its neural and its symbolic part (this will have a dependency on the encoded expert knowledge). Explanations will be visualized appropriately so that they are intuitive and easy to understand for non-expert users.

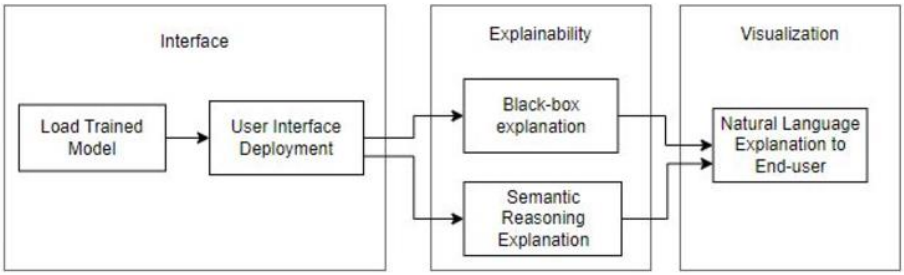

Figure 39. *The online pipeline.*

In the next months of this task we aim to enhance the two workflows by adding a user interaction feature where the (expert) user will be able to dynamically provide their knowledge in natural language text, out of which the symbolic rules guiding LTN training will be provided. A prerequisite for this is that the offline pipeline and especially the part related to model training happens fast enough to enable a usable interactive feedback loop, where the user can add knowledge, observe model behaviour during inference, add further knowledge to improve the model etc.

Funded by the European Union

4.5.3 Technologies

Table 10. *Table of the software used.*

logictensornetworks https://pypi.org/project/ltn/2.0/	Logic Tensor Networks framework written over tensorflow backend
LTNtorch https://pypi.org/project/LTNtorch/	Logic Tensor Networks framework in pytorch - additionally to the tensorflow one needed for adding LTN functionality to imported models using pytorch
Semantic Probabilistic Layer https://github.com/KareemYousrii/SPL	SPL implementation - with data loading and probabilistic circuit architecture altered to fit specific usecases and their corresponding rules
semantic-loss-pytorch https://github.com/lucadiliello/semantic-loss-pytorch	Used to test semantic rule representations in sdd format as needed by SPL, also helpful inclusion of tool for converting cnf to DIMACS logic formats
PySDD https://pysdd.readthedocs.io/en/latest/index.html	Used for obtaining .vtree and .sdd format from logic rules to be input to SPL
tabnet https://pypi.org/project/pytorch-tabnet/	An attention based deep learning model that works well with tabular data (written in pytorch)
outlines https://github.com/dottxt-ai/outlines	A library for extracting LLM responses in a predefined format e.g. json
langchain https://www.langchain.com/	Library for easier working with LLMs

Hardware Requirements:

System with 4 CPU cores, 16GB RAM, and an NVidia K80 GPU (optional - LTN on tabular data can run on CPU only).

4.6 Demo and Use Cases

To showcase the capabilities of NS AI, as well as build the scaffolding for what our application will look like, we build a small demo based on the Breast Cancer Wisconsin dataset from the UCI Machine Learning Repository. Albeit a small dataset both in numbers of samples and features, BCW is not that different from the dataset we have received from ISP regarding diabetic patients.

The UI of the demo consists of an Offline screen, loosely corresponding to the offline pipeline of the component architecture. In this screen (Figure 40) the user can select between different rules that are to be embedded to the LTN model during training.

Funded by the European Union

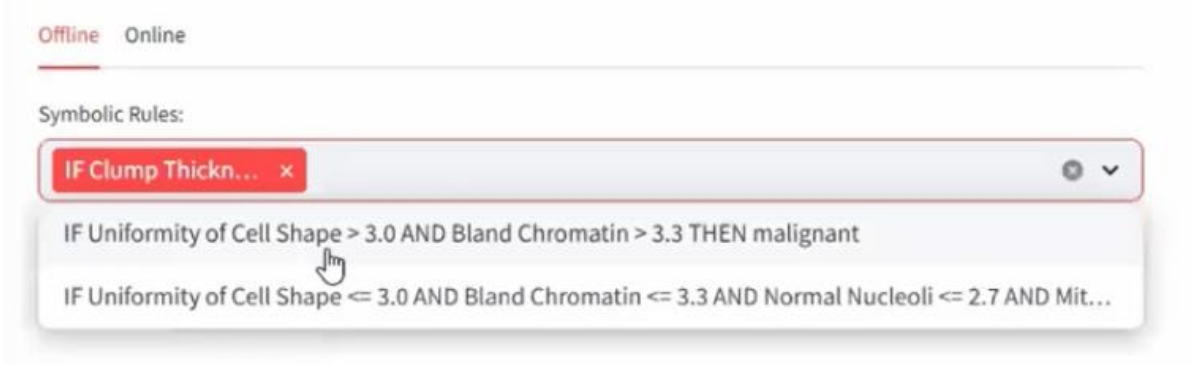

Figure 40. *Choosing symbolic rules.*

The user can also select how to seed the experiment, the amount of training data to use and the number of samples chosen to create an approximate global explainability measure as shown in Figure 41. The data ratio input is there to examine how the NeuroSymbolic model copes with different degrees of data scarcity. The global explainability measure tries to approximate the feature importances resulting from the training processes using the SHAP methods over a sample of instances from the training set.

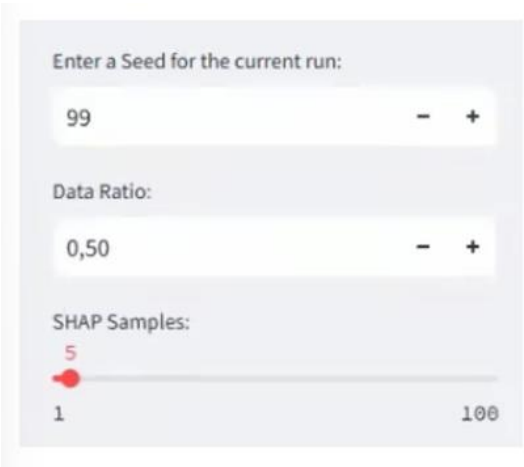

Figure 41. *Configuring SHAP.*

Once the user selects the rules to be used during training, they can click on the "Train Models" button below to initiate the training as shown in Figure 41. After training is complete, various metrics are presented comparing the NeuorSymbolic LTN method (MLP + Rules) against a simple neural network method (MLP) and against the Rules only. Various metrics are given including AUROC, F1 and Recall. Below each metric the improvement is calculated based on the previous method from left to right. In the experiment showcased in the screenshot below we see the NeuroSymbolic method showing a small advantage resulting from the combination of rules and data-driven learning generalising slightly better on the test set.

Funded by the European Union

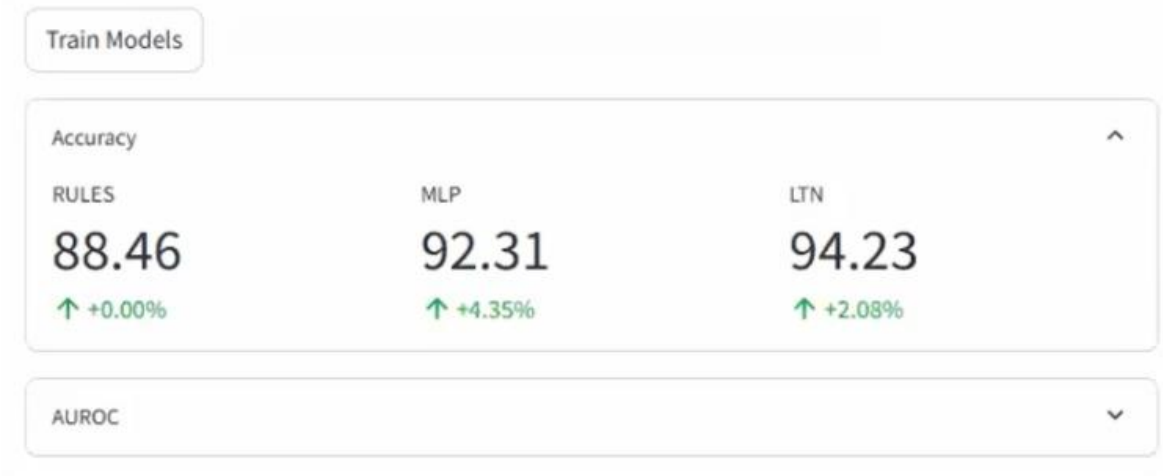

Figure 42. *Modeling results.*

The second screen, shown in Figure 43, of the demo UI showcases what loosely corresponds to the inference pipeline of the architecture. Here the user can select a random patient from the test set and view various results from the trained models.

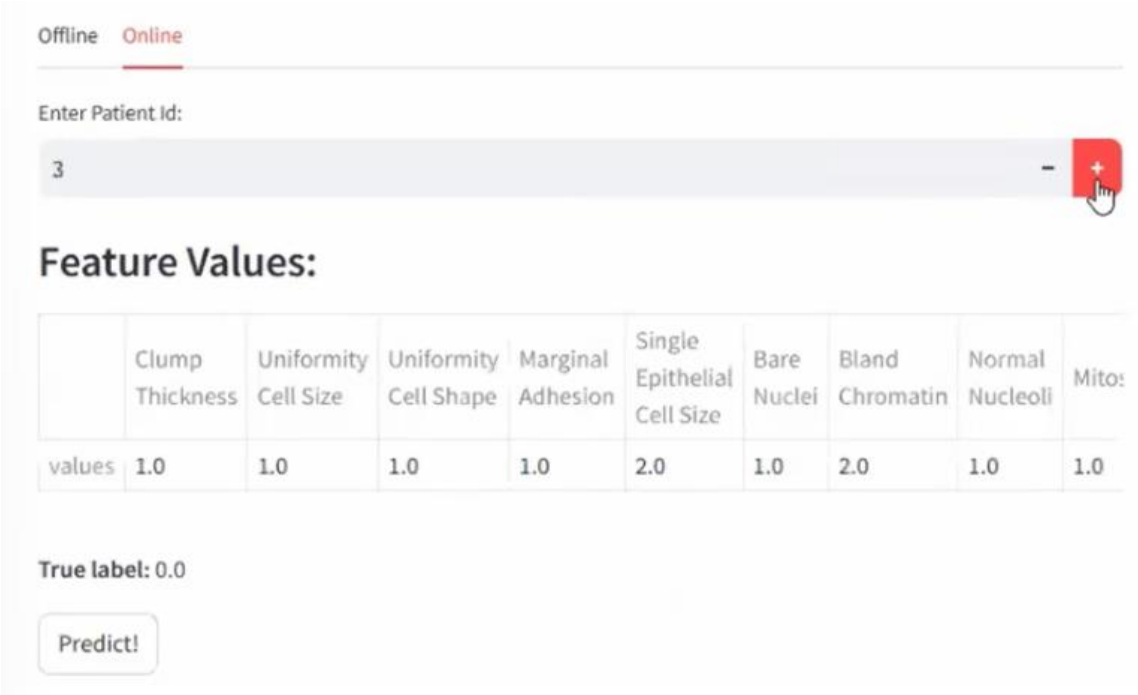

Figure 43. *Online prediction GUI.*

The first result is the predictions. The patient for which this screenshot presented in Figure 44 was taken has a true label of 1, meaning a malignant tumor. We see that the rules-only model which includes one or more rules satisfied by the feature vector and the assignment of the label 1 issues a certain correct prediction. On the other hand the MLP model, based on the training distribution mistakenly labels the patient as 0 (benign tumor). Because the LTN is trained with the same rules as the rule-based model in the loss function, its output is shifted towards the correct label, showcasing how NeuroSymbolic AI manages to generalise better for this instance and correct a mistake made by the model trained only from the data.

Funded by the European Union

Figure 44. *Visualization of predictions.*

Additionally, to the prediction values we show the rules that are triggered by the specific instance. For example, the patient for which the screenshot was taken satisfies two of the rules for the malignant label as presented in Figure 45.

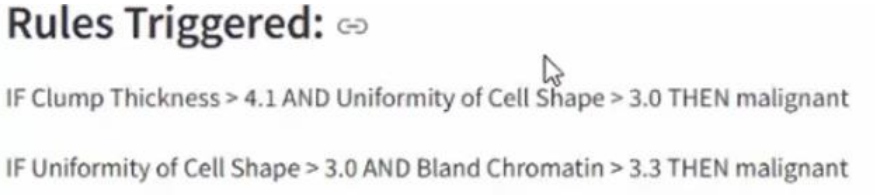

Figure 45. *Depiction of triggered rules.*

To also have a picture of the feature importances from the data-driven learning we include feature importance scores extracted by SHAP, presented in Figure 46. In the future we plan to correlate these features with the satisfiability of the rules to produce a unified and meaningful explanation over the whole NeuroSymbolic model.

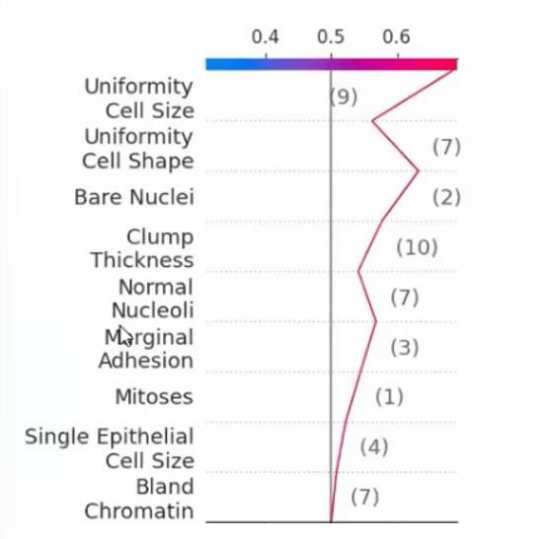

Figure 46. *SHAP feature importances.*

Finally to make the above demo more interoperable we are collaborating with task 3.4 to make the selection of rules by the user more user-friendly by enabling their automatic generation through natural language text. To achieve this we aim to utilise a LLM that will be constrained to produced json outputs describing first-order logic rules. This json will be an intermediate representation from which we will be able to transform the rules into the required representations needed for the underlying NeuroSymbolic model (for example SPL uses a very different format from LTN to encode the rules).

Funded by the European Union

4.7 Initial Results

4.7.1 Validation on Open Datasets

To assess the use of NeuroSymbolic AI, and specifically LTNs, we initially evaluated it on some open-source healthcare datasets: the BCW dataset, whose results were presented in the Use Case and Demo section, as well as the Chronic Kidney Disease (**https://www.kaggle.com/datasets/mansoordaku/ckdisease**) and Heart Failure (**https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data**) Datasets.

