AI4Gov

Trusted AI for Transparent Public Governance fostering Democratic Values

Deliverable 4.2

Trustworthy, Explainable, and unbiased AI V2

<31-12-2024>

Version 1.0

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 or the Agency. Neither the European Union nor the granting authority can be held responsible for them.

PROPERTIES	
Dissemination level	Public
Version	<1.0>
Status	< Final >
Beneficiary	
License	This work is licensed under a Creative Commons Attribution-No Derivatives 4.0 International License (CC BY-ND 4.0). See: https://creativecommons.org/licenses/by-nd/4.0/

AUTHORS		
	Name	**Organisation**
Document leader	Alenka Guček	JSI
Participants	Kostis Mavrogiorgos	UPRC
	Fabiana Fournier, Lior Limonad, Shlomit Gur	IBM
	George Manias	UPRC
	Matej Kovačič	JSI
Reviewers	Xanthi Papageorgiou	UBI
	Silvina Pezzetta	WLC

VERSION HISTORY				
Version	**Date**	**Author**	**Organisation**	**Description**
0.1	23/10/24	Alenka Guček	JSI	Initial ToC
0.2	08/11/24	Kostis Mavrogiorgos	UPRC	Updated Sections 4.4 and 4.5 based on the finalized implementation
0.25	28/11/24	Fabiana Fournier	IBM	IBM input section 3
0.3	09/12/24	George Manias	UPRC	Updates on Sections 4.3, 4.4 and 4.5
0.4	12/12/24	Alenka Guček	JSI	Updates on Section 2 and merger of contributions
0.5	16/12/24	Xanthi Papageorgiou	UBI	1st Internal Review
0.6	17/12/24	Silvina Pezzetta	WLC	1st Internal Review
1.0	18/12/24	Alenka Guček	JSI	Integration of comments

Table of Contents

List of figures

List of Tables

Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
AQA	Abstractive Question Answering
ARIMA	Autoregressive Integrated Moving Average
BERT	Bidirectional Encoder Representations from Transformers
BP	Business Process
BPIC	Business Process Intelligence Challenge
BPM	Business Process Management
CD	Causal Discovery
CDQA	Closed Domain Question Answering
CEP	Complex Event Processing
CRO	Caregiver-Reported Outcomes
DL	Deep Learning
DNN	Deep Neural Networks
EQA	Extractive Question Answering
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
LiNGAM	Linear Non-Gaussian Acyclic Model
LLM	Large Language Model
ML	Machine Learning
MSE	Mean Square Error
MSR	Minimal Sufficient Reasons
NLP	Natural Language Processing
ODQA	Open Domain Question Answering
OECD	Organisation for Economic Co-operation and Development
PD	Process Discovery
PLMs	Pre-trained Language Models
PRO	Patient-Reported Outcomes
RAG	Retrieval-Augmented Generation
QA	Question - Answering
SAX	Situation-Aware eXplainability
SAX4BPM	SAX for BPM
SENN	Self- Explainable Neural Network
SSC	Sustainable Smart Cities
SVG	Scalable Vector Graphics
VUF	Virtualized Unbiasing Framework
XAI	eXplainable Artificial Intelligence
XGBoost	eXtreme Gradient Boosting

Abstract

This document, D4.2 "Trustworthy, Explainable, and Unbiased AI V2", has been developed within the framework of WP4, "Trustworthy and Unbiased AI," and is a follow up to the D4.1 "Trustworthy, Explainable, and Unbiased AI V1". This deliverable offers an overview of the technical advancements within WP4 and will be complemented with the D4.4 that focuses on the visualization aspect of the proposed technical solutions, that is planned in M27.

The document details the following technologies:

• the Virtualized Unbiasing Framework (VUF) for AI & Big Data and Bias Detector Toolkit for data incompleteness detection,

• the SAX4BPM Library, and a novel framework for DNN classifiers is introduced, so-called Self-Explaining Neural Networks (SENNs) with Minimal Sufficient Reasons (MSRs)

• strategies to Improve Citizen Engagement and Trust utilizing Natural Language Processing (NLP) and improved functionality of the Multilingual Bias Classification component.

The above mentioned technologies are explained in detail, and the deliverable is complemented with demonstrators effectively showcasing the technologies application to the use cases.

1 Introduction

1.1 Purpose and scope

This deliverable is the results of the work conducted in M12-M24 for tasks 4.1, 4.2 and 4.3 under the Work Package 4. The WP started in M1 and continues until M27. This is the second version of this deliverable, while the first one was due in M12. At this point, the deliverable describes the work on the aforementioned technical tasks, specifically the methodologies and services that were developed for the use case scenarios, that were identified under D6.1 in M6 and further elaborated under D6.2 in M18.

This deliverable is being released on M24 of the project, and its main purpose is to present AI4Gov progress in the development of the Bias Detector Toolkit, the SAX4BPM library and the NLP-enhanced analytics methodologies and applications.

1.2 Document structure

The deliverable is structured as follows: Chapter 1 introduces the document, including the purpose and scope, and document structure. The following chapters are dedicated to individual tasks from WP4, focusing on a diverse range of aspects towards Trustworthy, Explainable and Unbiased AI. Chapter 2 provides detailed information on T4.1, Virtualized Unbiasing Framework (VUF) for AI & Big Data. Chapter 3 is focused on the progress of work for T4.2, XAI Library. Chapter 4 introduces methodologies developed for T4.3, Improve Citizen Engagement and Trust utilizing NLP. Chapter 5 concludes the deliverable, summarizing the findings and describing the next steps. Chapter 6 includes the reference list. Since this deliverable is a demonstrator, videos for each of the tasks are available here.

1.3 Updates with respect to previous version (if any)

This is the second of the two versions of this deliverable on Trustworthy, Explainable and Unbiased AI. This deliverable is an update of D4.1 from M12, and it provides the extended information on the technologies developed under T4.1, T4.2 and T4.3. For T4.1, new version of the implemented catalogue is presented, together with bias detection pipeline for rare disease and police data. Additionally, research on detection of bias in news is presented and updates on the implementation for the use cases are described. For T4.2, updates on Situation Aware eXplainability (SAX) and the SAX4BPM library are provided, and a novel framework for DNN classifiers is introduced, the so-called Self-Explaining Neural Networks (SENNs) with Minimal Sufficient Reasons (MSRs). For T4.3, the final implementation for all the use cases is presented alongside with additional functionalities that specifically aim to engage the citizens and encourage them to interact with the trained ML models and benefit from the said models. Moreover, the improved functionality of the Multilingual Bias Classification component is introduced based on its integration with a Retrieval-Augmented Generation (RAG) mechanism towards a two-fold objective. On one hand to successfully evaluate the performance of Pre-trained Language Models (PLMs) and improve the identification of types of biases in the text. While in parallel, the utilization of the RAG will enhance the explainability and trust on the final results as it will provide detailed justifications for the classification results by retrieving relevant information from a database of explanations.

2 Virtualized Unbiasing Framework (VUF) for AI & Big Data

This section of the deliverable introduces updates on Virtualized Unbiasing Framework, that consists of the Bias Detector Toolkit and specific tools to detect bias in AI systems. The Bias Detector Toolkit was in detail described in D4.1 and is designed to function as a visual catalogue synthesizing diverse tools tailored for detecting and mitigating biases in AI systems. Here we rename it to the Virtualized Unbiasing Framework Catalogue and provide updates on its design and implementation. As further development has been made on tools to detect bias, new section titled Bias Detection Toolkit is added and describes in detail the analytics pipeline to detect bias in two use cases. We also describe the detection of bias in a research context: analysis of news spreading regarding barriers. We conclude this section with updates on the use cases, which are in further detail explained in D6.4.

2.1 VUF Catalogue

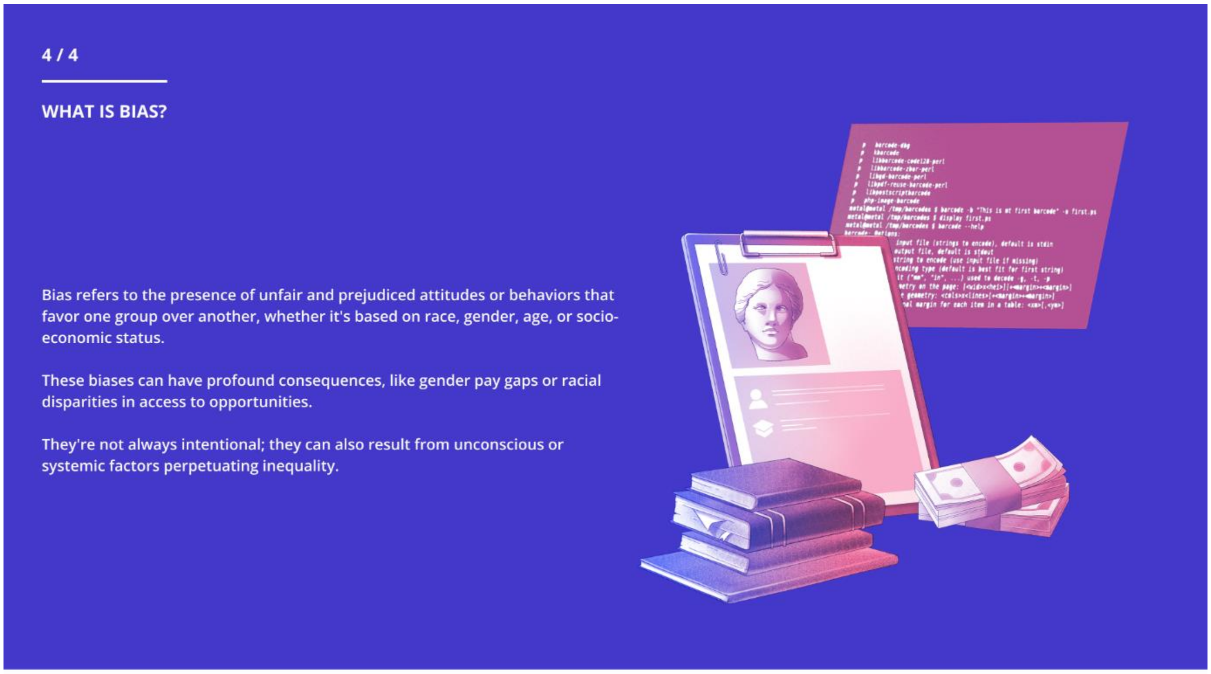
Virtualized Unbiasing Framework Catalogue (previously named a Bias Detector Toolkit) is designed to function as a visual catalogue synthesizing diverse tools tailored for detecting and mitigating biases in AI systems. As we've laid down the structure of the catalogue in D4.1, it consists of the following subcomponents:

- Scrollytelling narrative to introduce the complexity of bias.
- Real life examples that ensure that all stakeholders grasp the significance of bias mitigation.
- Stages of training of AI models.
- Catalogue of bias detection and mitigation strategies, structured as visual summary.

We additionally present a new design of the visual synthesis of tools provided by OECD. The VUF Catalogue has now been developed and the specific sections are explained in detail below.

2.1.1 Scrollytelling narrative to introduce the complexity of bias

Scrollytelling narrative has the aim to inform both general public and developers on the concepts of AI and bias. Employed as an educational strategy, scrollytelling guides audiences through the complexities of bias in a step-by-step manner, utilizing visual metaphors and scalable vector graphics (SVG) animations. In D4.1 we proposed a visual metaphor of a lighthouse, but during development we decided against it, simplifying the message by making it more approachable with visuals that better fit the concepts. Complexity of bias is described first, followed by what is AI and by the concepts of bias in AI (see example in Figure 1).



4 / 4

WHAT IS BIAS?

Bias refers to the presence of unfair and prejudiced attitudes or behaviors that favor one group over another, whether it's based on race, gender, age, or socio-economic status.

These biases can have profound consequences, like gender pay gaps or racial disparities in access to opportunities.

They're not always intentional; they can also result from unconscious or systemic factors perpetuating inequality.

```
p   barcode-dbg
p   kbarcode
p   libbarcode-code128-perl
p   libbarcode-zbar-perl
p   libgd-barcode-perl
p   libpdf-reuse-barcode-perl
p   libpostscriptbarcode
p   php-image-barcode
metal@metal /tmp/barcodes $ barcode -b "This is mt first barcode" -o first.ps
metal@metal /tmp/barcodes $ display first.ps
metal@metal /tmp/barcodes $ barcode --help
barcode: Options:
          input file (strings to encode), default is stdin
          output file, default is stdout
          string to encode (use input file if missing)
          ncoding type (default is best fit for first string)
          it ("mm", "in", ...) used to decode -g, -t, -p
          metry on the page: [<wid>x<hei>][+<margin>+<margin>]
          t geometry: <cols>x<lines>[+<margin>+<margin>]
          nal margin for each item in a table: <xm>[,<ym>]
```

Figure 1: Example of scrollytelling screen that explains what is bias

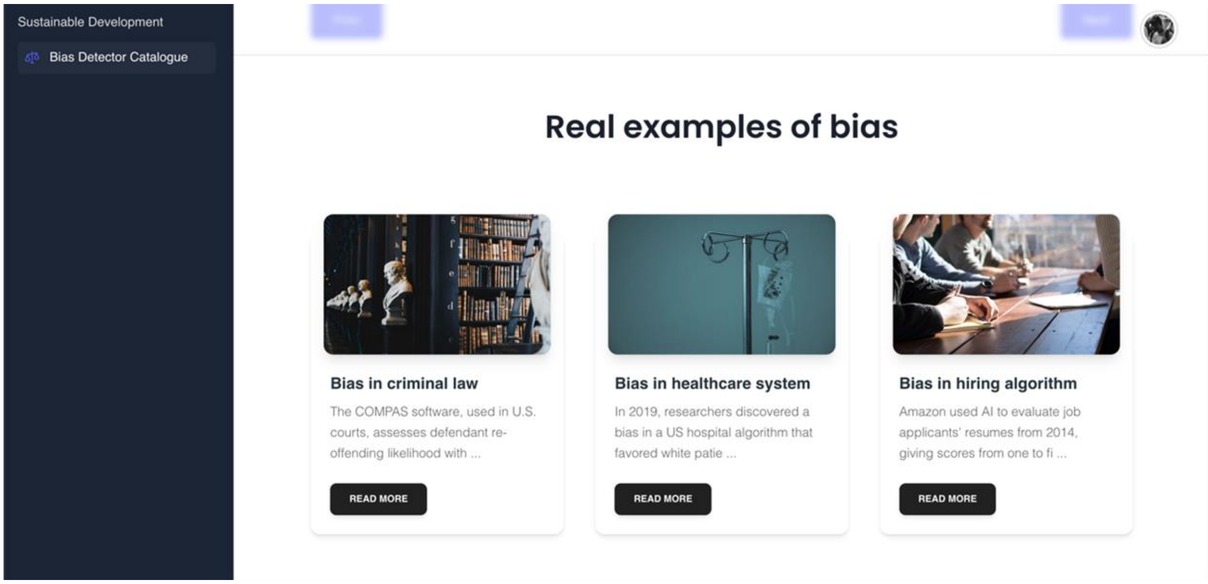
2.1.2 Examples of bias in AI

The second section provides real examples of bias occurrences in different business sectors (see Figure 2). Bias in real-world applications of ML has manifested in various forms (as we already explained in detail in D4.1), raising ethical concerns and highlighting the importance of responsible AI development. For each example (see Figure 3), there is:

- a short description that summarizes the problem,
- the solution that either solved or mitigated the problem,
- some reference material for further research on the topic.

Real-life examples of bias in policies serve as digestible illustrations that underscore the critical importance of understanding and addressing systemic inequalities. For each step, a short description is provided along with ways that bias can occur. The intended use for this section is to provide policy makers, stakeholders and ML engineers with the information needed in order to prevent the occurrence of bias in workflows.

The importance of recognizing these biases lies in the potential to empower a general audience. When individuals comprehend the tangible impacts of biased policies, they are better equipped to advocate for change, engage in informed discussions, and challenge discriminatory practices. By offering relatable examples, we empower the general audience to navigate and contribute to conversations about fairness, justice, and equitable policy reform in their communities and beyond.



Real examples of bias

Bias in criminal law

The COMPAS software, used in U.S. courts, assesses defendant re-offending likelihood with ...

READ MORE

Bias in healthcare system

In 2019, researchers discovered a bias in a US hospital algorithm that favored white patie ...

READ MORE

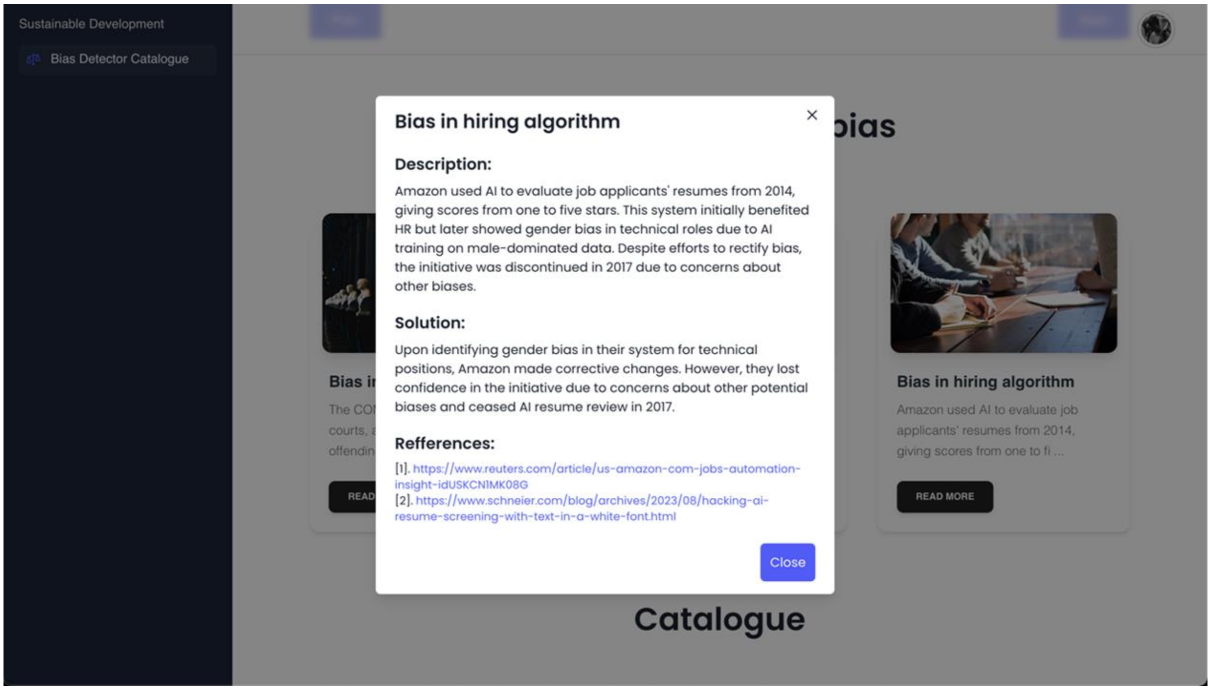
Bias in hiring algorithm

Amazon used AI to evaluate job applicants' resumes from 2014, giving scores from one to fi ...

READ MORE

Figure 2: Real life examples of bias in AI

These examples underscore the importance of addressing bias in ML systems to ensure fair and equitable outcomes and to provide motivation to do better.



Bias in hiring algorithm

Description:

Amazon used AI to evaluate job applicants' resumes from 2014, giving scores from one to five stars. This system initially benefited HR but later showed gender bias in technical roles due to AI training on male-dominated data. Despite efforts to rectify bias, the initiative was discontinued in 2017 due to concerns about other biases.

Solution:

Upon identifying gender bias in their system for technical positions, Amazon made corrective changes. However, they lost confidence in the initiative due to concerns about other potential biases and ceased AI resume review in 2017.

Refferences:

[1]. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G
[2]. https://www.schneier.com/blog/archives/2023/08/hacking-ai-resume-screening-with-text-in-a-white-font.html

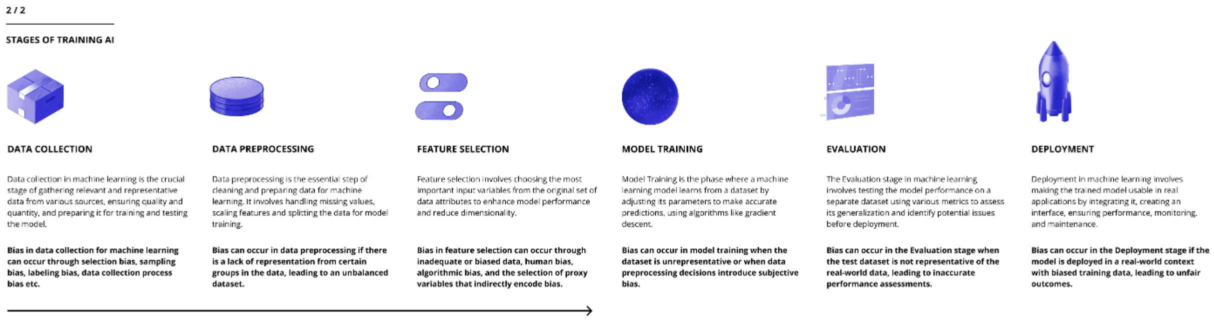
Bias in hiring algorithm

Amazon used AI to evaluate job applicants' resumes from 2014, giving scores from one to fi ...

Figure 3: Real life examples of bias in AI of a detailed explanation

2.1.3 Stages of training AI

The third section provides information on the training stages of AI (Figure 4). This is a prelude to the catalogue, that explains on a high level each stage and provides information of what types of biases can occur at each of the stages.



2 / 2

STAGES OF TRAINING AI

DATA COLLECTION

Data collection in machine learning is the crucial stage of gathering relevant and representative data from various sources, ensuring quality and quantity, and preparing it for training and testing the model.

