URBREATH [101139711]

Systemic Integration of Transformative Technical and Nature-based Solutions to Improve Climate Neutrality of European Cities and Regions and tackle Climate Change: the URBreath Approach

URBREATH

D3.10 URBREATH Tools for AI-based algorithms and Data management and monitoring - V1

Project Reference No	URBREATH – 101139711
Deliverable	D3.10 URBREATH Tools for AI-based algorithms and Data management and monitoring – V1
Work package	WP3: URBREATH data strategy and tools
Type	OTHER
Dissemination Level	PU - Public (fully open)
Date	18/12/2025
Status	Final
Editors	Christina Nichiforov (EXUS), Ettore Etenzi (EXUS)
Contributors	Giovanni Luca d'Acierno (ENG) Francesco Nudo (ENG) Stijn Vranckx (VITO) Thomas Adolphi (VCS) Faezeh Kazemihatami (LAT) Toni Rubio España (FICLIMA) Sakis Dalianis (ATC) Martina Forconi (Deda) Chiara Savoldi (Deda)
Reviewers	Henna-Mari Laurila (KAMK) Faezeh Kazemihatami (LAT) Giovanni Giacco (LAT)

Document description	Functional specifications of the AI-Based algorithms and tools that will be used in the project: The development of the algorithms and tools that will provide the following functionalities: (a) High Speed Big Data Semantic clustering and Mining; (b) Advanced Statistical Machine Learning (ML); (c) High Level Data Fusion and Reasoning; (d) Scheduling/ synchronisation and data assimilation. Additionally, the AI-Based algorithms and tools will be implemented in Front Runner Cities. This deliverable is linked to T3.6.

Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
0.1	17/11/2025	Added Data Acquisition, Aggregation and Harmonization chapter	Thomas Adolphi (VCS)
0.2	20/11/2025	Updated chapter Data Acquisition, Aggregation and Harmonization chapter	Giovanni Luca d'Acierno (ENG) Francesco Nudo (ENG)
0.3	23/11/2025	Added section about Land surface temperature model	Toni Rubio España (FICLIMA)
0.4	24/11/2025	Added sections about Water Infiltration Model and VIE-AI tools	Christina Nichiforov (EXUS)
0.5	27/11/2025	Updated section about Numerical Models for Nature Based Solutions	Stijn Vranckx (VITO)
0.6	28/11/2025	Visualization and Interfaces section	Sakis Dalianis (ATS)
0.7	01/12/2025	Introduction	Ettore Etenzi (EXUS)
0.8	02/12/2025	Conclusions	Ettore Etenzi (EXUS)
0.9	10/12/2025	Internal reviewing process	Henna-Mari Laurila (KAMK) Faezeh Kazemihatami (LAT) Giovanni Giacco (LAT)
1.0	11/12/2025	Comments on the sections related with the city of Leuven	Laura Dens (LEU)
1.1	17/12/2025	Comments addressed, final check	Ettore Etenzi (EXUS)
2.0	18/12/2025	Quality check by coordinator	Marcella Bonanomi (LC)
Final	18/12/2025	Version ready for submission to the EC Portal	LC

Disclaimer

The URBREATH project is co-funded by the European Union under grant agreement ID 101139711. The information and views set out in this document are those of the URBREATH Consortium only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

Executive Summary

Deliverable **D3.10 - URBREATH Tools for AI-Based Algorithms and Data Management and Monitoring -V1** presents the first integrated release of the data, AI, and explainability components developed within **WP3** of the URBREATH project. This initial version consolidates the architectural foundations, technical implementations, and analytical tools that form the backbone of the URBREATH digital ecosystem, supporting the project's mission to enable climate-neutral, resilient, and nature-based urban transformations across European cities.

A significant focus of this deliverable is the establishment of a robust, interoperable, and standards-compliant data acquisition, aggregation, scheduling, synchronization, and harmonization layer. This layer integrates heterogeneous data sources (including IoT observations, open data platforms, geospatial repositories, and environmental sensors) through technologies such as NGSI-LD, the SensorThings API, DCAT-AP, and OGC-compliant services. The result is a scalable and flexible data infrastructure capable of supporting real-time ingestion, semantic harmonization, and cross-platform data sharing.

On top of this infrastructure, the deliverable documents the first suite of **AI-based models and analytical tools** implemented in WP3. These include the Infiltration Prediction Model for assessing the hydrological performance of Nature-Based Solutions (NBS), the ensemble of weather, seasonal, and climate models supporting extreme event anticipation, and the downscaled Land Surface Temperature model for urban heat stress analysis. In parallel, the deliverable introduces the initial framework for integrating numerical NBS simulations and outlines the forthcoming use of agentic AI to improve user interaction with complex modelling workflows.

A key achievement of this first version is the deployment of **VIE-AI**, the project's dedicated explainability and interpretability tool. VIE-AI provides transparent, model-agnostic explanations through SHAP, LIME, and interactive diagnostic features, enabling domain experts, planners, and city stakeholders to understand the behaviour of AI models and build trust in their outcomes.

The deliverable also describes the design of the **Simulator** for scenario exploration, advancements in workflow orchestration, automated publication of geospatial outputs, and the implementation of new data pipelines in Leuven, Cluj-Napoca, and other pilot sites.

Finally, the document outlines the roadmap toward D3.10 -V2 (M48), which will consolidate the models presented here, incorporate new capabilities such as the Water Discharge and Flooding Prediction Model, enhance scalability, and improve real-world validation. Upcoming development cycles will focus on strengthening interoperability, extending model usability, enriching explainability features, and deepening integration with the URBREATH Toolbox and Digital Twin environment.

Overall, D3.10 V1 establishes a strong technical foundation for the AI and data-driven components of URBREATH.

Table of Contents

List of Figures

List of Tables

List of Terms and Abbreviations

Abbreviation	Definition
WP	Work Package
GA	Grant Agreement
AI	Artificial Intelligence
NBS	Nature-Based Solutions
FR	Front Runner

1 Introduction

1.1 Purpose and Scope

Deliverable D3.10 “URBREATH Tools for AI-based Algorithms and Data Management and Monitoring - V1” provides the first consolidated definition and technical specification of the AI-based components, data management mechanisms, and analytical tools developed within Work Package 3. As outlined in the Grant Agreement, URBREATH also aims to integrate advanced digital technologies, particularly AI, Local Digital Twins, and interoperable data ecosystems, to support the development, validation, and deployment of hybrid and NBS in European cities. Within this context, D3.10 describes how the project’s data and AI infrastructure is conceived, designed, and implemented during the Development Phase.

The scope of this deliverable encompasses the complete set of processes and tools that form the WP3 backbone, ranging from the foundational architectural elements to the concrete AI-based functionalities that enable evidence-based analysis and decision support. It includes the definition of the data acquisition, aggregation, scheduling, synchronization, and harmonization layer, which ensures that heterogeneous datasets (from sensors, external IT platforms, geospatial repositories, open data portals, and statistical sources) are collected, transformed, and served through interoperable and standards-based mechanisms. The deliverable also outlines the first set of AI algorithms and computational tools, such as semantic clustering, machine learning models, high-level reasoning mechanisms, and data fusion processes, which together enable advanced urban analytics and operational intelligence across the URBREATH ecosystem.

Because this deliverable is tied to the development trajectory of the entire WP3, it is inherently connected to all its tasks, including T3.1, which establishes the overarching architecture, integration logic, standards compliance framework, and operational requirements that guide all subsequent developments. The outputs of T3.1 provide the structural and methodological foundations on which T3.6 (and thus the contents of D3.10) are built. As a result, this deliverable reflects not only the technical outputs specific to AI and data management but also the architectural decisions, interoperability specifications, and integration pathways defined at the WP3 level.

Moreover, D3.10 serves as the initial version of a progressive and iterative process. It sets the baseline for the full implementation of AI-based tools and data workflows to be further refined, validated, and expanded in alignment with the Digital Twin development in WP4 and the real-world deployment in the Front Runner Cities under WP5. The deliverable therefore defines the current state of development while establishing the scope and direction for subsequent iterations leading to the final version (V2) at M48.

1.2 Approach for Work Package and Relation to other Work Packages and Deliverables

The approach followed in WP3 builds on a coherent and iterative development workflow that integrates architectural design, data management, and AI-based analytical capabilities into a unified technical framework. The work carried out for D3.10 represents the outcome of this coordinated process, grounded in the architectural foundations and interoperability principles established in T3.1 and progressively expanded through the remaining WP3 tasks. The development of the AI algorithms, harmonization workflows, and monitoring mechanisms described in this deliverable is therefore not an isolated effort, but the result of the cumulative and interdependent work of the entire work package.

WP3 operates as the core provider of the project's data and computational infrastructure, and its outputs naturally interface with several other work packages. The functional and governance requirements defined in WP2 guide the specification of data models, interoperability rules, and ethical principles embedded in the AI tools. WP3 supplies the harmonized data streams, semantic context, and analytical components that underpin the Digital Twin, simulation models, and city dashboards developed in WP4, enabling their advanced reasoning and visualisation functions. Furthermore, the tools and data workflows specified in D3.10 are progressively validated through interactions with the Front Runner Cities in WP5, where real-world datasets and operational needs drive refinement and testing. Finally, the methodological clarity and technical robustness established in this deliverable contributes to the replication and scalability activities addressed later in WP6.

In this way, D3.10 reflects both the internal cohesion of WP3 and its essential role within the broader URBREATH architecture, ensuring that AI-driven functionalities, data management strategies, and cross-cutting interoperability mechanisms are fully aligned with the project's objectives and integrated across its technical, demonstrative, and validation activities.

1.3 Methodology and Structure of the Deliverable

The deliverable is structured to present the work in a coherent and accessible manner. Following this introductory chapter, it provides a detailed description of the data acquisition, harmonization, and orchestration mechanisms underpinning the AI tools; outlines the functional specifications of the algorithms developed in this phase; and describes the explainability, visualization, and interoperability components supporting their use. Subsequent sections focus on integration efforts, preliminary validation, and the roadmap toward the final release, ensuring that the document captures both the current achievements and the planned evolution of the WP3 framework. D3.10 is structured into the following main chapters:

- Chapter 1: Introduction; this chapter presents purpose, scope, approach, and methodology (the current section).
- Chapter 2: Data Acquisition, Aggregation, Scheduling, Synchronization, and Harmonization; this chapter describes the data architecture, connectors, metadata management, orchestration, and harmonization mechanisms forming the backbone of AI tools
- Chapter 3: Functional Specifications of AI-Based Tools; this chapter details the AI models, algorithms, and analytical components developed in T3.6
- Chapter 4. Explainability, Visualization, and Interoperability; this chapter defines mechanisms supporting transparency, accountability, and integration with WP4's Digital Twin and dashboards
- Chapter 5: Integration and Implementation; this chapter provides early results, technical integration notes, and cross-WP interfaces

- Chapter 6: Roadmap to D3.10 V2 (M48); this chapter outlines the development, validation, and integration plan for the final deliverable version
- Chapter 7: Conclusions and Recommendations
- Chapter 8 and 9: References and Annexes; these chapters support technical specifications and documentation.

2 Data Acquisition, Aggregation, Scheduling, Synchronization, and Harmonization

The data management and monitoring layer defined in Task T3.6 provides the foundation for acquiring, aggregating, and harmonizing heterogeneous datasets from multiple sources, including satellite and Earth Observation data, open governmental repositories, crowdsourced and sensor-based observations, industrial data streams, and statistical datasets. These sources differ significantly in structure, semantics, and updating frequency, which creates challenges related to interoperability, data silos, and fragmented stewardship. The objective of this layer is to establish a decentralized, standards-based infrastructure that ensures transparent and secure data sharing while enabling advanced AI-driven functionalities.

Data acquisition begins with the integration of distributed sources through standardized protocols and open interfaces. The architecture relies on tools such as the FROST Server, which implements the OGC SensorThings API to support real-time ingestion of IoT and sensor data streams. This component enables spatio-temporal indexing of observations and synchronization of dynamic datasets, ensuring that high-frequency measurements from environmental sensors and crowd-sourced platforms are captured and made available for downstream processing. Complementing this, GeoNetwork operates as a geospatial metadata catalogue, supporting ISO 19115 and ISO 19139 standards for describing and discovering spatial resources. It provides advanced search and retrieval capabilities and facilitates metadata harvesting from distributed repositories, ensuring that spatial datasets are properly indexed and accessible for harmonization.