Chronic Kidney Disease
The data was taken over 2 months in India with 25 features (eg, red blood cell count, white blood cell count, etc). The problem is a binary classification problem, which is either that there is a chronic kidney disease, or no chronic kidney disease. The dataset has 400 rows and 62% of the rows has a chronic kidney diseas, and 38% has no chronic kidney disease.
The following rules were used for this problem:

Rule 1: IF hemoglobin > 12.95 AND specific gravity > 1.017 AND hypertension <= 0.5 THEN classification = 1 (disease)
Rule 2: IF serum_creatinine <= 29.5 AND packed_cell_volume > 0.95 AND hemoglobin <= 12.95 AND red_blood_cell_count <= 39.5 THEN classification = 0 (no disease)

Heart Failure
This dataset contains data of 299 patients with heart failure collected in 2015. It serves the need for early detection and management of heart issues for people with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidaemia or already established disease) wherein a machine learning model can be of great help. We apply several machine learning classifiers to predict the patient's survival. Regarding the dataset imbalance, the survived patients (death event = 0) are 203 (32.11%), while the dead patients (death event = 1) are 96 (67.89%). For this case the following rules used were:

Rule 1: IF time > 173.5 AND serum_creatinine <= 11.55 AND ejection_fraction > 27.5 THEN death_event = 1
Rule 2: IF time <= 73.5 AND serum_sodium <= 139.5 THEN death_event = 0

One of the principal uses of NeuroSymbolic AI is that it can augment a neural network when trained on limited and “complete” its knowledge. For this reason we experimented with different samples ratios of the original datasets (0.25, 0.5, 0.75 and 1.0) to simulate varying conditions of data scarcity. To highlight the benefits of a combined neural and symbolic approach we compare the LTN against i) a symbolic-only approach that predicts only according to the ruleset used or randomly if the instance does not trigger any of the rules and ii) a neural-only approach consisting of a simple MLP with the same architecture and parameterization as the LTN. The evaluation metric used was the F1 score as datasets vary in size and are imbalanced. The following figures show the obtained results for the kidney disease dataset and heart failure dataset respectively. The blue line shows the performance for the rules only model. This model is only using the rules, without any machine learning algorithm. The orange line shows the results for a multilayer perceptron that has been trained purely on the data. The green line shows the obtained results for the logic tensor network.

Funded by the European Union

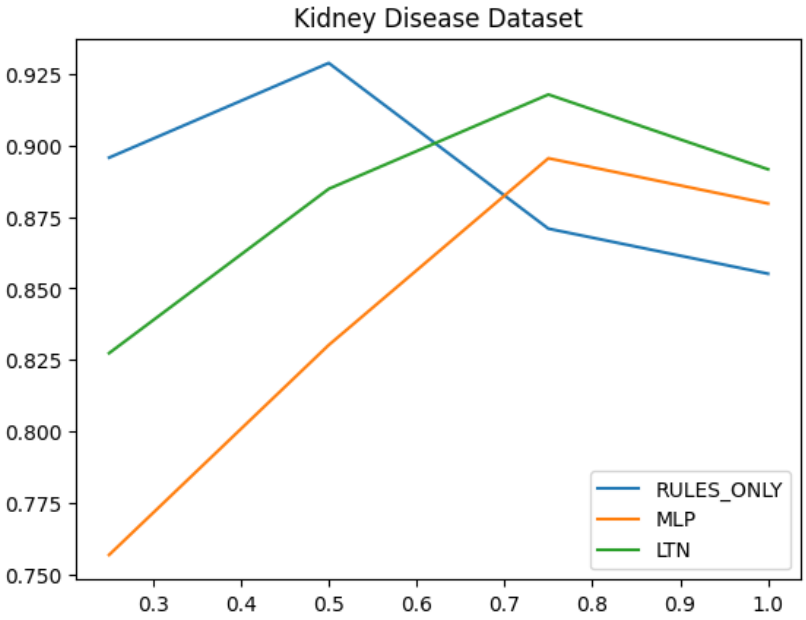

Figure 47. *Results in kidney disease dataset.*

In the kidney disease dataset the LTN shows higher f1-score across sampling ratios than the MLP. This difference becomes more pronounced as the data gets more scarce. In comparison to the rules-only model the LTN outperforms it for ratios 0.75 and 1.0, while when data scarcity is significant the heuristic model is more predictive.

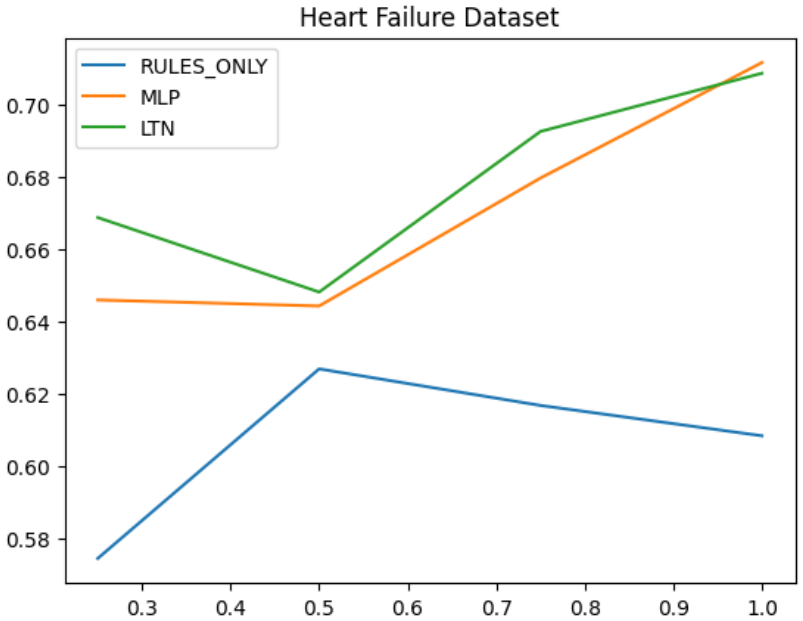

Figure 48. *Results in heart failure dataset.*

In heart failure the pattern has both differences and similarities to the kidney disease dataset. One difference stems from the lower coverage of the rules around 60% vs. around 90% for kidney. For this reason both MLP and LTN have higher f1 scores across all ratios. The LTN slightly outperforms the MLP in all cases except the full dataset where the MLP is slightly better. It could be the case that the low coverage rules do not offer an advantage where data is more abundant, making fully data-driven learning more efficient.

Funded by the European Union

4.7.2 ISP Dataset

Several initial results have been obtained through experimental investigation of the dataset provided by Innovation Sprint. This experiment was performed using three expert rules. First, the patient should sleep between 6 and 8 hours per night. Second, the patient should exercise at least 150 minutes per week. Third, the patient should at least walk 500 steps more than the baseline. These have been translated into rules to be used in the logic tensor network.

Rule 1
If the user follows at least two of the three recommendations, then the predicted SMBG value should not be on the highest 25% quantile.

Rule 2

If the SMBG is before a meal, the SMBG value should be higher than 90 and lower than 130.
If the SMBG is after a meal, the SMBG value should be higher than 120 and lower than 220.

Rule 3
The predicted SMBG value should be equal to the true SMBG value

The rules are normalized according to the time frame and a normalisation method is used for the respective features. This experiment yielded several results. The following table presents a comparison between a multi-layer perceptron (MLP) trained to minimize the mean absolute error (MAE) and an LTN of identical architecture and training parameters trained to maximize the Satisfiability of the rules. The table shows the mean squared error (MSE), root mean squared error (RMSE) and mean absolute error (MAE).

Table 11. *Results of the experiments.*

Multi layer perceptron	**MSE**	**RMSE**	**MAE**
Average	**431,4894**	**20,3800**	**16,1201**
Standard deviation	176,1878	4,4926	2,5006
Logic tensor network	MSE	RMSE	MAE
Average	**425,5827**	**19,7499**	**14,8533**
Standard deviation	248,2382	6,6638	3,7141

The figure below shows the predicted versus the actual values for the SMBG value of the LTN.

Funded by the European Union

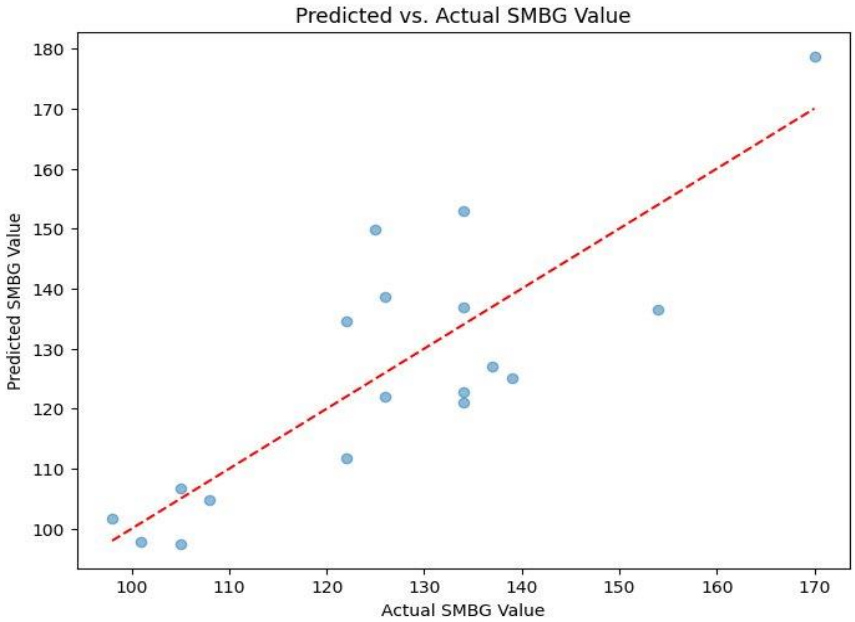

Figure 49. *Predicted versus the actual values for the SMBG value of the LTN.*

In general the LTN with the help of rules manages to stay close to the line with slope 1 where actual values match predictions which shows some early predictive capacity for the problem as well as some small improvement over the vanilla MFP. This result is of course not final and we will continue to work on different setups of learning for the diabetes use case as the data grows and the use case matures. An interesting future direction would be to study the effect of the addition of rules under varying levels of noise.

4.8 Conclusions and Next Steps

Future efforts will focus on experimenting with weekly and bi-weekly data aggregation as more patient information becomes available. Moreover, future efforts include the integration of data from the questionnaire and examination data into feature vectors will provide a more comprehensive view of patient health. Additionally, incorporating rules derived from these questionnaires and exams, along with common-sense guidelines, will be explored to improve the model's predictive capacity by embedding expert knowledge not present in the training data. Moreover, a future direction could be adopting a meta-cognitive approach, where rules are iteratively and interactively refined with user input, which will allow continuous improvement and ensure the model remains aligned with clinical insights. These strategies aim to advance the model's effectiveness and adaptability in the diabetes healthcare case.

4.9 References (NS-AI)

1. Kareem Ahmed, Stefano Teso, Kai-Wei Chang, Guy Van den Broeck, and Antonio Vergari. *Semantic probabilistic layers for neuro-symbolic learning.* In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems,* volume 35, pages 29944–29959. Curran Associates, Inc., 2022. URL

2. Samy Badreddine and Michael Spranger. *Injecting prior knowledge for transfer learning into reinforcement learning algorithms using logic tensor networks. ArXiv,* abs/1906.06576, 2019. **URL**

3. Samy Badreddine, Artur d'Avila Garcez, Luciano Serafini, and Michael Spranger. *Logic tensor networks. Artificial Intelligence,* 303:103649, 2022. ISSN 0004-3702. doi. **URL**

Funded by the European Union

4. Tim Bohne, Anne-Kathrin Patricia Windler, and Martin Atzmueller. *A neuro-symbolic approach for anomaly detection and complex fault diagnosis exemplified in the automotive domain.* In *Proceedings of the 12th Knowledge Capture Conference 2023,* K-CAP '23, pages 35–43, New York, NY, USA, 2023. Association for Computing Machinery. ISBN: 9798400701412. **doi**. **URL**

5. Tommaso Carraro, Alessandro Daniele, Fabio Aiolli, and Luciano Serafini. *Logic tensor networks for top-n recommendation.* In *AIxIA 2022 – Advances in Artificial Intelligence: XXIst International Conference of the Italian Association for Artificial Intelligence, AIxIA 2022, Udine, Italy, November 28 – December 2, 2022, Proceedings,* pages 110–123, Berlin, Heidelberg, 2023. Springer-Verlag. ISBN: 978-3-031-27180-9. doi. URL

6. Cristina Cornelio, Jan Stuehmer, Shell Xu Hu, and Timothy Hospedales. *Learning where and when to reason in neuro-symbolic inference.* In *The Eleventh International Conference on Learning Representations,* 2023. **URL**

7. Ivan Donadello, Luciano Serafini, and Artur S. d'Avila Garcez. *Logic tensor networks for semantic image interpretation.* In *International Joint Conference on Artificial Intelligence,* 2017. **URL**

8. Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J. R. Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, David Silver, Demis Hassabis, and Pushmeet Kohli. *Discovering faster matrix multiplication algorithms with reinforcement learning. Nature,* 610(7930):47–53, October 2022. ISSN 1476-4687. **doi**. **URL**

9. Jerry A. Fodor and Zenon W. Pylyshyn. *Connectionism and cognitive architecture: A critical analysis. Cognition,* 28(1–2):3–71, 1988. doi

10. Vladimir Golovko, Aliaksandr Kroshchanka, Mikhail Kovalev, Valery Taberko, and Dzmitry Ivaniuk. *Neuro-symbolic artificial intelligence: Application for control the quality of product labeling.* In Vladimir Golenkov, Victor Krasnoproshin, Vladimir Golovko, and Elias Azarov, editors, *Open Semantic Technologies for Intelligent System,* pages 81–101, Cham, 2020. Springer International Publishing. ISBN: 978-3-030-60447-9.