Bias in data collection for machine learning can occur through selection bias, sampling bias, labeling bias, data collection process bias etc.

DATA PREPROCESSING

Data preprocessing is the essential step of cleaning and preparing data for machine learning. It involves handling missing values, scaling features and splitting the data for model training.

Bias can occur in data preprocessing if there is a lack of representation from certain groups in the data, leading to an unbalanced dataset.

FEATURE SELECTION

Feature selection involves choosing the most important input variables from the original set of data attributes to enhance model performance and reduce dimensionality.

Bias in feature selection can occur through inadequate or biased data, human bias, algorithmic bias, and the selection of proxy variables that indirectly encode bias.

MODEL TRAINING

Model Training is the phase where a machine learning model learns from a dataset by adjusting its parameters to make accurate predictions, using algorithms like gradient descent.

Bias can occur in model training when the dataset is unrepresentative or when data preprocessing decisions introduce subjective bias.

EVALUATION

The Evaluation stage in machine learning involves testing the model performance on a separate dataset using various metrics to assess its generalization and identify potential issues before deployment.

Bias can occur in the Evaluation stage when the test dataset is not representative of the real-world data, leading to inaccurate performance assessments.

DEPLOYMENT

Deployment in machine learning involves making the trained model usable in real applications by integrating it, creating an interface, ensuring performance, monitoring, and maintenance.

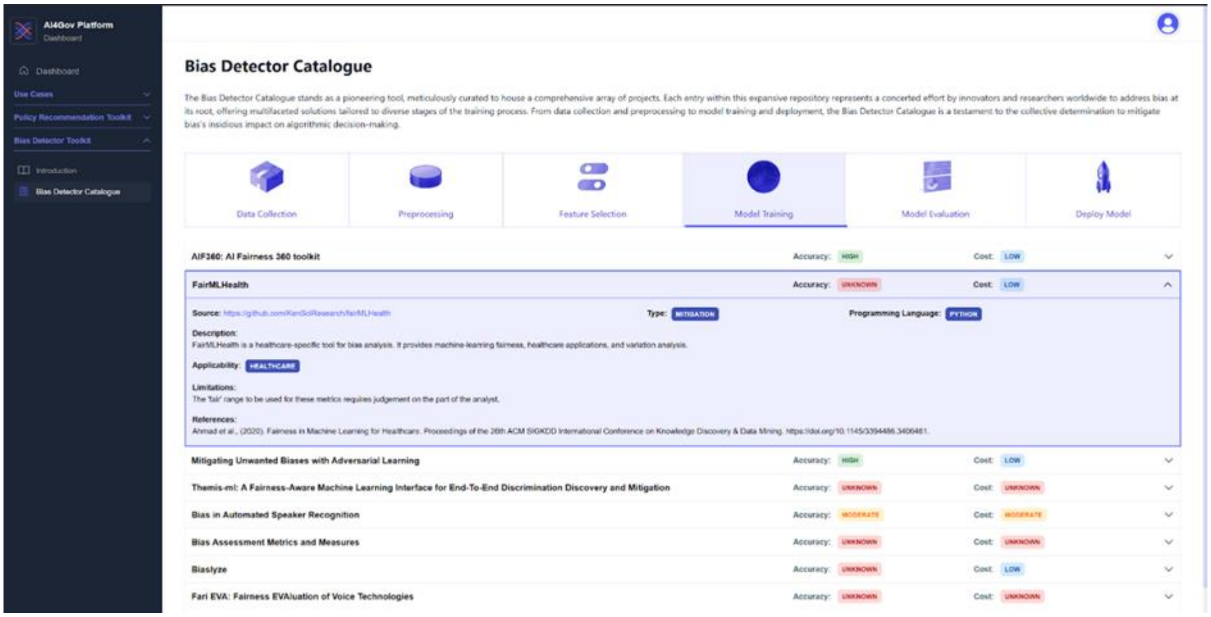
Bias can occur in the Deployment stage if the model is deployed in a real-world context with biased training data, leading to unfair outcomes.

Figure 4: Stages of training AI

Training starts with data collection, where relevant datasets are gathered to be used by the model. Afterwards, data pre-processing plays a crucial role, emphasizing the cleaning, normalization, and transformation of raw data to facilitate efficient and effective learning. Next is feature selection, where key attributes are identified to improve the model's performance and reduce its complexity. In the model training phase, the processed data is fed into the chosen algorithm or architecture, allowing it to recognize patterns and relationships. The trained model is then assessed with a separate validation set to measure its accuracy and ability to generalize. If the results are satisfactory, the model moves to deployment, making it functional for real-world use. Ongoing monitoring and updates may be required to maintain its effectiveness, adjusting to shifts in data distribution or evolving challenges. This iterative process of data-driven learning — from collection to deployment — forms the backbone of machine learning model development.

2.1.4 Bias Detection Catalogue

The Catalogue builds upon the above-mentioned training steps, providing tools and mitigation techniques for each of the steps. In D4.1 we reported to have collected the bias detection and mitigation tools, and in D4.2 we now present the Bias Detector Catalogue (Figure 5). As reported in D4.1, the central idea is not to create another text heavy framework, but to provide a visual summary of existing bias detection and mitigation strategies in an approachable and easy-to-grasp format. The user can therefore interact with the platform and choose for which of the stages they are looking for the solution. Tools are then presented in a visual table, with links and additional information. An example of a json input for the Catalogue can be seen in Figure 6.



Bias Detector Catalogue

The Bias Detector Catalogue stands as a pioneering tool, meticulously curated to house a comprehensive array of projects. Each entry within this expansive repository represents a concerted effort by innovators and researchers worldwide to address bias at its root, offering multifaceted solutions tailored to diverse stages of the training process. From data collection and preprocessing to model training and deployment, the Bias Detector Catalogue is a testament to the collective determination to mitigate bias's insidious impact on algorithmic decision-making.

AIF360: AI Fairness 360 toolkit Accuracy: HIGH Cost: LOW

FairMLHealth Accuracy: UNKNOWN Cost: LOW

Source: https://github.com/KenSciResearch/fairMLHealth **Type:** MITIGATION **Programming Language:** PYTHON

Description:
FairMLHealth is a healthcare-specific tool for bias analysis. It provides machine-learning fairness, healthcare applications, and variation analysis.

Applicability: HEALTHCARE

Limitations:
The 'fair' range to be used for these metrics requires judgement on the part of the analyst.

References:
Ahmad et al., (2020). Fairness in Machine Learning for Healthcare. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. https://doi.org/10.1145/3394486.3406461.

Mitigating Unwanted Biases with Adversarial Learning Accuracy: HIGH Cost: LOW

Themis-ml: A Fairness-Aware Machine Learning Interface for End-To-End Discrimination Discovery and Mitigation Accuracy: UNKNOWN Cost: UNKNOWN

Bias in Automated Speaker Recognition Accuracy: MODERATE Cost: MODERATE

Bias Assessment Metrics and Measures Accuracy: UNKNOWN Cost: UNKNOWN

Biaslyze Accuracy: UNKNOWN Cost: LOW

Fari EVA: Fairness EVAluation of Voice Technologies Accuracy: UNKNOWN Cost: UNKNOWN

Figure 5: Bias Detector Catalogue of tools

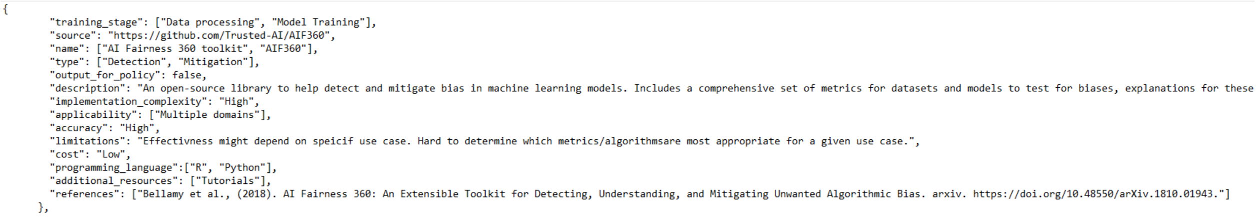

```
{
        "training_stage": ["Data processing", "Model Training"],
        "source": "https://github.com/Trusted-AI/AIF360",
        "name": ["AI Fairness 360 toolkit", "AIF360"],
        "type": ["Detection", "Mitigation"],
        "output_for_policy": false,
        "description": "An open-source library to help detect and mitigate bias in machine learning models. Includes a comprehensive set of metrics for datasets and models to test for biases, explanations for these
        "implementation_complexity": "High",
        "applicability": ["Multiple domains"],
        "accuracy": "High",
        "limitations": "Effectivness might depend on speicif use case. Hard to determine which metrics/algorithmsare most appropriate for a given use case.",
        "cost": "Low",
        "programming_language":["R", "Python"],
        "additional_resources": ["Tutorials"],
        "references": ["Bellamy et al., (2018). AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias. arxiv. https://doi.org/10.48550/arXiv.1810.01943."]
      },
```

Figure 6: Example of an input for the Catalogue in json format

2.1.5 OECD new design

As we developed the Bias Detection Catalogue in early 2024, OECD published Catalogue of Tools & Metrics for Trustworthy AI[1] with more than 900 tools and metrices. Our catalogue would seriously overlap with OECD's published Catalogue as the overall aim and method is almost the same, so we decided to discontinue the development of our Catalogue of bias detection and mitigation strategies, and rather focus on the visual synthesis of all the tools collected in the OECD Catalogue database.

OECD development team was kind enough to provide us with data dump of all the tools they have in their database, and we decided to create a design that would complement the tabular presentation OECH currently has on their webpage. As the tools are separated by approach into technical, education and procedural, we decided to follow along this separation and present them visually in a circular representation (Figure 7) where information can be condensed. To aid visual search, objective fields are listed around the central circle that denotes the training stages and

[1] https://oecd.ai/en/catalogue/tools

interactive lines connect the chosen tool with the objectives. When the tool is chosen, a tooltip pops up, that shows detailed information on the tool (see right side of Figure 7). Detailed view is presented in Figure 8.

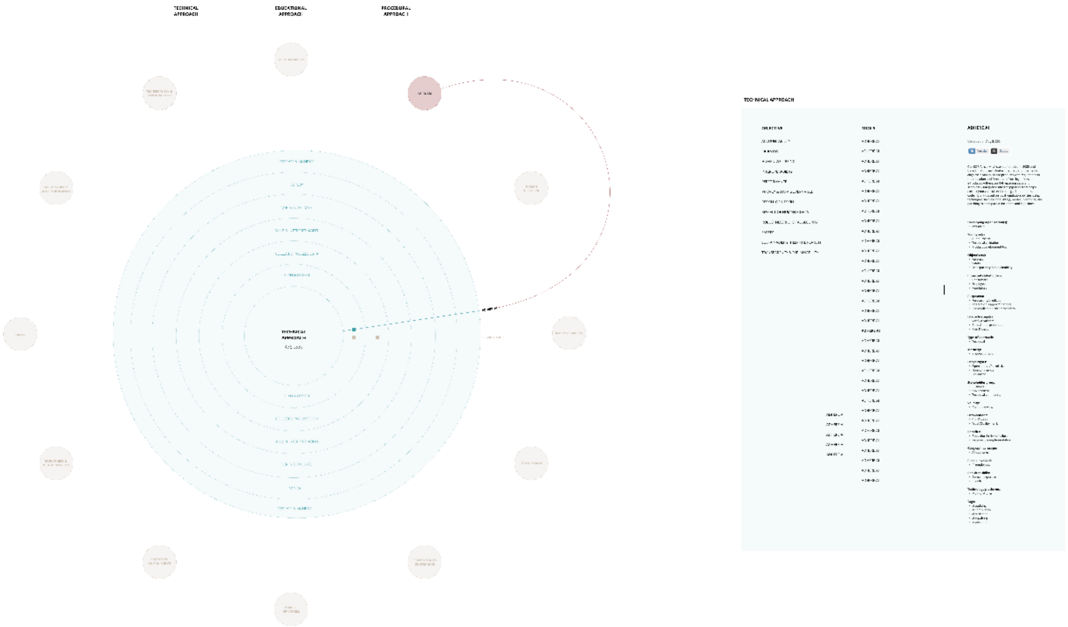


Figure 7: Wireframe design for visual synthesis of bias detection and mitigation tools

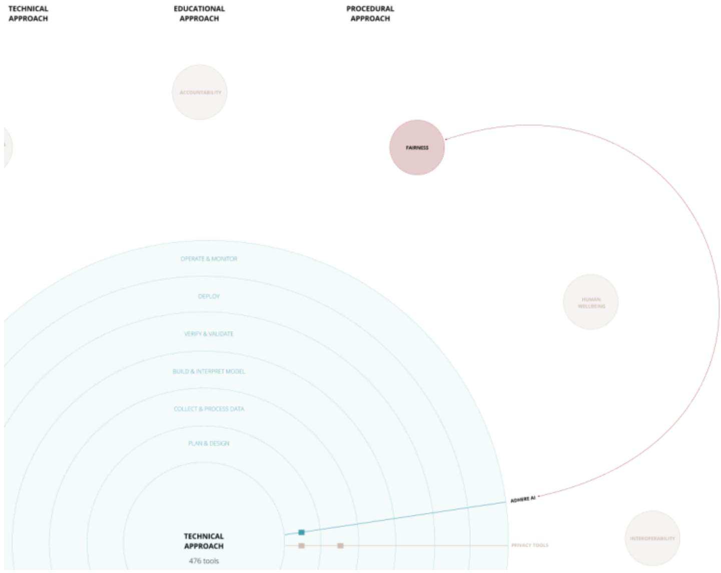


Figure 8: Detail of visual synthesis of bias detection and mitigation tools

The updated version of the catalogue is planned to be implemented M25-27 and will be presented in the D4.4.

2.2 Bias detection toolkit

As reported in D4.1, part of our work is to develop bias detection in specific datasets for SDG observatories. Here we present two strategies to detect bias that were developed in the scope of AI4Gov project.

2.2.1 Pipeline for bias detection in rare diseases

SDG3 is connected with health and wellbeing, and under its umbrella we have previously developed a Rare Disease Observatory[2], that collects heterogenous data from different sources and displays it in a coherent way. Rare Disease Observatory is currently in a pilot version and covers data on news, social media and scientific publications for 16 different rare diseases. As the plan is to expand the Observatory to cover all the diseases (>7000) outside of scope of AI4Gov, the ingested data needs to be as bias-free as possible.

2.2.1.1 Rare diseases and patient-reported outcomes

In rare diseases, limited scientific and clinical data often consist of single case reports and local data with small populations. To overcome this, global cohorts' data is crucial, making patient-reported outcomes (PROs) and caregiver-reported outcomes (CROs) repositories essential for insights on the patients. However, for the sake of equity, it is crucial to detect bias and data incompleteness in these repositories, as there are severe differences in data representation across continents. While a comprehensive understanding requires global data, the reality is that data predominantly comes from Western countries with more developed healthcare systems.

To this end we have developed a data incompleteness pipeline with a primary aim to develop and implement an advanced analytics pipeline for the automatic detection of bias and quantification of data incompleteness in patient-reported outcomes repositories, particularly addressing the geographic disparities in data representation to enhance the reliability and inclusivity of rare disease research.

2.2.1.2 Analytics pipeline

The analytics pipeline is developed to evaluate the bias inherent in PROs and CROs repositories. In the pilot version we've developed under AI4Gov, the pipeline is focused on the CROs data from the Genida registry for 83 different rare diseases. Its architecture is depicted in Figure 9 and the pipeline in Figure 10.

[2] https://rarediseases.ijs.si/

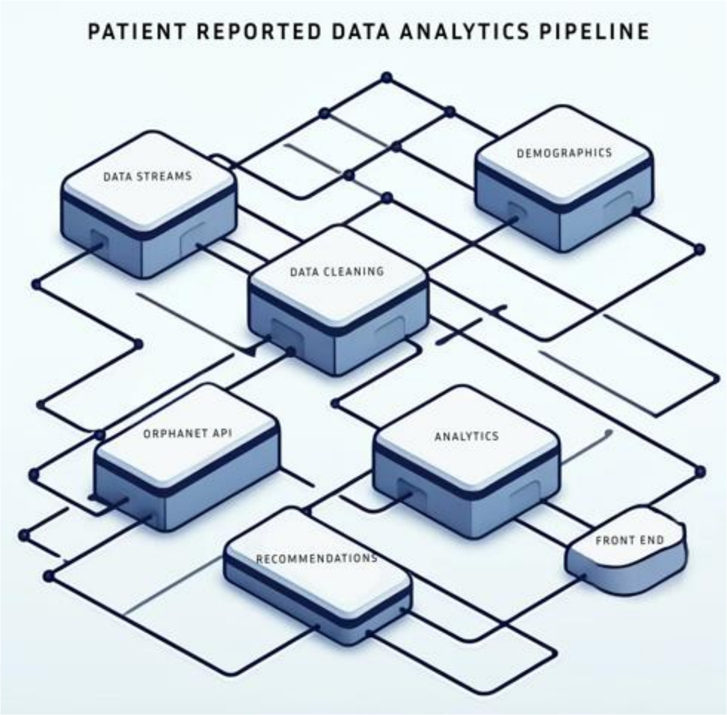


Figure 9: Architecture of PRO analytics pipeline

To calculate the data incompleteness, data from Genida is compared to prevalence collected through API from Orphanet[3], normalized to demographics and displayed for the whole world.

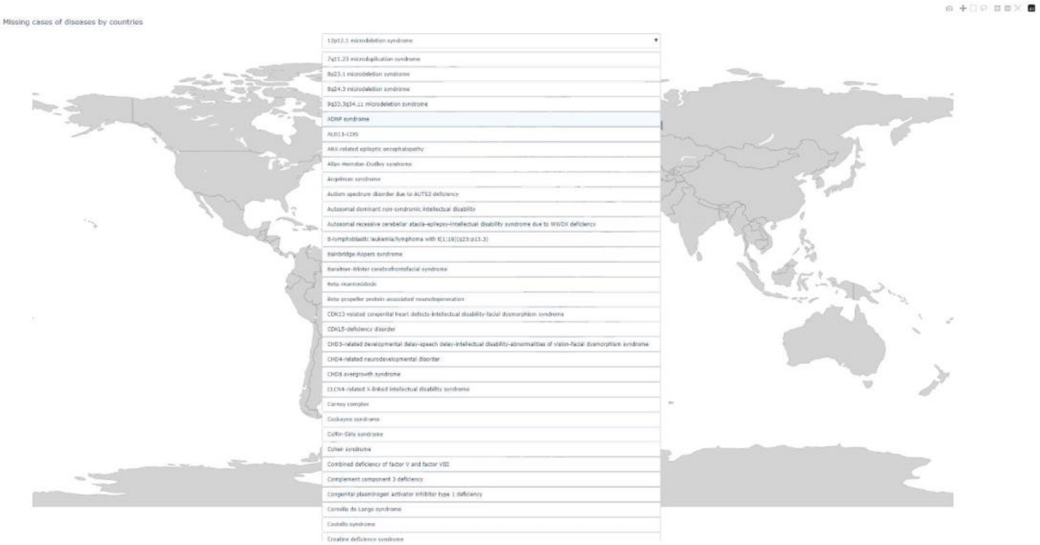

Missing cases of diseases by countries
17p12.1 microdeletion syndrome
7q11.23 microduplication syndrome
8q23.1 microdeletion syndrome
8q24.3 microdeletion syndrome
9q33.3q34.11 microdeletion syndrome
ADNP syndrome
ALG13-CDG
ARX-related epileptic encephalopathy
Allan-Herndon-Dudley syndrome
Angelman syndrome
Autism spectrum disorder due to AUTS2 deficiency
Autosomal dominant non-syndromic intellectual disability
Autosomal recessive cerebellar ataxia-epilepsy-intellectual disability syndrome due to WWOX deficiency
B-lymphoblastic leukemia/lymphoma with t(1;19)(q23;p13.3)
Bainbridge-Ropers syndrome
Baraitser-Winter cerebrofrontofacial syndrome
Beta-mannosidosis
Beta-propeller protein-associated neurodegeneration
CDK13-related congenital heart defects-intellectual disability-facial dysmorphism syndrome
CDKL5-deficiency disorder
CHD3-related developmental delay-speech delay-intellectual disability-abnormalities of vision-facial dysmorphism syndrome
CHD4-related neurodevelopmental disorder
CHD8 overgrowth syndrome
CLCN4-related X-linked intellectual disability syndrome
Carney complex
Cockayne syndrome
Coffin-Siris syndrome
Cohen syndrome
Combined deficiency of factor V and factor VIII
Complement component 3 deficiency
Congenital plasminogen activator inhibitor type 1 deficiency
Cornelia de Lange syndrome
Costello syndrome
Creatine deficiency syndrome

Figure 10: Data incompleteness analytics pipeline for 83 rare diseases

[3] https://www.orpha.net/

As one of the diseases is chosen (Figure 11), missing data is displayed, showing which countries have the most missing data cases. Tooltip provides detailed information for each country with the number of recorded cases, number of missing cases and the prevalence of the disease.