Table 1: Standards and Their Functions

Standard	Function
NGSI-LD	Linked data interoperability and semantic enrichment.
DCAT-AP	Dataset cataloguing and metadata harmonization.
SensorML	Standardization of sensor metadata and observation models.
OGC Services (WMS, WFS, WCS)	Interoperable geospatial data exchange and visualization.

Aggregation involves consolidating datasets from multiple domains into a unified representation. This process requires metadata harmonization, schema mapping, and temporal and spatial alignment to overcome inconsistencies in resolution, granularity, and semantics. The IDRA Catalogue plays a central role in this stage by acting as a federated discovery and indexing service. It enables cross-domain dataset harvesting and semantic clustering based on linked data principles, exposing harmonized datasets through NGSI-LD endpoints. GeoServer complements this functionality by publishing harmonized geospatial layers using OGC-compliant services such as WMS, WFS, and WCS. This ensures interoperability with analytical pipelines and visualization platforms, allowing harmonized datasets to be consumed by diverse applications.

Table 2: Tools and Their Roles

Tool	Role
IDRA Catalogue	Federated discovery and indexing of datasets; semantic clustering and NGSI-LD exposure.
FROST Server	Real-time ingestion and synchronization of IoT and sensor data streams using OGC SensorThings API.
GeoNetwork	Geospatial metadata catalogue supporting ISO standards; metadata harvesting and advanced search.
GeoServer	Publishing harmonized geospatial layers via OGC-compliant services (WMS, WFS, WCS).

Harmonization techniques are essential to achieve semantic and structural consistency across heterogeneous datasets. These techniques include ontology-based mapping using NGSI-LD for linked data representation, SensorThings API standardization for sensor metadata, and AI-driven semantic clustering for grouping similar datasets based on content and context. Data fusion mechanisms combine multi-source datasets into coherent knowledge graphs, enabling reasoning and advanced analytics. This harmonized data layer serves as the foundation for high-speed big data semantic clustering and mining, advanced statistical machine learning, and high-level data fusion and reasoning. It also supports scheduling and synchronization workflows for real-time data assimilation, which are critical for predictive modeling and decision support in Front Runner (FR) Cities.

Security and privacy considerations are integrated throughout the process. Data is anonymized to protect individual privacy, and encryption combined with role-based access control ensures confidentiality and integrity. Compliance with European initiatives such as DS4SSCC and DSSC guarantees alignment with federated data space principles, promoting trustworthy and transparent data sharing. The adoption of open standards and decentralized architecture ensures interoperability and scalability, enabling seamless interaction with heterogeneous systems and applications.

2.1 Architectural Overview

Data Acquisition, Aggregation and Harmonization Layer operate as the backbone of the URBREATH architecture, connecting the diverse functional domains that govern the platform's data lifecycle — from collection to publication and utilization.

Its architecture ensures that all incoming data are effectively acquired, managed, and made discoverable through standardized processes.

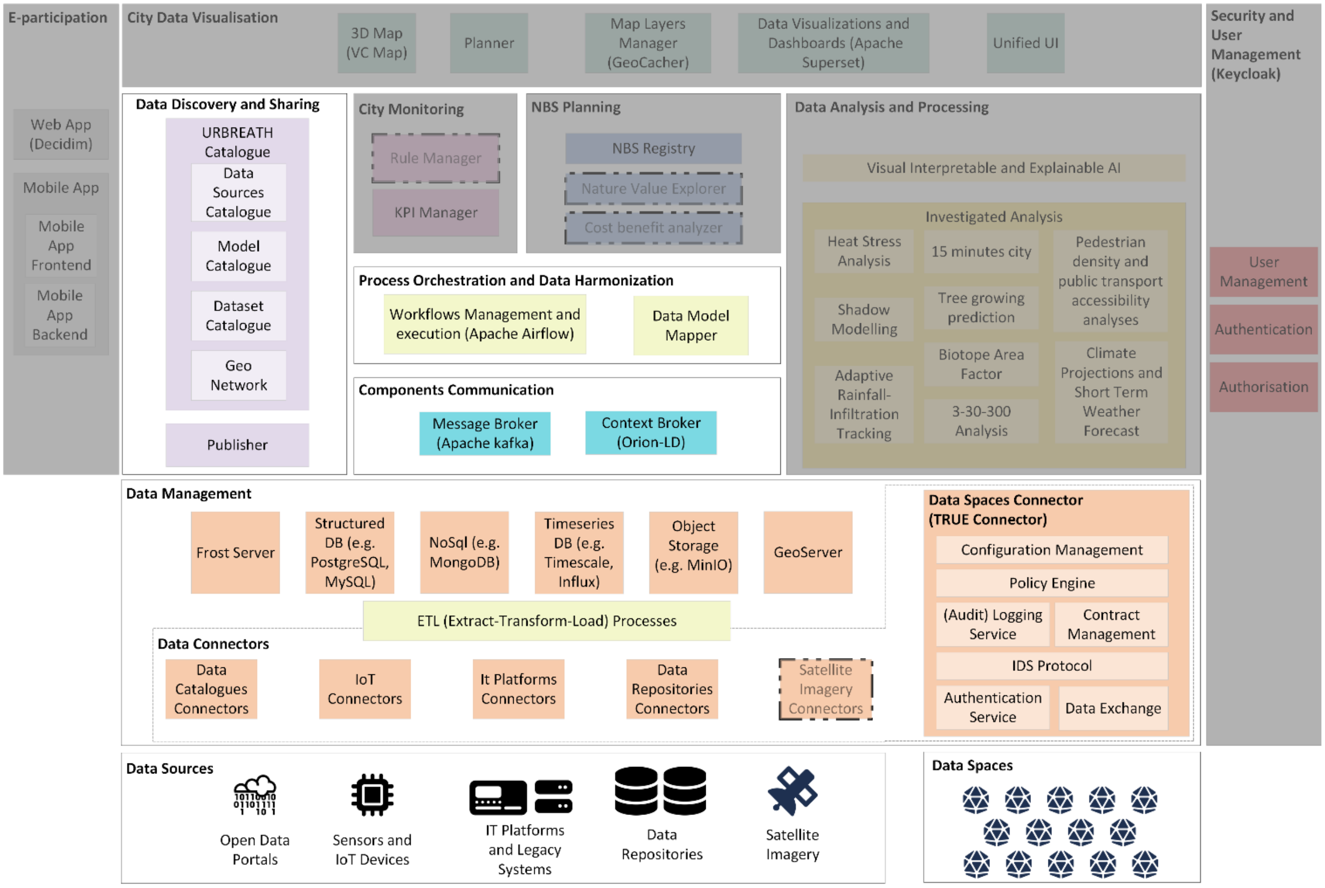


Figure 1: Data Acquisition, Aggregation and Harmonization Layer in URBREATH architecture

The layer is organized around a coherent set of interconnected subsystems:

- Data Management: allows integration of data from heterogeneous systems, harmonizes and aggregates it through tailored ETL processes and expose it via standard formats and APIs.
- Components Communication: manages interoperability and information exchange between modules, ensuring standardized message routing, and synchronization between services, allowing them to communicate consistently.
- Process Orchestration and Data Harmonization: using orchestration tools (Apache Airflow) and semantic alignment components (Data Model Mapper) to coordinate automated workflows for data ingestion and integration, ensuring heterogeneous data sources to be harmonized into data models processable and reusable across the platform.
- Data Discovery and Sharing: enables access to available datasets through the URBREATH Catalogue and allows users and services to explore datasets, metadata, and services.

The cooperation of these subsystems enables URBREATH to operate as an integrated environment where acquisition, harmonization, management, and sharing are aligned and ensuring not only data consistency and interoperability but also supporting scalability and integration with external data spaces and urban ecosystems.

Notably, the architecture now places “Visual, Interpretable, and Explainable AI” outside the Investigated Analysis layer, reinforcing transparency as an independent and cross-cutting capability, rather than a downstream step, to support in the evaluation of explainability of AI-based algorithms.

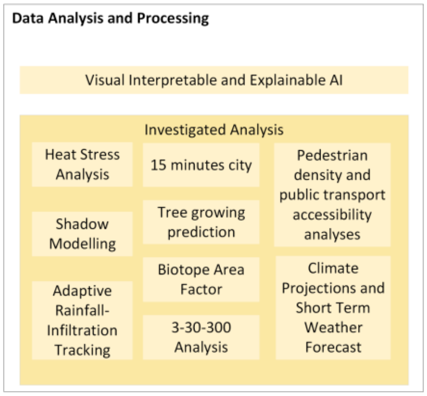

Data Analysis and Processing
Visual Interpretable and Explainable AI
Investigated Analysis
Heat Stress Analysis
15 minutes city
Pedestrian density and public transport accessibility analyses
Shadow Modelling
Tree growing prediction
Biotope Area Factor
Climate Projections and Short Term Weather Forecast
Adaptive Rainfall-Infiltration Tracking
3-30-300 Analysis

Figure 2: Updated Data Analysis and Processing layer

2.2 Data Sources

URBREATH collects and integrates information from multiple heterogeneous data origins, ensuring broad coverage of urban, environmental and social domains.

Main ones can be summarized as follows:

- Open Data Portals: publicly available datasets from municipal, regional, and national datasets.
- Sensors and IoT Devices: real-time data streams capturing information such as air quality, mobility, noise or temperature.
- IT Platforms and Legacy Systems: existing administrative and operational platforms integrated through connectors and APIs (including external digital systems such as Waze, providing dynamic traffic and mobility information, OpenStreetMap, offering geospatial reference data on the urban environments and GIS systems).
- Data Repositories: institutional and project-specific databases hosting curated datasets.

On a separate note, Satellite Imagery Connectors are not yet supported in the current implementation. They will be explored as a potential data source, with integration planned for future releases once the corresponding ingestion and preprocessing workflows for remote sensing data are defined.

2.3 Data Management and Connectors

Data Management is a critical design consideration due to the diverse and voluminous nature of processed data, which ranges from sensor time series to spatial and semantic metadata. The architecture utilizes several specialized storage components:

- **FROST STA Server**: This is an open-source implementation of the OGC SensorThings API (OGC STA). It provides a high-performance, resource-efficient platform for storing and accessing IoT sensor data via a standardized RESTful interface. The data model is compliant with OGC standards, and the server is recognized by the European Commission as a best practice for sharing measurement data under INSPIRE guidelines.
- **Structured, Timeseries, and NoSQL Databases (DB):** Three main categories of databases are used to handle different data types.
    - **Structured Database (RDBMS):** Used for tabular data with well-defined schemas, ensuring reliability and consistency of critical metadata. PostgreSQL (with PostGIS spatial extensions) serves as the unified data backend for dashboards, supporting complex queries and geospatial operations. MySQL is used for legacy modules or lightweight structured data storage.
    - **Timeseries Database:** Optimized for managing high-frequency, timestamped measurements generated by sensors and meteorological instruments. InfluxDB is used for fast data ingestion, time-based queries, and efficient long-term storage via retention policies.
    - **NoSQL Database:** Handles semi-structured and unstructured data, offering flexible schemas and horizontal scalability. MongoDB is chosen for storing items such as model outputs, configuration files, and user-generated content.
- **Object Storage (MinIO):** Provides the ability to handle large amounts of data files, including structured data formats (e.g., large CSV datasets) and unstructured data (e.g., multimedia files like images and videos). It stores data in immutable objects, ensuring stable access to raw and processed data for analysis and simulation activities. It offers S3-Compatible APIs, versioning, automated lifecycle policies, and rich security/access control features.
- **GeoServer:** An open-source server application that enables users to share, process, and edit geospatial data. GeoServer supports a wide range of data formats and standards, utilizing open standards from the Open Geospatial Consortium (OGC) to publish data from any major spatial data source.

2.3.1 Data Connectors

Data connectors are the connection points responsible for centralized discovery and access to heterogeneous and scattered data sources.

- **Data Catalogues Connectors:** Allow the internal Dataset Catalogue to "federate" with external catalogue repositories. These connectors interact with APIs exposed by various catalogue software systems, such as CKAN and SOCRATA, to collect metadata and uniform its representation according to DCAT-AP.