11. Greta Greco, Federico Alberici, Matteo Palmonari, and Andrea Cosentini. *Declarative Encoding of Fairness in Logic Tensor Networks,* September 2023. ISBN: 9781643684369. doi

12. Hu Huang, Bowen Zhang, Liwen Jing, Xianghua Fu, Xiaojun Chen, and Jianyang Shi. *Logic tensor network with massive learned knowledge for aspect-based sentiment analysis. Knowledge-Based Systems,* 257:109943, 2022. ISSN 0950-7051. doi. **URL**

13. Jindong Jiang and Sungjin Ahn. *Generative neurosymbolic machines. ArXiv,* abs/2010.12152, 2020. **URL**

14. Daniel Kahneman. *Thinking, fast and slow.* Macmillan, 2011.

15. Michael C. Kampffmeyer, Yinbo Chen, Xiaodan Liang, Hao Wang, Yujia Zhang, and Eric P. Xing. *Rethinking knowledge graph propagation for zero-shot learning. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pages 11479–11488, 2018. URL

16. Henry Kautz. *The third AI summer: AAAI Robert S. Engelmore Memorial Lecture. AI Magazine,* 43(1):105–125, March 2022. doi. URL

17. Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. *Building machines that learn and think like people. Behavioral and Brain Sciences,* 40, 2017. doi

18. Meiyi Ma, Ji Gao, Lu Feng, and John A. Stankovic. *STLNet: Signal temporal logic enforced multivariate recurrent neural networks.* In *Neural Information Processing Systems,* 2020. **URL**

19. Georgios Makridis, Spyros Theodoropoulos, Dimitrios Dardanis, Ioannis Makridis, Maria Margarita Separdani, Georgios Fatouros, Dimosthenis Kyriazis, and Panagiotis Koulouris. *XAI enhancing cyber*

Funded by the European Union

defence against adversarial attacks in industrial applications. In *2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS),* volume 5, pages 1–8, 2022. doi

20. Jiayuan Mao, Chuang Gan, Pushmeet Kohli, Joshua B. Tenenbaum, and Jiajun Wu. *The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. ArXiv,* abs/1904.12584, 2019. **URL**

21. Giuseppe Marra and Ondřej Kučelka. *Neural Markov logic networks.* In *Conference on Uncertainty in Artificial Intelligence,* 2019. **URL**

22. Giuseppe Marra, Michelangelo Diligenti, Francesco Giannini, Marco Gori, and Marco Maggini. *Relational neural machines.* In *European Conference on Artificial Intelligence,* 2020. **URL**

23. Lia Morra, Alberto Azzari, Letizia Bergamasco, Marco Braga, Luigi Capogrosso, Federico Delrio, Giuseppe Di Giacomo, Simone Eiraudo, Giorgia Ghione, Rocco Giudice, et al. *Designing logic tensor networks for visual Sudoku puzzle classification,* 2023.

24. Darian M. Onchis, Gilbert-Rainer Gillich, Eduard Hogea, and Cristian Tufisi. *Neuro-symbolic model for cantilever beams damage detection. Computers in Industry,* 151:103991, 2023. ISSN 0166-3615. **doi**. **URL**

25. Gabriele Picco, Hoang Thanh Lam, Marco Luca Sbodio, and Vanessa Lopez Garcia. *Neural unification for logic reasoning over natural language.* In *Conference on Empirical Methods in Natural Language Processing,* 2021. **URL**

26. Maximilian-Peter Radtke and Juergen Bock. *Expert knowledge induced logic tensor networks: A bearing fault diagnosis case study. PHM Society European Conference,* 7:421–431, June 2022. doi

27. Spyros Theodoropoulos, Patrik Zajec, Joe M. Roanec, Dimitrios Dardanis, Georgios Makridis, Dimosthenis Kyriazis, and Panayiotis Tsanakas. *Identifying novel defects during AI-driven visual quality inspection. IFAC-PapersOnLine,* 56(2):3738–3743, 2023. ISSN 2405-8963. **doi**. **URL**

28. Spyros Theodoropoulos, Patrik Zajec, Jože M. Rožanec, Dimosthenis Kyriazis, and Panayiotis Tsanakas. *On-the-fly image-level oversampling for imbalanced datasets of manufacturing defects. Machine Learning,* January 2024. ISSN 1573-0565. **doi**. **URL**

29. Geoffrey G. Towell, Jude W. Shavlik, and Michiel O. Noordewier. *Refinement of approximate domain theories by knowledge-based neural networks.* In *Proceedings of the Eighth National Conference on Artificial Intelligence - Volume 2,* AAAI'90, pages 861–866. AAAI Press, 1990. ISBN: 026251057X.

30. Benedikt J. Wagner and Artur d'Avila Garcez. *A neurosymbolic approach to AI alignment. Neurosymbolic Artificial Intelligence, Preprint,* pages 1–12, 2024. ISSN 2949-8732. **doi**. **URL**

31. Wenguan Wang, Yi Yang, and Fei Wu. *Towards data-and knowledge-driven artificial intelligence: A survey on neuro-symbolic computing,* 2022. URL

32. X. Wang, Yufei Ye, and Abhinav Kumar Gupta. *Zero-shot recognition via semantic embeddings and knowledge graphs.* In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pages 6857–6866, 2018. **URL**

33. Hua Wei, Paulo Shakarian, Christian Lebiere, Bruce Draper, Nikhil Krishnaswamy, and Sergei Nirenburg. *Metacognitive AI: Framework and the case for a neurosymbolic approach.* In Tarek R. Besold, Artur d'Avila Garcez, Ernesto Jimenez-Ruiz, Roberto Confalonieri, Pranava Madhyastha, and Benedikt Wagner, editors, *Neural-Symbolic Learning and Reasoning,* pages 60–67, Cham, 2024. Springer Nature Switzerland. ISBN: 978-3-031-71170-1.

34. Bowen Xi, Kevin Scaria, Divyagna Bavikadi, and Paulo Shakarian. *Rule-based error detection and correction to operationalize movement trajectory classification,* 2024. **URL**

Funded by the European Union

35. Yuan Yang and Le Song. *Learn to explain efficiently via neural logic inductive learning. ArXiv,* abs/1910.02481, 2019. URL
36. Dongran Yu, Bo Yang, Dayou Liu, Hui Wang, and Shirui Pan. *A survey on neural-symbolic learning systems. Neural Networks,* 166:105–126, 2023. ISSN 0893-6080. doi. URL

Funded by the European Union

5 Secure Data Collection and Management

The Secure Data Collection and Management task (T4.1) forms the backbone of the HumAIne platform. The system provides a repository for collecting, storing and accessing securely HumAIne's related data. The proposed implementation will collect data from the pilot use cases to assist the WP6 activities, as well as the AI learning paradigms.

5.1 Introduction

The rapid growth of digital data generation, driven by advancements in IoT ecosystems, cloud-based applications, and enterprise systems, has highlighted the importance of ensuring strong security mechanisms in storage and processing frameworks. The exponential growth of security incidents such as unauthorized access, data breaches and malware has led to costs up to 400 billion USD [1] to the global economy from cybercrime. Modern cloud-based architectures face multifaceted security challenges, ranging from sophisticated cyber threats to compliance with stringent regulatory requirements. To address these, organizations are adopting innovative strategies that incorporate adaptive security mechanisms and privacy-enhancing technologies.

5.1.1 State of The Art Analysis

Cryptography-Based Models

Cryptography remains central to securing sensitive information in cloud environments. Both symmetric and asymmetric encryption techniques are employed to achieve confidentiality, integrity, and secure communication. Kao et al. [2] introduced uCloud, a user-centric key management system that encrypts user data indirectly via the Rivest-Shamir-Adleman algorithm (RSA), leveraging public keys stored on mobile devices. The innovative use of 2D barcodes for expressing private keys enables secure decryption of sensitive data, enhancing usability and security. Al-Haj et al. [3] designed two cryptographic algorithms focusing on confidentiality, integrity, and authenticity. By combining a hash-based symmetric key cryptographic function with the elliptic curve digital signature algorithm and advanced encryption standard-Galois counter mode, their approach provides robust data protection. Liang et al. [4] proposed a Ciphertext-Policy Attribute-Based Proxy Re-Encryption Scheme to minimize computational and communication costs while enabling data owners to assign access rights. This is particularly effective for secure cloud data sharing, allowing re-encryption without heavy computation. Wang et al. [5] addressed hierarchical file sharing in cloud environments using a File Hierarchy Attribute-Based Encryption (FH-CP-ABE) scheme. Their model leverages a layered access structure to reduce storage costs and computational complexity for encryption and decryption. However, computation costs rise dynamically when common attributes or integrated ciphertext are computed repeatedly.

Differential Privacy with Machine Learning

Differential privacy with machine learning aims to protect sensitive data by making outputs indistinguishable for queries that differ in only one record. This approach uses a predefined privacy parameter (ε), where a smaller ε indicates stronger privacy protection. Statistical noise is embedded into datasets, ensuring privacy during machine learning processes, such as classification. Yonetani et al. [11] proposed a Doubly Permuted Homomorphic Encryption (DPHE) mechanism, which enables multi-party protected scalar products with reduced computational overhead. Although effective for visual recognition tasks, the model is limited to performing one operation at a time (multiplication or addition). Hesamifard et al. [12] introduced CryptoDL, a framework for applying deep neural network algorithms on encrypted data. Their model establishes a theoretical foundation for secure neural network computation under the constraints of homomorphic encryption schemes. However, their approach does not address the challenges of using multiple encryption keys for data from different owners. Li et al. [13] proposed a privacy-preserving Naive Bayes learning scheme that enables multiple providers to collaboratively train classifiers with ε-differential privacy. This approach maintains privacy

Funded by the European Union

for individual datasets during collaborative training but introduces vulnerabilities through potential collusions among data contributors.

Access Control Models

Access control frameworks ensure that only authorized users can access specific data. Nabeel and Bertino [6] introduced a privacy-preserving attribute-based key management scheme. This hybrid approach combines coarse-grained encryption by data owners and fine-grained encryption by cloud providers to minimize data owner overhead while maintaining data confidentiality. Zaghloul et al. [7] developed P-MOD, a Privilege-Based Multilevel Organizational Data-sharing model. It strengthens attribute-based encryption with privilege-based access structures to manage hierarchical datasets. Experimental analysis demonstrated superior efficiency compared to CP-ABE [8] and FH-CP-ABE [5] schemes, especially for multi-level organizations. Almutairi et al. [9] proposed a Role-Based Access Control (RBAC) policy for cloud environments. By measuring data sensitivity based on sharing levels, their method limits exposure of sensitive data in multi-tenant settings. Xu et al. [10] introduced a fine-grained access control scheme for dynamic user groups, defining and enforcing policies based on data attributes (Attribute Based Access Control). This approach supports key updates and allows untrusted cloud service providers to perform delegated computations without requiring a delegation key.

5.1.2 Implementation Rationale

During the beginning of the task, an initial round of gathering essential information regarding pilot data was performed by disseminating a Data Landscaping spreadsheet among the pilots, in order to better understand the data that they are planning to use. The contributions, containing information about the volume of the data, data format, data asset providers, etc., as well as a sample of the dataset, were stated in D1.2-Data Management Plan.

Driven by the results of this spreadsheet, where all pilot datasets contained information in different file formats, such as JSON, CSV, YAML, and PDF files, we decided to use MinIO as our main storage solution, which functions as an object storage platform. To ensure a secure and transparent way of exchanging data and information between the pilots and the learning paradigms, we opted for an Attribute-Based Access Control (ABAC) architecture. This architecture was implemented through eIDAS and Keycloak for authentication and user base management within the HumAIne platform.

To further elaborate, ABAC provides a flexible and fine-grained access control mechanism tailored to securely manage data from multiple pilot use cases. ABAC allows dynamic policy enforcement based on user attributes, such as roles and pilot group memberships. eIDAS ensures secure user authentication and verified attribute retrieval, which are mapped into Keycloak for policy assignment. MinIO serves as the main storage solution, enforcing bucket-level access control using these policies. Together, these technologies enable secure and seamless data management, ensuring compliance and scalability for the HumAIne platform.

This section focuses on the pivotal technologies used to fulfil the requirements of this task, eIDAS for sovereign identity management, Keycloak for identity access management and MinIO for the data storage.

5.2 Sovereign Identities

Sovereign identity management is an approach that empowers individuals with control over their personal data, ensuring secure and trustworthy authentication for accessing services. Unlike traditional identity systems, sovereign identity frameworks emphasize privacy, while allowing the usage of digital credentials that can be independently verified. By leveraging electronic identity solutions, secure and seamless authentication for users can be ensured.

In the context of digital ecosystems, the use of sovereign identities helps facilitate interoperability, strengthens compliance with data protection laws such as the General Data Protection Regulation

Funded by the European Union

(GDPR), and mitigates security risks associated with unauthorized access. The implementation of sovereign identity frameworks often involves integrating robust authentication protocols and attribute management systems to enable secure access to digital resources.

5.2.1 eIDAS - Sovereign Identity Solution Description

In the context of HumAIne, eIDAS (Electronic Identification, Authentication, and Trust Services) was selected as a sovereign identity solution, which is an exemplary framework developed by the European Union and adopts an electronic identification scheme that may be used as proof of identity for individuals across Europe. This tool allows for identity verification of citizens of a European Union Member State when using online services in other Member States, thus facilitating cross-border electronic transactions. The eIDAS network is composed by a number of eIDAS nodes, one per Member State implemented at the national level. Each node can either provide or request cross border authentication. In HumAIne, eIDAS will be used for managing user identities. Specifically, the eIDAS node will support the authentication process for individuals registering on the HumAIne platform, while ensuring secure and seamless access to the platform's services.

5.2.2 eIDAS Integration for User Registration and Authentication in the HumAIne Platform

For the facilitation of various operations of the eIDAS node, an eIDAS API has been implemented. The eIDAS node relies on plain text configuration files to record and store user information. To facilitate user management, we developed a Python API to handle modifications to these files, which enables the creation of new users in the test eIDAS node deployed for HumAIne. When registering a user, the API requires a set of mandatory attributes, including first name, last name, national identification number, and gender. The system also allows the addition of custom attributes for more flexibility. The API accepts these attributes through a dedicated endpoint.

The initial phase of user registration involves authentication through eIDAS. This process is supported by a Spring Boot back-end service, which communicates with the eIDAS node. The authentication back-end provides an endpoint for submitting a username and password, forwarding these credentials to the eIDAS node for verification.

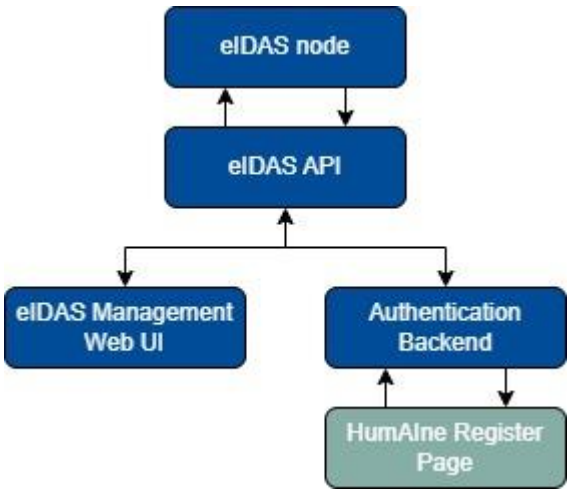

Figure 50. *Implementation of eIDAS component and various services.*

Upon successful user creation, the Python API returns a confirmation message stating, "User inserted Successfully." If the username or national identification number provided already exists in the system, the API responds with an error message, "User Already Exists." Meanwhile, the Spring Boot back-end provides an endpoint which handles authentication responses. On successful authentication, it returns

Funded by the European Union

a JSON object containing an 'eidas_response' that includes all associated user attributes stored in the eIDAS node, along with an assertion.