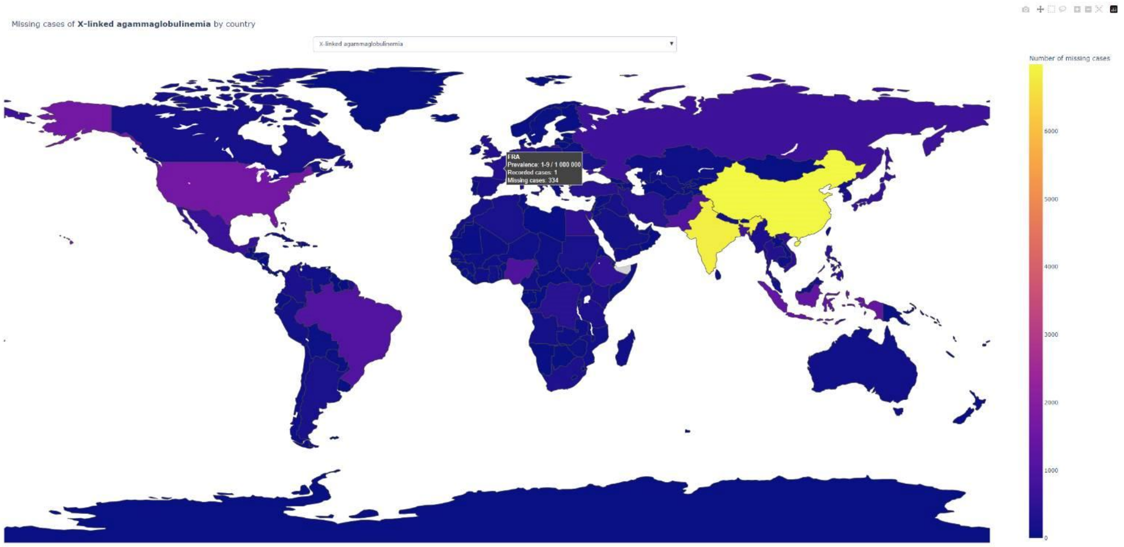

Missing cases of **X-linked agammaglobulinemia** by country
X-linked agammaglobulinemia ▾
FRA
Prevalence: 1-9 / 1 000 000
Recorded cases: 1
Missing cases: 334
Number of missing cases
6000
5000
4000
3000
2000
1000
0

Figure 11: Data incompleteness analytics pipeline for one of the diseases

2.2.2 Pipeline for bias detection in police data

To explore how we can generalize the data incompleteness pipeline we've developed for rare diseases, we added a new use case of police data for breathalyser test and traffic accidents, which is further explained in deliverables for WP6.

Based on the data, we visualised traffic accidents by administrative units per 10.000 inhabitants from January 2005 till November 2022 (93.170 cases), shown in Figure 12.

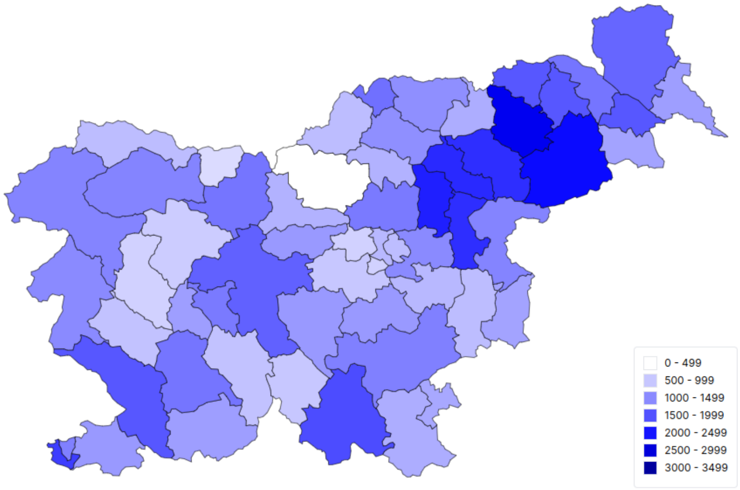


Figure 12: Traffic accidents by administrative units

The data was then compared for the same time period for cases when breathalyser shown more than 0.24g/kg of alcohol in breath (8.581 cases), shown in Figure 13.

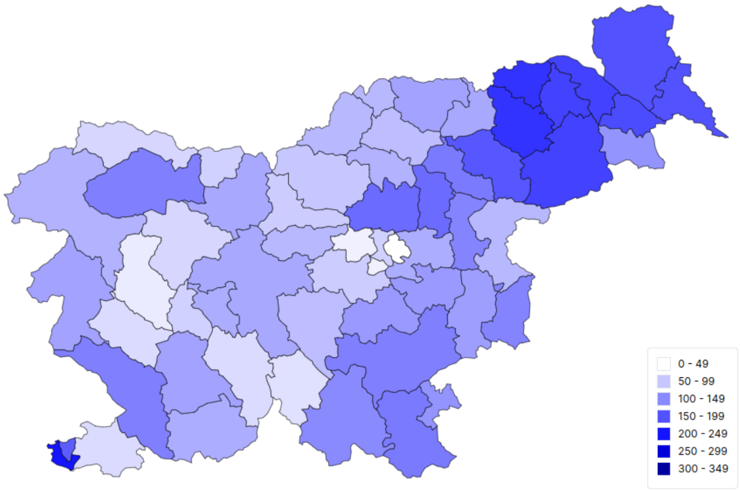


Figure 13: Traffic accidents by administrative units for >0.24g/kg of alcohol in breath

Visualisation of breathalyser tests ordered by police - by administrative units per 10.000 inhabitants from January 2005 till December 2022 (1.631.018 cases), is shown in Figure 14.

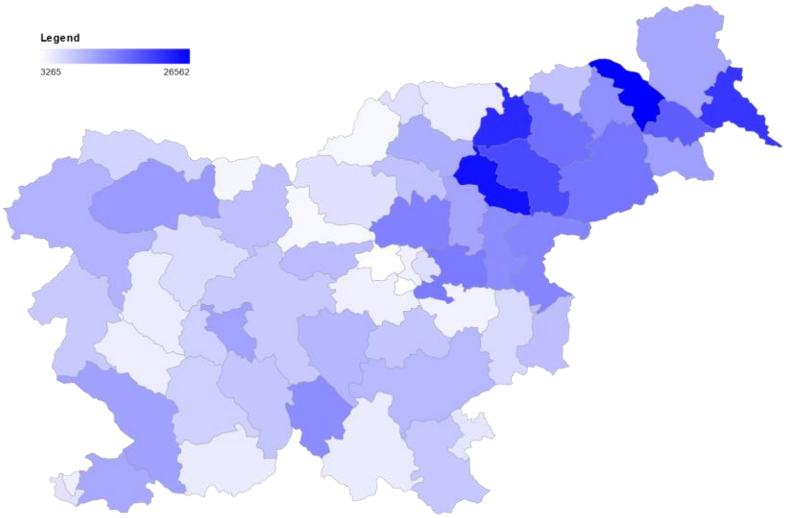


Figure 14: Breathalyser tests ordered by police by administrative units

From discussions with the police administrative we concluded we still need more data to be able to evaluate the potential bias in data and will report on this in further WP6 deliverables.

2.3 Research – understanding bias in news through barriers of news spreading

Here we provide a high-level overview on research regarding bias in news that has been submitted recently (Sittar, 2024, under revision).

Detecting bias in news helps ensure balanced public discourse by revealing how different perspectives shape the representation of events. This awareness promotes media literacy, enabling people to critically evaluate information and hold news sources accountable for fair reporting. The way societal events are reported is deeply influenced by biases that stem from geographical, economic, political, and cultural contexts. The platform *BAR-Analytics* provides a framework for identifying these biases by analyzing the spread, tone, and thematic focus of news coverage. When examining conflicts such as the Russia-Ukraine and Israel-Palestine wars, the platform reveals clear disparities in sentiment and emphasis. For instance, media coverage of the Israel-Palestine conflict tends to adopt a more negative tone across various regions compared to the relatively neutral or positive tone used for the Russia-Ukraine conflict. Additionally, biases emerge in the aspects of the conflicts that are highlighted: human rights violations are more frequently discussed in Israel-Palestine reporting, while election interference is emphasized in Russia-Ukraine coverage. These differences indicate that the framing of news is not merely reflective of events themselves but is shaped by the political and economic leanings of media outlets based on the barriers. Identifying these patterns of bias is crucial for understanding how news consumption influences public perception and how narratives are selectively constructed and propagated on a global scale.

2.4 Development for use cases

We provide a very short description for the work done on use cases, since more detailed information has already been provided in D6.2 and D6.3.

Development for SDG observatories and police data has been explained in sections 2.2.1 and 2.2.2.

For the Top 100 projects, checklists for both applicants and reviewers were developed, the findings can be found in D6.3. Based on the evaluation the updated checklist in collaboration with WP5 will be implemented for potential further calls of Top100 projects.

For the OECD documents, bias boards showcasing the sentiment analysis of the AI policy documents are being implemented in the SDG observatories.

2.5 Summary

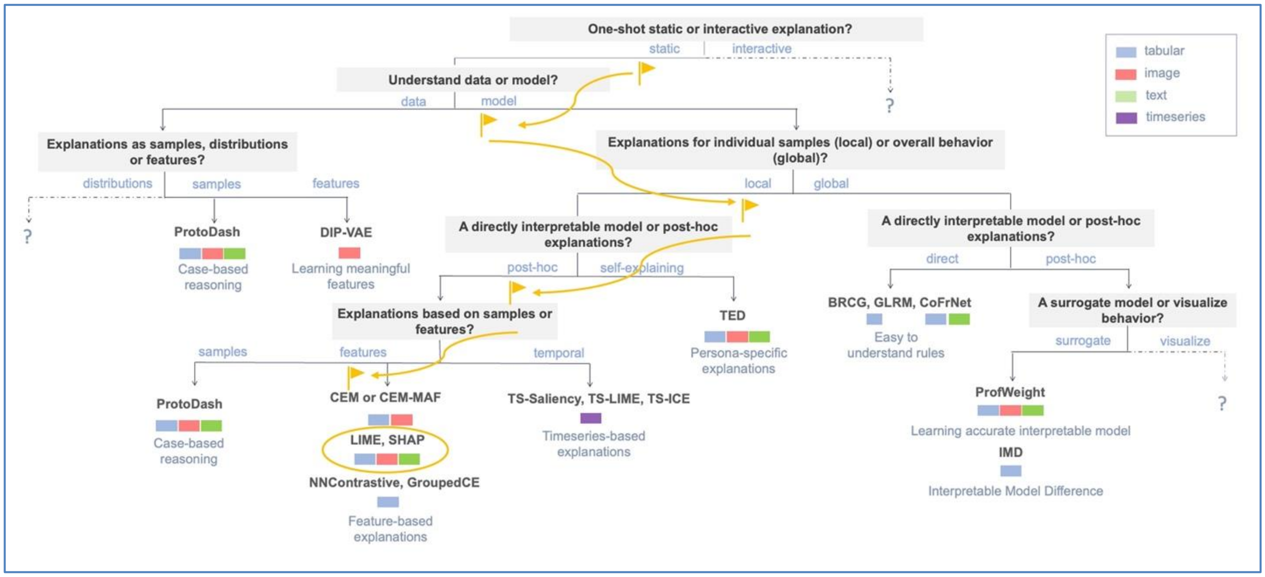
This section summarized the work done under T4.1. Our main contributions are VUF catalogue which provides awareness-raising materials for general public and specific tools for developers, with the intention for bias-free-by-design AI models and specific bias detection tools (for rare diseases and police data) which evaluate possible bias at the data level.

3 Situation Aware eXplainability

3.1 Sufficient Subset Training

The field of eXplainable Artificial Intelligence (XAI) aims to develop techniques that can provide human-understandable explanations to inner workings of AI models, thereby increasing the models' transparency and trustworthiness. As the use of Machine Learning (ML) models in general, and Deep Neural Networks (DNNs) in particular, is becoming more widespread and democratized in recent years, despite many of these models being black boxes, the field of XAI is rapidly evolving. While in some domains users would like to know how a decision was reached by the model, in other domains such understanding is crucial.

Initially, XAI focused primarily on feature attribution methods such as LIME (Ribeiro, 2016) and SHAP (Lundberg, 2017). As work in the field progressed, different taxonomies were introduced to organize the different techniques and better understand which technique one should use. Based on one such taxonomy (Figure 15), LIME and SHAP, for example, can now be further classified as *static, local, post-hoc, feature-attribution* explainability of a *model* (see orange flags and arrows in Figure 15). While SHAP is still a commonly used XAI technique, it has two main limitations. First, being a feature attribution method means that it is assumed the explained model behaves in an approximately linear fashion locally (Yeh, 2019), which is not the case for complex ML models (e.g., very deep DNNs). Second, it is a post-hoc method, meaning it requires time in addition to the explained model's inference time.



One-shot static or interactive explanation?
static
interactive
?
tabular
image
text
timeseries
Understand data or model?
data
model
Explanations as samples, distributions or features?
Explanations for individual samples (local) or overall behavior (global)?
distributions
samples
features
local
global
?
ProtoDash
Case-based reasoning
DIP-VAE
Learning meaningful features
A directly interpretable model or post-hoc explanations?
post-hoc
self-explaining
A directly interpretable model or post-hoc explanations?
direct
post-hoc
Explanations based on samples or features?
TED
Persona-specific explanations
BRCG, GLRM, CoFrNet
Easy to understand rules
A surrogate model or visualize behavior?
surrogate
visualize
samples
features
temporal
ProtoDash
Case-based reasoning
CEM or CEM-MAF
LIME, SHAP
NNContrastive, GroupedCE
Feature-based explanations
TS-Saliency, TS-LIME, TS-ICE
Timeseries-based explanations
ProfWeight
Learning accurate interpretable model
IMD
Interpretable Model Difference
?

Figure 15: XAI taxonomy (adapted from AIX360 github repository - methods-choice-updated.png)

In this work we introduce a framework to train a model that produces both a task output and an explanation at time of inference (Figure 16). The output is in the form of subset of input features that *together* explain the model's prediction for that input (section 3.1.1). This framework, as mentioned, inferences both the prediction and the explanation, thus not requiring additional time for the explanation beyond the explained model's inference time. Additionally, the type of explanation it produces does not assume the explained model behaves in an approximately linear manner locally.

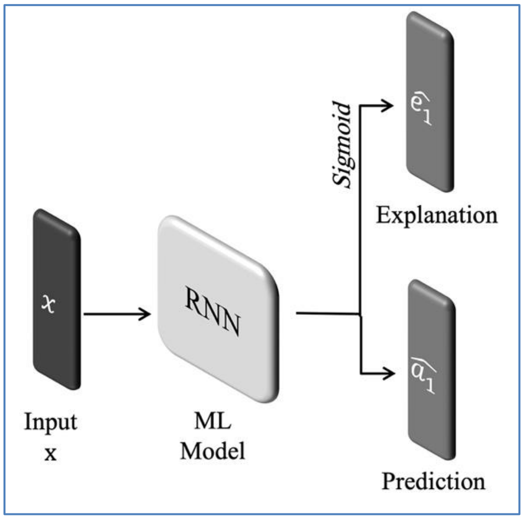


Figure 16: The two-head model

3.1.1 Probabilistic Sufficient Reasons

A *sufficient reason* (Barceló, 2020) (Darwiche, 2020) (Arenas, 2022) is any subset of input features S that ensures the prediction of a *classifier* remains constant. That is, keeping the values of the features in S fixed and assigning new values to the features in the complement $\bar{S}$ (within reason), will result in the same prediction by the *classifier*. A *probabilistic sufficient reason* entails sampling $\bar{S}$ from a specified distribution (e.g., $U(0,1)$). Formally speaking, given a **classification** model f on an n-dimentional input x, and a distribution $\mathcal{D}$ over the input features, $S \subseteq \{1,2,\dots,n\}$ is a *probabilistic sufficient reason* of $\langle f, x \rangle$ iff

$$Pr_{z\sim\mathcal{D}}\left[\underset{j}{\operatorname{argmax}} f(z)_j = \underset{j}{\operatorname{argmax}} f(x)_j \,| z_s = x_s\right] \geq 1-\delta$$

where $z_s = x_s$ denotes fixing the features of S in z to their corresponding values in x.

3.1.2 Self-Explaining Neural Networks (SENNs) with Minimal Sufficient Reasons (MSRs) and Probabilistic Masking

We introduce *SENNs with probabilistic MSRs*, a framework for DNN classifiers. We developed an algorithm to train the SENN in a way that balances between prediction accuracy and minimal sufficiency of explanation output.

In Figure 17 we provide illustrative examples of SENNs with probabilistic MSRs in two domains. In (A) is an example of the next activity prediction task from Business Process Management (BPM) domain. In this example, a four-event trace from the BPIC 2017 dataset (van Dongen, 2017) is used to predict the next activity ('*cancelled*'), and with the prediction, an MSR for it is provided. In this case data are tabular and thus their MSRs are in the form of subsets of features. In this case: ‘time in day’ of event 2, ‘time difference’ and ‘time in day’ of event 3, and ‘time in week’ of event 4. In (B) are two examples from MNIST (Deng, 2012), a benchmark dataset in computer vision domain. On the left are the original images (six in the top image and seven in the bottom image) and on the right are their respective MSRs. In vision, the MSRs are usually in the form of pixels (as in this case) or in the form of super-pixels (usually used for higher-resolution – i.e., greater DPI - images).

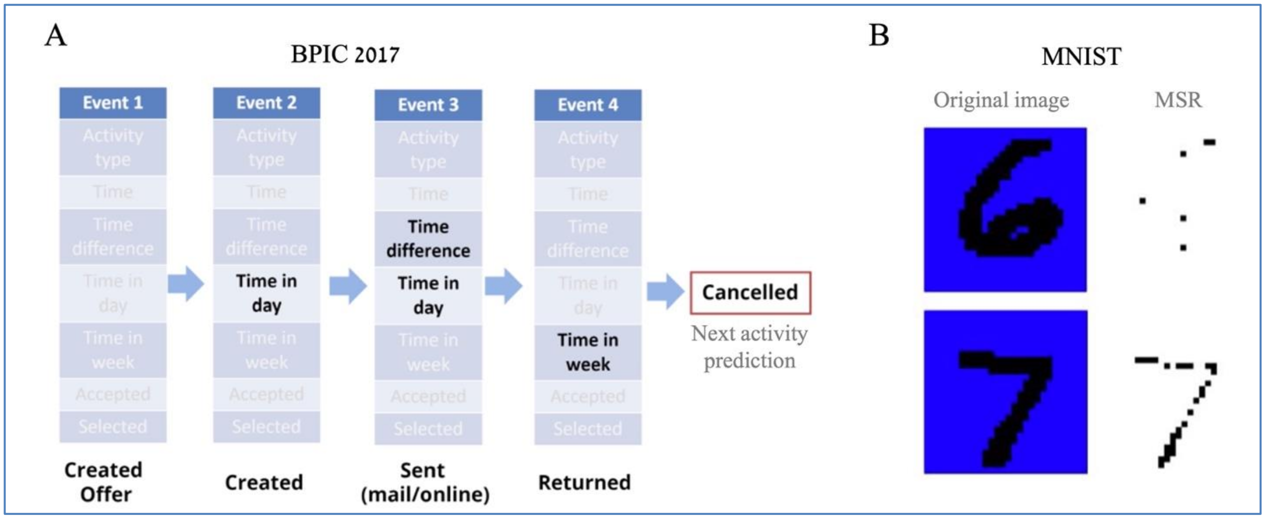


Figure 17: Illustrative examples from BPM (A) and Computer Vision (B) domains

3.1.3 Application to Regression

As the data in the water management pilot require a regression model rather than a classification model, we expanded the above work to regression LSTM-based RNNs by modifying the objectives to use a regression-specific metric, Mean Squared Error (MSE) $:= \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$. We use MSE to evaluate both the performance of f in the regression task and the sufficiency of the MSR output.

Further details on the pilot-specific implementation can be found in section 4.

3.2 Assessing the quality of explanations as perceived by users

3.2.1 Introduction

At the heart of Situation Aware eXplainability (SAX) is the perception that explainability in business processes (BPs) promotes trust and adoption of the automation technology. As already mentioned in D4.1 - “Trustworthy, Explainable, and unbiased AI V1, SAX seeks for explanations of business processes outcomes and conditions that are sound and interpretable taking advantage of the BP definitions, contextual environment, and full runtime process traces. Furthermore, they are expected to embed the ability to go beyond a local reasoning context, handle a large variety of situations, and facilitate the (automatic or by humans) tracking of execution consistency for a better understanding of process flows and process outcomes. One way to provide automatic explanations is by exploiting Large Language Models (LLMs) capabilities.

Situation Aware eXplainability (SAX) including motivation and main concepts, as well as the SAX4BPM library architecture and services were introduced in D4.1 in section 3, therefore we refrain ourselves from repeating the content here. More specifically, the following concepts: Business process, eXplainable AI, process discovery, causal discovery and Large Language Models (LLMs) are explained in section 3.1 in D4.1, sound explanation is described in section 3.2., and SAX4BPM library is detailed in section 3.4. The SAX4BPM library consists of a set of services that aids with the automatic derivation of SAX explanations leveraging existing LLMs. We would like to stress that since the submission of D4.1 to today, several enhancements and extensions have been implemented in the SAX4BPM library which was released to the open source and can be accessible at: https://github.com/IBM/sax4bpm.

To assess the quality of textual explanations produced by our SAX4BPM library as perceived by users, IBM carried out a user study. Our main objective was to leverage LLMs to give explanations that users grasp as adequately with respect to certain process conditions.

To this end, our approach consisting of blending a variety of BP-related views i.e., process, causal, and eXplainable AI, employing state-of- the-art techniques to ensure the correctness of these ingredients. The combination of these views (i.e., "knowledge ingredients") forms the fuel for the automatic generation of narratives that serve as input for prompt engineering of LLMs to achieve better perceived process outcome explanations as illustrated in Figure 18.

We henceforth describe the main constructs and results of our user study and its implications to AI4GOV. A complete description can be found in (Fahland, 2024).