- **IoT Connectors:** Ensure transparent communication and real-time sensor data collection from a wide range of IoT devices. They support multiple communication protocols (e.g., MQTT, LoRaWAN, HTTP) and translate protocol-specific payloads into standardized data formats like NGSI-LD or the OGC SensorThings API, making data consumable by the Toolbox regardless of the device origin.
- **IT Platform Connectors:** A set of functional blocks providing guidelines for secure, standard, and interoperable communication with external IT systems (e.g., municipal databases, legacy systems). This mechanism acts as a bridge to retrieve and deliver data without disrupting existing infrastructure. Key functional blocks include Authentication and Authorization, API provision (RESTful), and Semantic Adaptation, which translates exchanged data formats into the semantic models used by the URBREATH knowledge framework.
- **Data Repositories Connectors**: Used to access data from multiple sources efficiently for integrated urban analytics. This capability is primarily provided by Presto DB, an open-source, distributed SQL query engine that enables fast federated querying across disparate systems (e.g., PostgreSQL, MongoDB, Amazon S3) using standard SQL, without the need to move or duplicate the data.
- **Data Spaces Connector:** Crucial for safe, uniform, and policy-driven data sharing within a trusted zone or Data Space. It adheres to the rules of the International Data Spaces (IDS) plan, ensuring data control and usage checks. It can act as both a data giver and a data receiver and uses Policy-Aware Interaction based on predefined contracts.

2.3.2 GeoNetwork Connector and Flanders Implementation

As part of its Data Catalogue Connectors, URBREATH adopts a special integration with GeoNetwork, the open-source platform widely used for publishing and managing geospatial metadata. It harvests metadata records coming from these external sources, transforms them into URBREATH's internal metadata profile based on DCAT-AP, and ensures consistency within the platform. Rather than duplicating the original datasets, the connector maintains dynamic links to the source systems so that users can discover and access external resources through URBREATH without compromising data provenance.

This federation mechanism supports distributed data governance with enhanced discoverability and interoperability, particularly within cross-border or multi-agency urban contexts. In such a context, URBREATH is also able to extend its catalogue with high-value spatial data, remaining under the control of the original publisher, following FAIR principles, and avoiding redundancy.

The GeoNetwork Connector, developed within the URBREATH Catalogue (based on Idra; see D4.7 URBREATH NBS ICT integrated solution, Section 4.2.1), enables URBREATH to federate external catalogues that expose metadata through an OGC-compliant CSW endpoint (Catalog Service for the Web https://www.ogc.org/standards/cat/), compliant with ISO 19139 schema and INSPIRE directive.

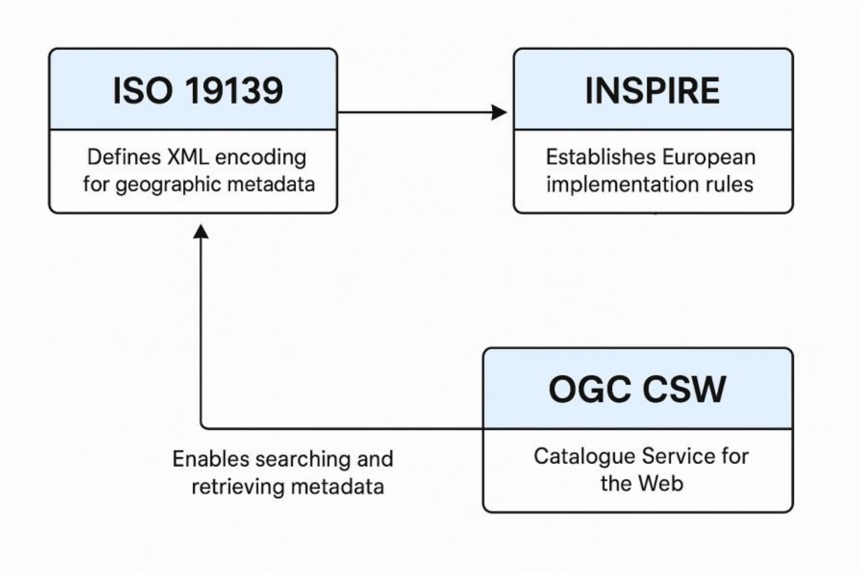

ISO 19139
Defines XML encoding for geographic metadata
INSPIRE
Establishes European implementation rules
OGC CSW
Catalogue Service for the Web
Enables searching and retrieving metadata

Figure 3: Dependencies between OGC CSW, ISO19139 and INSPIRE

The GeoNetwork connector has been developed taking into account the official catalogue of geospatial data managed by the Flanders region, which provides a rich source of spatial and environmental datasets (https://www.dov.vlaanderen.be/geonetwork/srv/dut/catalog.search#/home).
Further details about the mapping on DCAT-AP on annex A (page 68).

2.4 Components Communication

To ensure effective share of information through the integrated system, Components Communication Layer focuses on enabling data exchange among different tools.
This system is built upon two components: the Context Broker handles the management and dissemination of contextual information, the Message Broker focuses on efficient event streaming and processing, both leveraging a publish-subscribe mechanism in different ways.
The Context Broker manages life cycle of context information; it leverages a *publish-subscribe* mechanism to distribute real-time updates to subscribed users or components. Users can define subscriptions to specific data entities or sets and are automatically notified whenever updates occur or when new information becomes available. Based on Orion LD it provides an implementation of the NGSI-LD APIs, managing contextual information in a linked data environment.
The Message Broker provides event streaming and real-time data processing; it uses a *publish-subscribe* communication pattern too, but rather than contextual entities, applications subscribe to *topics*. Implemented in Apache Kafka, that offers an open-source platform supporting scalable event streaming pipelines, ideal for real-time data processing and analytics.

2.5 Process Orchestration and Data Harmonization

Process Orchestration and Data Harmonization macro area's main goal is to provide mechanisms that enable the definition, configuration, execution and monitoring of interconnected processes, ensuring coordinated operation and overall process control. Through this macro area, URBREATH can orchestrate data flows and interactions across different tools, guaranteeing the smooth integration of functionalities and efficient use of computational resources.

This layer includes two main components: Firstly, Workflow Management and Execution provides the necessary capabilities for defining, managing, and executing workflow processes to connect different tools, implemented through Apache Airflow (an open-source platform that enables users to define workflows as Directed Acyclic Graphs, DAGs, using Python). It provides scheduling and monitoring features, making it well suited for orchestrating data processing pipelines.

On the other hand, the layer has been expanded to encompass data harmonization functions through the Data Model Mapper, recognizing the increasing importance of semantic and structural alignment in data workflows.

2.5.1 Data Model Mapper

Developed in Node.js, The Data Model Mapper component replaces the previously considered Data Mashup Editor and focuses on data processing and transformation operations, provides users with an interface to design and implement data transformation processes, supporting the integration, harmonization, and mapping of data.

This component enables the conversion of various file types, such as CSV, JSON, and GeoJSON, into NGSI-LD compliant with Data Models. The input files may include rows, JSON objects, or GeoJSON features, each representing an element to be transformed into an NGSI entity in accordance with the selected Data Model. The tool operates by parsing the input file and converting it into a uniform stream of objects, which are then translated into intermediate representations. These intermediate objects are subsequently mapped to NGSI-LD entities using a JSON mapping specification tailored to the target Data Model.

Once mapped, the resulting entities are validated against the corresponding JSON schema using the AJV JSON Schema Validator. After validation, entities can either be forwarded directly to the configured Orion Context Broker or stored locally as files. The DMM can be executed either as a command-line application or as a REST server, in which case a dedicated graphical user interface is also available.

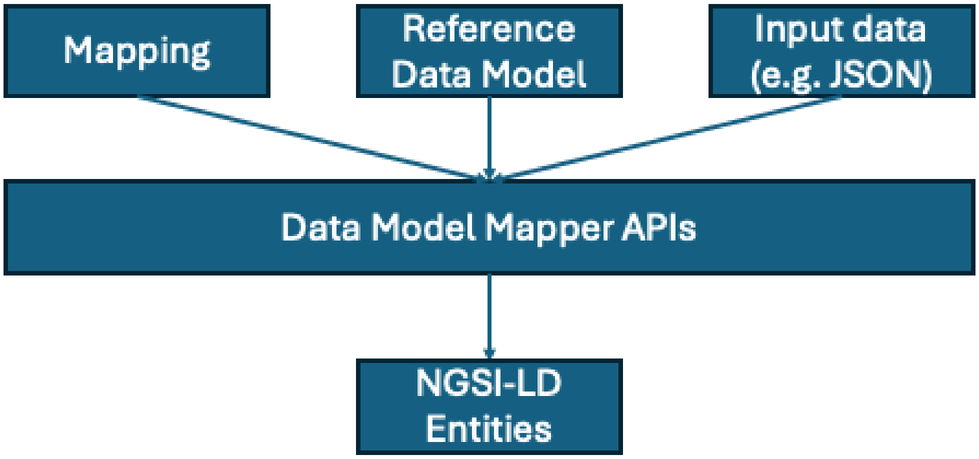


Figure 4: Data Model Mapper Interactions

2.6 Data Discovery and Sharing

Data Discovery and Sharing (formerly *Data Access and Sharing*) provide a unified point for accessing, searching, and discovering the information managed within the URBREATH Toolbox. It is used for both data gathered from external sources (integrated through Data Management macro area) and data generated internally by the Toolbox (analysis results and processed datasets) ensuring users to easily identify, access, and reuse available information, promoting interoperability and knowledge sharing across users.

This area is composed of two main elements: the URBREATH Catalogue and the Publisher. The URBREATH Catalogue provides the functionalities for accessing, searching, and discovering information within the Toolbox. It comprises two main subcomponents, the Model Catalogue and the Dataset Catalogue. Following the decision to fully merge the former NBS Catalogue into the NBS Registry, creating a single authoritative reference point for all NBS datasets and metadata. The Model Catalogue enables exploration and access of available models, algorithms, and computational resources within the Toolbox, supporting collaboration and reuse across departments or projects. The Dataset Catalogue manages datasets coming both from external and from internal analyses, operating with the Data Catalogue Connectors of the Data Management area.

The Publisher component complements the URBREATH Catalogue by enabling users to contribute new information to the Toolbox ecosystem. It supports the publication of diverse content such as new datasets, analytical results, and 3D city models that can feed the Digital Twin visualizations within the City Data Visualization macro area. By facilitating structured data publication and discovery, the Data Discovery and Sharing macro area strengthens the overall data governance of the URBREATH Toolbox, ensuring that information is findable, accessible, interoperable, and reusable across all participating systems and stakeholders.

2.7 Workflow for Acquisition and Management of Data from External Systems

URBREATH's data integration relies on a well-defined workflow for acquiring, transforming, and managing data coming from external systems, ensuring compatibility and interoperability with standardized data models.

The general workflow includes the following key stages:

- Source Analysis: Understanding the external system's API structure, data semantics, and update frequency.
- Data Mapping and Modelling: Selecting or defining the appropriate Smart Data Model compatible with the NGSI-LD standard of the context broker.
- ETL: Implementing the Extract, Transform, and Load logic within automated pipelines.
- Context Management: Injecting the transformed entities into the Orion Context Broker, which manages their lifecycle and contextual relationships.
- Integration and Storage: Depending on the data type and use case, entities can be routed to other systems such as the FROST Server for SensorThings API entities or directly consumed by other URBREATH components.

The following subchapters illustrate this process for two specific external sources: Waze and OpenStreetMap.

2.7.1 Acquisition of Traffic Data from Waze

An example of data flow within the URBREATH Toolbox is the acquisition of street and traffic data from the Waze API, its ingestion into the Context Broker, and, through that, into the FROST Server. The process involves a sequence of automated and interoperable steps:

- **Data acquisition**: the process starts with a detailed analysis of the Waze data feed to understand its structure, content, and update frequency. The initial testing was performed using data from the city of Cluj-Napoca, which already had an active upload system that enabled the provision of real-time traffic updates.
- **Data modelling**: A standard Smart Data Model compatible with Orion-LD is chosen to represent this information within the URBREATH ecosystem (PointOfInterest to describe spotted data about hazards and accidents, https://github.com/smart-data-models/dataModel.PointOfInterest/tree/master/PointOfInterest and TrafficFLowObserved for anomalies in traffic flows, https://github.com/smart-data-models/dataModel.Transportation/tree/master/TrafficFlowObserved).
- **Mapping and transformation**: Using the Data Model Mapper, raw source data is transformed into NGSI-LD-compliant entities according to the defined mapping rules (Annex 9.2).
- **ETL implementation**: The extraction and transformation pipeline is implemented as a Python DAG script.
- **Context injection**: Once converted into NGSI-LD entities, the data is published to the Orion Context Broker, which manages updates and ensures contextual integrity.