Adding an eIDAS user Information Flow

The process of adding a user to the eIDAS pool, in order to be able to authenticate and register to the HumAIne platform, involves an eIDAS administrator accessing the eIDAS Management Web UI. As shown in Figure 51. , the administrator inputs the required user credentials and information into a form, which is then sent to the eIDAS API. The API modifies the configuration files within the eIDAS node to register the new user. Once the user is successfully registered, the administrator receives a notification through the web interface.

The described implementation, which is graphically represented in Figure 50. , has been tested locally and may require modifications, according to the needs of HumAIne platform.

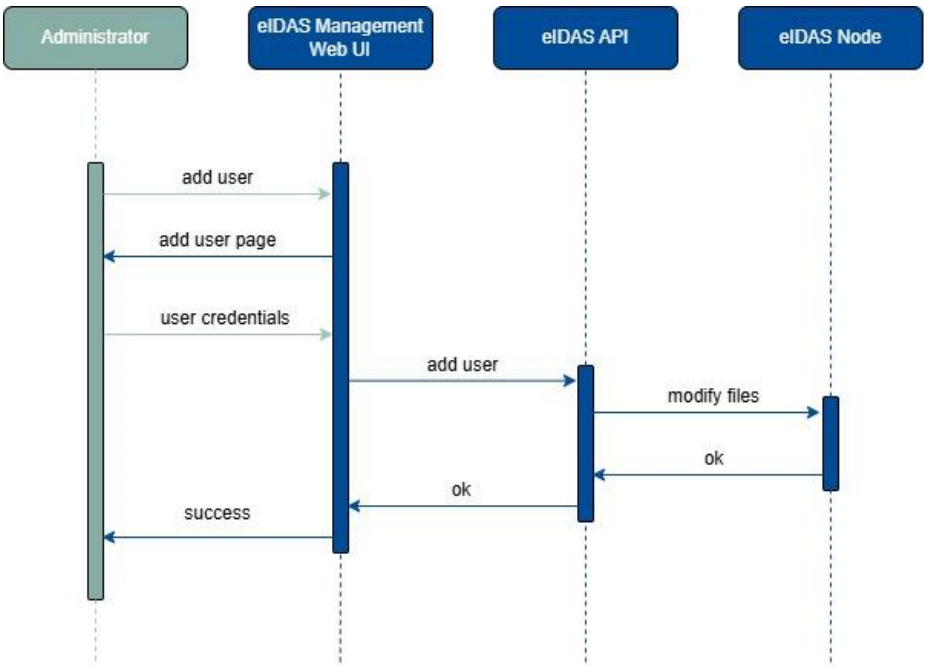

Figure 51. *UML Diagram of the administrator adding a new eIDAS user to the system and the communication of the eIDAS management front-end, eIDAS API and eIDAS node.*

HumAIne Registration

The HumAIne registration process, begins through the HumAIne common front-end application, where users are presented with an eIDAS login form. Here, they input their eIDAS credentials, which are transmitted from the front-end to the authentication back-end service. The authentication back-end then forwards these credentials to the eIDAS node for verification. If the credentials correspond to an existing eIDAS user, the eIDAS node issues an access token. This token is returned to the authentication back-end and subsequently forwarded to the common front-end. At this point, the involvement of the eIDAS component concludes, and the user is allowed to proceed with creating their HumAIne account.

Funded by the European Union

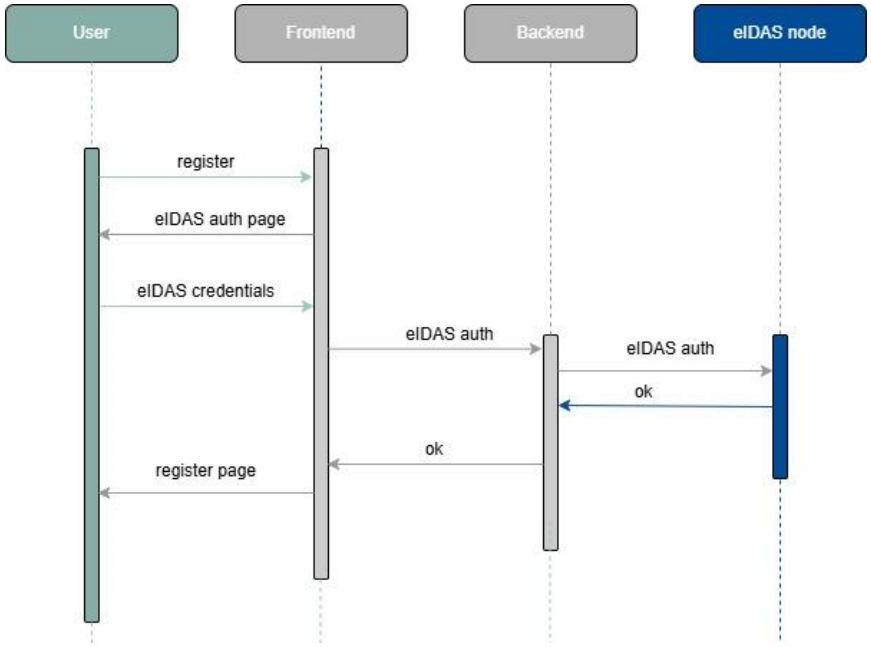

Figure 52. *UML Diagram of the user registration process and the communication of the front-end, authentication back-end, and eIDAS services.*

5.3 Identity Access Management Mechanism

To ensure that only authenticated and authorized users have access to data resources and different applications, the HumAIne project incorporates a robust Identity Access Management (IAM) system. Keycloak will serve as the primary IAM solution due to its extensive support for user authentication, authorization, and seamless integration with various identity and data storage services.

5.3.1 Keycloak: IAM Solution

Keycloak is an open-source identity access management solution that provides comprehensive capabilities regarding user authentication, authorization and session management. Keycloak supports standard identity protocols such as OpenID Connect (OIDC), OAuth 2.0 and SAML 2.0, making it versatile for various integration needs within the HumAIne project. By leveraging these protocols, Keycloak can interface seamlessly with other HumAIne components/systems.

In the context of the HumAIne project, Keycloak serves as the userbase management system for managing user identities, groups and access policies, as well as providing authentication for the different HumAIne tools in the HumAIne platform. Its primary functionalities include:

- **User Authentication**: Keycloak validates user credentials and facilitates secure login processes. This is especially important in a collaborative AI environment where multiple users across different pilots need controlled access.
- **User Attributes Management**: Keycloak stores and manages user attributes that are received from eIDAS, such as basic user information as described below. The administrator can use the Admin Console web UI to manage users, realms, groups and user attributes as well as configure and manage security related settings.
- **Single Sign On (SSO) functionality**: Keycloak provides SSO capabilities, allowing users to log in once and access various services in the HumAIne project without needing to authenticate repeatedly. This will simplify the authentication process allowing access to data and tools across multiple pilots with a single pair of credentials.

Funded by the European Union

5.3.2 Internal Architecture

Keycloak employs a PostgreSQL database to store the HumAIne user base and manage its configuration settings. It provides external applications with the option to interact with it either through its REST API or via libraries like the Keycloak Java API. For user registration and login operations, the HumAIne authentication back-end utilizes the Keycloak Java API.The authentication page on the HumAIne front-end interacts with the system through API endpoints exposed by the authentication back-end, ensuring seamless communication and functionality across the platform.

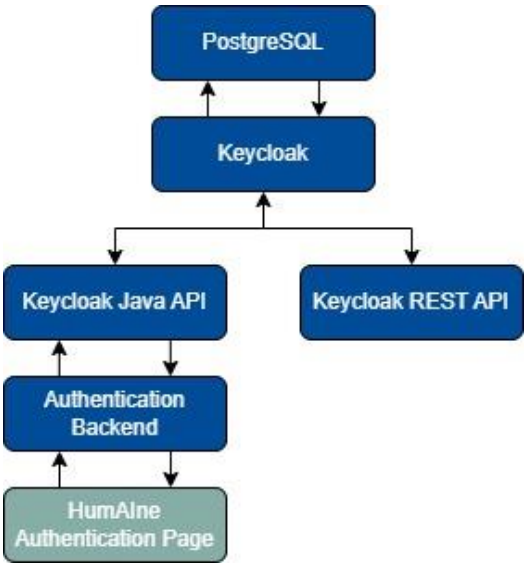

Figure 53. *Internal architecture of the Keycloak component and related libraries and services.*

5.3.3 Keycloak Input/Output

The Keycloak identity manager integrates with the back-end to provide essential authentication and authorization functionalities. It communicates exclusively with the back-end during both the registration and login phases.

Register

During the user registration process, the back-end propagates user attributes retrieved from the eIDAS node to Keycloak to facilitate the creation of new users. The following details are provided as input to Keycloak for user registration:

- **Username**: The username chosen by the user during registration process, after having authenticated with their eIDAS credentials. It is always the same as the user's eIDAS username.
- **First Name**: The user's first name, extracted from the access token returned by the eIDAS node.
- **Last Name**: The user's last name, also retrieved from the access token issued by the eIDAS node.

Additionally, four custom attributes are included in the current implementation and sent to Keycloak:

- **assertion**: The encoded assertion returned by the eIDAS node, stored in its original format.
- **birthdate**: The user's date of birth, obtained from the access token issued by the eIDAS node.
- **gender**: The user's gender, as specified in the access token issued by the eIDAS node.
- **person_identifier**: The user's national identification number, retrieved from the eIDAS node's access token.

Funded by the European Union

After the eIDAS authentication has been completed and the user proceeds to register in the HumAIne platform through the common frontend, the user is required to select which pilot they are participating in, through a dedicated menu option (e.g. Smart Energy, Smart Manufacturing etc.). This information is then forwarded to the backend. Each user has the option to select multiple pilots. For example, developers that are working on the active learning paradigm may select multiple pilots, such as smart cities, smart energy and smart finance pilots.

Moreover, Keycloak has the capability to accept and store a password for the user account once it has been successfully created. Keycloak returns the “201 Created” response code to indicate that the user creation process has successfully completed. This process ensures seamless integration of user data from the eIDAS node into Keycloak during the registration phase of HumAIne.

Login

The user login process for HumAIne can be handled using two methods involving Keycloak. Keycloak provides a native API endpoint for authenticating users, which can be used directly. Alternatively, the back-end implements a dedicated endpoint that communicates with Keycloak on behalf of the system, serving as an intermediary between the web UI and Keycloak. Both methods ensure seamless interaction for user authentication.

In either approach, Keycloak accepts the following fields as input:

- **username**: The user’s username, provided during registration.
- **password**: The user’s password, set during registration.
- **grant_type**: A constant value, "password," indicating that the client is requesting an access token using the user’s credentials.
- **client_id**: A constant value, "humAIne-login," specifying the application name through which authentication is performed.

These methods ensure secure and efficient user login while integrating Keycloak into the HumAIne authentication workflow.

When Keycloak verifies that the provided credentials are valid and match a user stored within the relevant realm, it returns the following data:

- **access_token**: A token associated with the user. This token can be decoded to access account details, Keycloak roles, access permissions to Keycloak resources, and user information stored during registration (e.g., last name).
- **expires_in**: The number of seconds remaining before the access token expires.
- **refresh_token**: A token that can be used to obtain a new access token.
- **refresh_expires_in**: The number of seconds remaining before the refresh token expires.
- **token_type**: The type of token being issued, typically "Bearer."
- **not-before-policy**: The number of seconds indicating the earliest time the token becomes valid.
- **session_state**: The identifier for the user's session or the token associated with it. This identifier links the session to the user in Keycloak.
- **scope**: Specifies the resources the client is authorized to access on behalf of the user using the access token.

This information allows HumAIne to manage user sessions securely and efficiently while integrating with Keycloak for authentication and authorization.

Funded by the European Union

5.3.4 Keycloak Information Flow

In HumAIne 6 Keycloak groups were created, 1 group per pilot. Each HumAIne user can either be assigned to 1 or multiple Keycloak groups. Each group has an attribute attached to it with the key 'policy', and the value corresponds to the name of the group (i.e. smart-energy-policy). This policy attribute is later used by MinIO to allow for restrictive access to users, depending on the pilot they are working on in the context of ABAC. More information about the policies is provided in the corresponding MinIO section.

Register

The registration process in HumAIne begins through the front-end application, where users are presented with an eIDAS login form to input their eIDAS credentials. These credentials are transmitted from the front-end to the authentication back-end service, which then forwards them to the eIDAS node for authentication. If the credentials are valid and correspond to an existing eIDAS user, the eIDAS node issues an access token, which the authentication back-end returns to the front-end.

Following this, the user is presented with a registration form to create their HumAIne account. The submitted HumAIne credentials, along with the pilot attribute selected and the access token issued by eIDAS, are sent to the back-end. The back-end decodes the token to extract user information such as name, surname, gender, and other relevant attributes. Using this data, the back-end uses the Keycloak Java API to create a new user account and set the account password. After the account creation, the backend adds the user to the pilot Keycloak groups according to what the user selected previously. As mentioned above, each user can be assigned to either one or multiple groups.

Upon successful completion of this process, the front-end notifies the user, informing them that they can now log in with their created HumAIne account.