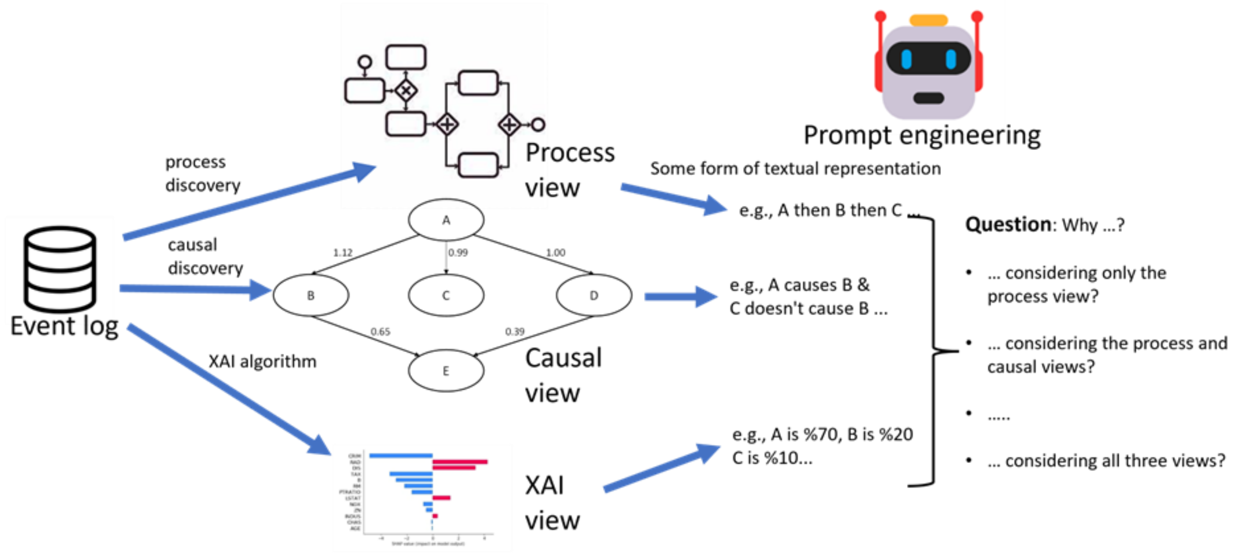

process discovery
causal discovery
XAI algorithm
Event log
Process view
A
1.12
0.99
1.00
B
C
D
0.65
0.39
E
Causal view
XAI view
Prompt engineering
Some form of textual representation
e.g., A then B then C ...
e.g., A causes B & C doesn't cause B ...
e.g., A is %70, B is %20 C is %10...
Question: Why ...?
• ... considering only the process view?
• ... considering the process and causal views?
• .....
• ... considering all three views?

Figure 18: General approach

3.2.2 Methodology

With the goal of leveraging LLMs for generating SAX explanations, we focused on the following research question:

Research Question: *How does informed knowledge of business processes affect the perceived quality of LLM-generated explanations regarding business process conditions?* Where a business process (BP) condition refers to any state or outcome arising during BP execution.

To investigate this, we utilized the SAX4BPM library to manipulate knowledge inputs and designed an experiment to evaluate their impact on the perceived quality of explanations. We operationalized "perceived quality of explanations" through two constructs from prior literature: **fidelity** and **interpretability**, as detailed below. These constructs informed the formation of measurement dimensions. Based on this framework, we developed two hypotheses:

Hypothesis 1 (H1): *Explanations generated by LLMs informed with business process knowledge will be perceived as having higher **fidelity** than those generated by uninformed LLMs.*

Hypothesis 2 (H2): *Explanations generated by LLMs informed with business process knowledge will be perceived as having higher **interpretability** than those generated by uninformed LLMs.*

Method

We conducted a controlled experiment to test our hypotheses, using between-group manipulations of knowledge ingredients related to various BPs. These inputs were presented to an LLM, which generated textual explanations for specific BP conditions. The resulting explanation variants were embedded in questionnaires designed to measure perceived explanation quality through rating scales. User ratings were analysed to determine the effects of knowledge manipulation across groups.

The target population included members of the global IT community who might use LLMs for automating explanation generation in their domains. Responses were collected through an online questionnaire disseminated via three channels: graduate students at TU/e’s computer science faculty, IBM Research employees, and the Reddit r/SampleSize community. Participation was voluntary and anonymous, with informed consent obtained. Ethical approval was granted by the TU/e Ethical Review Board (#ERB2023MCS42).

A total of 50 participants were recruited, with 49 completing the survey (one did not consent). Of these, 28 (57.1%) identified as men, 19 (38.8%) as women, and 2 (4.1%) chose not to specify gender. Age distribution was as follows: 23 (46.9%) over 45, 15 (30.6%) aged 25–35, 6 (12.2%) under 25, and 5 (10.2%) aged 36–45.

Scale development

To evaluate the quality of explanations generated by the LLM, we developed scale metrics based on various measurement dimensions. Currently, no universally accepted framework or metrics exist for evaluating perceived explanations. The literature lacks consensus on a definition for explainability, the properties that make explanations effective and understandable, and standardized methods for assessing explanation quality (Carvalho, 2019), (Elkhawaga, 2023), (Lage, 2018), (Markus, 2021), (Sokol, 2020), (Vilone, 2021), and (Zhou, 2021).

Our focus was on assessing SAX explanations synthesized by an LLM using different knowledge ingredients about business processes. Unlike traditional XAI methods, we concentrated on the intrinsic qualities and content of the explanations rather than their hedonic or exogenous qualities tied to user interaction. While we included some attitude-based factors, these were treated as moderating variables influencing perceptions of explanation quality.

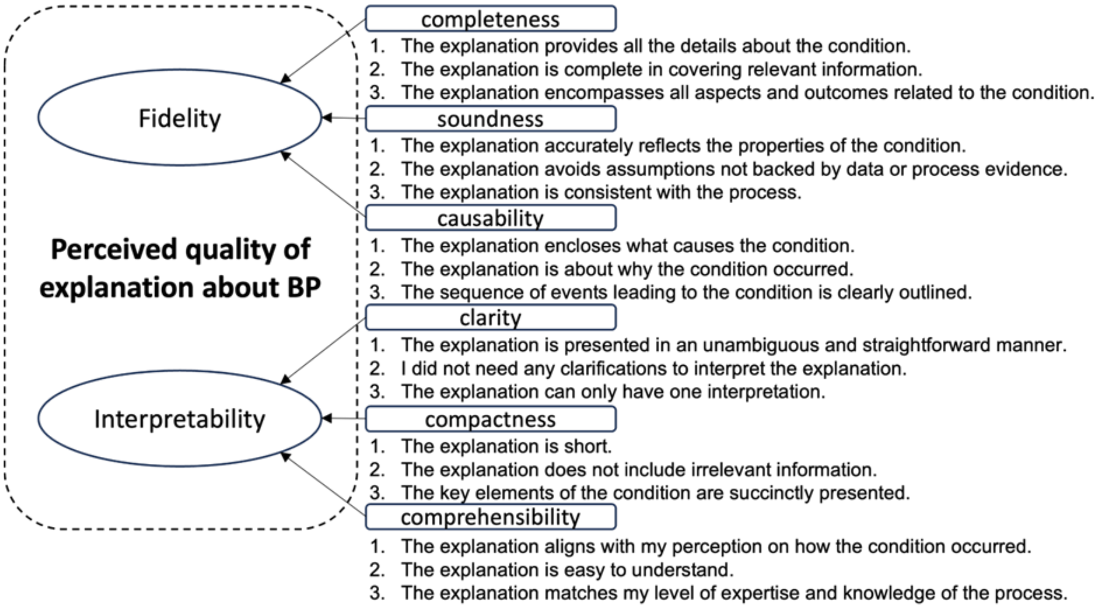
Following standard practices (Kulesza, 2013), (Lage, 2018), and (Markus, 2021), we conducted a user study using a tailored evaluation scale. Common dimensions from the literature were reviewed and adapted to suit our needs. Table 1 lists the key dimensions reviewed, and Table 2 provides a summary of the final constructs, dimensions, and their definitions used in our assessment.

Table 1: Key constructs and dimensions for quality of explanations reviewed in prior literature

Ref.	Latent construct	Dimension	Definition in reference
(Elkhawaga, 2023)		Robustness	An explanation can withstand small perturbations of the input that do not change the output prediction. Consequently, robustness expresses a low sensitivity of the XAI method to changes in inputs.
		Fidelity	The XAI method should preserve the internal concepts and original behavior of the black box ML model whenever there is a need to mimic that model.
		Causality	The XAI method should maintain causal relationships between inputs and outputs. An ML model is perceived as being more human-like whenever it provides such causal explanations. Therefore, causality is fundamental to achieving a human understanding of the ML model.
		Trust	The extent that the outcomes of an XAI method enable gaining confidence that the ML model acts as intended.
		Fairness	The extent an explanation enables humans to ensure unbiased decisions of the employed ML models.
(Velmurugan, 2021)		Soundness	Measures how truthful an explanation is with respect to the underlying predictive model.
		Completeness	The extent an explanation generalizes well beyond the particular case in which the explanation was produced.
		Contextfullness	The degree the explanation is accompanied by all the necessary conditions for it to hold, critiques (i.e., explanation oddities) and its similarities to other cases.
		Interactiveness	The explanation process should be controllable and interactive.
		Actionable	An explanation that the users can treat as guidelines towards the desired outcome.
		Chronology	The extent an explanation inherently possesses time ordering, typically with users preferring expla- nations that account for more recent events as their cause, i.e., proximal causes.
		Coherence	The extent an explanation is aligned with background knowledge and beliefs held by the users.
		Novelty	The extent an explanation contains surprising or abnormal characteristics.
		Complexity	The degree an explanation matches the skills and background knowledge of explainees.
		Personalisation	The extent an explainability technique is adjusted to model the users' background knowledge and mental model.

		Parsimony	The extent an explanation is selective and succinct enough to avoid overwhelming the explainee with unnecessary information.
(Stevens, 2021)		Causability	The extent an explanation makes the causal relationships between the inputs and the model’s predictions explicit.
		Interpretability	The capacity to provide or bring out the meaning of an abstract concept.
		Understandability	The capacity to make the model understandable by end users.
(Wickramanayake, 2023)	Interpretability	Response time for understanding	Time taken to answer questions about the explained situation.
		Accuracy of understanding	Number of correctly answered questions about the explained situation.
		Subjective satisfaction	Self-reported degree of satisfaction.
(Galanti, 2023; Bozorgi, 2021)	Interpretability		How understandable an explanation is for humans.
		Clarity	The degree an explanation is unambiguous.
		Parsimony	The extent an explanation is not too complex, i.e., presented in a compact form.
(Galanti, 2023; Bozorgi, 2021; Bozorgi, 2020)	Fidelity		How accurately an explanation describes model behavior, i.e., how faithful an explanation is to the model.
		Completeness	The extent an explanation describes the entire dynamic of the ML model.
		Soundness	The degree an explanation is correct, i.e., truthful to the model.
(Zhou, 2021)		Causability	The measurable extent to which an explanation achieves a specified level of causal understanding.
	Interpretability	Clarity	The extent an explanation is unambiguous.
		Parsimony	The extent an explanation is presented in a simple and compact form.
		Broadness	How generally applicable the explanation is.
(Bozorgi, 2023)		Satisfaction	The degree to which users feel that they sufficiently understand the AI system or process being explained to them.
		”Goodness”	A set of yes/no questions reflecting explanation correctness, comprehensiveness, coherence, and usefulness.
(Künzel, 2019)		Causability	The extent to which an explanation of a statement to a user achieves a specified level of causal understanding with effectiveness, efficiency, and satisfaction in a specified context of use.

Based on our literature review, we adopted **fidelity** and **interpretability** as two primary latent constructs, each encompassing measurable dimensions. **Fidelity** includes completeness, soundness, and causability, while **interpretability** comprises clarity, compactness, and comprehensibility. Causability reflects the correctness of the model and relates to fidelity, while comprehensibility, tied to understanding, connects to interpretability. Figure 19 illustrates these six dimensions and their corresponding survey questions.



completeness

1. The explanation provides all the details about the condition.
2. The explanation is complete in covering relevant information.
3. The explanation encompasses all aspects and outcomes related to the condition.

soundness

1. The explanation accurately reflects the properties of the condition.
2. The explanation avoids assumptions not backed by data or process evidence.
3. The explanation is consistent with the process.

causability

1. The explanation encloses what causes the condition.
2. The explanation is about why the condition occurred.
3. The sequence of events leading to the condition is clearly outlined.

clarity

1. The explanation is presented in an unambiguous and straightforward manner.
2. I did not need any clarifications to interpret the explanation.
3. The explanation can only have one interpretation.

compactness

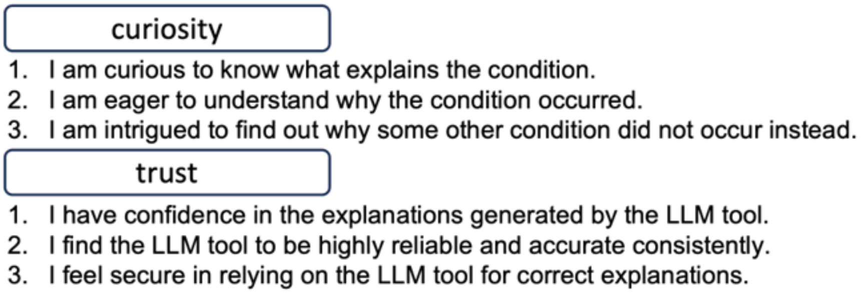
1. The explanation is short.
2. The explanation does not include irrelevant information.
3. The key elements of the condition are succinctly presented.

comprehensibility

1. The explanation aligns with my perception on how the condition occurred.
2. The explanation is easy to understand.
3. The explanation matches my level of expertise and knowledge of the process.

Figure 19: (Perceived) explanation quality – a total manifestation of 18 questions populated for the measurement dimensions, corresponding to each of the two high-level (latent) constructs

To address potential biases from participant interest, we included **curiosity** (desire for knowledge) and **trust** (confidence in the LLM's reliability) as background factors. Trust stems from the clarity and capability of automation to achieve user goals (Lee, 2004), while curiosity influences engagement with explanations (Hoffman, 2023). These factors were treated as covariates or moderators affecting interaction, rather than intrinsic explanation qualities, as shown in Figure 20.



curiosity

1. I am curious to know what explains the condition.
2. I am eager to understand why the condition occurred.
3. I am intrigued to find out why some other condition did not occur instead.

trust

1. I have confidence in the explanations generated by the LLM tool.
2. I find the LLM tool to be highly reliable and accurate consistently.
3. I feel secure in relying on the LLM tool for correct explanations.

Figure 20: (Perceived) moderating factors (covariates) - an additional manifestation of 6 questions

In total, our framework operationalizes two latent constructs through six measurement dimensions and incorporates two additional background factors. Table 2 summarizes the adapted definitions.

Table 2: Definitions for all experimental constructs, measurement dimensions, and background factors

Dependent Variables	**Measurement Dimensions**	**Definition**
Fidelity		How faithful an explanation is to the condition explained.
	Completeness	The extent to which an explanation describes all the information relevant to the condition explained.
	Soundness	How truthful the explanation is with respect to the condition explained.
	Causability	The explanation provides the reasons for the occurrence of the condition explained.
Interpretability		How understandable an explanation is for humans.
	Clarity	The extent the explanation is unambiguous.
	Compactness	The extent the explanation is presented in a simple and compact form.
	Comprehensibility	The explanation aligns with my understanding of the problem and how to react to it.
Background Factors		
Curiosity		The general desire to acquire knowledge about the reason for a certain condition.
Trust		An attitude toward the LLM that affects reliance on its explanations.

We developed an initial set of 40 questions (5 per dimension) for the 8 measurement dimensions and refined them using a card sorting procedure followed by reliability analysis that resulted with 3 items per dimension of a total set of 24 items (Figure 19 and Figure 20).

3.2.3 User study

In our experiment, we explored how different types of explanatory knowledge about BPs could be integrated into LLM prompts before interacting with the models to explain process execution outcomes. We used three between-group manipulations as independent variables, based on knowledge types: process, XAI (feature importance), and causal. All input combinations are listed in Table 3.

In our experiment, BP knowledge was used in all manipulations. The first manipulation involved augmenting this baseline with XAI knowledge (related to factors influencing the process) to using only XAI knowledge. The second manipulation examined the effect of combining process and XAI knowledge versus integrating all three knowledge types. Finally, we explored the impact of adding causal knowledge about execution dependencies, comparing it to using only the baseline process knowledge.

Table 3: Three manipulations of input knowledge type

Problem domain	**Group 1**	**Group 2**
Pizza delivery	Process and Feature-Importance (XAI)	Feature-Importance (XAI)
Parking fines	Process and Feature-Importance (XAI)	Process, Feature-Importance (XAI), and Causal
Loan approval	Process and Causal	Process

Respective to the various manipulations depicted in Table 3, each of the above hypotheses was further instantiated for more concrete testing resulting with a total manifest of three combinations per hypotheses as illustrated in Figure 21.

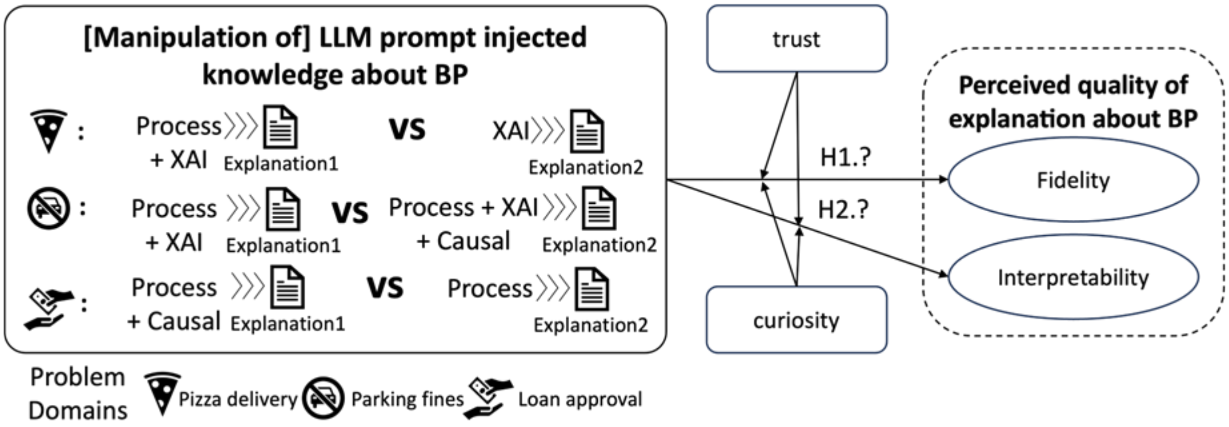


Figure 21: Experiment model

3.2.4 Domains

Each manipulation was implemented within a distinct problem domain to derive explanations for specific conditions. In particular, the first manipulation was developed within the 'pizza delivery' domain, focusing on creating explanations for delays in pizza delivery. The second manipulation is related to the 'parking fines' domain, targeting the development of explanations for the lateness in fine processing. Finally, the third manipulation, was instantiated in the 'loan approval' domain, with a focus on elucidating the potential reduction in loan approval times.

Corresponding to each domain, we generated the knowledge ingredients about the process model, the causal model, and the XAI feature importance using the SAX4BPM library services. This was applied to two synthesized process data in the pizza delivery and parking fines domains, and to the loan approval domain using an open dataset[4].

3.2.5 Results

Data was inspected for reliability and outliers and domain-wise analysis of the effects was carried out. We henceforth describe the main results received for each of the data sets.

Pizza delivery domain

The results support the expectation that adding process-related knowledge to XAI knowledge as an input to the LLM can potentially yield a significant effect on both perceptions of fidelity and interpretability. However, this effect is moderated by the degree to which the user is curious

[4] https://data.4tu.nl/articles/dataset/BPI_Challenge_2017/12696884

about the condition explored and trusts the LLM tool. Stretched beyond our original hypotheses, the results also identified trust as a background factor. This factor not only moderates the effect of the manipulated input to the LLM but could also directly mask this effect.

Parking fines domain

The results support the expectation that adding causal-related knowledge to process and XAI knowledge as an input to the LLM can potentially yield a significant effect on both perceptions of fidelity and interpretability. In the case of interpretability, the effect is present regardless of any particular moderation, whereas in the case of fidelity, the effect is moderated by the user's curiosity about the explored condition and trust in the LLM tool. Beyond our original hypotheses, it is also observed that curiosity has a direct effect on interpretability, and trust has a direct effect on both perceptions, interpretability, and fidelity.

Loan approval domain

The results support the expectation that adding causal-related knowledge to process knowledge as an input to the LLM can potentially yield a significant effect on both perceptions of fidelity and interpretability. The effect on both is moderated by the user's curiosity about the explored condition and trust in the LLM tool. Beyond our original hypotheses, it is also observed that curiosity has a direct effect on fidelity, and that trust has a direct effect on both perceptions, interpretability and fidelity.

Direction of effect

Careful observation of the study results revealed that, regardless of the significance of the effects, an important outcome that stands out across all three domains — one which we did not fully anticipate in our hypotheses — is related to the direction of the effects on fidelity and interpretability. Our original thought was that informing the LLM prompt with additional knowledge, whether process-related (as in the pizza domain) or causal-related (as in the two other domains), would lead to a consistent effect direction on both dependent constructs, potentially improving both. However, this was, not the result. The actual effect directions, as illustrated in Figure 22, not only demonstrate an interaction between the two constructs (supported by prior literature (Markus, 2021)) but also show that the addition of either type of knowledge ingredient (process or causal) works towards improving the perceived fidelity of the explanation while compensating for the perceived interpretability of the explanation. Concerning the size of the effect, in the pizza delivery and loan approval domains, the improvement in fidelity was larger than the loss in interpretability. However, in the parking fines domain, the opposite was true. Hence, we highlight this particular caveat to be watched for in specific problem domains where one might attempt a similar type of intervention to 'improve' the perceived quality of generated explanations.