- **Downstream integration**: Since Orion and FROST Server manage different types of entities, a subscription mechanism in Orion triggers an endpoint to receive relevant updates. This endpoint translates the NGSI-LD data into SensorThings API-compliant entities and forwards them to the FROST Server. Additional connections to other analytical or storage systems can be configured as needed.

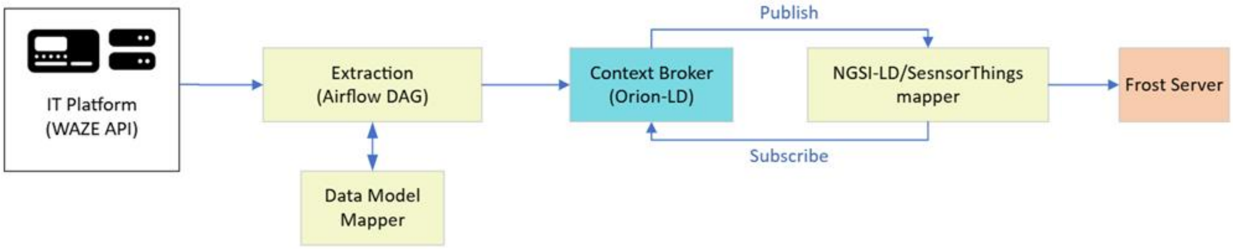


Figure 5: Waze data acquisition workflow

2.7.2 Acquisition of Geospatial Data from OpenStreetMap

In the case of OpenStreetMap, the workflow follows the same conceptual structure but with a simplified integration pathway, as no FROST Server is involved.

The data of interest is mainly related to urban infrastructure elements and is retrieved and transformed only once or on demand, rather than continuously:

- **Data extraction**: A dedicated script executed once retrieves OpenStreetMap data through available APIs or local extracts in formats such as GeoJSON or shapefile. The collection focused on information about all cities participating in URBREATH, covering public fountains, hospitals, platforms, parks, and waterway drains.
- **Data mapping**: The extracted data is analysed, and a suitable Smart Data Model (PointOfInterest) is selected to represent its attributes and geometry in NGSI-LD format.
- **Transformation**: The script performs the conversion from OpenStreetMap structures to NGSI-LD entities, ensuring alignment with URBREATH semantic models.
- **Context injection**: The resulting entities are sent directly to the Orion Context Broker, which stores and manages their contextual information.
- **Data access**: Applications such as the GeoCacher module interact directly with the Context Broker to retrieve the required geospatial information without intermediate storage layers or continuous synchronization processes.

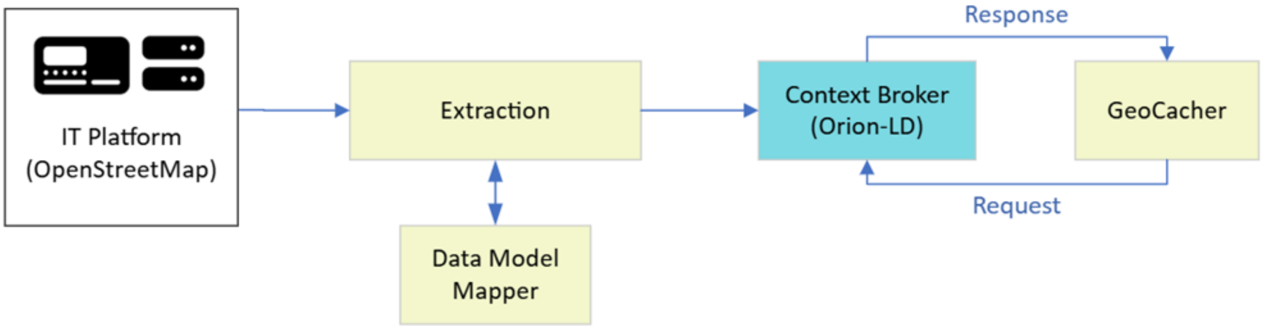


Figure 6: OpenStreetMap data acquisition workflow

This workflow demonstrates a more static integration pattern, suitable for base geospatial datasets that do not require frequent updates. It highlights the URBREATH Toolbox's flexibility in accommodating both dynamic real-time feeds, as in the Waze case, and static reference data, as in the OpenStreetMap case, within a coherent data management framework.

2.7.3 Automated Mobility Data Pipeline for Leuven (Telraam Network)

A specific data pipeline has been implemented to interface with the Telraam sensor network for monitoring traffic dynamics and mobility patterns in Leuven. This workflow is orchestrated via Apache Airflow, ensuring a scheduled, reliable, and secure extraction of traffic data on a monthly basis. The implementation follows a two-stage logic: *raw data ingestion* followed by *analytical processing*.

The pipeline is scheduled to execute automatically on the first day of every month. To maintain security standards within the URBREATH infrastructure, the workflow utilizes Airflow Variables to manage sensitive credentials (API keys and MinIO access tokens) and configuration parameters, such as the list of monitored segment IDs). This decouples configuration from the codebase, facilitating scalability across different sensors without code modification.

The first stage of the ingestion pipeline connects to the Telraam API to retrieve hourly traffic counts, the system iterates through the defined sensor segments, requesting data for the previous month. Instead of processing data in-memory immediately, the raw JSON responses are converted to CSV format and stored directly in the project's centralized MinIO object storage.

Data is organized hierarchically under Leuven/Mobility/KPIs/<timestamp>/raw_data/. This preserves the original dataset in an immutable state, allowing for future re-processing or auditing if required. The second stage retrieves the raw files from MinIO to perform data cleaning and aggregation using Pandas library.

The core analytical task involves calculating the distribution of transport modes, the algorithm computes the percentage shares for four distinct categories: pedestrians, cyclists, cars, and heavy vehicles. The service then generates aggregated metrics, summing traffic counts by date to provide a clear overview of daily trends.

Lastly, the processed results are saved back to MinIO in a dedicated folder (processed_data), providing structured CSV files (e.g., total_shares.csv, summed_data.csv) that are ready to be consumed by visualization dashboards or further spatial analysis tools.

2.7.4 Acquisition of Environmental Data into FROST Server (Cluj-Napoca)

Weather, air quality and noise levels data from Cluj-Napoca are available on the Romanian Open Data portal (https://data.gov.ro/dataset/calitate_aer). Eight monitoring stations used in the pilot site are considered, providing measurements for eleven parameters, including temperature, humidity, atmospheric pressure, PM10, PM2.5, PM1, VOC, noise levels, CO_2, O_3, and CH_2O.
Historical measurements are provided as CSV files, while real-time observations, updated every thirty minutes, are accessible through APIs (e.g. https://data.e-primariaclujnapoca.ro/calitate_aer/?id_senzor=82000496).

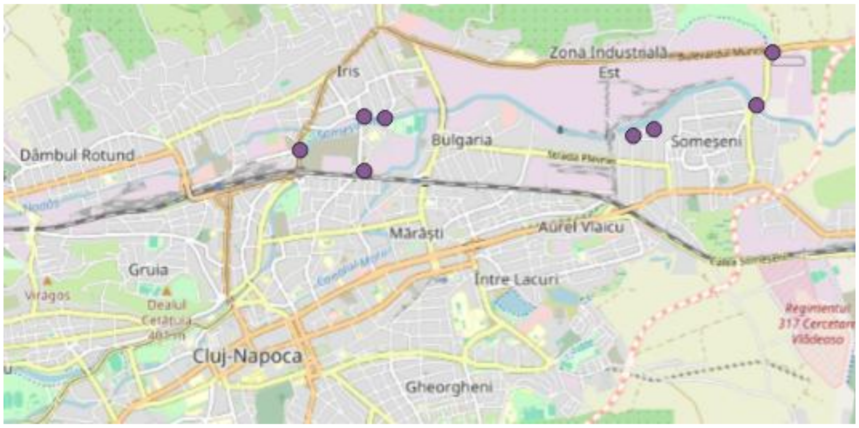


Figure 7: Maps of monitoring stations

A one-off ingestion procedure loads the historical CSV datasets into the URBREATH FROST Server. A dedicated pipeline manages the ingestion of real-time observations.
The integration process involves these steps:

- **Data Extraction**: Historical observations are provided as CSV files, while real-time observations are read periodically from the portal through an automated script.
- **Data Mapping**: The extracted data is processed to create SensorThings entities for each station, including Things, Locations, Features of Interest, Observed Properties, Sensors, Datastreams, and Observations. Only one uRAD sensor with metadata (https://www.uradmonitor.com/products/) has been created and a total of 88 Datastreams (8 Things × 11 parameters). Custom properties (id_sensor, pilot = "Cluj-Napoca") are added to the entities to ensure consistent traceability.
- **Transformation**: For historical datasets, a Python script converts the CSV files into JSON files, which are then uploaded to the FROST Server via API requests. *phenomenonTime* is derived from the *time* field (Unix timestamp converted to Zulu time), while *resultTime* is derived from *momentul_citirii* (local reading time converted to Zulu time).

- **Context Injection**: *Historical Observations are uploaded to the FROST Server using POSTMAN requests. For real-time data, the script (under finalisation) reads the measurements via APIs and sends them to the FROST Server via ita APIs.*
- **Data Access**: Analytical modules and dashboards access both historical and real-time datasets for environmental assessment, time-series analysis, and integration with other URBREATH components.

This workflow integrates Cluj-Napoca environmental data into the URBREATH infrastructure, enabling use of historical and real-time datasets.

2.8 Technical Implementation of Geospatial Data Ingestion and Publication Services

A dedicated pipeline has been developed to automate the ingestion and publication of geospatial datasets (e.g. produced as results of an analysis). The solution leverages GeoServer as the core map server, enhanced by two custom microservices designed to bridge the gap between the project's object storage (MinIO) and the OGC-compliant dissemination services (in this specific case WMS). The implementation relies on a containerized environment orchestrated via Docker, ensuring reproducibility and scalability.

2.8.1 GeoServer Synchronization Service (geoserver-sync)

The first component, identified as geoserver-sync, ensures the continuous alignment between the centralized MinIO object storage and the local file system accessible by the GeoServer instance.

This microservice operates as a background daemon that periodically scans specific buckets within the MinIO infrastructure. It compares the remote state of geospatial assets (timestamps and file sizes) with the local volume mounted to the GeoServer container.

Developed in Python, the service utilizes the official MinIO SDK to handle secure data transfers. A key feature of the implementation is the intelligent handling of multi-file geospatial formats; specifically, it groups and synchronizes Shapefile components (.shp, .shx, .dbf, .prj, etc.) as coherent units, while also managing single-file formats such as GeoTIFFs, GeoJSONs, and SLD styles.

Its primary role is to ensure that the data volume (/geoserver_data_volume) remains an exact mirror of the cloud storage, eliminating the need for manual file transfers to the server environment.