Funded by the European Union

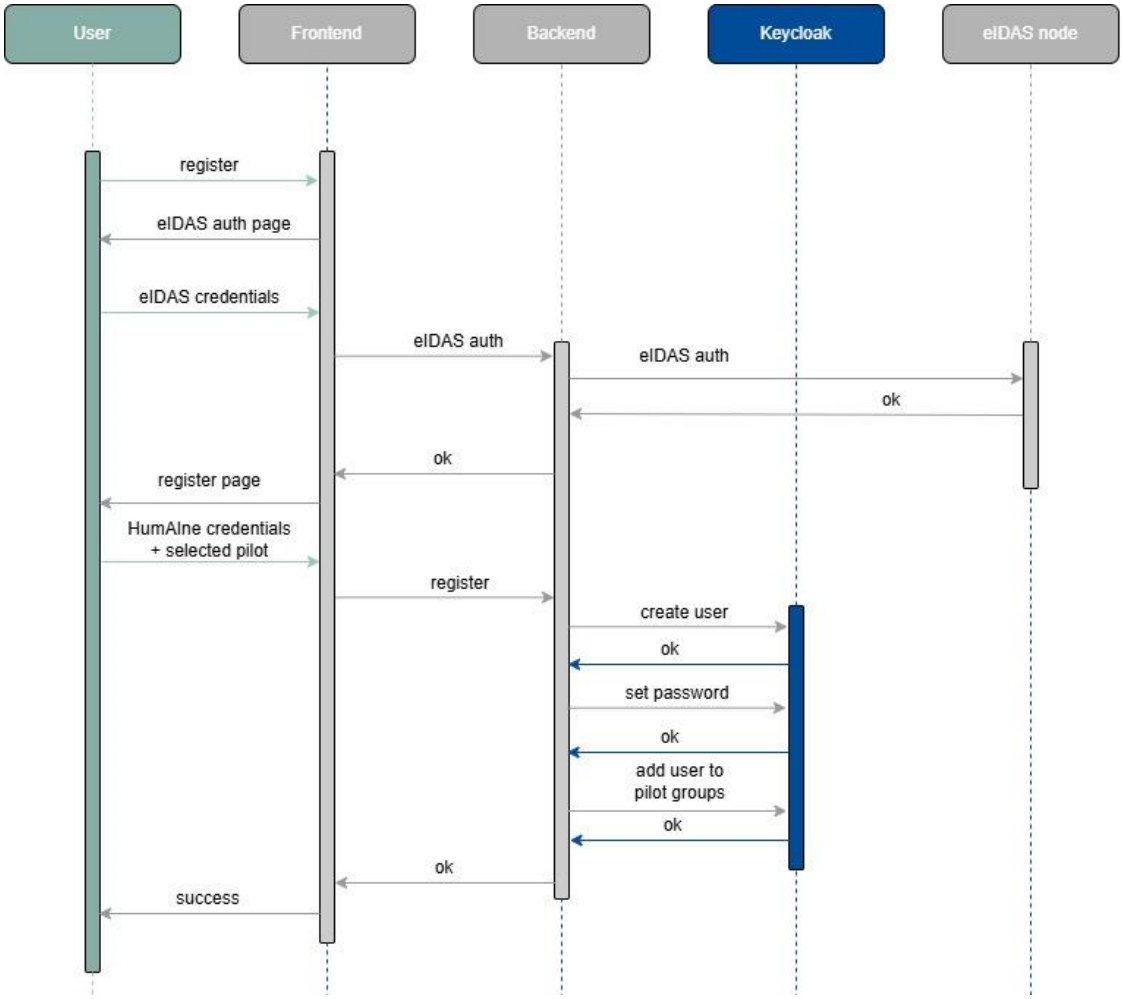

Figure 54. *UML Diagram of the registration process and the communication of the front-end, authentication back-end, Keycloak and eIDAS services.*

Login

The login process in HumAIne begins through the front-end application, where users are presented with a login form to enter their username and password. When the credentials are submitted, a request is sent to the authentication back-end, which interacts with Keycloak to verify if the user exists and whether the provided credentials are valid. If authentication is successful, an access token is issued by Keycloak and returned to the back-end. The back-end then forwards this token to the front-end, enabling the user to access the HumAIne application.

Funded by the European Union

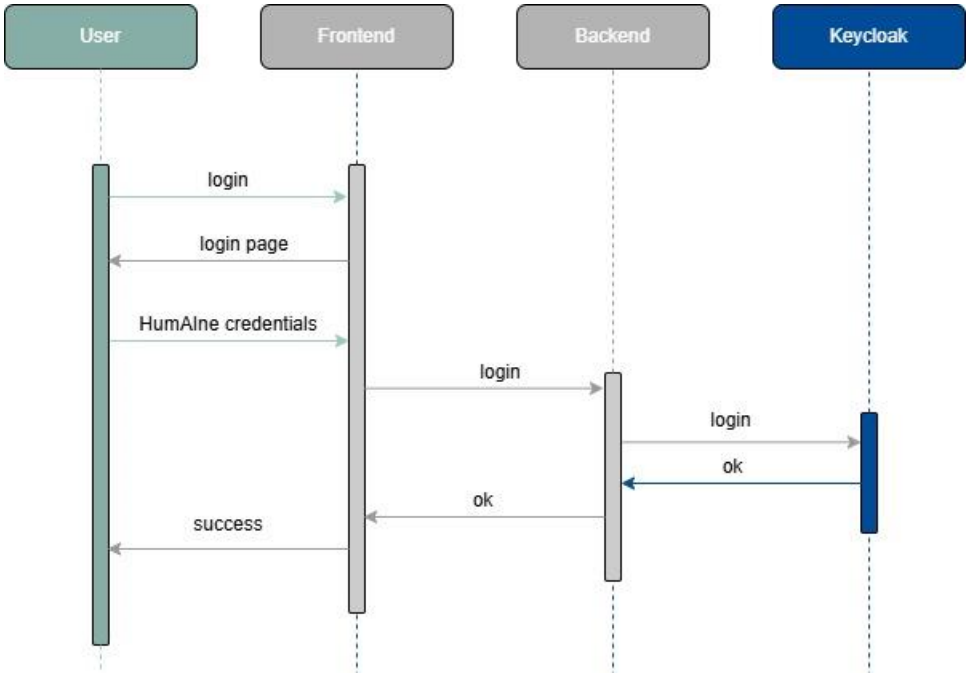

Figure 55. *UML Diagram of the login process and the communication of the front-end, authentication back-end and Keycloak services.*

5.4 Object Storage Mechanism

To store, manage, and provide secure access to data resources and AI models generated by the HumAIne project, an object storage system was selected. MinIO serves as the primary object storage solution due to its compatibility with modern cloud-native architectures, support for hybrid data formats, and seamless integration with the project's IAM and identity verification systems. This section delves into the technical architecture, workflows, and integration of MinIO within the HumAIne project, emphasizing its role in enabling secure, traceable, and efficient data and model storage.

5.4.1 MinIO: Object Storage Solution

MinIO is an open-source object storage system that supports the storage of unstructured data such as text, images, and machine learning models. An object storage is a storage architecture that manages data as objects rather than files, or blocks. It is fully compatible with the Amazon S3 API, and is designed for high performance, supporting scalable and efficient data retrieval for AI workflows.

In the HumAIne project, MinIO is used as the platform for storing and managing the data, models, and results associated with the different pilot use cases. Its primary functionalities include:

- **Bucket Management**: MinIO organizes data into buckets, with each pilot in the HumAIne project allocated dedicated buckets for storing raw or processed data, models, and results.
- **Metadata Management**: MinIO allows metadata to be attached to each object, providing additional information for organizing, searching, and tracking data and models. Metadata attributes include object identifiers, timestamps, version details and performance metrics ,regarding AI models, and more.
- **Access Control**: MinIO integrates with Keycloak to enforce Attribute-Based Access Control (ABAC) through the access policies it utilizes. This ensures that access to buckets and stored objects is restricted based on user attributes as defined by the IAM system (Keycloak).
- **Versioning**: MinIO supports object versioning, allowing for multiple iterations of data and models to be stored, tracked, and retrieved as needed.

Funded by the European Union

- **Encryption**: MinIO provides encryption for data at rest and in transit, ensuring compliance with security requirements and safeguarding sensitive information if needed.

MinIO is integrated with Keycloak and eIDAS to ensure that only authenticated and authorized users can interact with the stored data and models. This integration enables secure data access workflows tailored to the requirements of each pilot project within HumAIne.

5.4.2 MinIO Architecture

As stated above, MinIO is deployed as the primary object storage solution for all pilot data in the HumAIne project. MinIO's architecture revolves around buckets. Given the pilot and the AI paradigm requirements, we decided to structure our storage so as each one of the 6 pilots in the HumAIne project will have 3 dedicated buckets available, categorized into:

- **Data Buckets**: For uploading and storing their datasets.
- **Model Buckets**: For storing AI and machine learning models developed within the HumAIne project.
- **Results Buckets**: For storing outputs generated by these models.

For example, the smart-energy pilot will utilize the buckets 'smart-energy-data', 'smart-energy-models' and 'smart-energy-results'. Similar bucket names have been created for the rest of the pilots. In addition, more buckets may be created according to the project needs regarding storage for the rest of HumAIne's technologies. For example, Benchmarking Suite's results can be stored and accessed either in the corresponding pilot-results buckets, or by creating separate buckets dedicated to Benchmarking Suite. The same principle can be followed for the rest of HumAIne technologies if needed.

Model Registry and Metadata

Users can upload and store their datasets, models, and results to dedicated storage buckets designed to facilitate streamlined management and collaboration. There are 2 ways users can upload their files. Either manually through a user-friendly web interface that MinIO offers or programmatically using API or CLI commands (through MinIO Client mc command line tool). Moreover, custom APIs will be created to assist with the pilot specific needs related to the MinIO usage if needed.

To maintain consistency and improve accessibility, users are required to include relevant metadata when uploading or storing files. Metadata about the AI models should describe key attributes related to datasets used, the model metadata, learning paradigm specific, performance and traceability metadata. An initial suggested format of metadata provided by the learning paradigms is presented below:

Active Learning

- **Dataset UUID <string>:** UUID of the dataset in the data lake.
- **Dataset Version <string>**: The version or timestamp of the dataset used.
- **Model type <string>**: The type of model (e.g., neural network, SVM, random forest).
- **Model Version <string>**: Version of the model or training pipeline used.
- **Hyperparameters <JSON>**: All hyperparameter settings (e.g., learning rate, batch size).
- **Training Dataset <[integer]>**: Subset of data used for training (indices, IDs).
- **Validation Dataset <[integer]>**: Subset of data used for validation (indices, IDs).

Funded by the European Union

- **Query Strategy <string>**: The strategy used to select samples (e.g., uncertainty sampling, diversity sampling) and metrics or criteria for choosing the queried samples (e.g., entropy, margin of confidence).
- **Iteration Samples <integer>**: The number of samples queried in each iteration.
- **Seed Set <[integer]>**: The initial labelled dataset used to start active learning.
- **Iteration Metrics <[iteration: integer, JSON: { "accuracy": float, ...}]>**: Accuracy, precision, recall, F1 score, and other relevant metrics at each iteration. The last record represents the final model performance metrics.
- **Timestamps* <JSON: [{ "action": string, "timestamp": timestamp}]>**: For each action, including dataset preparation, training, querying, and annotation.
- **Logging and Auditing* <string (blob)>**: Logs of decisions made by the algorithm and any manual overrides.
- **Version Control* <string>**: Git hashes or other versioning identifiers for datasets, models, and code.
- **Experiment Parameters [<string>]**: Unique IDs or names for each experiment run.

Swarm Learning

- **Participant ID<string>**: Unique identifier for each participant in the swarm.
- **Node Roles[<string>]**: Role in the swarm (e.g., leader, member).
- **Dataset UUID <string>:** UUID of the dataset in the data lake.
- **Dataset Version<string>**: The version or timestamp of the dataset used.
- **Data Sensitivity<string>**: Privacy and security requirements for the participant's data.
- **Model Architecture<onnx> or <JSON>**: Shared architecture agreed upon by participants.
- **Global Hyperparameters<JSON>**: Global settings (e.g., learning rate, batch size).
- **Global Model Version<string>**: Version of the globally aggregated model.
- **Aggregation Algorithm<string>**: Technique used for combining local models (e.g., federated averaging, weighted aggregation).
- **Weight Contributions<JSON>**: Proportion of each node's contribution to the global model.
- **Aggregation Frequency<string>**: How often aggregation occurs.

Neurosymbolic Learning

- **Dataset UUID <string>:** UUID of the dataset in the MinIO data lake.
- **Dataset Version <string>**: The version or timestamp of the dataset used.
- **Transformation Steps <[string]>**: Custom labels representing the preprocessing and transformations applied to raw data.
- **Rules File <string>**: Path to the JSON containing a representation of the rules used for the current training run. The file name will include an ID for the training run.
- **Model Type <string>**: Type of neural network (e.g., CNN, RNN, Transformer).
- **Model Version <string>**: Versioning of model referring to the HumAIne model repository (most likely model in ONNX format).
- **Layer Details<[JSON]>**: List of json objects with layer index, layer types, layer configuration arguments, next layer indices.

Funded by the European Union

- **Hyperparameters<JSON>**: Json object containing various float/string values such as learning rate, batch size, epochs, etc.
- **Optimizer<JSON>**: Type(e.g., Adam, SGD) and configuration, loss metric
- **Loss Metric<string>**: Loss metric that is minimized (i.e. name of the loss function)
- **Epoch Statistics<JSON>**: Contains loss metric value for specific epoch checkpoints.
- **Knowledge Representation<String>**: Knowledge representation for the rules to be useable by the model OR name of the converter method used to convert the rules from standard into model-specific format (e.g., Probabilistic SDD for the SPL method).
- **Satisfiability Scores <JSON>**: Satisfiability of ruleset at specific checkpoints during training.
- **Seed Value<integer>:** seed used for training
- **Training Indices <[integer]> or <JSON>**: data instances used for training (perhaps per cross-validation fold)
- **Validation Indices <[integer]> or <JSON>**: data instances used for validation (perhaps per cross-validation fold)
- **Post-training Evaluation Metrics <JSON>**: Evaluation metrics calculated post-training on validation set(s).
- **Training Time <Float>:** Time spent during training.
- **Training Time per Epoch <Float>:** Time spent per epoch during training.
- **Training Failure Cases <[Integer]>:** Indices of mislabeled training set predictions after training
- **Validation Failure Cases <[Integer]>:** Indices of mislabeled validation set predictions after training

All the metadata per learning paradigm stated above can be enriched/modified accordingly during the integration phase, if needed.

5.4.3 MinIO Access Control via Policies

To facilitate secure connection and secure data management between the users and different components of the HumAIne platform, a Single Sign On (SSO) functionality has been implemented between Keycloak and MinIO. This connection has been configured using Keycloak's OpenID Connect (OIDC) protocol which is also supported by MinIO so that MinIO seamlessly interacts with Keycloak for identity and access management. To achieve this, a Keycloak Client has been created with the Client ID 'minio' and the Client Type set to OpenID Connect along with a set of other configurations.