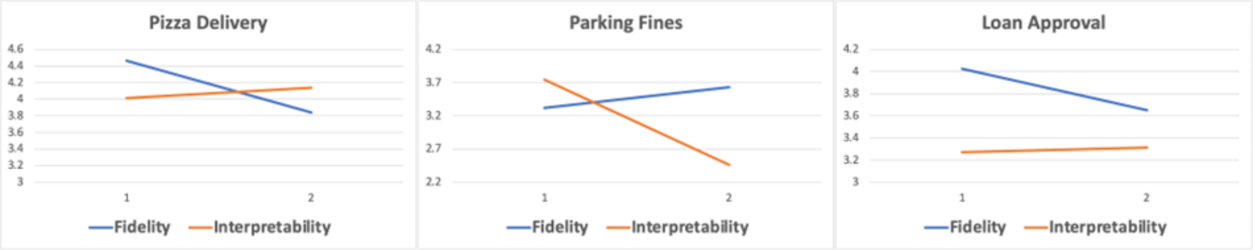


Figure 22: Constructs interaction

3.2.6 Value of developed instrumentation for government institutions

We present a realistic scenario that highlights the potential value of our developed instrumentation as a tool for a government organization aiming to establish a technological platform that enhances the efficiency of tax refund processing while encouraging usage among residents. Consider the case where the CIO of the National Tax Agency is exploring the deployment of a new AI-powered system designed to expedite tax return handling by automating the submission of annual tax returns through web-based agents accessible via smartphones. This system assists taxpayers throughout the submission process, facilitating a seamless tax refund experience.

Just before the process concludes, each taxpayer receives a notification that includes an explanation (see step 6 highlighted in the Annual tax refund process listed below) of the approved refund and any adjustments made to their submitted application. The proposed system incorporates a newly developed module that leverages state-of-the-art LLM technology to generate these explanation texts. This module aims not only to clarify the reasons for the adjustments but also to foster trust in the system, thereby encouraging continued usage. Given that LLM technology is relatively new, the infrastructure is designed to allow easy switching between different LLM engines "under the hood". At the time of deployment, the CIO must select the specific LLM engine version to be implemented. As for the writing of this report, the decision under consideration involves choosing between GPT-4.0 and LLAMA2. Each LLM has its own "style" affecting the reception of the refund decision. Indirectly, influencing the user perception about the governmental entity running the service.

How can the CIO make a thoughtful choice between these two competing technologies in this case? Our proposal is that, using our developed instrumentation, the CIO could initiate a small pilot program in two representative residential regions. In this pilot, participants will be presented with explanation narratives generated by the two engines as described below and subsequently will be asked to complete a brief survey based on our developed instrumentation. The feedback gathered from these surveys would provide a quantifiable comparison, reflecting participants' preferences and their level of appreciation of the explanations made by each engine. This data would assist the CIO in selecting the engine that achieves the higher score.

Annual Tax Refund Process

1. Start: Taxpayer Files Tax Return

- Input: Taxpayer's financial documents (e.g., income statements, receipts for deductions).
- Activity: Taxpayer submits the annual tax return to the national tax authority.
- Output: Filed tax return.

2. Validation of Tax Return

- Input: Filed tax return.
- Activity: The tax authority reviews the return for:
    - Completeness.
    - Accuracy.
    - Missing or invalid information.
- Decision Point:
    - If valid: Proceed to Step 3.
    - If invalid: Notify the taxpayer to correct and resubmit.

3. Assessment of Refund Eligibility

- Input: Validated tax return.
- Activity: The tax authority evaluates eligibility based on the following criteria:
    1. Income Level: Determines applicable tax brackets and identifies potential overpayments.
    2. Deductions: Considers eligible deductions such as:
        - Mortgage interest.
        - Educational expenses (e.g., tuition fees).
        - Medical expenses, where applicable.
    3. Tax Credits: Applies credits, such as:
        - Child tax credit.
        - Disability tax credit.
        - Green energy credits.
    4. Tax Payments Made: Compares total tax payments (e.g., withholding, estimated payments) against tax liability.
    5. Residency Status: Adjusts eligibility for specific national or regional tax benefits:
        - Non-residents may have limited credits/deductions.
        - Residents may qualify for additional benefits.
    6. Special Circumstances: Considers unique cases, such as:
        - Disability benefits.
        - Disaster relief provisions.
- Decision Point:
    - If eligible for a refund: Proceed to Step 4.

- If not eligible or additional tax is owed: Notify taxpayer and end process.

4. Refund Calculation

- Input: Results from refund eligibility assessment.
- Activity: The tax authority calculates the refund amount by:
  1. Overpaid Taxes: Identifying if taxes paid exceed the liability.
  2. Applicable Deductions: Applying eligible deductions (e.g., mortgage, education, or medical).
  3. Tax Credits: Applying refundable and non-refundable credits (e.g., child tax credit, green energy credits).
  4. Residency Adjustments: Accounting for residency-specific benefits or limitations.
  5. Special Circumstances: Incorporating any additional benefits (e.g., disaster relief).
  6. Interest on Overpayments: Adding interest to refunds in cases of overpayment or delays.
- Output: A finalized refund amount is determined.

5. Issuance of Refund

- Input: Finalized refund amount.
- Activity: Refund is issued to the taxpayer:
  - Method: Direct deposit to bank account or mailed cheque.
- Output: Refund issued to taxpayer.

6. Notify Taxpayer

- Input: Refund status or owed taxes.
- Activity: Tax authority sends a notification with:
  - Confirmation of refund (amount, method, and any interest).
  - Explanation of adjustments, if applicable.
  - Details on tax owed, if no refund is issued.
- Output: Taxpayer informed of their refund or tax obligation.

7. End Process

- Output:
  - Process ends with either the issuance of a refund or notification of additional tax owed.
  - Any unresolved corrections or disputes may initiate additional cycles of the process.

Two explanation examples made by two LLMs are shown below. Note the style difference in the responses between the two LLMs.

Example 1: Adjustment for Ineligible Deduction

Explanation to the Taxpayer: "*During the processing of your tax return, we noticed that you claimed a deduction for educational expenses. However, our review revealed that the claimed expense does not meet the criteria for an eligible deduction under the applicable tax code (e.g., expenses must be for courses leading to a degree or certification). As a result, your taxable income has been adjusted upward by €2,000, which reduced your refund from €1,500 to €1,200.*"

Explanation to the Taxpayer: "*Upon reviewing your tax return, we identified that you claimed a deduction for educational expenses. However, this expense does not qualify as an eligible deduction under the tax code, which requires such expenses to pertain to courses leading to a degree or certification. As a result, your taxable income has been increased by €2,000. Consequently, your refund amount has been reduced from €1,500 to €1,200.*"

Example 2: Adjustment for Incorrect Tax Credit Application

Explanation to the Taxpayer: "*Our review identified an inconsistency in the tax credit for child dependents you applied for. While you claimed a credit for two children, the supporting documentation verified only one qualifying dependent. Accordingly, your tax credit has been adjusted from €800 to €400. This adjustment has reduced your refund amount from €2,000 to €1,600.*"

Explanation to the Taxpayer: *“Our review uncovered a discrepancy in the tax credit for child dependents you claimed. While you applied for a credit for two dependents, supporting documentation confirms eligibility for only one. Therefore, your tax credit has been revised from €800 to €400, reducing your refund from €2,000 to €1,600*."

3.2.7 Summary

Our main contribution is the creation of a novel scale for pragmatic assessment of the quality of explanations as perceived by users of business processes. Our study explores the understandability of business process explanations in natural language, testing whether LLM-generated text is inherently clear and suitable for users. A key innovation in our study is the use of causal execution knowledge to generate more interpretable explanations. The results reveal a trade-off between fidelity and interpretability. Specifically, adding causal execution or temporal sequencing knowledge enhances perceived fidelity, but this effect may diminish with low user trust in the LLM or limited curiosity about the problem. Caution is advised, as higher fidelity may reduce perceived interpretability.

Trust is an essential component in the adoption of AI based systems and their integration into the operational systems of organizations and institutions. Explainability in this regard may be seen as the glue between the different strategies driven by such entities and the intention of the users to use and abide to the regulations that underlie the operation of such system. Explanations can also serve to either expose, and sometimes also mitigate, the consequences of applying different policies and changes in a strategy. Consequently, trustworthiness is formed by the perceived quality of the explanations produced by the systems.

The capability to assess the consequences of applying different policies and regulations by organizations can only be assessed via feedback from those who interact with the AI-based system and “consume the explanation”. Hence, methodological instrumentation to measure such feedback is eminent for controlling the effect of potential interventions.

We have shown the suitability of our developed instrumentation to the governmental sector in the classical tax refund process.

4 Policy-Oriented AI and NLP Algorithms

In this section of the deliverable, the "Policy-Oriented AI and NLP algorithms" component is thoroughly specified, whilst extensive information about the source code and the corresponding user manual are provided. As also mentioned in D2.3 - Reference Architecture and Integration of AI4Gov Platform V1, the "Policy-Oriented AI and NLP algorithms" component is being developed in the context of T4.3 – "Improve Citizen Engagement and Trust utilising NLP" and consists of two (2) sub-components, namely *Policy-Oriented Analytics and AI Algorithms* and *Adaptive Analytics Framework*. The *Policy-Oriented Analytics and AI Algorithms* aims to develop several NLP algorithms in order to analyse large volumes of text data and also assist the respective AI experts. This particular subcomponent consists of the following mechanisms:

- Question Answering Service
- Time Series Analyser
- Multilingual Bias Classification

The scope of the *Adaptive Analytics Framework* subcomponent is to develop the needed ML models for performing predictive analytics and optimised resource allocation to satisfy the needs of the pilots and assist policy makers.

It should be mentioned that the aforementioned sub-components interact with each other in specific use cases, thus acting as input to one another and enriching the provided predictions. These interactions will be further analysed in the sections below. It is also worth mentioning that all the above should be executed in an efficient manner, utilising the least possible number of resources. Furthermore, based on a comprehensive literature review that was conducted (Mavrogiorgos K. K., 2024), great emphasis was given on several aspects of the implementation in order to mitigate possible bias that could originate from the datasets, the algorithms and hyperparameters utilized, as well as the metrics that were used to evaluate the said algorithms.

The following sections are organized as follows. First a state-of-the-art analysis is conducted in terms of the AI algorithms used for policy making, as well as current advancements in NLP, focusing on question-answering (QA) systems and approaches. Then, for both the subcomponents mentioned above, the architecture and internal workflow are thoroughly described. Furthermore, additional details are provided with regards to the engagement of citizens and the way that can interact and benefit from the aforementioned subcomponents. Moreover, for every subcomponent, the baseline technologies used are provided, whilst the availability and the key points for the source code are also examined. Last but not least, a user guide for each of the two (2) subcomponents is provided, focusing on the installation and use.

4.1 AI Algorithms for Policy Making

There have been several advancements regarding the fields of policy-oriented AI. To begin with, AI has been widely utilized in a variety of sectors in order to assist the corresponding policy makers. For instance, in the healthcare domain and during the recent COVID-19 pandemic, a variety of approaches (Rahman, 2021) were introduced in order to help public health experts form specific policies, protocols and interventions that could potentially protect the citizens from the pandemic. In those approaches, a variety of data analytics took place, and a vast number of ML algorithms were developed (Mavrogiorgou, 2021), in order to monitor the progress of the pandemic and predict certain outcomes, thus enabling policy makers to make the right decisions. Towards the same direction, i.e., assisting in data-driven decision making and, thus, creating more useful policies, AI is also being utilized in a variety of sectors such as e-Governance (Alexopoulos, 2019), transportation (Ağbulut, 2022) and the environment (Hettinga, 2023).

With regards to the AI4Gov's pilots and the corresponding sectors, which are water management, sustainability and tourism, there have also been some advancements in the literature. More specifically, regarding water management, the authors (Gino Sophia, 2020) proposed a genetic algorithm based on a fitness function that effectively manages water distribution with regression of 98%. In (Bhardwaj, 2022). The authors proposed a machine learning–based framework for the assessment of water quality by utilizing a variety of ML-based algorithms such as logistic regression, naïve Bayes, ensemble-based approaches such as Random Forest and XGBoost to classify the data in appropriate classes and predict turbidity in a water sample. As for sustainability, a variety of approaches have been proposed, such as the ones presented in (Shafiq., 2020) and aim to the classification of Sustainable Smart Cities (SSC) network traffic. Those approaches are based on several ML techniques; most of them are more traditional while very few approaches utilize more advanced techniques such as reinforcement learning, ensemble learning and genetic algorithms. Regarding tourism, several approaches have been developed that aim to address different use case scenarios, mostly for predicting tourism flows and performing sentiment analysis such as the ones presented in (Xie, 2021), (Li, 2021) and (Puh, 2023).

4.2 Current Advancements in NLP and QA

As for NLP, there have been tremendous advancements, since the high availability of resources and data have provided new opportunities, including Large Language Models (LLMs) that are capable of performing a vast variety of tasks, such as question-answering (QA), in different domains. QA is generally divided into two (2) subcategories named Closed Domain Question Answering (CDQA) and Open Domain Question Answering (ODQA). Based on the type of answers that are provided, and more specifically, whether the answer is extracted from the text, or it is generated based on the context of the text, QA can be divided into Extractive Question Answering (EQA) and Abstractive Question Answering (AQA).

CDQA, as its name implies, refers to QA approaches that answer questions of a specific domain. This is achieved through having a domain-specific knowledge base (e.g., a knowledge base that consists of articles about the environment) (Cortes, 2022). ODQA is about approaches that aim to answer questions regardless of the domain that they refer to. This is achieved by having a "global" knowledge base such as the whole Wikipedia (Zhong, 2022). ChatGPT and Google Bard[5] could be

[5] https://bard.google.com

identified as indicative examples of widely used ODQA tools. Depending on the user requirements and the volume of the knowledge base, both CDQA and ODQA systems may use EQA or AQA (Yoon, 2022).

Of course, answers provided by ODQA tools are known to be inaccurate, since the knowledge itself could be inaccurate (i.e., the Internet) and could also promote, among others, prejudices, fake information and hate speech, thus posing a threat to the users (Ray, 2023). CDQA tools could also have the same disadvantages, but at a very lower scale, since the knowledge base is domain-specific, smaller in size and probably not utilizing data just from the Internet (Antoniou, 2022). To this end, both the *Adaptive Analytics Framework* and the *Policy-Oriented Analytics and AI Algorithms* seek to implement appropriate AI techniques and NLP algorithms that satisfy the needs of the AI4Gov pilots whilst taking into consideration any potential challenges and ensuring the development of unbiased ML models.

4.3 Multilingual NLP

Our multilingual and multicultural societies express the need for the introduction of cross lingual and language-agnostic solutions. At the same time, the tremendous growth in the popularity and usage of social media, such as X (former Twitter), and of applications that support citizens in their interactions with public authorities and services, as well as gather their feedback, has resulted in an immense increase in user-generated data, as mainly represented by the corresponding texts in users' posts and complaints. However, the analysis of these specific data and the extraction of actionable knowledge and added value out of them is a challenging task due to the domain diversity and the high multilingualism that characterizes these data. Hence, leveraging the potentials that can be derived from the cross lingual analysis of them is crucial in the modern policy-making domain. In that direction, researchers are constantly trying to develop the most comprehensive multilingual systems.

Several AI research teams from major pioneers, such as Google AI and Facebook AI Research, have introduced multilingual tools, corpora and sentence encoding models that are able to cover any language, thus overcoming the limitations imposed by the lack of labelled data in all languages (Zorrilla, 2022). The successful design and implementation of multilingual solutions rely on incorporating multilingual sentence embeddings and employing multilingual classifiers, both built upon pre-trained models and the principles of transfer learning (Artetxe., 2019). Current methodologies extend beyond word embeddings to enhance multilingual NLP and capture deeper semantic meaning by utilizing embeddings for higher-level structures, such as sentences or even paragraphs. Existing approaches for generating such embeddings, like LASER (Artetxe, 2019) or MUSE (Lample, 2017), rely on parallel data, mapping a sentence from one language directly to another language to encourage consistency between the sentence embeddings. In addition, the authors in (Feng, 2020) present a multilingual BERT embedding model, called LaBSE, that produces language-agnostic cross-lingual sentence embeddings for 109 languages that is highly effective even on low-resource languages for which there is no data available during training.

One of the latest milestones in the NLP field is the introduction of BERT that enables transfer learning with large language models reaching the state-of-the-art for a great number of NLP tasks and applications (Devlin, 2018). In this context, several research works have proposed multilingual models based on the utilization of BERT for a wide range of cross-lingual text classification tasks. More specifically, advances in multilingual language models such as multilingual BERT (mBERT) (Pires, 2019) and XLM-RoBERTa that are trained on a huge corpus in over 100 languages indicate promising approaches and solutions for the implementation of multilingual applications and have been characterized as benchmarks and introduced remarkable results in Multilingual Text Classification tasks (Wang, 2021). To leverage the potential of these approaches, a Multilingual Bias Classification (MBC) tool is introduced under the scopes of the AI4Gov project. This tool is planned to be evaluated and validated in the context of the OECD Scenario and dataset as it has been described in the context of D6.1 - "Specification of UC Scenarios and Planning of Integration and Validation Activities V1".

Moreover, the growing use of AI in policy analysis and decision-making process has highlighted critical concerns regarding trustworthiness, explainability, and bias in AI systems. These challenges are particularly significant in critical domains such as governance, where decisions informed by biased or unreliable AI systems can result on major implications and consequences, including reinforcing unfair systems, losing public trust, and making poorly informed decisions.

Trustworthiness in AI requires stakeholders to have confidence that the system produces reliable, accurate, and fair outputs. In the context of policy analysis, this involves ensuring that AI systems do not favour particular groups, perspectives, or interests based on implicit or explicit biases embedded in their training data or algorithms. Explainability, another cornerstone of trustworthy AI, refers to the ability of the system to provide transparent and comprehensible reasoning for its outputs. This is particularly crucial in policy contexts, where stakeholders must not only trust the AI's conclusions but also understand the reasoning behind them to ensure accountability and align decisions with ethical and societal norms. Bias, both implicit and explicit, poses a unique challenge, as it risks reinforcing stereotypes, marginalizing vulnerable populations, and undermining the legitimacy of policy decisions.

In that context, the initially implemented and evaluated MBC component has revised and enhanced through the utilization of a Retrieval-Augmented Generation (RAG) mechanism as further detailed in later sections.

4.4 Adaptive Analytics Framework

As also mentioned above the *Adaptive Analytics Framework* component is being developed in the context of T4.3 – "Improve Citizen Engagement and Trust utilising NLP". The scope of this component is to develop the needed ML models for efficiently performing predictive analytics and optimised resource allocation to satisfy the needs of the pilots and assist policy makers.

In the following subsections further information about this component are provided, including the architecture and the internal workflow, the baseline technologies, the source code and a user guide.

4.4.1 Architecture and Internal Workflow

The architecture of the *Adaptive Analytics Framework* is depicted in Figure 23 and thoroughly analyzed below.

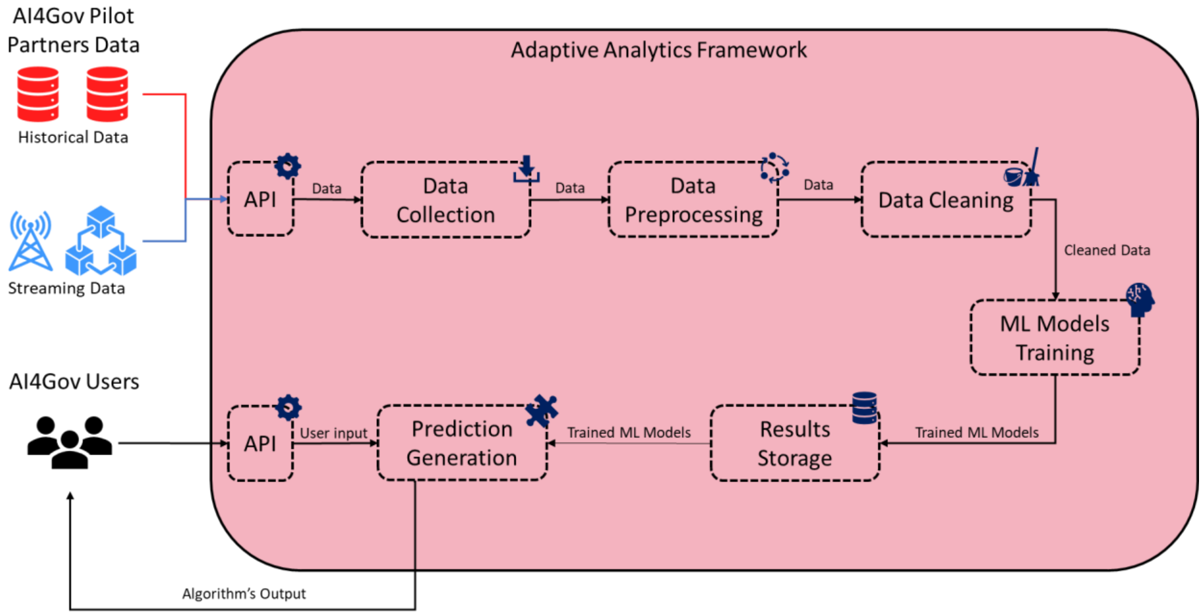


Figure 23: Architecture of Adaptive Analytics Framework Component

As shown in the figure above, the *Adaptive Analytics Framework* retrieves both historical data and streaming data from the corresponding AI4Gov pilot partners in order to analyze them and offer the appropriate predictive ML models. This component is being utilized in Pilot 1 (i.e., Policies for Sustainable Water Cycle Management at a Large Scale) and Pilot 3 (i.e., Tourism-driven multi-domain policy management and optimisation), thus it collects and analyzes the corresponding data. More specifically, dedicated APIs (Application Programming Interfaces) have been developed in order to enable the component to retrieve the data from the project's interim repository where all the data from the corresponding data sources are being stored. After the data collection step, the data preprocessing and data cleaning steps take place. In those specific steps, the appropriate techniques are utilized in order to preprocess the data, remove/replace any erroneous values that may affect the training of the ML models, and finally reshape the datasets that are to be trained in a desired form. Since the data are cleaned, the training of the corresponding ML algorithms occurs. In this case, there have been developed a variety of ML algorithms that are trained on the datasets and the ones with the best results in terms of evaluation metrics are then stored in order to allow the users to make predictions based on the trained ML model. Those algorithms include variations of the Decision Tree algorithm, the KNN algorithm, the K-means algorithm and a set of algorithms that is used for solving the vehicle routing problem.