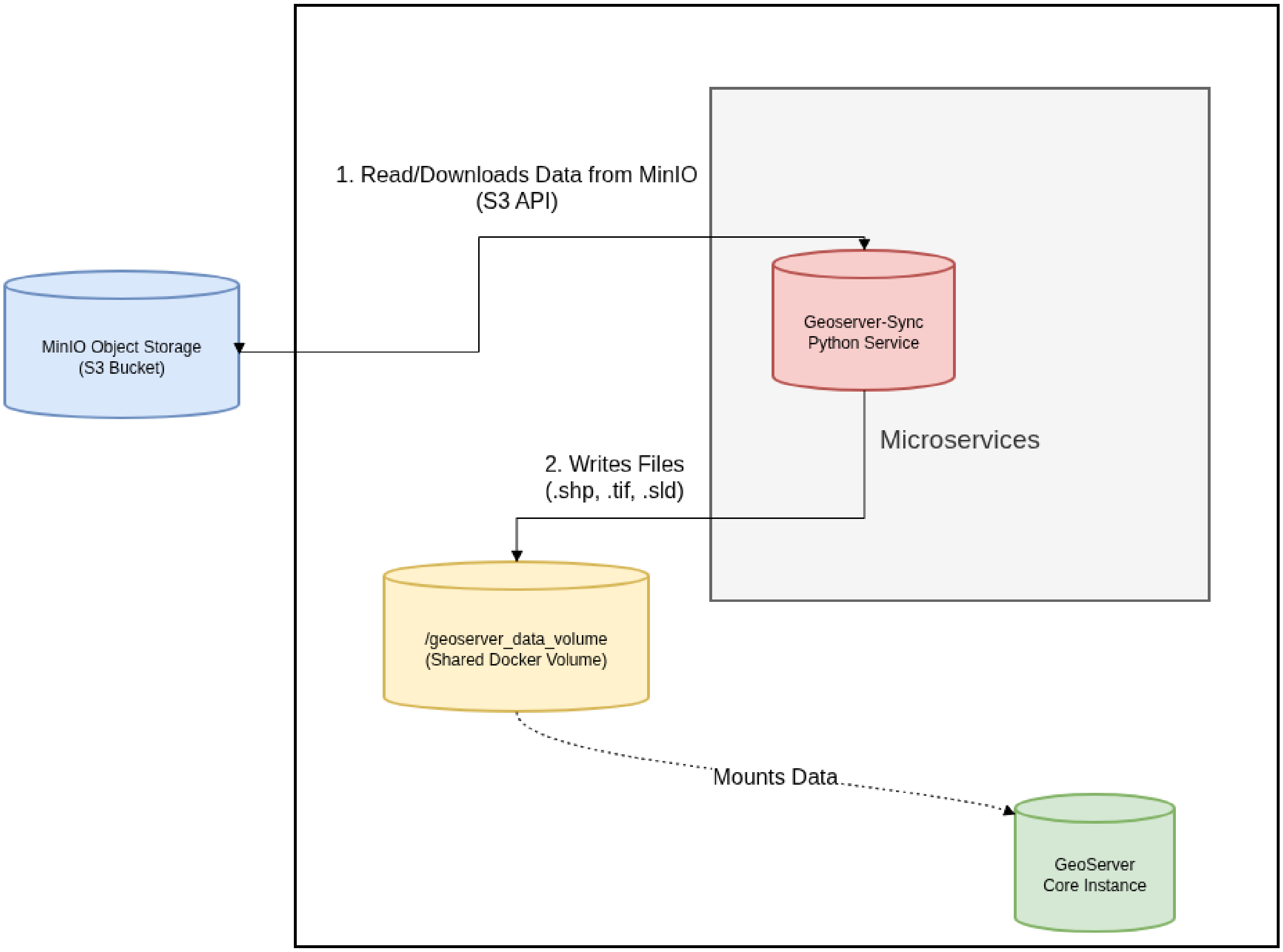


Figure 8: GeoServer Synchronization Service (geoserver-sync)

2.8.2 Automated Publication Service (geoserver-publisher)

The second component, geoserver-publisher, creates an automated interface for registering data layers within GeoServer without requiring manual interaction with the GUI.

This service monitors the storage for specific configuration triggers (files named _publish.json). Upon detection, it parses the configuration to identify the target workspace, data store parameters, and styling requirements.

The service dynamically checks for the existence of the target workspace (e.g., specific to a pilot city like Tallinn or Madrid) and creates it via the GeoServer REST API if it is missing.

It supports the automatic publication of both Vector DataStores (Shapefiles) and CoverageStores (Raster/GeoTIFFs).

A distinctive feature is the automated association of cartographic styles. The service can upload Styled Layer Descriptors (SLD), update existing styles if an override is requested, and link them immediately to the published layers.

Once a layer is successfully published, the configuration file is renamed (to _published.json) to prevent redundant processing, ensuring an idempotent workflow.

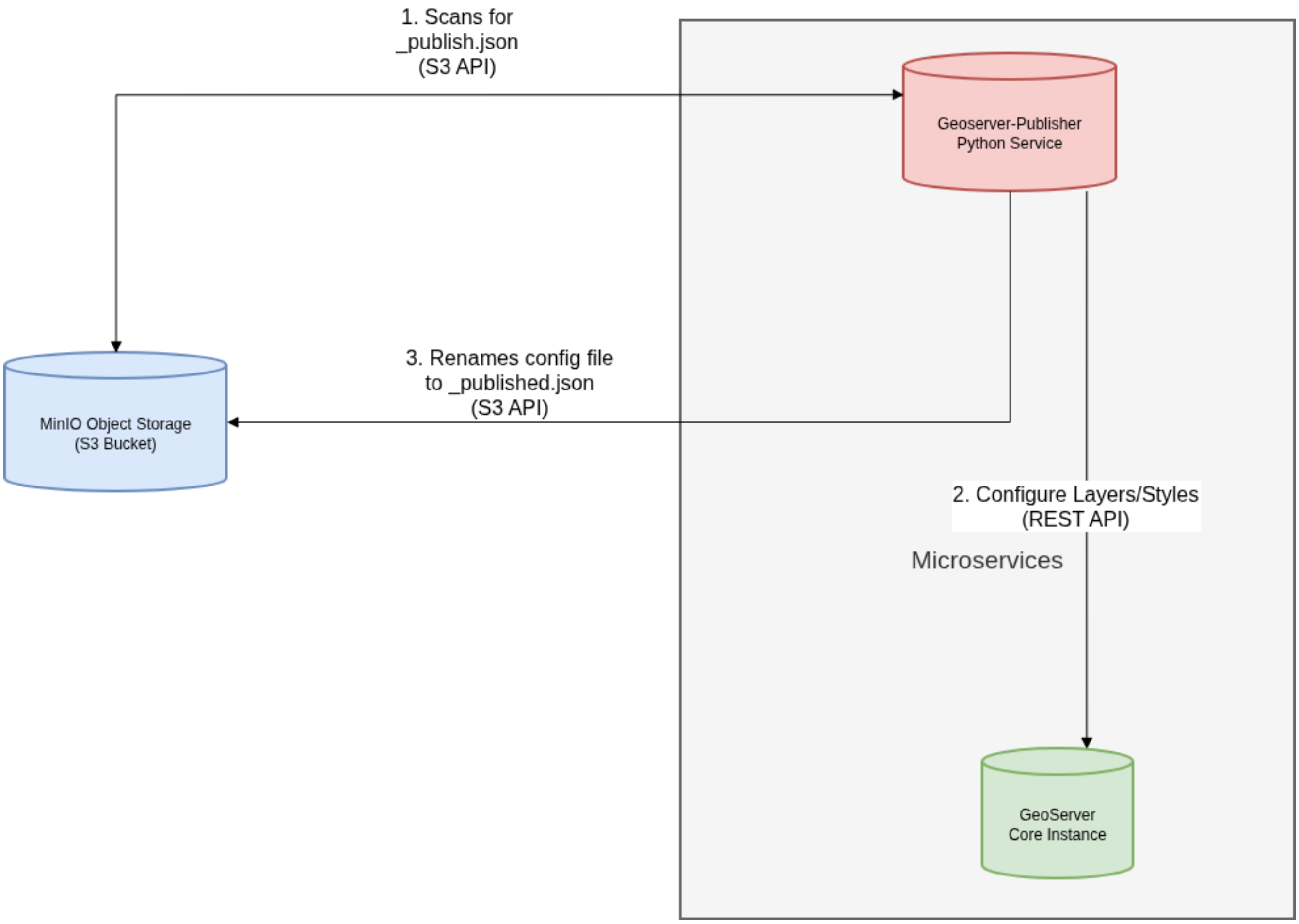

1. Scans for _publish.json (S3 API)
Geoserver-Publisher Python Service
3. Renames config file to _published.json (S3 API)
MinIO Object Storage (S3 Bucket)
2. Configure Layers/Styles (REST API)
Microservices
GeoServer Core Instance

Figure 9: Automated Publication Service (geoserver-publisher)

2.8.3 Integration and Deployment

The technical deployment is defined through a Docker Compose orchestration that couples the standard geoserver and postgis services with the custom Python microservices. A shared volume strategy is employed: the geoserver-sync service writes data to a volume that is mounted read-only by the geoserver container. The geoserver-publisher service then instructs the GeoServer instance (via its REST API) to load data from this shared local path. This architecture decouples data transport from data serving, enhancing the robustness of the URBREATH spatial data infrastructure.

3 Functional Specifications of AI-Based Tools

3.1 AI-Based Tools

The AI-based tools developed within WP3 constitute the analytical core of the URBREATH digital ecosystem, transforming heterogeneous environmental, climatic, and geospatial data into actionable insights for urban planning and Nature-Based Solutions deployment. Building on the harmonized data infrastructure established in earlier chapters, these tools combine advanced machine learning models, statistical forecasting techniques, and spatial analytics to support decision-making across the project's Living Labs. This section presents the first set of AI models implemented in D3.10 -V1 (including infiltration prediction, weather and seasonal forecasting, climate projections, and land surface temperature estimation) together with their methodological foundations, technical specifications, and integration within the broader URBREATH platform.

3.1.1 Infiltration Prediction Model

The infiltration prediction model developed within URBREATH represents a hybrid machine learning approach that currently combines real soil data from the Leuven region with synthetic weather scenarios to predict soil infiltration rates under various environmental conditions. This model serves as the foundation for evaluating Nature-Based Solutions (NBS) interventions aimed at improving urban stormwater management and flood resilience.

The model predicts infiltration rates (mm/hr) based on soil properties, meteorological conditions, and temporal factors, enabling stakeholders to assess the potential effectiveness of different NBS implementations across the urban landscape

3.1.1.1 Data Sources and Processing

3.1.1.1.1 Soil Data Acquisition

Currently, the model utilizes real soil data from the DOV (Databank Ondergrond Vlaanderen) database, comprising comprehensive soil measurements from the Leuven study area. The dataset includes multiple soil property sources: grondmonsters (soil samples), boring (drilling records), CPT measurements (cone penetration tests), lithology data, and classification records.

A complex data processing pipeline consolidates these heterogeneous sources through weighted coordinate assignment and spatial interpolation. Direct boring URL matching links soil samples to geographic locations, while weighted spatial distribution assigns coordinates to remaining records based on proximity to known sample locations. This multi-source integration process achieved coordinate assignment for the complete dataset with spatial coverage spanning the Leuven metropolitan area.

3.1.1.1.2 Soil Property Extraction

The data processing workflow extracts both measured and derived soil properties. Measured properties include organic content (humusgehalte), bulk density (volumemassa), calcium carbonate content (kalkgehalte), and Atterberg limits (plastic and liquid limits). Each property undergoes range validation to remove outliers - for instance, organic content is constrained to 0-50%, while bulk density must fall between 0.8-2.2 g/cm³.

Derived soil texture parameters employ multiple complementary methods. Clay content is estimated from Atterberg limits using established pedotransfer relationships, with additional texture information extracted from Dutch/Flemish soil classification descriptions. The final texture components (clay, sand, silt) are normalized to sum to 100%, ensuring physical consistency.

Moreover, other advanced soil structural properties are calculated from basic measurements. Soil organic carbon (SOC) derives from organic content using the standard 0.58 conversion factor. Porosity is computed from bulk density and particle density, accounting for organic matter effects on particle density. Mean weight diameter (MWD) and water-stable aggregates (WSA) are estimated using empirical relationships incorporating organic content, texture, and bulk density.

3.1.1.2 Model Architecture and Training

3.1.1.2.1 Training Data Generation

A hybrid approach generates training data by combining real soil measurements with synthetic weather scenarios. For each of the real soil sample locations, the system generates multiple weather scenarios representing Belgian climatic conditions. Temperature exhibits seasonal sinusoidal variation with base values around 10°C and peak summer temperatures reaching 18°C, modified by random daily variation. Precipitation follows exponential distribution with seasonal probability modulation, capturing typical Belgian rainfall patterns. Humidity varies seasonally between 40-95%, while antecedent precipitation (7-day cumulative) is sampled from a normal distribution.

The model adjusts infiltration rates based on weather conditions. Cold temperatures reduce infiltration because frozen soil (below 0°C) blocks water entry, allowing only 10% of normal infiltration, while cold soil (0-5°C) permits 50% infiltration. Wet soil from previous rainfall also reduces infiltration capacity—the model uses an exponential relationship where saturated soil infiltrates at only 20% of dry soil capacity. Seasonal changes affect infiltration through vegetation growth and soil conditions throughout the year. Heavy rainfall creates a surface crust that blocks pores and reduces infiltration rates.
This hybrid approach generated ~8000 training samples combining real spatial variability in soil properties with realistic temporal variability in weather conditions.

3.1.1.2.2 Ensemble Machine Learning Approach

The model is an ensemble of two complementary machine learning algorithms to maximize prediction accuracy and robustness. Random Forest Regressor captures complex nonlinear relationships between features without requiring feature scaling, making it suitable for the mixed-type input data (soil properties, weather variables, spatial coordinates). The final model configuration uses 200 trees with maximum depth of 15, minimum samples split of 5, and minimum samples per leaf of 2.