Funded by the European Union

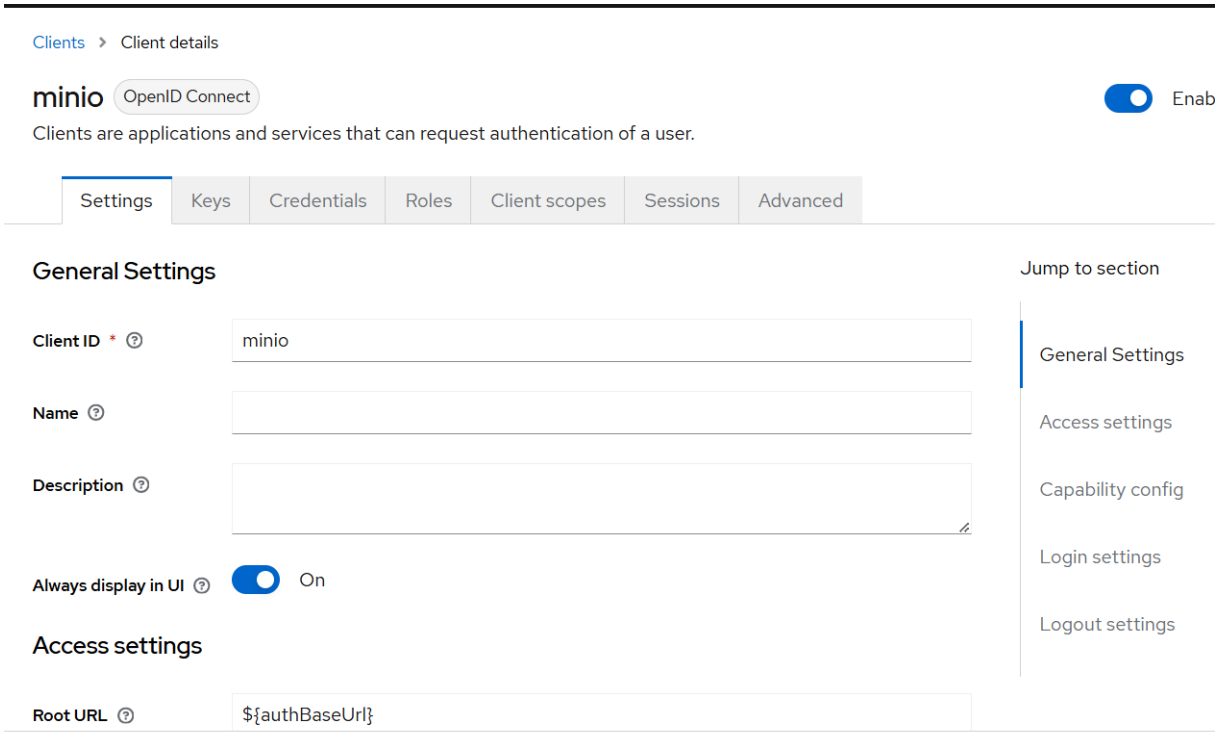

Figure 56. *Configuration of minio Client in Keycloak*

Additionally, a Client Scope named 'minio-authorization' has also been created for the 'minio' Client, which allows to map user attributes as part of the JWT returned in authentication requests. This allows MinIO to reference those attributes when assigning policies to the user. This step creates the necessary client scope to support MinIO authorization after successful Keycloak authentication. Finally, an OPENID Configuration was created in the MinIO Client, accepting details such as the configuration URL to match the Keycloak HumAIne Realm, Client ID, Client Secret etc.

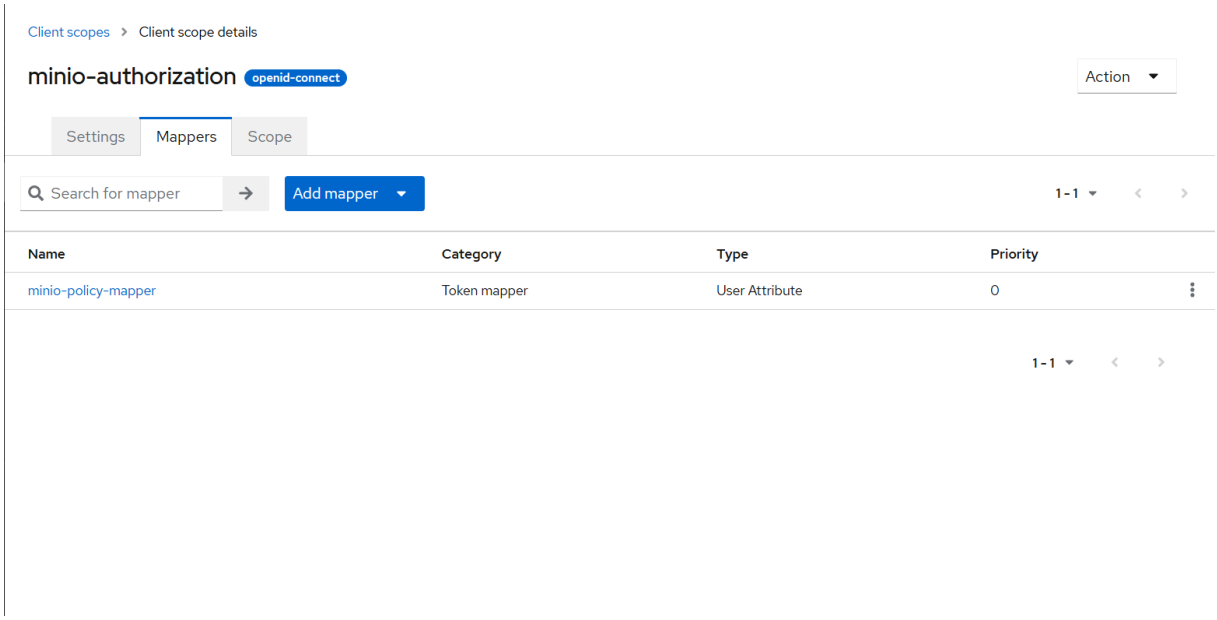

Figure 57. *Configuration of minio-authorization Client Scope in Keycloak*

Moreover, MinIO enforces access control using a policy-based mechanism that defines user permissions for specific buckets, objects, or actions. These policies allow administrators to configure

Funded by the European Union

granular access control, ensuring that only authorized users can perform specific operations on data and models stored in MinIO. MinIO has already some predefined policies, such as 'consoleAdmin', 'diagnostics', 'readonly', 'writeonly' and 'readwrite'.the context of HumAIne project, we decided to create custom policies to serve our ABAC implementation needs. Specifically, 6 custom policies were created, one policy per pilot that will grant users restricted access based on the pilot they are working on. The aforementioned policies are the following: 'smart-finance-policy', 'smart-healthcare-diabetes-policy', 'smart-healthcare-oncology-policy', 'smart-cities-policy', 'smart-manufacturing-policy' and 'smart-energy-policy'. The difference between them is the resources each custom policy allows access to.

Overall, MinIO policies specify permission at the bucket or object level, including actions such as GetObject (read), PutObject (write) and DeleteObject (delete). They also follow the S3-compatible JSON structure, allowing for easier integration and management with other systems using S3-based workflows. In our case, each policy allows for access to the specific buckets each one corresponds to. For example, the 'smart-energy-policy' allows access only to the buckets 'smart-energy-data', 'smart-energy-models' and 'smart-energy-results'. Some of the main actions that can be performed to the buckets and the objects contained in these buckets, specified by the Action section of the 'policy.JSON' file, include: list objects in a bucket, list all versions of objects in a versioned bucket, retrieve and apply bucket tags, retrieve and enable bucket logging settings, enable versioning, retrieve and configure bucket notification settings, retrieve, upload and delete an object and more. While users can upload, retrieve, and manage their data, bucket deletion is strictly restricted to prevent accidental loss.

5.5 Components Integration and Usage

This section details the end-to-end workflow of how the previously described components integrate and perform ABAC to ensure a seamless and secure authentication and authorization process in order to access the HumAIne data repository.

5.5.1 Process Description

The initial setup begins with the eIDAS pool registration. In the beginning, the eIDAS administrator adds to the test eIDAs node the HumAIne users along with their credentials and attributes through the eIDAS Web Management UI. This is done in order to simulate a real eIDAS node implementation where the users and their attributes (such as personal information) would already exist in this user pool.

During registration to the HumAIne platform, the user navigates to the HumAIne common frontend registration tab where they are presented with the eIDAS authentication form. After providing their credentials, these are sent to the authentication back-end which communicates with the eIDAS node to verify that they correspond to an existing eIDAS user. An access token is then issued and returned to the authentication backend and forwarded to the frontend.

In the next step, the user is required to provide their password one more time along with choosing the pilot that they are working on to be able to access the corresponding pilot buckets. The user has the option to choose between multiple pilots if they are participating in more than one pilot (i.e. learning paradigm developers work on multiple pilots). Through the Keycloak Java API a new user account is created, and a password is set. The attributes received from eIDAS are mapped as Keycloak user attributes and are being saved in a PostgreSQL along with the credentials.

As mentioned before, we created 6 different Keycloak groups that correspond to each pilot respectively. An extra group, managers group, was also created for the system administrators. Depending on the input by the user during registration, the backend assigns each user to their corresponding user group(s). Each pilot group has already an attribute attached to it. This attribute has the key 'policy' and a value which is similar to the group's name (e.g. smart-manufacturing-policy). Every Keycloak user that belongs to a Keycloak group, automatically inherits all the policies attached to the group as extra attributes.

Funded by the European Union

Additionally, an SSO functionality is configured between Keycloak and MinIO using Keycloak's OIDC protocol. As MinIO uses policies in order to authorize access to users and specifically to the buckets and objects stored, 6 different policies where created (one per pilot) and configured in MinIO, whose names are equivalent to the attributes attached to the Keycloak groups. An additional policy 'ConsoleManager' was created for administrative purposes.

Finally, 3 buckets per pilot were created to access and store data related to datasets, AI models and their respective results. Each user has 2 options to access the data repository after registering to the HumAIne platform. The first one is to submit their credentials to the HumAIne common frontend login page and navigate to the MinIO Web Console manually and the second one is to use MinIO APIs. In both cases, the JWT token issued from Keycloak contains the policy value, which is forwarded to MinIO to authorize access. Depending on the value, each user has access only to the data that the corresponding policy value refers to. For example, a user that selected during registration the energy pilot, is being assigned to the energy-pilot Keycloak group and inherits the 'energy-pilot-policy' as user attribute, so they can access only the energy pilot related buckets.

The integration between the components used to achieve ABAC are graphically presented in Figure 58.

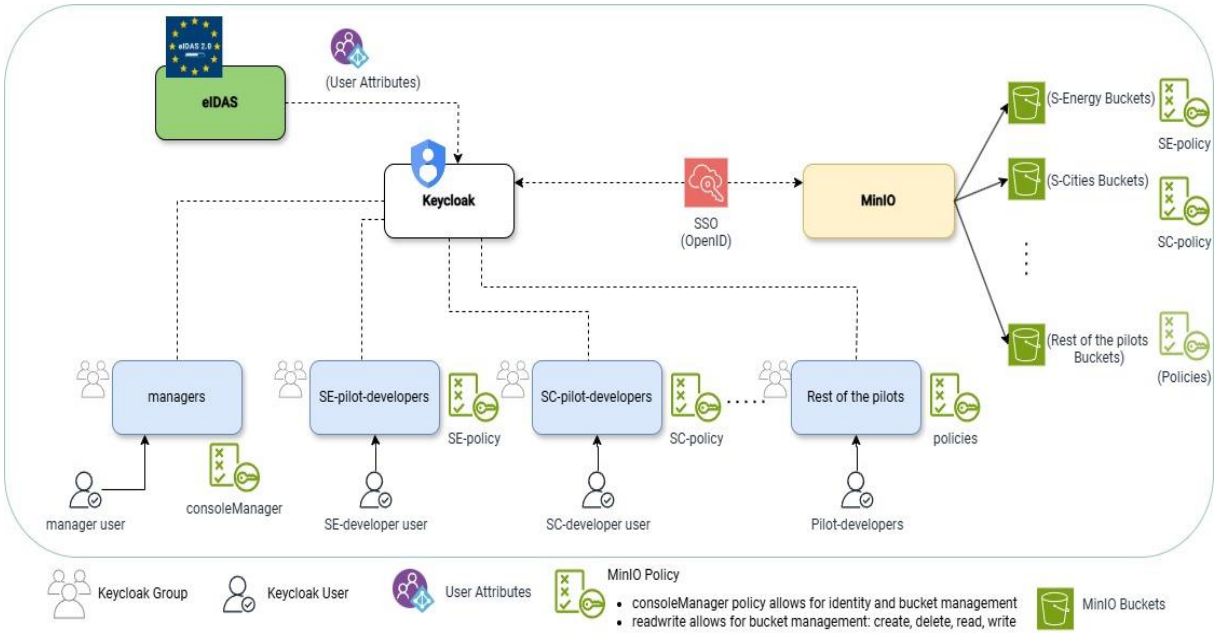

Figure 58. *Secure Data Collection and Management architecture for the HumAIne data repository.*

5.5.2 HumAIne Platform Infrastructure Deployment

Currently, both MinIO and Keycloak are installed on a Kubernetes cluster, which is the reference infrastructure for the HumAIne platform. The integration between the rest of the component's mentioned have been tested locally so that the information flow would be defined, and integration is going to be performed in the near future.

Several configurations have been made to ensure a seamless integration between these two components and other similar configurations will have to be made to integrate other components that will have to be included in the HumAIne platform, such as Kubeflow, to realize the complete expected architecture and ensure full functionality and support for all use cases. In particular, to make MinIO available for the HumAIne platform, it was decided to create a specific Tenant using the MinIO Operator console as a system administrator. In this way, it was possible to create the MinIO servers through a simple guided procedure using a UI, avoiding possible errors due to manual configurations.