In deeper detail and with regards to Pilot 1, the KNN algorithm is used as a data cleaning approach in order to predict the missing values that are present in the provided datasets so that they can then be utilized by the *Policy-Oriented Analytics and AI Algorithms sub-component to train specific* *models for time series forecasting.* As for Pilot 3, a modified version of K-means is used to split geographical areas into clusters and the Decision Tree and KNN algorithms are used to perform classification tasks in the context of both the use cases of this specific pilot.

The users of the AI4Gov project are able to communicate with the component through dedicated APIs, either by utilizing the user interface of the Visualization Workbench, since those two components are integrated, or by performing specific HTTP requests with the appropriate parameters themselves. Based on the provided input by the user (i.e., specific parameters to use for prediction) and the trained ML model utilized, a prediction is generated which is then provided back to the user. At this point, it is also worth mentioning that in most cases, the users not only retrieve a prediction but also, based on the requirements of the corresponding use case, interactive diagrams and maps. More details about the usage of the component will follow in section 4.4.5.

4.4.2 Citizens' Engagement

The *Adaptive Analytics Framework* is also a citizen-centric component. This means that specific functionalities have been implemented, in order to encourage the citizens to interact with the trained ML models and benefit from the said models. More specifically, there exist three (3) functionalities that can be utilized by the citizens, which have been developed in the context of the 3rd Pilot. The first one is called "Check Traffic Congestion" where citizens/visitors can be informed of the predicted traffic congestion in several roads on a specific date and time, thus being able to avoid the roads that are prone to high traffic congestion in specific times of the day and choose alternative routes. Similarly, the second functionality is called "Check Roads Safety". In this case, citizens/visitors can be informed of roads thar are more prone to traffic accidents on a specific date and time, thus enabling them to be more cautious. Lastly, the third functionality, which is also available in the *Policy-Oriented Analytics and AI Algorithms*, refers to the implementation of a feedback mechanism, which enables the citizens/visitors to provide their feedback with regards to the provided predictions and the way that they are presented to them. By doing so, the citizens are actively participating in the improvement of the ML models that have been trained to improve their everyday lives.

4.4.3 Baseline Technologies

Adaptive Analytics Framework is based on specific technologies. First of all, the component has been developed with the use of the Python[6] programming language, since there exists a plethora of modules available that are ideal for several ML tasks. Moreover, JavaScript[7] has also been utilized in order for the component to generate the interactive diagrams and maps that were mentioned above. Moreover, the component is also available as Docker Image[8], thus allowing its

[6] https://www.python.org

[7] https://developer.mozilla.org/en-US/docs/Web/JavaScript

[8] https://www.docker.com

flawless integration with the other components of the AI4Gov platform. A complete list of the Python modules that are currently being utilized by the *Adaptive Analytics Framework,* can be found below.

- *certifi*[9]*: provides validation of SSL certificates.*
- *Flask*[10]*, Flask_Cors*[11]*, Requests*: utilized for developing APIs.
- *GeoPy*[12]*:* manage and analyse geographical data.
- *matplotlib*[13]*, plotly*[14]: used for creating plots, maps, diagrams both static and interactive.
- *numpy*[15]: used for performing mathematical operations.
- *ortools*[16]*:* provides appropriate tools for solving optimization problems.
- *pandas*[17]: used for data manipulation.
- *scikit_learn*[18]*, scipy*[19]*, tensorflow*[20]: provide appropriate tools for the development of ML algorithms.
- *ydata-profiling*[21]*: provided appropriate tools for performing descriptive analysis on data.*

[9] https://pypi.org/project/certifi/

[10] https://flask.palletsprojects.com/en/3.0.x/

[11] https://flask-cors.readthedocs.io/en/latest/

[12] https://geopy.readthedocs.io/en/stable/

[13] https://matplotlib.org

[14] https://plotly.com/python/

[15] https://numpy.org/

[16] https://developers.google.com/optimization

[17] https://pandas.pydata.org

[18] https://scikit-learn.org/stable/

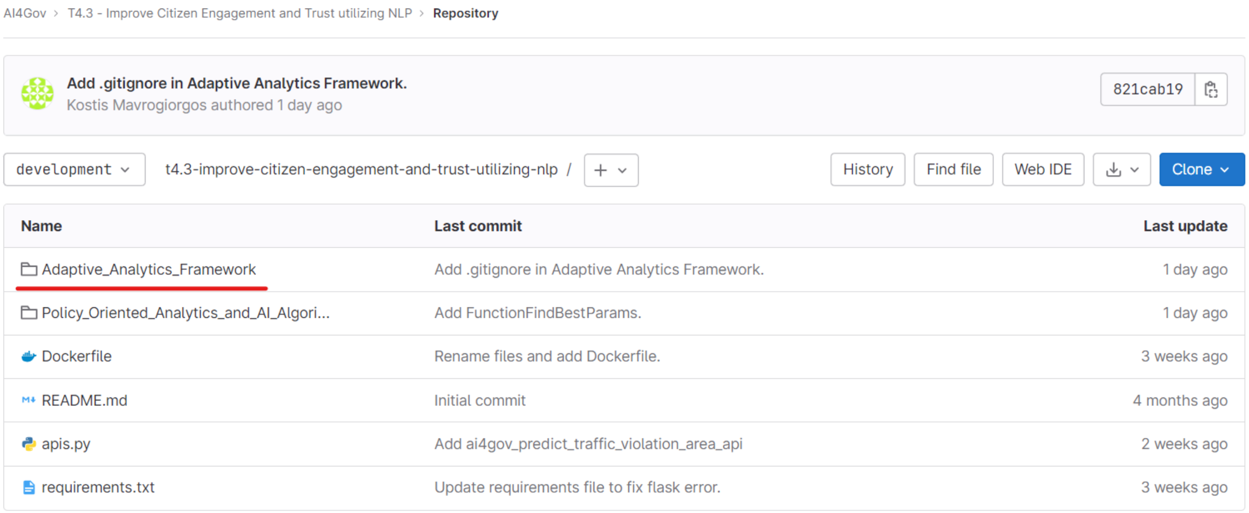
[19] https://scipy.org

[20] https://www.tensorflow.org

[21] https://github.com/ydataai/ydata-profiling

4.4.4 Source Code - Availability and Key Points

The source code of the *Adaptive Analytics Framework* is available on the project's GitLab repository under the GitLab project named "T4.3 - Improve Citizen Engagement and Trust utilizing NLP", as shown in Figure 24.



AI4Gov > T4.3 - Improve Citizen Engagement and Trust utilizing NLP > **Repository**

Add .gitignore in Adaptive Analytics Framework.
Kostis Mavrogiorgos authored 1 day ago

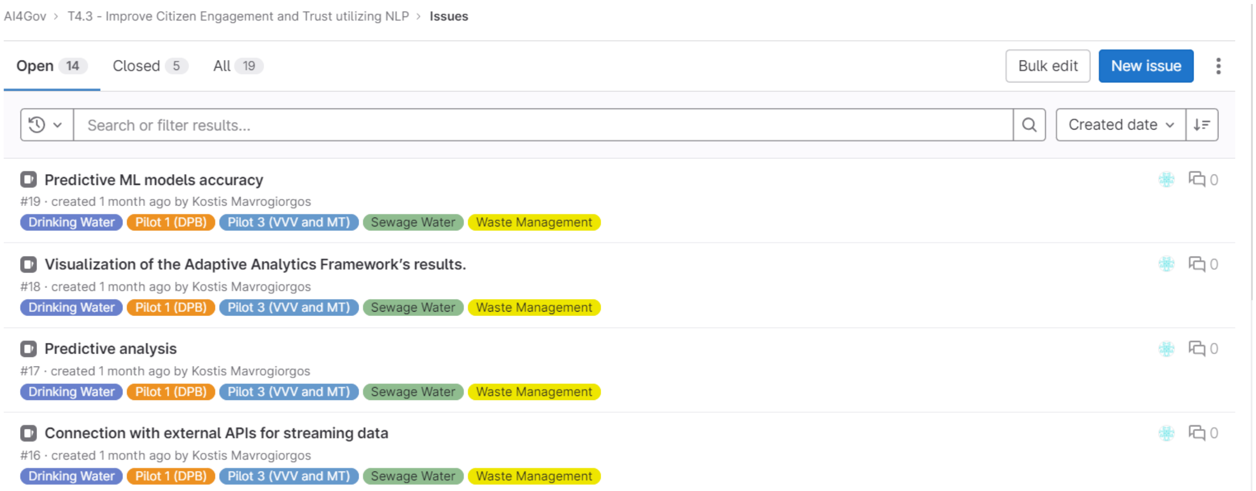
821cab19

development · t4.3-improve-citizen-engagement-and-trust-utilizing-nlp /

Name	Last commit	Last update
Adaptive_Analytics_Framework	Add .gitignore in Adaptive Analytics Framework.	1 day ago
Policy_Oriented_Analytics_and_AI_Algori...	Add FunctionFindBestParams.	1 day ago
Dockerfile	Rename files and add Dockerfile.	3 weeks ago
README.md	Initial commit	4 months ago
apis.py	Add ai4gov_predict_traffic_violation_area_api	2 weeks ago
requirements.txt	Update requirements file to fix flask error.	3 weeks ago

Figure 24: Adaptive Analytics Framework Source Code on GitLab

It is also worth mentioning that in order to simplify the development process and ensure that all the user requirements are met, corresponding issues have been created in the GitLab repository, as shown in Figure 25.



AI4Gov > T4.3 - Improve Citizen Engagement and Trust utilizing NLP > **Issues**

Open 14 · Closed 5 · All 19

Predictive ML models accuracy
#19 · created 1 month ago by Kostis Mavrogiorgos
Drinking Water · Pilot 1 (DPB) · Pilot 3 (VVV and MT) · Sewage Water · Waste Management
0

Visualization of the Adaptive Analytics Framework's results.
#18 · created 1 month ago by Kostis Mavrogiorgos
Drinking Water · Pilot 1 (DPB) · Pilot 3 (VVV and MT) · Sewage Water · Waste Management
0

Predictive analysis
#17 · created 1 month ago by Kostis Mavrogiorgos
Drinking Water · Pilot 1 (DPB) · Pilot 3 (VVV and MT) · Sewage Water · Waste Management
0

Connection with external APIs for streaming data
#16 · created 1 month ago by Kostis Mavrogiorgos
Drinking Water · Pilot 1 (DPB) · Pilot 3 (VVV and MT) · Sewage Water · Waste Management
0

Figure 25: Sample of GitLab Issues Created for the Adaptive Analytics Framework

4.4.5 User Guide – Installation and Use

Regarding the installation of the *Adaptive Analytics Framework,* this is a straightforward process since the component has been containerized (i.e., a corresponding Docker container has been created) and is available through the AI4Gov GitLab's container registry.

When the installation is complete, the component will be available through a specified port, and the following APIs will be available for use.

Table 4: Description of the ai4gov_interactive_density_mapbox_api

Endpoint Name	**ai4gov_interactive_density_mapbox_api**
Endpoint URL	http://[SERVER IP]:PORT/ai4gov_interactive_density_mapbox_api
Description	This endpoint is responsible for creating interactive density mapboxes that showcase the geospatial evolution of data in the corresponding use cases for a specific range of dates that are defined by the "date_from" and "date_to" variables shown below. This endpoint is used in the context of the 3rd pilot (Municipality of Vari-Voula-Vouliagmeni and Ministry of Tourism)
HTTP Method	POST
Parameters	date_from: date in YYYY-MM-DD format date_to: date in YYYY-MM-DD format use_case: available value is "parking_tickets". If "parking_tickets" is selected, then an interactive density mapbox is generated that showcases the geospatial evolution of traffic violations tickets issued for a specific type of traffic violation in the given time range. violation_type: the type of violation to use for filtering the data and providing the corresponding interactive density mapbox when the "use_case" parameter is equal to "parking_tickets"
Response	.json file containing the corresponding interactive density mapbox

Table 5: Description of the ai4gov_routing_optimization_api

Endpoint Name	ai4gov_routing_optimization_api
Endpoint URL	http://[SERVER IP]:PORT/ai4gov_routing_optimization_api
Description	This endpoint is responsible for solving the capacitated vehicle routing problem for the optimization of the garbage trucks' routes. This endpoint is also being used in the context of the 3rd pilot (Municipality of Vari-Voula-Vouliagmeni and Ministry of Tourism)
HTTP Method	POST
Parameters	date: date in YYYY-MM-DD format for which the optimization of routes should take place
Response	.json file containing the optimal routes for each vehicle

Table 6: Description of the ai4gov_routing_visualization_api

Endpoint Name	ai4gov_routing_visualization_api
Endpoint URL	http://[SERVER IP]:PORT/ai4gov_routing_visualization_api
Description	This endpoint is responsible for visualizing the routes that result from solving the capacitated vehicle routing problem for the optimization of the garbage trucks' routes. This endpoint is also being used in the context of the 3rd pilot (Municipality of Vari-Voula-Vouliagmeni and Ministry of Tourism)
HTTP Method	GET
Parameters	bin_category: can either be "Organics Smart Bin" or "Green Smart Bin"
Response	.json file containing the corresponding visualization

Table 7: Description of the ai4gov_predict_traffic_violation_area_api

Endpoint Name	**ai4gov_predict_traffic_violation_area_api**
Endpoint URL	http://[SERVER IP]:PORT/ai4gov_predict_traffic_violation_area_api
Description	This endpoint is responsible for predicting the area in the municipality of VVV where it is more likely a specified traffic violation might occur
HTTP Method	GET
Parameters	algorithm_name: the name of the algorithm that will be used for making the prediciton part_of_day: available values are 0 and corresponds to morning, 1 and corresponds to midday, 2 and corresponds to afternoon, 3 and corresponds to night), 'Violation' (– there is a dictionary that maps the integer number to the corresponding violation, in case you need it just let me know), 'Month', 'Week' (0-> Weekday, 1-> Weekend) violation: integer number corresponding to the type of the traffic violation month: integer number ranging from 0 (January) to 11(December) week: boolean value, 0 corresponds to weekday and 1 corresponds to weekend
Response	.html file containing an interactive map showcasing the area where it is more likely the specified traffic violation might occur

Table 8: Description of the ai4gov_routing_visualization_api

Endpoint Name	**ai4gov_check_traffic_congestion_api**
Endpoint URL	http://[SERVER IP]:PORT/ai4gov_check_traffic_congestion_api
Description	This endpoint is responsible for visualizing a map where roads that are predicted to have high traffic congestion are marked with red and roads that are predicted to have low traffic congestion are marked with green. This endpoint is also being used in the context of the 3rd pilot (Municipality of Vari-Voula-Vouliagmeni and Ministry of Tourism)
HTTP Method	POST

Parameters	datetime_str: a DateTime string (possibly from a Datepicker) related to the date and time that the citizen would like, for example, to find a parking space
Response	.json file containing the corresponding prediction and visualization

Table 9: Description of the ai4gov_check_roads_safety_api

Endpoint Name	**ai4gov_check_roads_safety_api**
Endpoint URL	http://[SERVER IP]:PORT/ai4gov_check_roads_safety_api
Description	This endpoint is responsible for visualizing a map where roads that are predicted to be more prone to traffic accidents are marked with red and roads that are predicted to less prone to traffic accidents are marked with green. This endpoint is also being used in the context of the 3rd pilot (Municipality of Vari-Voula-Vouliagmeni and Ministry of Tourism)
HTTP Method	POST
Parameters	datetime_str: a DateTime string (possibly from a Datepicker) related to the date and time that the citizen would like to travel
Response	.json file containing the corresponding prediction and visualization

Table 10: Description of the ai4gov_store_feedback_api

Endpoint Name	**ai4gov_store_feedback_api**
Endpoint URL	http://[SERVER IP]:PORT/ai4gov_store_feedback_api
Description	This endpoint is used in order to provide feedback with regards to a provided answer
HTTP Method	POST
Parameters	user_id: the ID of the user that provided the feedback user_role: the role of the user (e.g., citizen) use_case: the name of the use case to which this feedback refers to

	related_question: the related question that was asked by the user related_answer: the corresponding answer that was provided by the platform feedback: the user’s feedback
Response	Status code related to the successful submission of the feedback

A video demonstrating the abovementioned functionalities can be found here.

4.5 Policy-Oriented Analytics and AI Algorithms

As also mentioned previously, *Policy-Oriented Analytics and AI Algorithms* is being developed in the context of T4.3 – “Improve Citizen Engagement and Trust utilising NLP”. Its aim is to develop several NLP algorithms in order to analyse large volumes of text data and also assist the respective AI experts. This particular component consists of the following subcomponents:

- Question Answering Service: this service provides the necessary tool for allowing the AI experts, developers, and policy makers to perform queries on the OECD papers regarding, among others, raising awareness among them of AI solutions.
- Time Series Analyser: this tool supports the analysis of time series and historical data in order to discover possible trends that will support the corresponding users in the water management cycle and the parking tickets monitoring use cases.
- Multilingual Bias Classification: this tool supports the multilingual identification and classification of bias in the OECD papers, providing all stakeholders information and enhanced knowledge of the types of bias that governments and public authorities take into consideration in their AI policies. This tool is a standalone component that is directly integrated and exposed through the Visualization Workbench of the project offering an interface for the. The finalization of its integration will be further reported in D4.4 – ”Policies Visualization Services V2”.

Lastly, the aforementioned subcomponents follow the guidelines proposed by the Bias Detector Toolkit component in order to address possible bias in the whole workflow of the *Policy-Oriented Analytics and AI algorithms* component.

In the following subsections, further information about this component is provided, including the architecture and the internal workflow, the way that the citizens are able to interact and benefit from the component, the baseline technologies, the source code and a user guide.

4.5.1 Architecture and Internal Workflow

The architecture of the *Policy-Oriented Analytics and AI Algorithms* is depicted in Figure 26 and thoroughly analyzed below. For the sake of completeness, a simplified version of the architecture, showcasing the internal workflow of this component is also provided below.

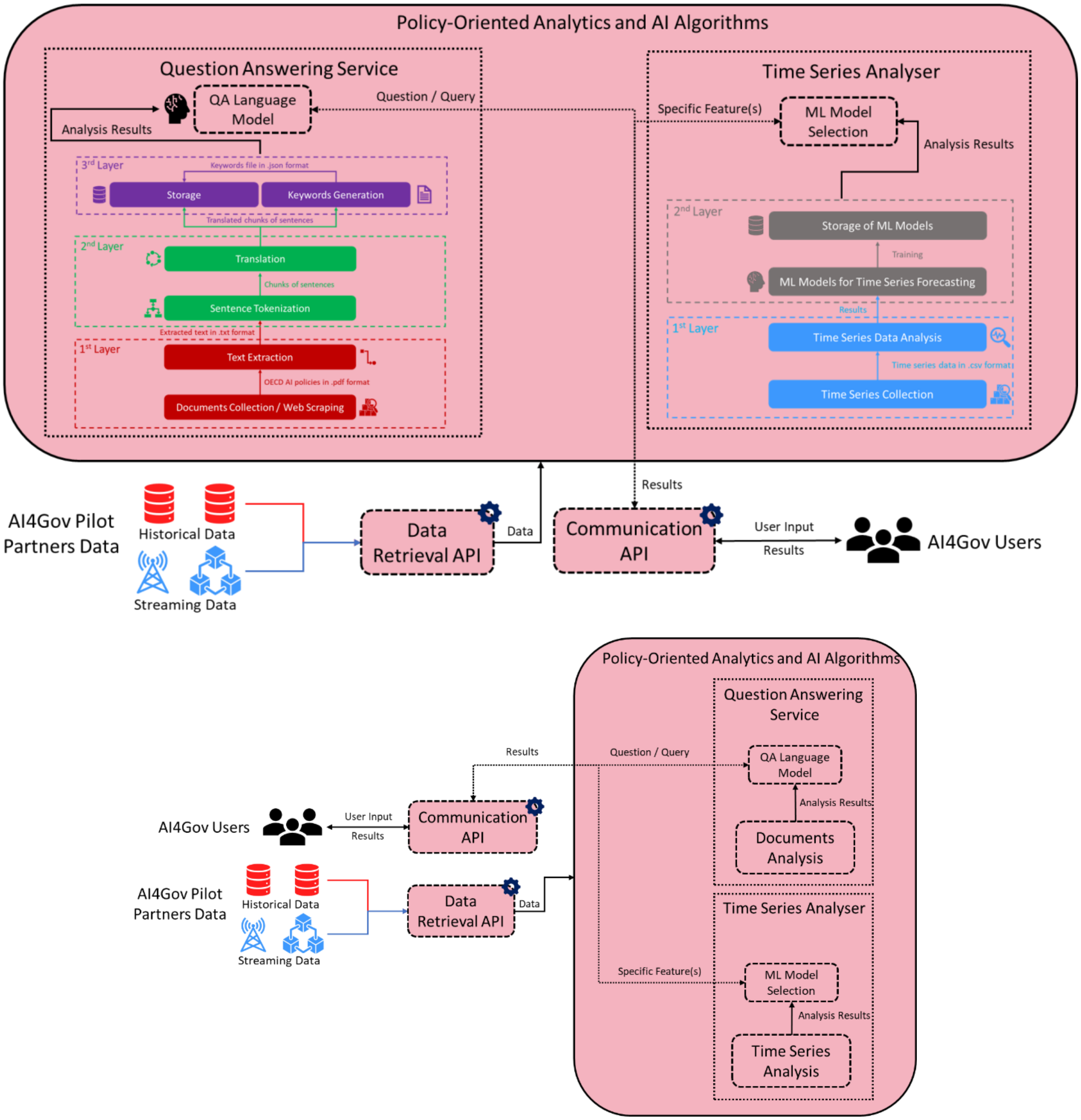


Figure 26: Architecture of Policy-Oriented Analytics and AI Algorithms Component

As mentioned previously and is also shown in figure above, the *Policy-Oriented Analytics and AI Algorithms* consists of two (2) integrated sub-components, namely the *Question Answering Service* and the *Time Series Analyser,* while the *Multilingual Bias Classification sub-component* will be integrated in later stages. The *Question Answering Service* is being utilized in the 2nd pilot (i.e., Sustainable Development and the European Green Deal) and, more precisely, the OECD policy papers use case. The component seeks to provide the ability to ask AI-related questions to a large knowledge base that consists of official and trustworthy policies and documents of several OECD countries, which refer to the way that each country introduces AI in governance, including potential dangers and methods to mitigate them. As depicted in the figure above, this sub-component consists of three (3) layers, each of them consisting of two (2) steps. Regarding the first layer, it consists of the "Documents Collection / Web Scrapping" step and the "Text Extraction" step. The AI policy documents that construct the knowledge base of the mechanism are publicly available in the OECD AI Policy Observatory[22]. For every available country, there exists a web page that contains a number of useful information about the corresponding policy, except for the policy itself, which consists of one hundred (100) pages on average, such as a short description of it, the name of the responsible governmental body and the objectives of the policy document. In the context of the proposed approach, the policy document should be collected, as well as the URL (Uniform Resource Locator) where it is available. In order to retrieve the aforementioned publicly available information, and since there is no dedicated API (Application Programming Interface), the proposed mechanism performs web scraping, thus retrieving the policy documents in PDF format and the corresponding URLs. As long as the documents are stored, the text extraction step takes place. The policy documents not only contain text, but also images, tables and other text styling that are not useful for the development of the knowledge base and, thus, have to be removed. In the text extraction step, as its name implies, the text from the policy documents is extracted and then stored in plain text files, so that it can later be efficiently manipulated by the rest of the mechanism.