Gradient Boosting Regressor provides complementary predictive power through sequential error correction. Operating on standardized features, the gradient boosting model uses 200 estimators with learning rate 0.1 and maximum depth 8. The standardization preprocessing (zero mean, unit variance) ensures optimal performance for this algorithm.

The final ensemble prediction combines Random Forest and Gradient Boosting outputs through simple averaging, leveraging the strengths of both approaches while reducing individual model biases. Prediction uncertainty is quantified as half the absolute difference between the two model predictions, providing users with confidence intervals.

3.1.1.2.3 Input Features

The model uses 17 input features organized into three categories:

- Soil Properties (11 features): organic content (%), bulk density (g/cm³), clay content (%), sand content (%), silt content (%), soil organic carbon (%), porosity (%), mean weight diameter (mm), water-stable aggregates (%), and spatial coordinates x and y (Lambert 72 projection).
- Weather Conditions (4 features): temperature (°C), precipitation (mm/day), relative humidity (%), and 7-day antecedent precipitation (mm).
- Temporal Factors (2 features): season (1-4 representing winter through fall) and day of year (1-365)

3.1.1.2.4 Model Performance

The ensemble model achieved strong predictive performance on held-out test data. The coefficient of determination (R^2) is 0.95, indicating that the model explains over 95% of variance in infiltration rates. Root Mean Squared Error (RMSE) is 1.3 mm/hr, while mean absolute error (MAE) is approximately 0.9 mm/hr. These metrics demonstrate reliable prediction across the full range of infiltration rates observed in the Leuven study area.

Model validation uses a hold-out testing approach with an 80/20 train-test split. The test set (20% of data) evaluates model performance on unseen samples to assess generalization capability.

Other quality checks include predicted vs. actual scatter plots and residual plots. The predicted vs. actual plot shows strong agreement with the 1:1 reference line ($R^2 = 0.945$). The residuals plot displays errors centered around zero without systematic patterns across the prediction range, indicating the model captures relevant relationships in the data.

Feature importance analysis identifies which input variables drive predictions. This analysis shows weather conditions (particularly 7-day antecedent precipitation and temperature) dominate model behavior, accounting for approximately 83% of predictive importance.

3.1.1.2.5 Feature Importance Analysis

Feature importance analysis reveals that weather conditions dominate infiltration prediction, accounting for approximately 83% of total feature importance. The 7-day antecedent precipitation emerges as the single most influential feature (~40% importance), followed by temperature (~35%). Temporal factors (day of year) contribute approximately 12% of importance, capturing seasonal patterns in soil conditions. Soil texture properties (sand, clay, silt) collectively account for about 9% of importance, while spatial coordinates contribute only 3%. This distribution indicates that weather conditions and their temporal patterns are the primary drivers of infiltration variability in the model, with soil properties playing a secondary but still meaningful role.

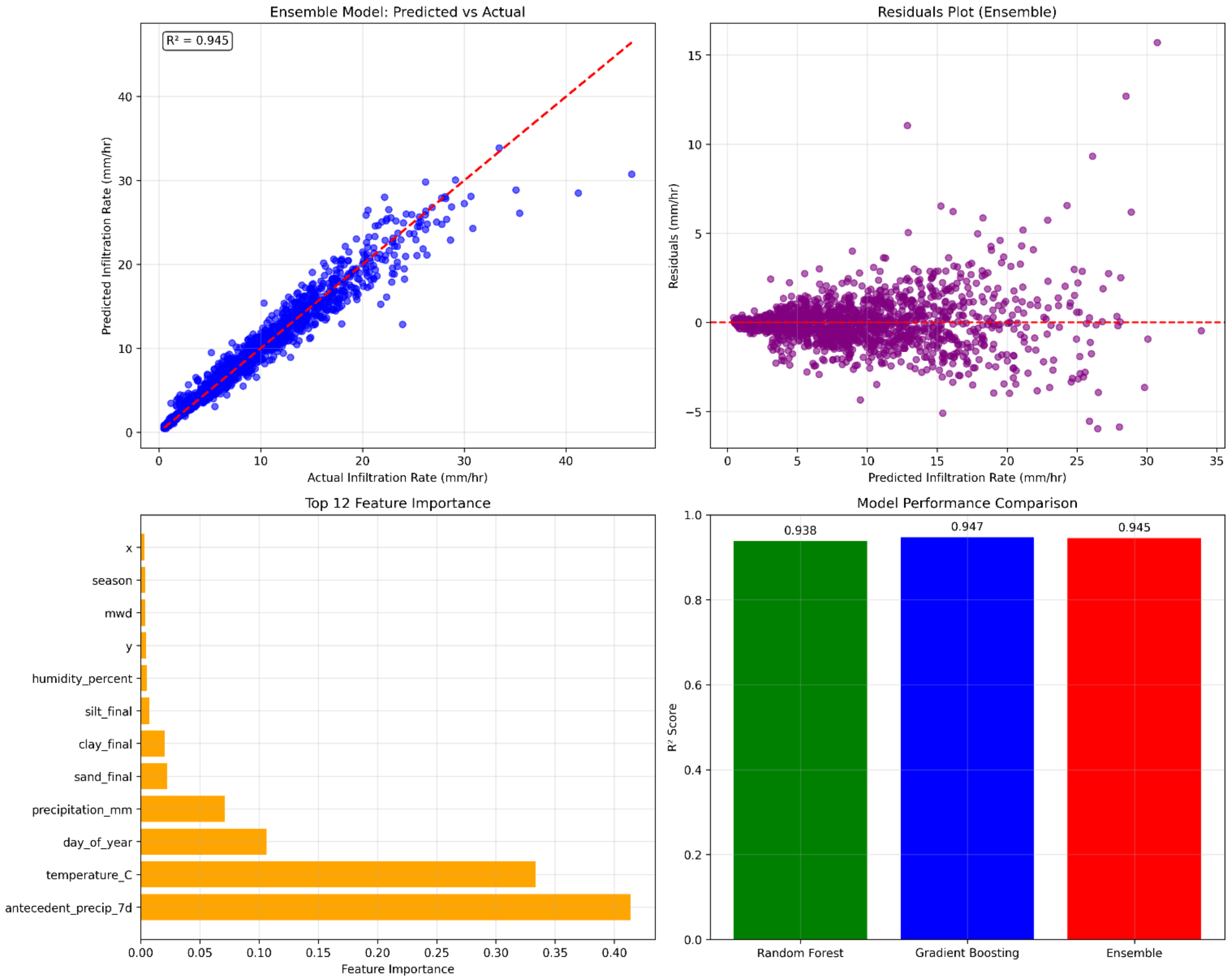


Figure 10: Model Validation

3.1.1.3 Nature-Based Solutions Integration

3.1.1.3.1 NBS Scenario Framework

The infiltration model integrates with a comprehensive NBS evaluation framework that simulates soil modifications induced by various green infrastructure interventions. Eight NBS types are currently implemented: rain gardens, bioswales, permeable pavement, extensive green roofs, constructed wetlands, urban forests, and infiltration trenches, plus existing conditions as baseline.

Each NBS scenario specifies soil property modifications relative to baseline conditions. Modifications can be absolute overrides (e.g., rain gardens specify organic content = 8%), multiplicative factors (e.g., bioswales increase organic content by 1.8×), or additive adjustments (e.g., urban forests reduce bulk density by 0.15 g/cm^3). Direct infiltration multipliers (1.4× to 4.0×) account for enhanced hydraulic conductivity beyond what soil properties alone would predict.

Some NBS types include additional features in the model beyond soil modifications. Permeable pavement has a surface storage parameter (25mm). Green roofs include an evapotranspiration factor (30%). Infiltration trenches have a storage capacity parameter (200mm). These parameters adjust the final infiltration calculations for each NBS type.