The following screenshots show the MinIO Tenant configurations for the HumAIne platform inside the Operator console

Funded by the European Union

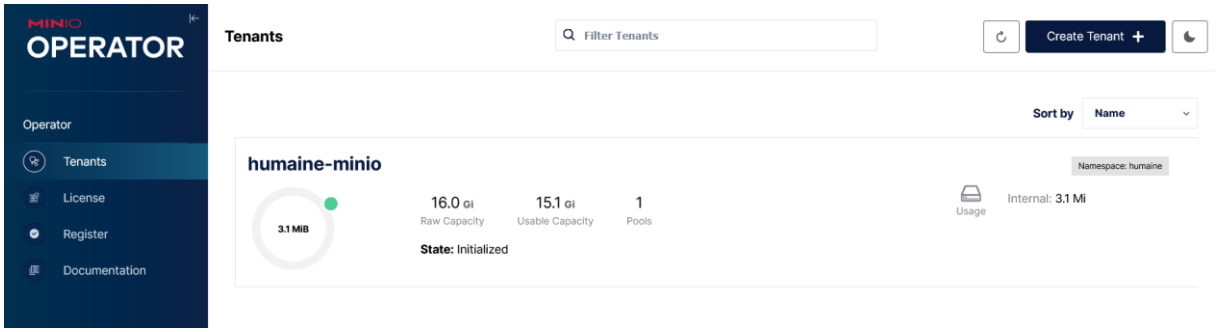

Figure 59. *MinIO Tenant Capacity Configurations*

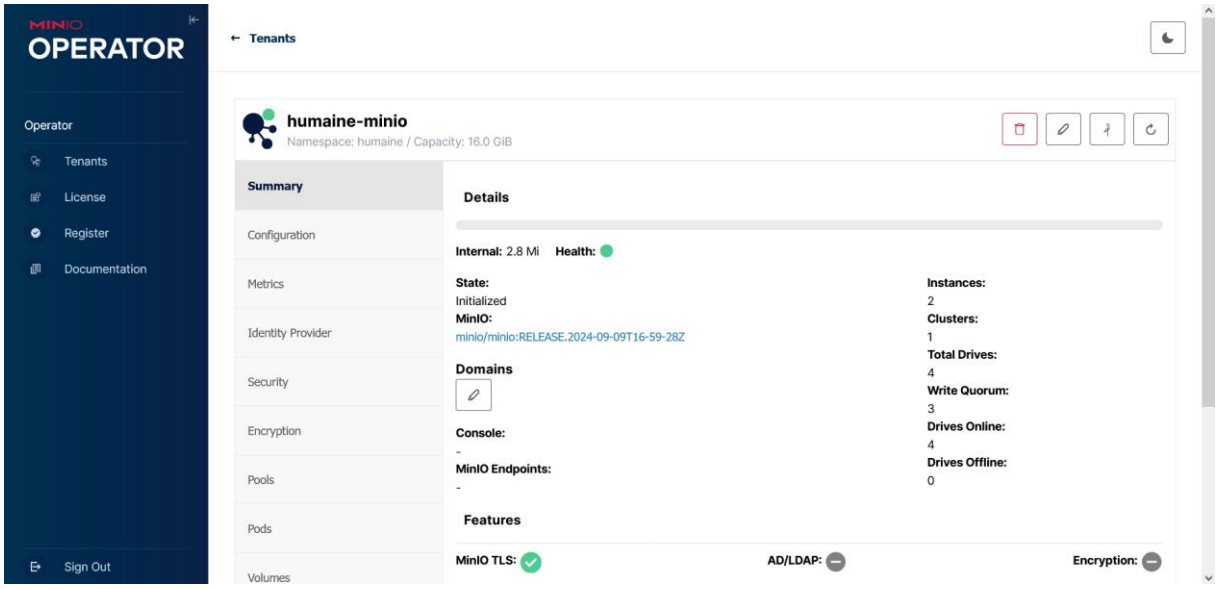

Figure 60. *MinIO Tenant Configurations Summarized.*

5.6 References for T4.1

1. Fischer, E. A., *Federal laws relating to cybersecurity: Overview of major issues, current laws, and proposed legislation*, 2014.
2. Kao, Y., Huang, K., Gu, H., & Yuan, S., "UCloud: A user-centric key management scheme for cloud data protection," *IET Information Security*, vol. 7, no. 2, pp. 144–154, Jun. 2013.
3. Al-Haj, A., Abandah, G., & Hussein, N., "Crypto-based algorithms for secured medical image transmission," *IET Information Security*, vol. 9, no. 6, pp. 365–373, Nov. 2015.
4. Liang, K., Au, M. H., Liu, J. K., Susilo, W., Wong, D. S., Yang, G., Yu, Y., & Yang, A., "A secure and efficient ciphertext-policy attribute-based proxy re-encryption for cloud data sharing," *Future Generation Computer Systems*, vol. 52, pp. 95–108, Nov. 2015.
5. Wang, S., Zhou, J., Liu, J. K., Yu, J., Chen, J., & Xie, W., "An efficient file hierarchy attribute-based encryption scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1265–1277, Jun. 2016.
6. Nabeel, M., & Bertino, E., "Privacy-preserving delegated access control in public clouds," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2268–2280, Sep. 2014.
7. Zaghloul, E., Zhou, K., & Ren, J., "P-MOD: Secure privilege-based multilevel organizational data-sharing in cloud computing," *IEEE Transactions on Big Data*, vol. 6, no. 4, pp. 804–815, Dec. 2020.

Funded by the European Union

8. Bethencourt, J., Sahai, A., & Waters, B., "Ciphertext-policy attribute-based encryption," in *Proceedings of the IEEE Symposium on Security and Privacy (SP)*, May 2007, pp. 321–334.
9. Almutairi, A., Sarfraz, M. I., & Ghafoor, A., "Risk-aware management of virtual resources in access-controlled service-oriented cloud datacenters," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 168–181, Jan. 2018.
10. Xu, S., Yang, G., Mu, Y., & Deng, R. H., "Secure fine-grained access control and data sharing for dynamic groups in the cloud," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2101–2113, Aug. 2018.
11. Yonetani, R., Boddeti, V. N., Kitani, K. M., & Sato, Y., "Privacy-preserving visual learning using doubly permuted homomorphic encryption," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2040–2050.
12. Hesamifard, E., Takabi, H., Ghasemi, M., & Wright, R. N., "Privacy-preserving machine learning as a service," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 3, pp. 123–142, Jun. 2018.
13. Li, T., Li, J., Liu, Z., Li, P., & Jia, C., "Differentially private Naive Bayes learning over multiple data sources," *Information Sciences*, vol. 444, pp. 89–104, May 2018.
14. Hong, J., Xue, K., Xue, Y., Chen, W., Wei, D. S. L., Yu, N., & Hong, P., "TAFC: Time and attribute factors combined access control for time-sensitive data in public cloud," *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 158–171, Jan. 2020.

Funded by the European Union

6 Platforms Customization and Bootstrapping (GFT)

6.1 Introduction

Platforms Customization and Bootstrapping focuses on adapting and integrating the components of WP4 to meet the unique requirements of the HumAIne project's pilots.

This Task (4.5) bridges the gap between the foundational infrastructure and the specific needs of each use case, ensuring that the solutions provided are tailored, robust, and capable of delivering real-world value.

Task 4.5 bridges WP4 paradigms - Active Learning, Neuro-Symbolic Learning, Swarm Learning, and Explainable AI - with pilot objectives, serving as a cornerstone in transforming the HumAIne vision into practical, impactful solutions.

This task also addresses critical aspects of data storage, model management, and trustworthiness, ensuring compliance with ethical standards and readiness for seamless collaboration among all partners. Through a systematic approach, Task 4.5 delivers a scalable, secure, and interoperable platform, laying the foundation for impactful AI-driven human-centric solutions.

6.2 Requirements Analysis

The customization and deployment of the WP4 platforms are driven by the specific requirements identified in WP2, in particular the deliverable D2.2.

Each pilot - spanning smart manufacturing, smart cities, healthcare, finance, and energy - presents distinct needs, from privacy-preserving data management to real-time decision-making capabilities. Task 4.5 analyses these requirements to ensure alignment between platform capabilities and pilot objectives.

Key considerations include the availability and structure of data, whether anonymized or synthetic, and its readiness for development. The analysis also evaluates storage solutions for both data and models, prioritizing traceability and security.

For the Smart Manufacturing pilot, the requirements emphasize a secure and privacy-preserving scheduling process using federated learning, dynamic multi-objective optimization for production, and real-time performance metrics to guide decisions.

The Smart Cities pilot requires automated data validation for council services, integration of Active Learning for continuous improvement, and Explainable AI to ensure transparency in decision-making processes.

The Smart Healthcare - Diabetes pilot focuses on personalized remote care supported by AI-driven virtual coaching, utilizing Neuro-Symbolic AI and Swarm Learning to combine domain expertise with patient data for enhanced healthcare outcomes. For the Smart Healthcare - Oncology pilot, requirements include tools for AI-driven medical reporting, image analysis for monitoring lesion progression, and multidisciplinary collaboration to support team-based decision-making.

In the Smart Finance pilot, the focus shifts to a Human-AI Resolution System (HAIRS) that ensures secure data handling, compliance with regulatory requirements, and transparent AI insights to support financial decision-making.

Lastly, the Smart Energy pilot demands real-time collaboration between AI tools and human operators for grid management, Explainable AI for trust and understanding, and continuous refinement through Active Learning.

By addressing these critical aspects, the Platforms Customization and Bootstrapping Task ensures that the infrastructure not only meets technical needs but also adheres to the trustworthiness and

Funded by the European Union

transparency principles integral to the HumAIne project. This analysis serves as a foundation for creating customized solutions, enabling the seamless integration of AI paradigms into each use case and ensuring readiness for iterative development and deployment.

6.3 Status of the Platform and Tools Configurations

The configuration of foundational components marks a significant milestone in establishing the HumAIne platform, setting the stage for its continued growth and evolution.

This section provides an overview of the progress made in configuring two critical platform elements: **MinIO** and **Keycloak**. These components serve as the initial building blocks, enabling robust data management and secure identity and access control within the platform.

MinIO, a high-performance object storage solution, ensures scalable and efficient handling of structured and unstructured data, a cornerstone for supporting diverse pilot requirements. Meanwhile, Keycloak, a flexible identity and access management tool, establishes secure and streamlined user authentication and authorization processes. Together, these configurations form the backbone of the HumAIne infrastructure, providing a reliable foundation for integrating advanced AI paradigms and facilitating seamless collaboration across project stakeholders.

The following subsections delve deeper into the work accomplished, highlighting technical configurations and the challenges addressed.

6.3.1 Kubernetes: Platform Solution

The HumAIne platform infrastructure, in which MinIO and Keycloak components are currently installed and configured, is based on a Kubernetes cluster deployed on the AWS cloud platform. Because Kubernetes is infrastructure agnostic, the HumAIne platform can be migrated to another cloud provider, public or private, or even to an on-premises infrastructure with little or no adjustments and without substantial architectural changes.

Kubernetes is an open source system for automating deployment, scaling, orchestration and management of containerized applications, and is used in this project to create a platform that ensures a high degree of security, reliability, scalability, following the DevOps, DevSecOps and GitOps best practices, using state-of-the-art tools to support all phases of CI/CD workflows, allowing application source and configuration versioning, building, deployment and runtime execution monitoring.

For the HumAIne platform, a dedicated namespace is set up to ensure proper isolation of installed components and complete management and control of resources, in terms of used CPU, memory and storage, by setting appropriate constraints and limits and thus without affecting the rest of the cluster.

The high-level architectural view of the HumAIne platform including MinIO and Keycloak components installed on Kubernetes is shown in the following picture. As depicted, Keycloak uses PostgreSQL database to store its configurations, while MinIO uses a proper physical storage infrastructure to store objects and other data

Funded by the European Union

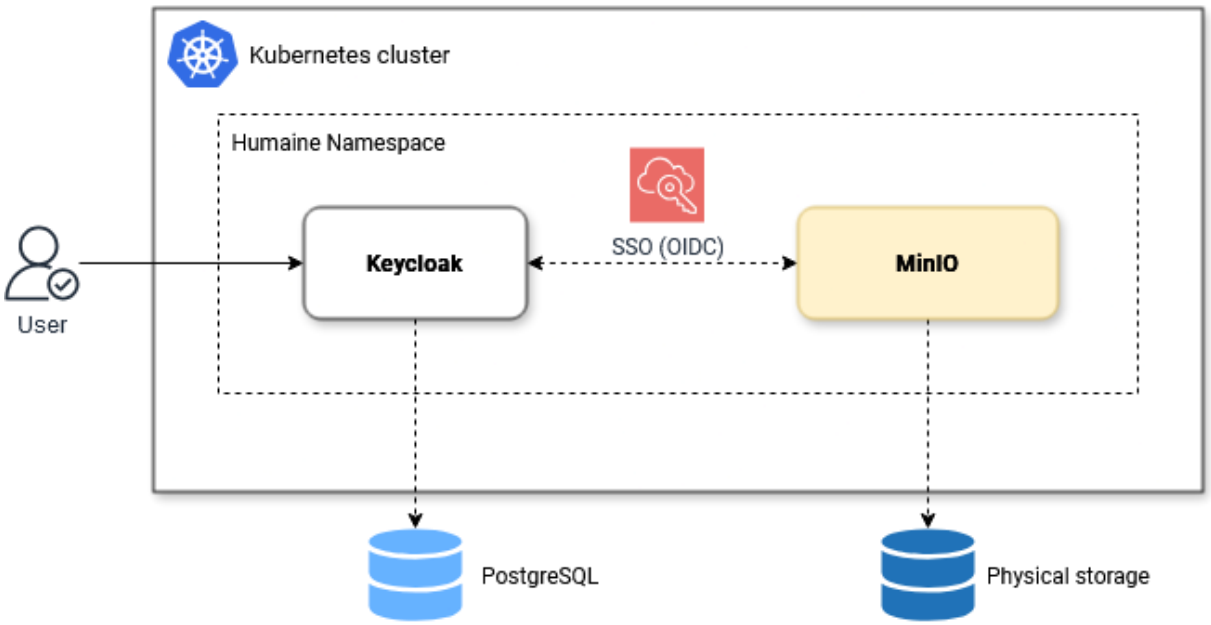

Figure 61. *Deployment of the system in Kubernetes cluster.*

Both MinIO and Keycloak provide web administration console exposed on Internet, and their DNS configuration are set in Route 53, the AWS highly available and scalable cloud domain name system (DNS) service.

In Keycloak, a specific realm named ***humaine*** is set in order to store all configuration settings needed for proper IAM management in the platform. Both MinIO and Keycloak must be configured to allow Single Sign On (SSO) authentication using OpenID Connect (OIDC) protocol. To achieve this goal, Keycloak is configured as OpenID provider for MinIO. All needed configurations for OIDC can be made using the administration console UI for both components.

The following picture shows the configuration of Keycloak OpenID provider in MinIO.