Given the fact that the first layer is complete, the sentence tokenization and the translation of those sentences occur. The policy documents are multilingual, meaning that either they should be translated into a specific language, or different language models should be used when performing the QA task. The latter would be quite costly in terms of resources and, as a result, it is preferred to translate all those text to a certain language, and most specifically English, since most of them are already in that particular language. However, the volume of the text extracted from each policy document is massive, so attempting to translate a whole policy text would be costly or even impossible in terms of resources. As a result, in the proposed mechanism, every text extracted from a policy document is being tokenized by sentences. Then, those sentences are translated into chunks, which is much more efficient in terms of execution time and resource allocation. It should be highlighted that the tokenization of the text is by sentence in order to ensure that the meaning of the text is not being altered in any way. For example, if it was being performed based on a default size of text chunks, then it would be highly possible that a lot of sentences would be trimmed, thus affecting the accuracy of the translation. Moreover, it should be mentioned that for the translation part, opensource APIs are utilized, which are capable of automatically identifying the language of the text and then translating it to English. The selection

[22] https://oecd.ai/en/

of those APIs, instead of using dedicated machine translation models locally, was made in order to reduce the cost of the mechanism in terms of resources.

Since the texts have been tokenized and translated in chunks, the third layer of the mechanism follows. This layer aims to reduce the response time of the mechanism when a user performs a question. Without this layer, the mechanism should open and search all the translated text chunks in order to find the corresponding answer, which would be catastrophic for the performance of the mechanism and could even lead to bottlenecks. In order to address this issue, the "Keywords Generation" step takes place. In this step, keywords are extracted from every chunk of the translated texts. More specifically, words of high importance such as nouns and verbs are extracted, excluding words that are not of high value, such as conjunctions. The keywords found for every chunk file, along with the corresponding chunk file name, are stored in a JSON (i.e., JavaScript Object Notation) file.

Having the aforementioned JSON file as an index means that whenever a user performs a question on the OECD knowledge base, this JSON file is searched based on the words that the question contains and, as a result, only the text chunks that contain the relative keywords are retrieved and searched for providing the answer. In order to extract the answer from the text chunks, the "deepset/roberta-base-squad2" language model[23] that was retrieved from the Hugging Face repository[24] was utilized. This model was selected because it was trained for QA tasks and it is the finetuned version of "roberta-base" language model[25], meaning that it is quite efficient for limited resources. In order to ensure that the extracted answers are in an understandable by the user's form, a large language model (i.e., LLM) is also being utilized in order to rephrase the extracted answer. After several tests and given the constraints in terms of resources, the LLM that was chosen was the "microsoft/Phi-3-mini-4k-instruct-gguf"[26], since it is specifically finetuned for running only on CPU. It should also be mentioned that an answer may be found in numerous text chunks. In that case, the proposed mechanism provides the answer with the highest accuracy. However, all the available answers, along with the text chunks that contain them, can also be retrieved by the user, in case the answer with the highest accuracy is not sufficient. It is also worth noting that the above-mentioned approach has been accepted and published in a corresponding scientific conference (Mavrogiorgos, 2023).

Regarding the *Time Series Analyzer,* it is being utilized in the first and the third pilot of the AI4Gov project, where time series data are available from the pilot partners. In order to retrieve the time series data in real time and periodically from the pilot partners, a dedicated API is available. This API is not directly connected to the data sources but to the interim repository of the AI4Gov project, where all the data are available for all the technical components of the project. Since the data are retrieved from the interim repository, they are firstly being sent to the *Adaptive Analytics*

[23] https://huggingface.co/deepset/roberta-base-squad2

[24] https://huggingface.co/tasks/question-answering

[25] https://huggingface.co/roberta-base

[26] https://huggingface.co/microsoft/Phi-3-mini-4k-instruct-gguf

Framework in order to predict the missing values that may be present in the data. Then, the cleaned data are sent back to the *Time Series Analyzer* in order to train the appropriate algorithms for time series forecasting. Having experimented with several algorithms and architectures, the most appropriate ones were selected for each pilot use case scenario.

More specifically, with regards to the drinking water use case of the 1st pilot, a long short-term memory recurrent neural network (i.e., LSTM RNN) was developed in order to perform time series forecasting on the corresponding data. This LSTM predicts the values of the variables of interest with a forecast horizon of six (6) hours, given twenty-four (24) hours from the past. The developed neural network architecture also incorporates a layer, which is the result of the integration of the tasks 4.2 and 4.3, that enables it to provide "sufficient reasons". More details with regards to the theoretical background of "sufficient reasons" are available in section 3. More specifically, the "sufficient reasons" layer provides further insights with regards to the way that the LSTM operates by identifying the features (in respective time points) that affect the predicted value. For instance, assuming that the historical data consist of four (4) variables named pH, Clorides, Water Level and Instant Output Quantity and the feature that needs to be predicted six hours into the future is the pH, alongside with the LSTM's prediction, the plot shown below is also provided. The following plot shows four types of features in the 24 hours prior to the pH prediction, used to make the pH prediction. In green are features (in respective time points) that, together, are sufficient for the pH prediction. That is, fixing their values, the values of the red features can reasonably change, and the pH prediction will remain similar. The generated prediction, as well as the generated plot (i.e., explainability report) are also anchored to the AI4Gov project's blockchain at the time that they are created by the *Policy-Oriented Analytics and AI Algorithms.* That way, the users are able to validate what they see in front of them with what is anchored to the blockchain, thus ensuring the validity of the prediction and the explainability report and the fact that they have not been tampered by any third-party entity. Further details with regards to the utilization of blockchain in the context of the AI4Gov project can be found in the corresponding deliverable D3.2 – "Decentralized Data Governance, Provenance and Reliability V2".

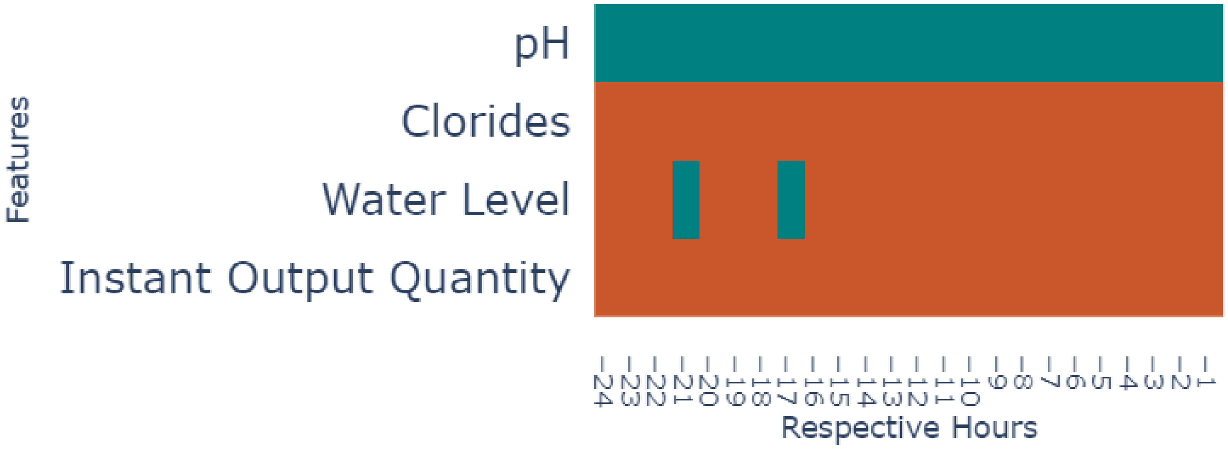


Figure 27: Example of sufficient reasons in the context of the drinking water use case

Similarly, in the context of the second use case of the first pilot (i.e., sewage water), another LSTM RNN has been designed and trained. This LSTM predicts the values of the variables of interest with a forecast horizon of one (1) day, given seven (7) days from the past. The developed neural network architecture also utilizes the “sufficient reasons” layer which provides an output similar to what was described in the drinking water use case. Similar to the drinking water use case, the generated prediction, as well as the generated plot (i.e., explainability report) are also anchored to the AI4Gov project’s blockchain at the time that they are created by the Policy-Oriented Analytics and AI Algorithms.

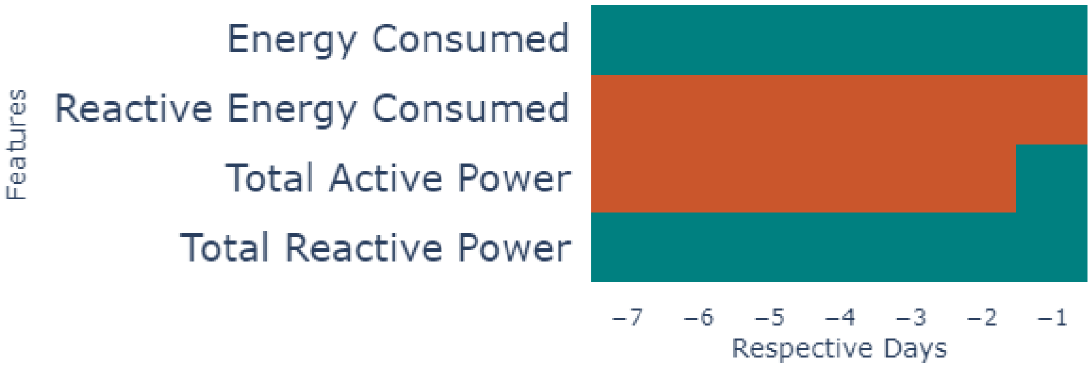


Figure 28: Example of sufficient reasons in the context of the sewage water use case

As for the 3rd pilot, the Timeseries Analyzer is being utilized in the context of the waste management use case, in order to train an ML model for predicting the fill levels of the garbage bins and also predict the citizens’ flows based on the rate that the bins are getting filled. The latter is based on the fact that the fill level of the bins that are located in areas with many visitors is increasing at a higher rate than the fill level of the bins that are located in the areas that have less visitors. In the context of this use case, another LSTM RNN has been developed that has been trained on data coming from all the smart bins of the Municipality of Vari-Voula-Vouliagmeni. Currently, this model has a forecast horizon of twelve (12) hours, given twenty-four (24) hours from the past but this can be increased given the fact that more data are available in the future. As long as the forecasts have been made by the model, the corresponding results are available to the clustering algorithms of the *Adaptive Analytics Framework,* in order to split the municipality in areas of interest in terms of the rate at which the bins get filled, thus providing insights with regards to the number of people that live and/or visit each area. The forecasts are also available to the optimization algorithms that are responsible for providing the optimal route for the garbage trucks, since this optimization relies on the predicted fill level of every bin. It is worth noting that the implementation of the Timeseries Analyzer and some initial results have also been accepted and published in a related conference paper (Mavrogiorgos K. K., 2024).

As concerns the *Multilingual Bias Classification* subcomponent (Figure 29), it has been utilized in the context of the second pilot and, more specifically, in the OECD dataset, where the identification of bias and the classification of the different policy documents based on it is of highest importance. In the context, of this second version of this series of deliverables, the MBC component has been integrated with a RAG mechanism. The MBC component focuses on detecting and classifying biases in OECD textual data, leveraging state-of-the-art PLMs fine-tuned on the StereoSet dataset (Nadeem, 2020). The categorization of biases into specific types such as gender, race, or profession, enables a comprehensive analysis of how biases manifest in policy-related texts. This classification is vital for identifying areas of concern, as well as for providing a foundation for understanding systemic patterns of bias across diverse datasets.

The RAG component enhances the methodology by addressing the challenge of explainability. While the MBC component identifies biases, RAG provides detailed and contextually relevant justifications for these classifications. This is achieved by retrieving information from curated knowledge bases that include academic literature, historical contexts, and region-specific cultural considerations. By presenting this information alongside bias classifications, RAG ensures that stakeholders are equipped with actionable insights that go beyond mere identification. For instance, if a policy document disproportionately emphasizes male leadership roles, RAG can retrieve evidence on the impact of gendered language in perpetuating workplace inequalities, thereby contextualizing the classification and supporting informed discussions on potential revisions.

The latter provides added value and improved information to the AI experts and all stakeholders by indicating the type of bias that each government takes into account in its AI-related policy actions, by also justifying and elaborating on the final result towards improving the transparency of the model.

The internal architecture and workflow are presented and described below to depict its overall functionality and internal pipeline in terms of data processing, analysis and models used so far for an initial evaluation and comparison in terms of their performance and accuracy.

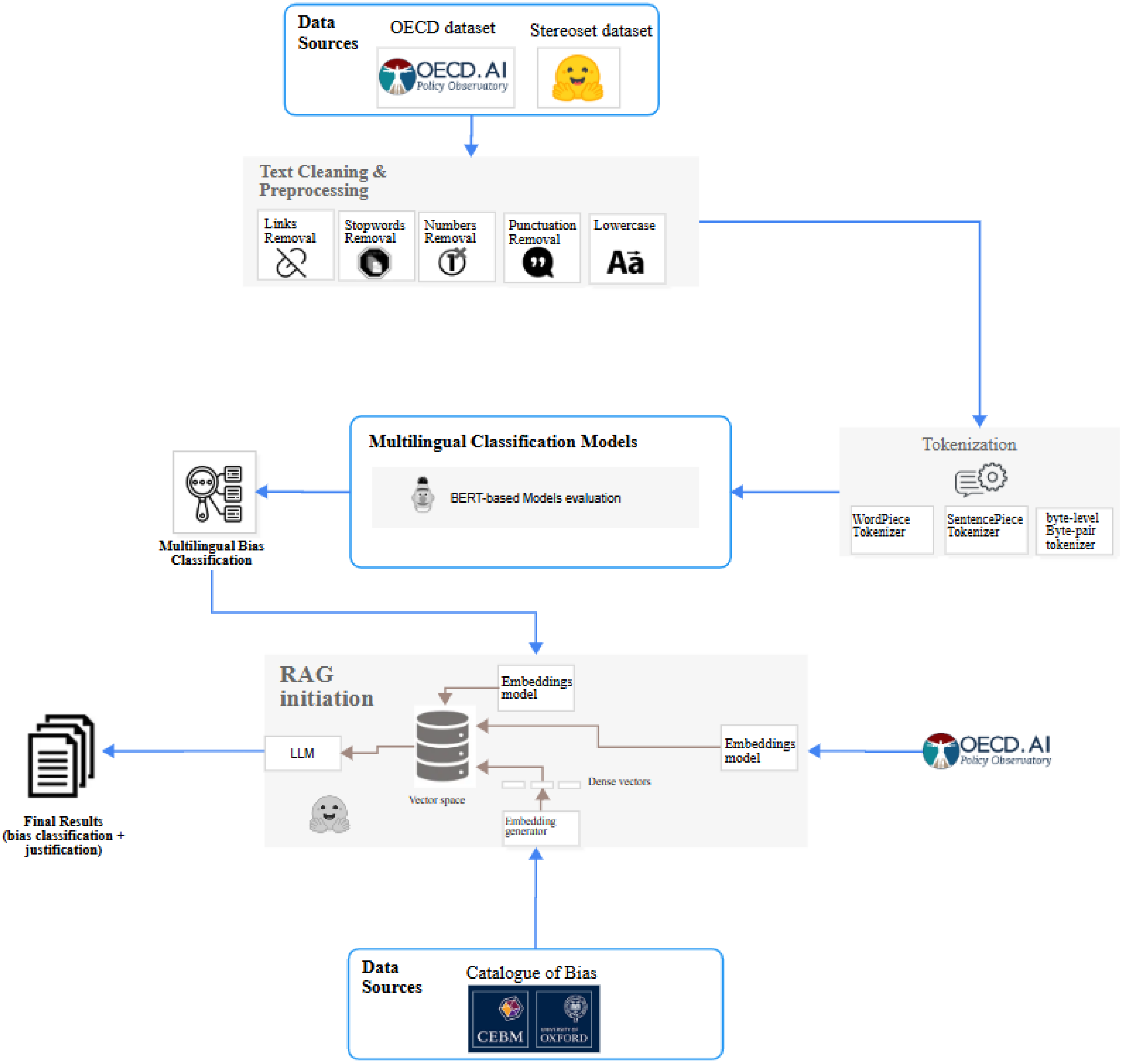


Figure 29: Multilingual Bias Classification Tool

Its main concept is based on the principles of transfer learning where the models are initially trained on a labelled dataset and then are applied on an unlabeled dataset, while some of the models are also evaluated for their performance in a zero-shot classification without any prior training. This sub-component incorporates in its different steps techniques from the fields of ML and NLP, starting from the pre-processing and cleaning of the data, where NLP techniques, such as stopwords, punctuation, lowercasing and links removal, are applied in order to provide cleaner and quality assured data. Afterwards, the tokenization phase is implemented through the utilization of multilingual word and sentence embedding and tokenization techniques by applying Byte-level Byte-pair, WordPiece and SentencePiece tokenization through the Hugging Face library. The selection of the right tokenizer is highly based on the Multilingual Classification model that is utilized and evaluated each time, as the different models handle the words and sentences in different ways. It should be noted that, so far, five (5) different BERT-based models have been evaluated for their performance providing initial results. The four (4) first models are the mBERT, XLM-R, DistilmBERT, and mDeBERTa, by utilizing their corresponding through the Hugging Face library, i.e., *"bert-base-multilingual-cased"*[27], "*xlm-roberta-base*"[28], "*distilbert-base-uncased*"[29], and "*mdeberta-v3-base*"[30] respectively. All these four (4) models were initially trained on the Stereoset dataset and then applied to the project's OECD dataset following a transfer learning approach. With regards to the fifth model, it is a different version of the XLM-R model, namely the "*xlm-roberta-large-xnli*"[31], that is trained on a larger corpus (i.e., the XNLI), than the XLM-R. This model was evaluated for its zero-shot classification without prior training in the Stereoset dataset to showcase the applicability of such a model in a more generic and policy-oriented task.

The training and evaluation of these models on the Stereoset showcased that the mBERT model outperformed the others and captures more efficiently the differences and nuances expressed into biased sentences, while the zero-shot utilization of the XLNI did not provide remarkable results and performance. Hence, the mBERT model is further used and applied on the project's OECD dataset that is an unlabelled dataset of policy briefs and documents sourced from governments and policy think tanks worldwide. The fine-tuned model successfully identifies key biases, including those related to gender, race, and socioeconomic status. The classifications are enriched with RAG-generated explanations, offering actionable insights into potential biases within the documents. This robust and holistic approach ensures both accuracy in detection and transparency in interpretation, empowering stakeholders to make informed decisions about the texts under analysis. This approach is widely generalizable and applicable to other scenarios where bias is incorporating into policy texts.

The nature of this unlabelled dataset further highlights the significance of this methodology, as policy-related documents from over 30 countries are analyzed to detect and explain biases, ensuring that insights are culturally sensitive and contextually appropriate. This is particularly crucial for fostering inclusivity and equity in global policy discussions. The identifying patterns of systemic bias, the methodology lays the foundation for developing unbiased policy recommendations that promote equitable decision-making processes.