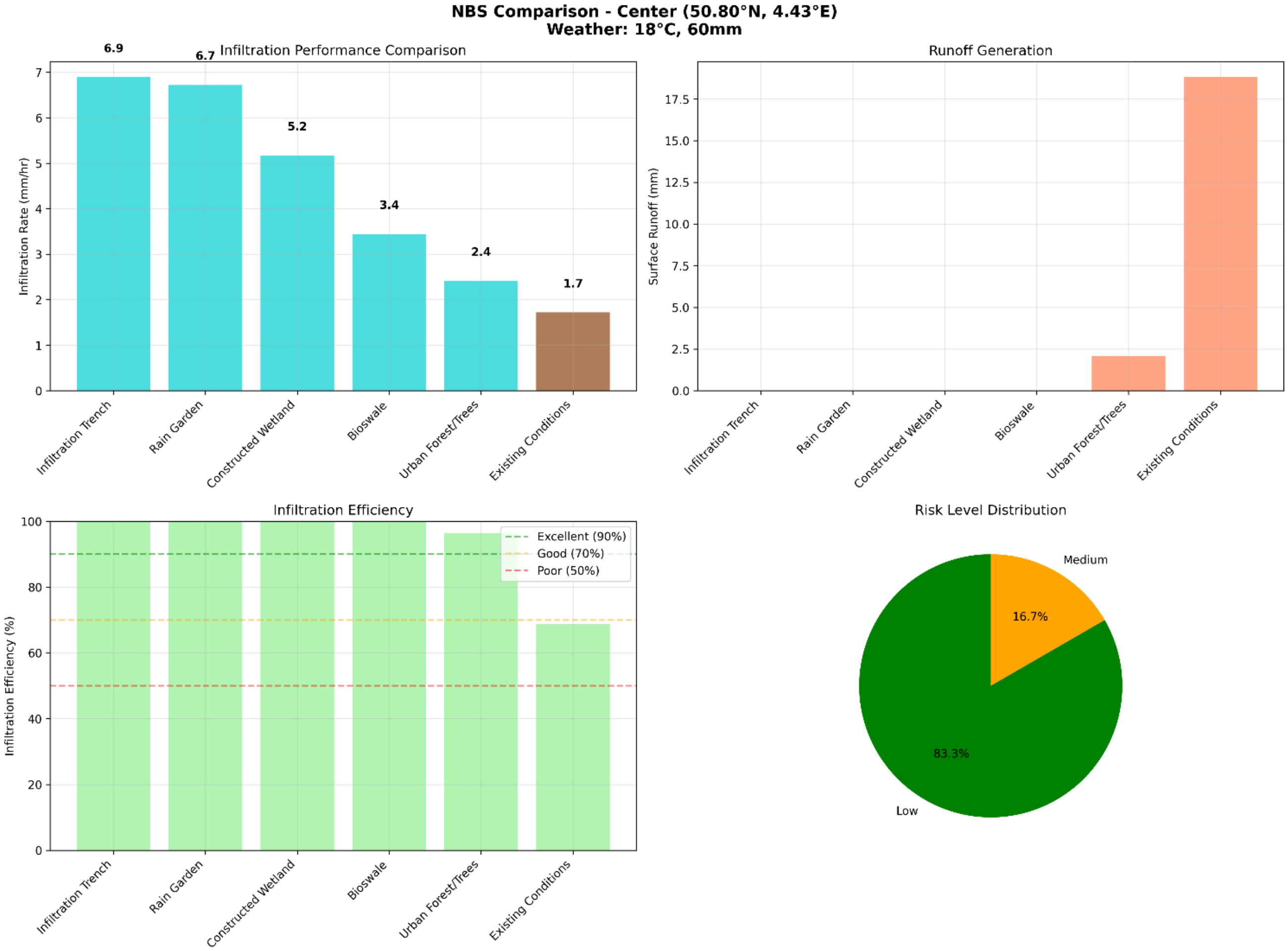


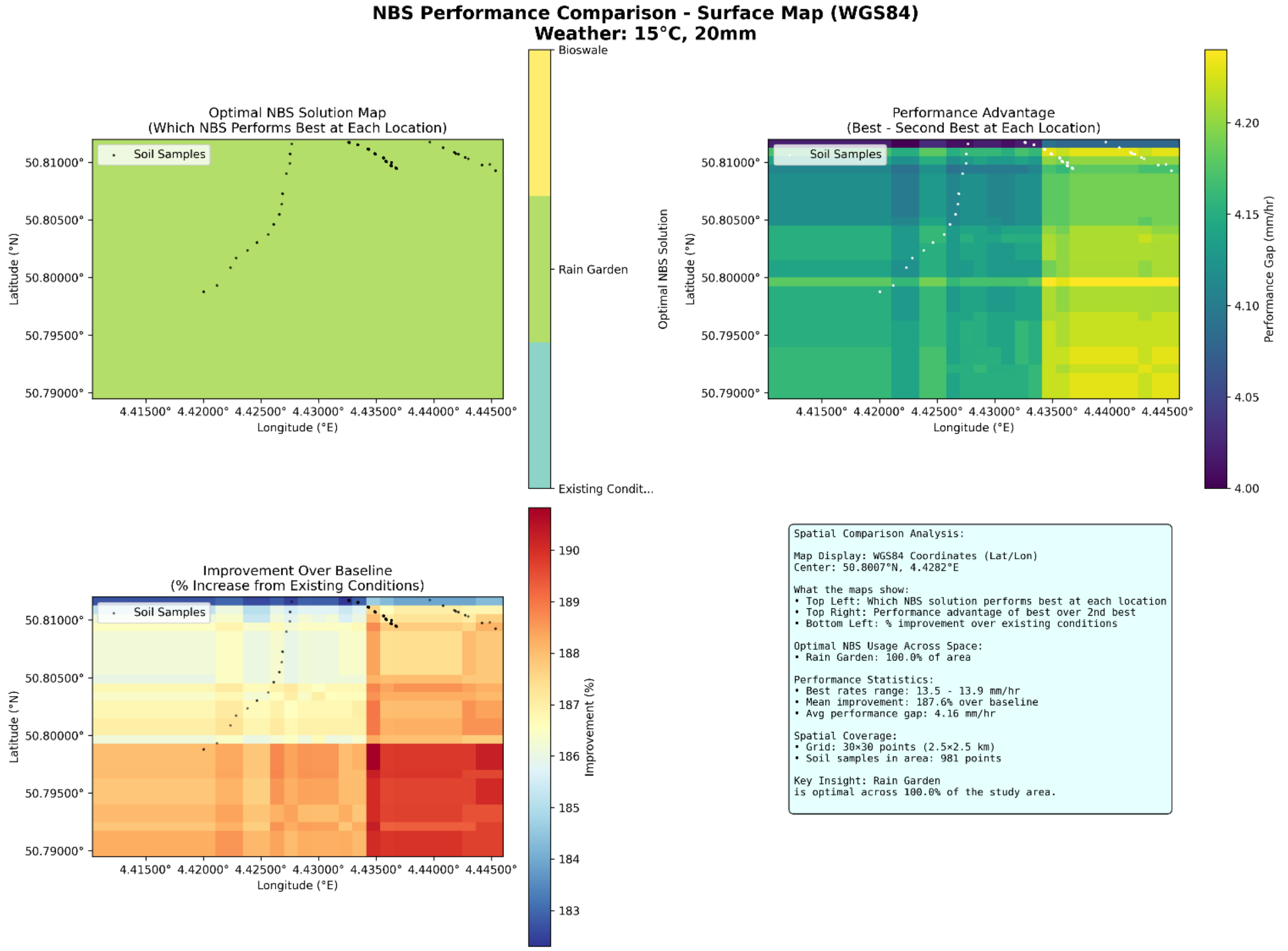
Figure 11: Example of infiltration rate comparison across NBS scenarios for a representative location. The plots show: (top left) infiltration rate performance across six NBS types, (top right) surface runoff generation, (bottom left) infiltration efficiency with quality thresholds, and (bottom right) flood risk classification distribution

3.1.1.3.2 Prediction Workflow for NBS Assessment

When predicting infiltration for an NBS scenario, the system first retrieves baseline soil properties at the target location through k-nearest-neighbor (KNN) spatial interpolation (k=5) with inverse distance weighting. The specified NBS modifications are then applied to these baseline properties, followed by recalculation of derived parameters (porosity, MWD, WSA) using the same empirical relationships employed during training.

The modified soil properties combine user-specified weather conditions and temporal information to form a complete feature vector. The ensemble model predicts baseline infiltration rate, which is then scaled by the NBS-specific infiltration multiplier. Additional NBS features (storage capacity, evapotranspiration) adjust the final hydrological performance calculations.

Output metrics include infiltration rate (mm/hr), daily infiltration capacity (mm/day), actual infiltration given the precipitation amount, surface runoff, infiltration efficiency percentage, and flood/overflow risk level.



```
Spatial Comparison Analysis:

Map Display: WGS84 Coordinates (Lat/Lon)
Center: 50.8007°N, 4.4282°E

What the maps show:
• Top Left: Which NBS solution performs best at each location
• Top Right: Performance advantage of best over 2nd best
• Bottom Left: % improvement over existing conditions

Optimal NBS Usage Across Space:
• Rain Garden: 100.0% of area

Performance Statistics:
• Best rates range: 13.5 - 13.9 mm/hr
• Mean improvement: 187.6% over baseline
• Avg performance gap: 4.16 mm/hr

Spatial Coverage:
• Grid: 30×30 points (2.5×2.5 km)
• Soil samples in area: 981 points

Key Insight: Rain Garden
is optimal across 100.0% of the study area.
```

Figure 12: Example of *spatial analysis of NBS performance across the study area*. The plots show: (top left) optimal NBS solution map showing which NBS performs best at each location, (top right) performance advantage of the best solution over the second-best option, (bottom left) percentage improvement over existing conditions, (bottom right) some statistics information

3.1.1.3.3 Spatial Prediction Capabilities

The model supports spatially-explicit predictions across the entire Leuven study area. A spatial tree data structure (cKDTree) enables efficient nearest-neighbor searches for soil property interpolation at arbitrary locations. This capability facilitates generation of continuous infiltration rate maps and identification of optimal NBS placement locations.

Spatial interpolation uses inverse distance weighting with five nearest neighbors, ensuring predictions reflect local soil conditions while maintaining smooth spatial transitions. The coordinate system uses Belgian Lambert 72 projection, compatible with standard Flemish geospatial datasets.

3.1.1.4 Analysis Types and Use Cases

The infiltration model provides two primary analysis modes to support different user workflows and use cases. Currently, all analyses are executed through command-line interfaces, with plans for integration into the graphical user interface for improved accessibility of the users.

3.1.1.4.1 Interactive Analysis Mode

The interactive mode guides users through step-by-step configuration for single-location assessments. Users provide:

- **Location selection**: Coordinates can be entered directly (WGS84 format) or selected from predefined locations (e.g., Leuven center)
- **Weather conditions**: Temperature (°C), precipitation (mm), humidity (%), and 7-day antecedent precipitation
- **NBS scenarios**: Selection from available interventions including rain gardens, bioswales, green roofs, permeable pavement, constructed wetlands, urban forests, infiltration trenches, and existing conditions baseline

The system validates inputs, generates predictions for all selected NBS types, and produces comparison visualizations showing infiltration rates, runoff generation, efficiency metrics, and flood risk classifications. Results are displayed interactively and can be saved as HTML reports with embedded plots.

The interactive mode includes predefined location presets for common sites in Leuven, automatic coordinate system conversion between Lambert 72 and WGS84 formats, real-time input validation with acceptable value ranges, and display of model performance metrics (R^2 score) for transparency. All NBS scenarios are automatically compared against existing conditions baseline to quantify improvement factors.

3.1.1.4.2 Batch Analysis Mode

Batch mode enables automated processing of predefined analysis workflows. Rather than interactive prompts, users run analysis scripts that systematically evaluate NBS scenarios across multiple conditions.

The batch mode executes analyses by calling the model's prediction functions directly with specified parameters. Typical batch workflows include:

- Running all NBS scenarios for a fixed location and weather condition
- Generating spatial comparison maps across grid points
- Creating comprehensive HTML reports with all scenarios included

Batch processing automatically generates complete result sets including comparison visualizations, spatial maps, and statistical summaries without requiring user interaction during execution. This mode is suitable for generating standardized reports, running systematic comparisons, or producing documentation outputs.

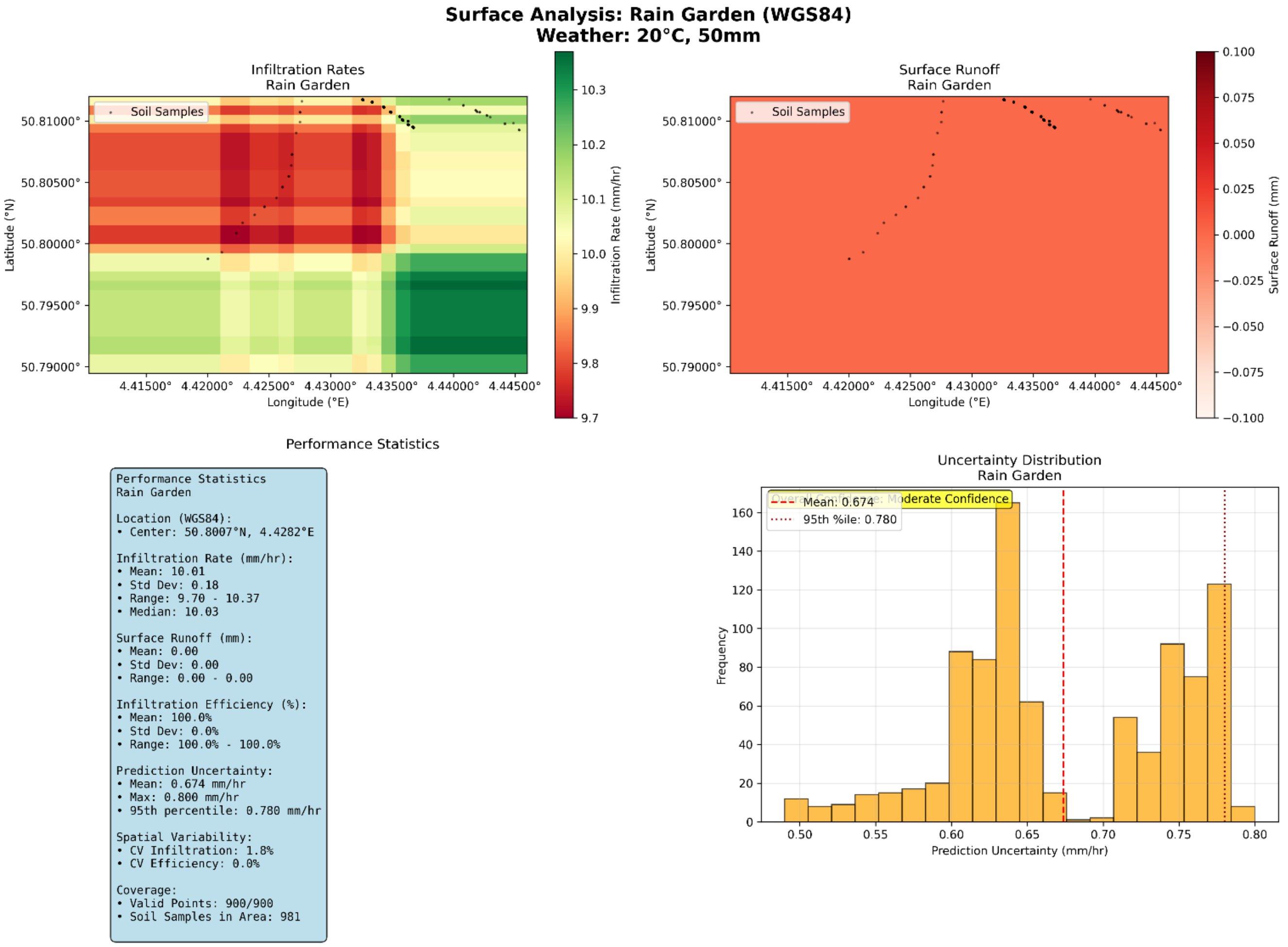
3.1.1.5 *Model Outputs and Applications*

3.1.1.5.1 *Primary Outputs*

For each prediction, the model provides different hydrological performance metrics:

- Infiltration rate with uncertainty bounds (mm/hr)
- Daily infiltration capacity (mm/day)
- Actual infiltration given precipitation input (mm/day)
- Surface runoff volume (mm/day)
- Infiltration efficiency (percentage)
- Flood risk classification (Low/Medium/High)
- Individual model predictions (Random Forest, Gradient Boosting) for transparency

The system also automatically generates HTML reports for NBS scenario comparisons. These interactive reports include visualization of infiltration performance across different NBS types, comparison tables, and spatial prediction maps. The HTML format can be used for sharing information easily with stakeholders and integration into project documentation.



```
Performance Statistics
Rain Garden

Location (WGS84):
• Center: 50.8007°N, 4.4282°E

Infiltration Rate (mm/hr):
• Mean: 10.01
• Std Dev: 0.18
• Range: 9.70 - 10.37
• Median: 10.03

Surface Runoff (mm):
• Mean: 0.00
• Std Dev: 0.00
• Range: 0.00 - 0.00

Infiltration Efficiency (%):
• Mean: 100.0%
• Std Dev: 0.0%
• Range: 100.0% - 100.0%

Prediction Uncertainty:
• Mean: 0.674 mm/hr
• Max: 0.800 mm/hr
• 95th percentile: 0.780 mm/hr

Spatial Variability:
• CV Infiltration: 1.8%
• CV Efficiency: 0.0%

Coverage:
• Valid Points: 900/900
• Soil Samples in Area: 981
```

Figure 13: Example of spatial analysis for Rain Garden implementation (20°C, 50mm rainfall). Maps show spatial distribution of infiltration rates and surface runoff with soil sample locations overlaid

3.1.1.5.2 Technical Implementation

The model is implemented in Python using scikit-learn for machine learning algorithms, pandas and NumPy for data manipulation, and SciPy for spatial operations. The complete trained model (including ensemble estimators, feature scalers, and metadata) is serialized using joblib for efficient storage and loading. Model files are deployed within Docker containers alongside the VIE-AI dashboard for integrated explainability analysis.