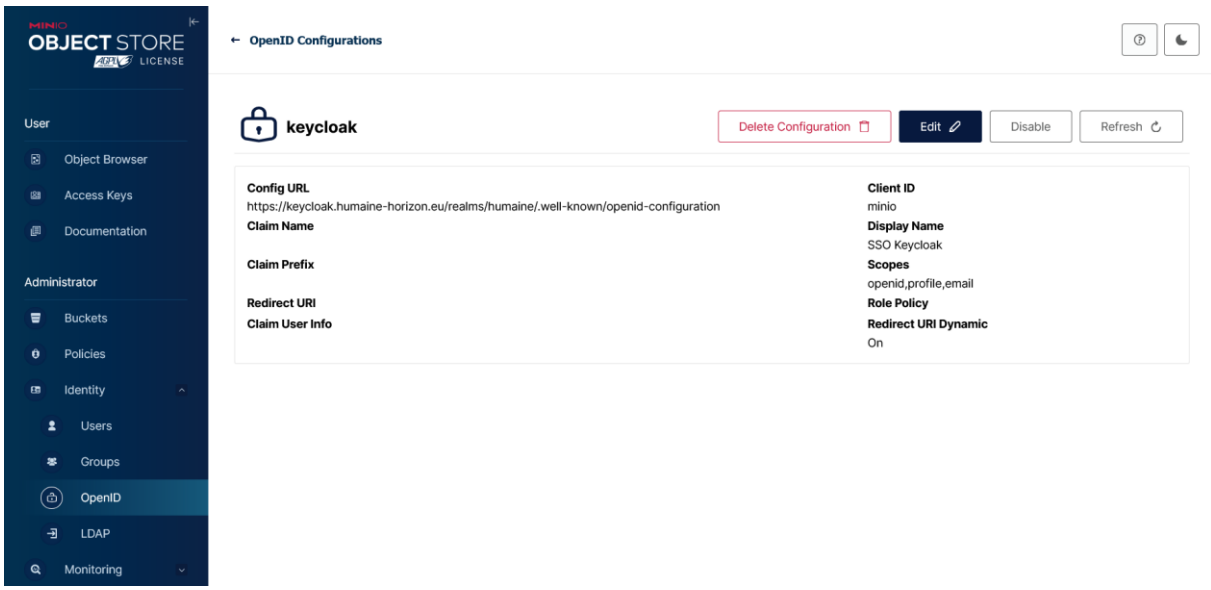

Figure 62. *Screenshot of the configuration of Keycloak OpenID provider in MinIO.*

Funded by the European Union

The procedure to configure MinIO for authentication using Keycloak can be found in the official MinIO documentation[6].

Moreover, the platform exposes the MinIO REST API to allow management of buckets and objects. The S3 API compatibility can be found in the official MinIO documentation[7]

6.3.2 Kubernetes Components Deployment and Configuration

In Kubernetes, the HumAIne platform includes several components, such as the already installed Keycloak and MinIO, which are composed of a set of specific resources. Some of these are namespace-bound resources while others are cluster resources.

Some typical resources used to deploy HumAIne platform components are the following:

- *Pod*, the smallest computation unit. Pods can include one or more containers, and each container represents a microservice.
- *Service*, in charge of traffic routing and load balancing between associated pods
- *Persistent Volume Claim*, containing a request for a storage needed by the application
- *Persistent Volume*, which maps and manages physical storage
- *Ingress*, the component that manages external access by providing http/s routing rules to the services within a cluster

The following picture shows the current deployment architecture of the HumAIne platform that includes MinIO and Keycloak, with their respective resources. The picture highlights the resources associated with the HumAIne namespace and those that are cluster scoped.

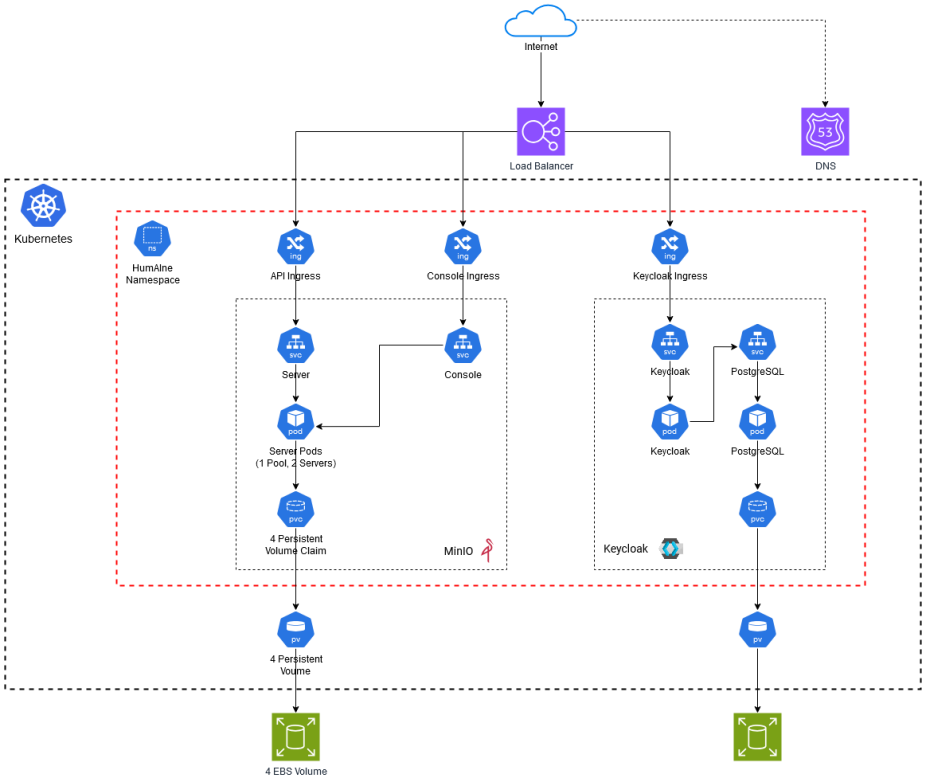

Figure 63. *Current deployment architecture of the HumAIne platform.*

6 https://min.io/docs/minio/kubernetes/upstream/operations/external-iam/configure-keycloak-identity-management.html

7 https://min.io/docs/minio/linux/reference/s3-api-compatibility.html

Funded by the European Union

MinIO uses two different Ingress, one to expose the REST API and the other for the console. The two Ingresses point to two different and dedicated Services, which in turn reference the same Pods, using different ports.

Since a pool with two servers has been configured on MinIO, two separate Pods are needed, one for each server. Each server manages two different volumes and therefore there are 4 volumes in total. For this reason, each Pod will use two different Persistent Volume Claims, and each claim will be bound to a specific Persistent Volume resource which in turn remaps the physical storage. Currently, each volume has a capacity of 4Gi, resulting in a total pool capacity of 16Gi.

The following pictures are screenshots taken from the MinIO Operator console and show the current configurations of the pool, servers and their related volumes.

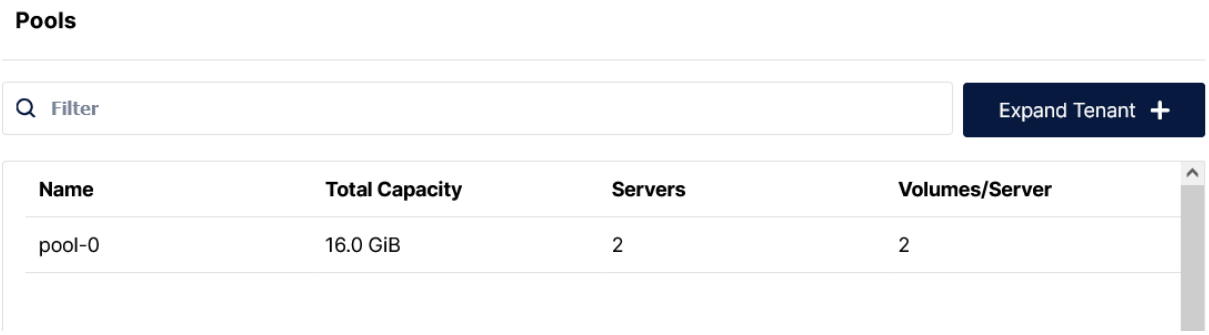

Figure 64. *Current configuration of the pool.*

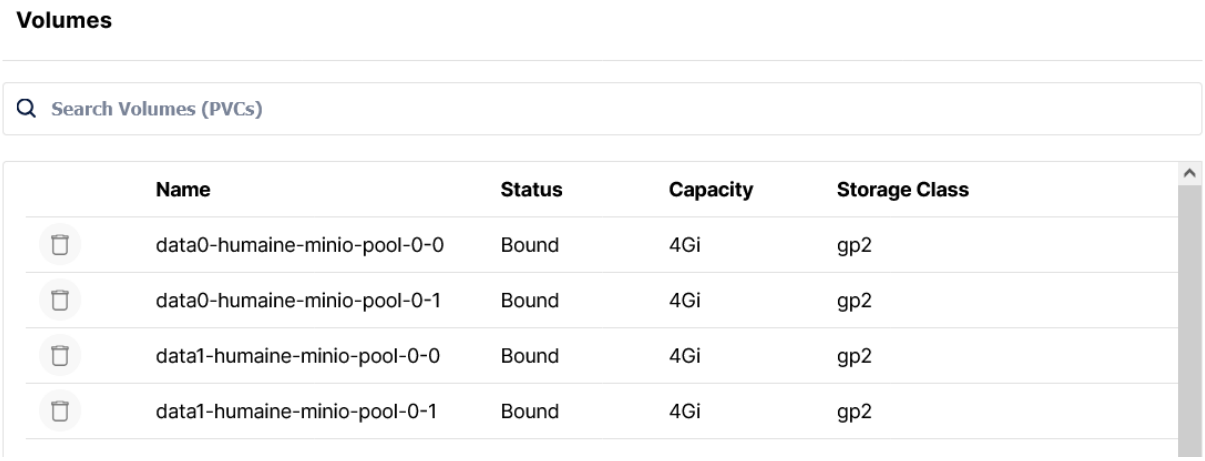

Figure 65. *Current configuration of servers and their related volumes.*

The Keycloak component is installed along with its supporting PostgreSQL database. Only one Ingress is needed to expose the Keycloak administration console, while there are two distinct Services, for the Keycloak and PostgreSQL Pods respectively. The PostgreSQL Pod uses a Persistent Volume Claim that in turn references a Persistent Volume with a current capacity of 1GiB. Communication between the Keycloak Pod and the supporting database is done through the PostgreSQL Service.

6.3.3 Kubernetes Implementation in AWS

As described above, the Kubernetes cluster for HumAIne project is installed on the AWS platform, currently located in the EU-central region (Frankfurt), and distributed across two availability zones to ensure high availability. The use of a region located in Europe ensures compliance with European legal regulations regarding data processing and privacy.

Funded by the European Union

Currently, the cluster is made up of 6 worker nodes, which are EC2 instances of type t3a.xlarge, with 4 VCPUs and 16GiB of memory. The cluster control plane is fully managed by the EKS service.

Referencing the deployment architecture in the previous picture, other AWS accessory services are used to complete the HumAIne platform. There is an ELB (Elastic Load Balancer) component that is responsible for distributing traffic and routing it appropriately to the different Ingress components in the cluster. This load balancer is a high availability component managed directly by AWS, with autoscaling capabilities to manage any traffic peaks.

The Route 53 service is also used to provide the DNS. This service keeps track of all DNS records belonging to the *humaine-horizon.eu* domain, including those for specific components and applications.

Finally, physical storage layer is currently implemented by scalable and high-performance EBS (Elastic Block Storage) volumes, but the system can also be evolved as needed to also use different storage components, such as EFS (Elastic File System), a fully elastic file storage able to share file data without provisioning or managing storage capacity and performance in advance.

Funded by the European Union

7 Conclusion

This deliverable details the design and implementation of three core AI platforms—**Active Learning, Swarm Learning, and Neuro-Symbolic AI**—designed to empower human-AI collaboration for enhanced decision-making in dynamic settings. The deliverable examines these platforms' development and initial software implementations, emphasizing the integration of multiple AI paradigms within applications and highlighting the need for trustworthy AI through Explainable AI (XAI). Additionally, the deliverable reports on the work achieved in supporting tasks, developing a secure data collection and management platform as well as its deployment.

Key Achievements:

- **Active Learning Platform**: The initial version of the Active Learning platform is now implemented. It has been tested in various pilot programs, showcasing its adaptability to diverse domains like Smart Cities, Smart Healthcare, Smart Finance, and Smart Energy. Notably, the platform achieved an impressive **acceleration of knowledge acquisition exceeding 130% in the Smart Finance scenario**, significantly surpassing the initial target of 50% set for M36. This success indicates the platform's potential to optimize model training and enhance human-AI communication by actively selecting the most informative data points.
- **Swarm Learning Platform**: A stand-alone Swarm Learning platform has been successfully developed and tested, demonstrating its effectiveness in collaborative learning environments. In the Manufacturing pilot, the platform achieved a **10% increase in accuracy compared to decentralized training without Swarm Learning**, highlighting its ability to enhance decision support by securely sharing knowledge between distributed clients. Furthermore, the platform exhibits **scalability, reducing training time by over 25% compared to centralized training algorithms**, making it a promising solution for handling large-scale, real-world applications.
- **Neuro-Symbolic AI Platform**: The Neuro-Symbolic AI platform, applied to the diabetes care use case, is in its early developmental stages. This platform leverages **synthetic patient data and expert knowledge to generate AI-driven advice for diabetes management**, focusing on integrating data-driven insights with logical reasoning to support human-centered systems. The platform incorporates a **human-in-the-loop mechanism** for advice review and approval by medical professionals, ensuring safety, compliance with medical standards, and responsible AI implementation within sensitive healthcare domains.
- **Secure data collection and management:** Initial version of the platform has been deployed in Kubernetes cluster, platform enables attribute-based access to the data as well as secure identification of the users.

Looking Ahead:

Future work on the Active Learning, Swarm Learning, and Neuro-Symbolic AI (NSAI) platforms within the HumAIne project will focus on enhancing usability, scalability, and integration across various pilots. All the paradigms will focus on the finalization of standalone implementations and provide support for integrated demos. **Active learning** will explore named entity recognition in medical scenarios as well as active learning based on natural language processing (NLP), including large language models (LLMs), and finally mimicking complex numerical models with data-driven approach. **Swarm Learning** will tackle scalability challenges in manufacturing and oncology pilots by increasing participant numbers, aiming for a higher forecast accuracy improvement, and optimizing communication protocols for efficiency. The **NSAI** platform will enhance symbolic reasoning, specifically for the diabetes pilot, improving

Funded by the European Union

user interaction and automating symbolic rule generation using constrained LLMs. Task 4.1 will finalize the integration of secure data management components (Keycloak, eIDAS, and MinIO) in the HumAIne platform to ensure privacy, transparency/trustworthiness and accessibility, while Task 4.5 will provide support for streamlining the learning paradigms integration.

Crucial points:

- KPIs for evaluation of the learning paradigms include arbitrary percentages for improvements, which greatly depend on the dataset and problem. KPIs need to be rewritten to better reflect the contributions of the methodologies.

Funded by the European Union