The RAG mechanism enhances the bias classification process by combining state-of-the-art retrieval and generation capabilities into a unified pipeline. Initially, RAG retrieves contextually relevant information from a curated knowledge base that includes descriptions and examples of various types of biases and has been created based on the Catalogue of Bias[32], ensuring the system has access to authoritative and detailed sources. When a specific bias is identified by the

[27] https://huggingface.co/google-bert/bert-base-multilingual-cased
[28] https://huggingface.co/FacebookAI/xlm-roberta-base
[29] https://huggingface.co/distilbert/distilbert-base-uncased
[30] https://huggingface.co/microsoft/deberta-v3-base
[31] https://huggingface.co/joeddav/xlm-roberta-large-xnli
[32] https://catalogofbias.org/biases/

MBC component, RAG dynamically fetches entries from the knowledge base that are semantically aligned with the flagged text. These entries are then processed by a language generation model, such as GPT-4, to create a clear, human-readable explanation tailored to the context of the identified bias. For instance, the sentence “Its projects include the Levantamento do PretaLab initiative that focuses on Afro-Brazilian women and seeks to raise awareness about algorithmic biases and their potential to reinforce discrimination” is flagged for race and detection bias, RAG retrieves information about stereotypes in leadership contexts and generates a justification explaining how such biases influence underrepresentation in senior roles, as indicated below.

Question: Its projects include the Levantamento do PretaLab initiative that focuses on Afro-Brazilian women and seeks to raise awareness about algorithmic biases and their potential to reinforce discrimination.

```
{
  "sentence": "Its projects include the Levantamento do PretaLab initiative that focuses on Afro-Brazilian women and seeks to raise awareness about algorithmic biases and their potential to reinforce discrimination.",
  "primary_bias_type": "Algorithmic Bias",
  "secondary_bias_types": ["Racial Bias", "Gender Bias"],
  "justification": [
      "  The question is about the use of the term “Algorithmic bias” in the context of the study aligning it with systemic problems caused by algorithms.",
      "It highlights Afro-Brazilian women, indicating a potential overlap with racial and gender biases."
  ]
}
```

This step ensures that the system does not only label text as biased but provides actionable insights that stakeholders can understand and use to make informed decisions. The seamless integration of the retrieval and generation transforms bias classification step from a black-box process into a transparent and interpretable workflow that fosters trust and accountability.

This approach provides significant value in building trust in AI-driven systems. The integration of the MBC robust bias detection model with an explainability layer provides to the stakeholders, including policymakers, researchers, and the public, improved confidence in the AI’s outputs. The transparency offered by RAG enables users to understand the reasoning behind the model’s classifications, fostering accountability and reducing resistance to AI insights. Moreover, the methodology’s adaptability across different languages and cultural contexts ensures that it remains relevant and equitable in global applications. It also demonstrates how advanced AI methodologies can address the challenges of bias detection, explainability, and trustworthiness in sensitive domains such as policy analysis. The provision of accurate classifications and detailed justifications enhances the transparency and accountability of AI systems. Furthermore, it sets the ground for the adoption of trustworthy AI solutions in global governance, contributing to the development of more equitable and inclusive policies. This approach represents a significant step toward ensuring that AI systems are effective and aligned with ethical principles and societal values.

Among the future steps, and until the finalization of T4.3 – "Improve Citizen Engagement and Trust utilizing NLP" on M27, is the incorporation of additional datasets to improve classification performance across more biases and the further fine-tuning of the model for multilingual capabilities to address biases in the non-English documents of the OEDC dataset. Moreover, a frontend application will be implemented and integrated with the Visualization Workbench to offer a tool to the stakeholders for enhancing their understanding on biases and to promote adoption of unbiased AI solutions. The latter is foreseen to be reported in the context of D4.4 – "Policies Visualization Services V2".

4.5.2 Citizens' Engagement

The *Policy-Oriented Analytics and AI Algorithms* is also a citizen-centric component. This means that specific functionalities have been implemented, in order to encourage the citizens to interact with the trained ML models and benefit from the said models. More specifically, there exist two (2) functionalities that can be utilized by the citizens, which have been developed in the context of the 3rd Pilot. The first one is called "Check My Bin" where citizens can be informed of the predicted fill level of a bin of their choice and whether this bin will be emptied in the next garbage collection cycle. It is worth noting that in the context of this functionality, the LLM that was previously mentioned is also being used in order to provide an easier to understand explanation with regards to whether or not the selected bin should be emptied. The second functionality, which is also available in the *Adaptive Analytics Framework*, refers to the implementation of a feedback mechanism, which enables the citizens to provide their feedback with regards to the provided predictions and the way that they are presented to them. By doing so, the citizens are actively participating in the improvement of the ML models that have been trained to improve their everyday lives.

4.5.3 Baseline Technologies

Policy-Oriented Analytics and AI Algorithms is based on specific technologies. First of all, the component has been developed with the use of the Python programming, since there exists a plethora of modules available that are ideal for several ML tasks. Moreover, JavaScript has also been utilized in order for the component to generate the interactive diagrams and maps that were mentioned above. Moreover, the component is also available as Docker Image, thus allowing its flawless integration with the other components of the AI4Gov platform. A complete list of the Python modules that are currently being utilized by the *Policy-Oriented Analytics and AI Algorithms,* can be found below.

- *certifi: provides validation of SSL certificates.*

- *Flask, Flask_Cors, Requests*: utilized for developing APIs.
- *matplotlib, plotly*: used for creating plots, maps, diagrams both static and interactive.
- *numpy*: used for performing mathematical operations.
- *pandas*: used for data manipulation.
- *nltk*[33], *pytorch*[34], *transformers*[35]: used for applying several NLP-related algorithms and techniques.
- *langdetect*[36]: provides detection of the language in which a given text is written to.
- *PyPDF2*[37]: provides the necessary methods for extracting text from .pdf files.
- *scikit_learn, scipy, tensorflow, darts*[38]: provide appropriate tools for the development of ML algorithms.
- *ydata-profiling: provided appropriate tools for performing descriptive analysis on data.*

4.5.4 Source Code - Availability and Key Points

The source code of the *Policy-Oriented Analytics and AI Algorithms* is available on the project's GitLab repository under the GitLab project named "T4.3 - Improve Citizen Engagement and Trust utilizing NLP", as shown in Figure 30.

[33] https://www.nltk.org

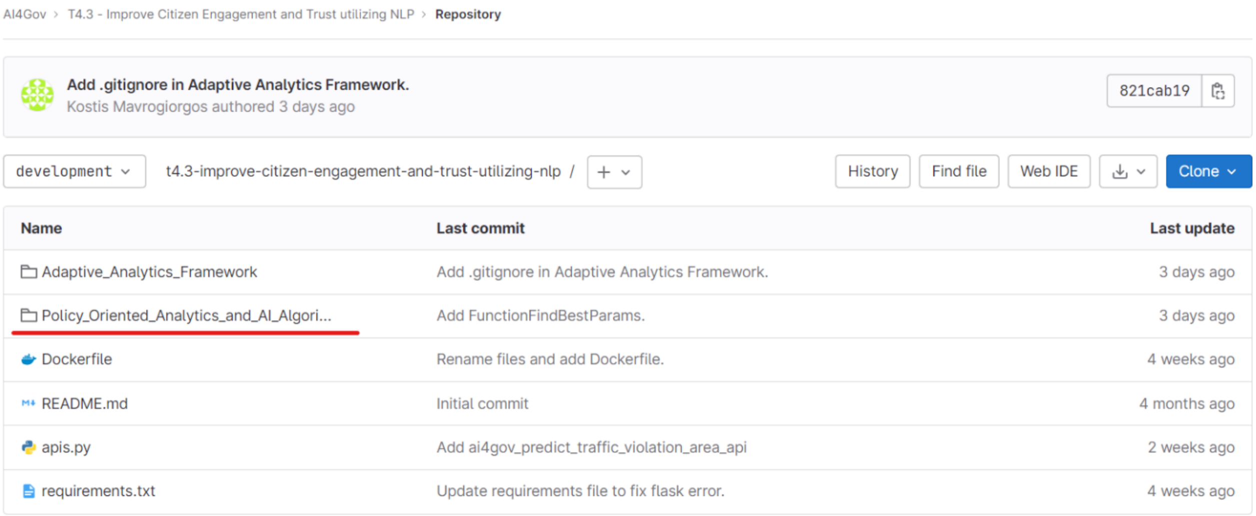
[34] https://pytorch.org

[35] https://huggingface.co/docs/transformers/index

[36] https://pypi.org/project/langdetect/

[37] https://pypi.org/project/PyPDF2/

[38] https://unit8co.github.io/darts/



AI4Gov > T4.3 - Improve Citizen Engagement and Trust utilizing NLP > Repository

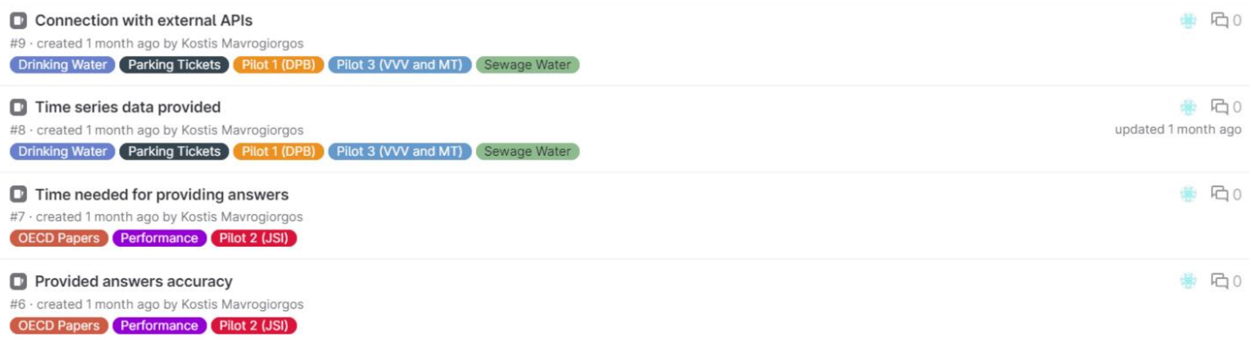
Add .gitignore in Adaptive Analytics Framework.
Kostis Mavrogiorgos authored 3 days ago

821cab19

Name	Last commit	Last update
Adaptive_Analytics_Framework	Add .gitignore in Adaptive Analytics Framework.	3 days ago
Policy_Oriented_Analytics_and_AI_Algori...	Add FunctionFindBestParams.	3 days ago
Dockerfile	Rename files and add Dockerfile.	4 weeks ago
README.md	Initial commit	4 months ago
apis.py	Add ai4gov_predict_traffic_violation_area_api	2 weeks ago
requirements.txt	Update requirements file to fix flask error.	4 weeks ago

Figure 30: Policy-Oriented Analytics and AI Algorithms Source Code on GitLab

It is also worth mentioning that in order to simplify the development process and ensure that all the user requirements are met, corresponding issues have been created in the GitLab repository, as shown in Figure 31.



Connection with external APIs
#9 · created 1 month ago by Kostis Mavrogiorgos
Drinking Water | Parking Tickets | Pilot 1 (DPB) | Pilot 3 (VVV and MT) | Sewage Water
0

Time series data provided
#8 · created 1 month ago by Kostis Mavrogiorgos
Drinking Water | Parking Tickets | Pilot 1 (DPB) | Pilot 3 (VVV and MT) | Sewage Water
0
updated 1 month ago

Time needed for providing answers
#7 · created 1 month ago by Kostis Mavrogiorgos
OECD Papers | Performance | Pilot 2 (JSI)
0

Provided answers accuracy
#6 · created 1 month ago by Kostis Mavrogiorgos
OECD Papers | Performance | Pilot 2 (JSI)
0

Figure 31: Sample of GitLab Issues Created for the Policy-Oriented Analytics and AI Algorithms

4.5.5 User Guide – Installation and Use

Regarding the installation of the *Policy-Oriented Analytics and AI Algorithms,* this is a straightforward process since the component has been containerized (i.e., a corresponding Docker container has been created) and is available through the AI4Gov GitLab's container registry.

When the installation is complete, the component will be available through a specified port and the following APIs will be available for use.

Table 11: Description of the ai4gov_qa_api_api

Endpoint Name	ai4gov_qa_api_api
Endpoint URL	http://[SERVER IP]:PORT/ai4gov_qa_api_api
Description	This endpoint is responsible for performing question answering on the OECD policies papers. As mentioned above, this API will be utilized in the context of the 2nd pilot and the OECD use case.
HTTP Method	POST
Parameters	countries: the list of the countries whose policy papers the component should search in order to provide the answer to the user. question: the question that the user asks
Response	JSON response that contains the extracted answer and the corresponding text from which it was extracted

Table 12: Description of the ai4gov_analytics_llm_api

Endpoint Name	ai4gov_analytics_llm_api
Endpoint URL	http://[SERVER IP]:PORT/ai4gov_analytics_llm_api
Description	This endpoint is used to interact with the LLM of the platform that is used on the context of the 2nd and 3rd pilots.
HTTP Method	POST
Parameters	prompt: the prompt that is provided to the LLM
Response	JSON response containing the LLM's answer

Table 13: Description of the ai4gov_store_feedback_api

Endpoint Name	ai4gov_store_feedback_api
Endpoint URL	http://[SERVER IP]:PORT/ai4gov_store_feedback_api
Description	This endpoint is used in order to provide feedback with regards to a provided answer
HTTP Method	POST
Parameters	user_id: the ID of the user that provided the feedback user_role: the role of the user (e.g., citizen) use_case: the name of the use case to which this feedback refers to related_question: the related question that was asked by the user related_answer: the corresponding answer that was provided by the platform feedback: the user’s feedback
Response	Status code related to the successful submission of the feedback

Table 14: Description of the ai4gov_water_management_api

Endpoint Name	ai4gov_ water_management_api
Endpoint URL	http://[SERVER IP]:PORT/ai4gov_water_management_api
Description	This endpoint is used for utilizing the corresponding ML models that are used for providing the timeseries forecasts in the context of the 1st pilot
HTTP Method	POST
Parameters	use_case: available values are "drinking water - quality" or "sewage water - WWTP energy consumption" entity: "ATALAYA" or "HIGUERA_LA_REAL" or "VALVERDE_DE_LLERENA" if use_case = "drinking water - quality" and "Analizadordered_EDAR_Cheles" or "Analizadordered_EDAR_Oliva_de_la_Frontera" or "Analizadordered_EDAR_Torremayor" if use_case = "sewage water - WWTP energy consumption"

Response	JSON response containing the predicted value, alongside the sufficient reasons plot

Table 15: Description of the ai4gov_waste_management_api

Endpoint Name	**ai4gov_ waste_management_api**
Endpoint URL	http://[SERVER IP]:PORT/ai4gov_waste_management_api
Description	This endpoint is used for predicting the fill level of the bins so that they can be used as input in solving the routing optimization problem and provide insights with regards to citizens' flows. This is used in the context of the 3rd pilot
HTTP Method	POST
Parameters	None
Response	JSON response containing the predicted fill levels

Table 16: Description of the ai4gov_predict_flows_api

Endpoint Name	**ai4gov_predict_flows_api**
Endpoint URL	http://[SERVER IP]:PORT/ai4gov_predict_flows_api
Description	This endpoint is used to split the Municipality of Vari-Voula-Vouliagmeni into cluster based on the citizens' flows
HTTP Method	POST
Parameters	n_areas: number of areas to split the municipality into n_clusters: number of clusters to split the bins based on their predicted fill level
Response	JSON response containing the corresponding visualization

A video demonstrating the abovementioned functionalities can be found here.

5 Conclusions

This deliverable has provided an overview of the work done for T4.1, T 4.2 and T4.3 in the period of months 12 till 24. We have covered the progress on Virtualized Unbiasing Framework, Situation Aware eXplainability and Policy-Oriented AI and NLP algorithms. Demonstrators for the abovementioned functionalities can be found here.

In the context of the Virtualized Unbiasing Framework, the developed application with educational and catalogue part has been demonstrated. Progression of use cases has been explained, particularly focusing on the methodology applied for SDG observatories, with the data incompleteness pipelines.

Concerning Situation Aware eXplainability, creation of a novel scale for pragmatic assessment of the quality of explanations as perceived by users of business processes has been demonstrated.

For Policy-Oriented Analytics and NLP Algorithms the final implementation for all the use cases is presented alongside with additional functionalities that specifically aim to engage the citizens and encourage them to interact with the trained ML models and benefit from the said models.

We have progressed with the development of these technical tasks, and in close collaboration with WP6 waiting the evaluation of co-designed solutions for the pilots wait for potential further improvements.

6 References

Ağbulut, Ü. (2022). Forecasting of transportation-related energy demand and CO2 emissions in Turkey with different machine learning algorithms. Sustainable Production and Consumption, (pp. 141-157).

Alexopoulos, C. L. (2019). How machine learning is changing e-government. 12th international conference on theory and practice of electronic governance. (pp. 354-363).

Antoniou, C. &. (2022). A survey on semantic question answering systems. The Knowledge Engineering Review.

Arenas, M. B. (2022). On computing probabilistic explanations for decision trees. Advances in Neural Information Processing Systems, 35, 28695-28707.

Barceló, P. M. (2020). Model interpretability through the lens of computational complexity. Advances in neural information processing systems 33, 15487-15498.

Berg, V. d. (2022). Artificial Intelligence and the Future of Public Policy. Retrieved from https://ec.europa.eu/jrc/communities/sites/jrccties/files/06_berg.pdf

Bhardwaj, A. D. (2022). Smart IoT and machine learning-based framework for water quality assessment and device component monitoring. Environmental Science and Pollution Research.

Carvalho, D. V. (2019). Machine learning interpretability: A survey on methods and metrics, doi:10.3390/electronics8080832. Electronics 8(8):832.

Cortes, E. G. (2022). A systematic review of question answering systems for non-factoid questions. Journal of Intelligent Information Systems, 1-28.

Darwiche, A. &. (2020). On the reasons behind decisions. ECAI (pp. 712-720). IOS Press.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, 29(6), 141–142.

Elkhawaga, G. (2023). Evaluating Explainable Arti- ficial Intelligence Methods Based on Feature Elimination: A Functionality-Grounded Approach doi:10.3390/electronics12071670. Electronics (Switzerland) 12 (7).

Fahland, D. (2024). How well can large language models explain business processes? Retrieved from https://arxiv.org/abs/2401.12846

Gino Sophia, S. G. (2020). Water management using genetic algorithm-based machine learning. Soft computing.

Hettinga, S. V. (2023). Large scale energy labelling with models: The EU TABULA model versus machine learning with open data.

Lage, I. (2018). Conference on Neural Information Processing Systems (NeurIPS) Workshop on Correcting and Critiquing Trends in Machine Learning. Conference on Neural Information Processing Systems (NeurIPS) Workshop on Correcting and Critiquing Trends in M, (pp. 1-7).

Li, X. L. (2021). Machine learning in internet search query selection for tourism forecasting. Journal of Travel Research, 1213-1231.

Lundberg, S. M.-I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing system 30, (pp. 4765-4774).

Markus, A. (2021). , The role of explainability in creating trust- worthy artificial intelligence for health care: A comprehensive survey of the terminology, design choices, and evaluation strategies, doi:10.1016/j.jbi.2020.103655. Journal of Biomedicine Informatics.

Mavrogiorgos, K. K. (2023). A Question Answering Software for Assessing AI Policies of OECD Countries. 4th European Symposium on Software Engineering, (pp. 31-36).

Mavrogiorgos, K. K. (2024). Bias in Machine Learning: A Literature Review. Applied Sciences.

Mavrogiorgos, K. K. (2024). Mitigating Bias in Time Series Forecasting for Efficient Wastewater Management. 2024 7th International Conference on Informatics and Computational Sciences (ICICoS), (pp. 185-190).

Mavrogiorgou, A. K. (2021). beHEALTHIER: A microservices platform for analyzing and exploiting healthcare data. 2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS) (pp. 283-288). IEEE.

Nadeem, M. B. (2020). StereoSet: Measuring stereotypical bias in pretrained language models. arXiv:2004.09456.

Puh, K. &. (2023). Predicting sentiment and rating of tourist reviews using machine learning. Journal of Hospitality and Tourism Insights, 1188-1204.

Rahman, M. M. (2021). Machine learning on the COVID-19 pandemic, human mobility and air quality: A review. Ieee Access.

Ray, P. P. (2023). ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. Internet of Things and Cyber-Physical Systems.

Ribeiro, M. T. (2016). " Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, (pp. 1135-1144).

Shafiq, M. T. (2020). Data mining and machine learning methods for sustainable smart cities traffic classification: A survey. Sustainable Cities and Society.

Sittar, A. (2024). BAR-Analytics: A Web-based Platform for Analyzing Information Spreading Barriers in News. Submitted manuscript.

Sokol, K. (2020). Explainability fact sheets: A framework for systematic assessment of explainable approaches, doi:10.1145/3351095.3372870. FAT* 2020 - Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, (pp. 56-57).

van Dongen, B. (2017). BPI Challenge 2017. Retrieved from 4TU Research Data: https://data.4tu.nl/articles/dataset/BPI_Challenge_2017/12696884/1

Vilone, G. (2021). Notions of explainability and evaluation approaches for explainable artificial intelligence, Information Fusion 76. doi:10.1016/j.inffus.2021.05.009.

Xie, G. Q. (2021). Forecasting Chinese cruise tourism demand with big data: An optimized machine learning approach. . Tourism Management.

Yeh, C. K. (2019). On the (in) fidelity and sensitivity of explanations. Advances in neural information processing systems 32.

Yoon, W. J. (2022). Sequence tagging for biomedical extractive question answering. In Bioinformatics (pp. 3794-3801).

Zhong, W. H. (2022). Reasoning over hybrid chain for table-and-text open domain question answering. International Joint Conference on Artificial Intelligence (IJCAI), (pp. 4531-4537).

Zhou, J. (2021). Evaluating the quality of machine learning explanations: A survey on methods and metrics doi:10.3390/electronics10050593. Electronics 10(5):593.