The implementation relies on standard, well-supported open-source libraries that ensure reproducibility and maintainability. Pandas and NumPy are used for tabular data handling, feature engineering, and efficient numerical computation, while SciPy supports interpolation and spatial operations required for generating continuous infiltration surfaces from irregular soil sampling points. Scikit-learn provides the core machine learning functionality, including Random Forest and Gradient Boosting estimators, train–test splitting, and model evaluation utilities, with joblib used to serialize the full pipeline (models, feature scalers, and configuration) into compact binaries for deployment. All components are orchestrated within a Dockerized environment that bundles Python dependencies, and model artefacts enabling consistent execution across development machines, servers, and future deployments.

The codebase follows modular design principles with separate components for data processing, model training, NBS scenario management, and spatial prediction. This architecture enables easy extension to additional NBS types or integration of new soil data sources, so it can be easily adaptable for different cities/case studies (Figure 14).

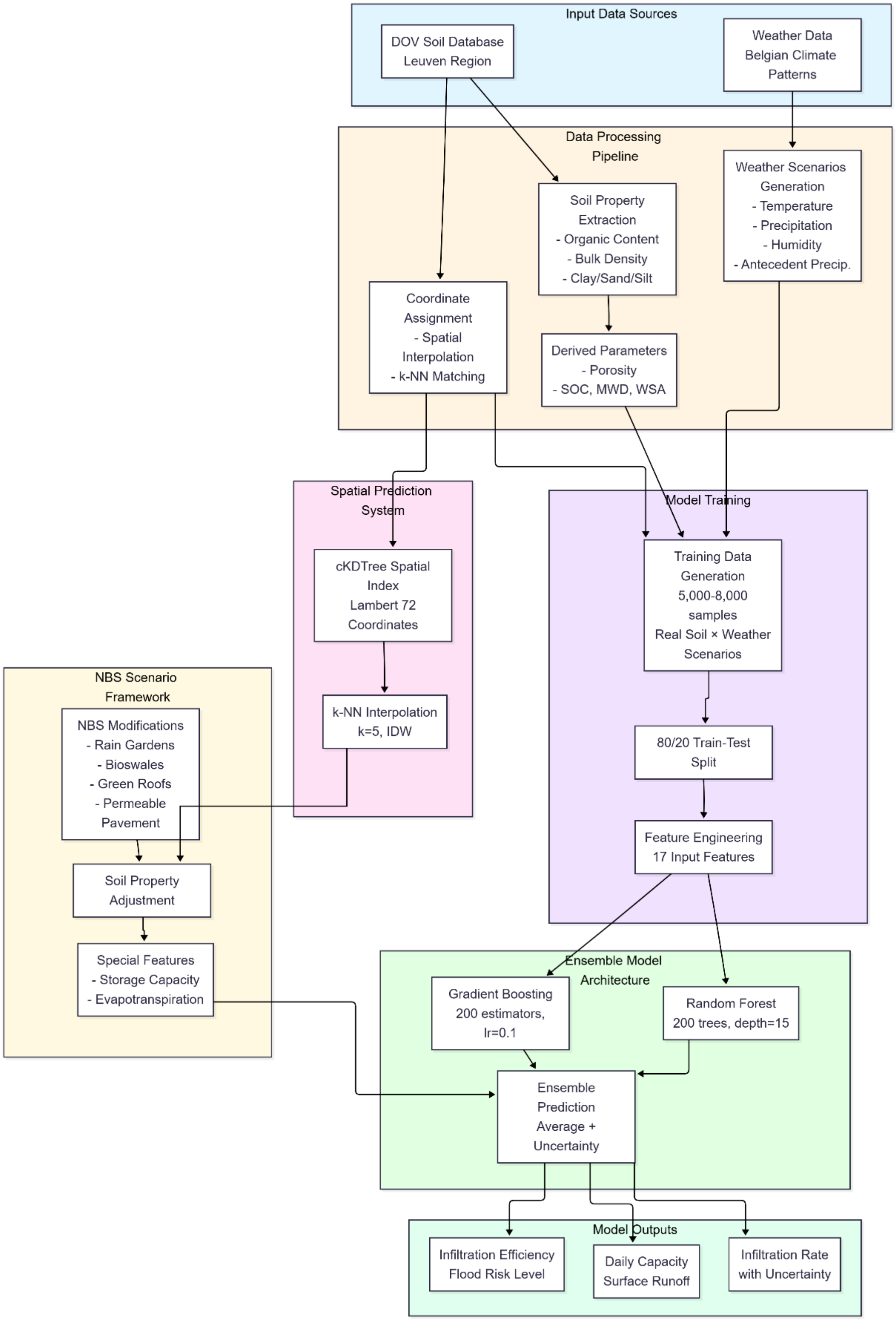

Input Data Sources
DOV Soil Database Leuven Region
Weather Data Belgian Climate Patterns
Data Processing Pipeline
Soil Property Extraction - Organic Content - Bulk Density - Clay/Sand/Silt
Weather Scenarios Generation - Temperature - Precipitation - Humidity - Antecedent Precip.
Coordinate Assignment - Spatial Interpolation - k-NN Matching
Derived Parameters - Porosity - SOC, MWD, WSA
Spatial Prediction System
cKDTree Spatial Index Lambert 72 Coordinates
k-NN Interpolation k=5, IDW
Model Training
Training Data Generation 5,000-8,000 samples Real Soil × Weather Scenarios
80/20 Train-Test Split
Feature Engineering 17 Input Features
NBS Scenario Framework
NBS Modifications - Rain Gardens - Bioswales - Green Roofs - Permeable Pavement
Soil Property Adjustment
Special Features - Storage Capacity - Evapotranspiration
Ensemble Model Architecture
Gradient Boosting 200 estimators, lr=0.1
Random Forest 200 trees, depth=15
Ensemble Prediction Average + Uncertainty
Model Outputs
Infiltration Efficiency Flood Risk Level
Daily Capacity Surface Runoff
Infiltration Rate with Uncertainty

Figure 14: System Architecture Diagram

3.1.1.6 Limitations and Future Work

The model's primary limitation is its reliance on synthetically generated data to augment the training dataset. While this hybrid approach enables broad coverage of soil-weather combinations, it cannot fully replicate the complexity and interactions present in real-world measurements. Future improvements should incorporate empirical field measurements combining simultaneous soil property, meteorological, and infiltration observations to validate and refine the synthetic data generation approach.

The NBS modifications are based on literature values rather than site-specific measurements. Field monitoring of implemented NBS projects would enable validation and refinement of the modification parameters.

Spatial resolution is constrained by the density of original DOV soil samples. Integration of remote sensing data or geophysical surveys could enhance spatial detail, particularly in under-sampled areas.

The current command-line interface, while functional for technical users, presents accessibility barriers for urban planners and policy makers. Future development will integrate the infiltration model into the graphical user interface to improve usability.

3.1.2 Weather and seasonal AI-based forecasts and climate projections

The weather - seasonal forecasts and climate projections are not tools, but the modelling results are a key input for the weather, seasonal and climate visualization URBREATH tools, among other related models and tools. Detailed and updated descriptions can be found in D3.1 (February 2025) and D3.2 (December 2025) documents. Here is a concise, self-contained summary of each model.

3.1.2.1. Weather forecast

For this model, probabilistic weather forecasting was adopted through Ensemble Prediction Systems (EPS) as the optimal approach for short- to medium-term prediction (1-10 days). This decision, agreed upon at the Madrid General Assembly, aligns URBREATH with leading meteorological institutions, offering significant advantages for urban resilience by quantifying forecast uncertainty, improving risk communication, and enhancing predictive skill for extreme events.

Raw EPS outputs often contain systematic biases and under-dispersion, especially at the high spatial resolutions required for urban environments. To bridge this gap, URBREATH integrates AI-based post-processing techniques alongside established statistical methods. The existing statistical correction framework uses both climate-based and operative (10-day preceding data) linear regression corrections, applied selectively based on forecast variability, and will complement the advanced AI methods.

The project is committed to exploring and implementing various AI methodologies—including Neural Networks (NNs) for bias correction, Quantile Regression Forests (QRF) / EMOS for probabilistic reliability, and Generative AI for precipitation enhancement-to optimize forecast skill, the updated methodology and results will be delivered in the D3.2 document at the end of December 2025. . The final operational system, performing at *ENG-Municipia* & *VCS* tools environment, became available for cities in early December 2025. It will use continuous daily downloads from selected EPS providers (best from three different models) and integrate near-real-time predictions into the AI framework. The historical data evaluation (2023-2024) will be used to train the primary AI algorithm (a Neural Network) to produce a forecast specifically tailored to each study case, thereby ensuring the reliability and sustainability of the operational data pipeline. Weather forecast is delivered to the plotting tools partners via WMS.

3.1.2.2. Seasonal forecast

Seasonal forecast modeling (0-6months) has been developed adopting a novel statistically driven approach methodology known as the Teleconnection Wavelet-ARIMA (TeWA) method, modified and extended through Artificial Intelligence (TeWA-CNN). This strategy is chosen due to the inherent unpredictability of seasonal variability in Mediterranean climates and the non-linear sensitivity of current physical models.

The core of the methodology is a hybrid statistical model built on three pillars: AI-Driven Predictor Selection (using CNNs to dynamically analyze global atmospheric and oceanic variables), Wavelet Analysis (to isolate predictable frequency components), and ARIMA Modeling (to capture residual autocorrelated behavior). The final output is a hybrid forecast that dynamically weights the contributions of the CNN-driven teleconnection component and the ARIMA component.

While this approach is computationally costly and is highly sensitive to the stationarity (coupling) between predictors and predictands - often resulting in "decoupling" during specific seasons, particularly in Southern Europe - preliminary findings validate its adoption. Even prior to incorporating CNNs, the original TeWA method demonstrated significantly high performance, showing up to a 70% improvement over ECMWF’s SEAS5 model for precipitation and up to 60% improvement for temperature. Initial estimations from integrating CNNs in the first stages indicate a further 50% improvement in the short-term forecast, confirming the strategic value of this complex, tailored methodology for the project. The updated methodology and results will be delivered in the D3.2 document at the end of December 2025.

The seasonal forecast is being delivered to the plotting tools partners via WMS. Thus, the final operational system, performing at *ENG-Municipia* & *VCS* tools environment, became available for cities in early December 2025.

3.1.2.3. Climate projections

Assessing climate risk requires defining local-scale impacts based on the future evolution of meteorological variables (temperature, precipitation, wind). The URBREATH project provides this local climate information using a novel two-step statistical regionalization methodology developed by the Foundation for Climate Research, applicable to both maximum/minimum temperature and precipitation. This methodology falls within the scope of AI techniques, utilizing principles related to pattern recognition and knowledge transfer between spatial scales.

The process is structured around analogs and transfer functions. The first step involves Analogue Stratification, where the method identifies the most similar historical atmospheric configurations (analogs) from a reference database. This pattern recognition phase is critical, and machine learning algorithms (such as neural networks) can be employed to optimize the identification of relevant analogs based on synoptic forcing variables (e.g., geopotential or wind). The second step applies to Local Transfer Functions to adjust the large-scale climate model outputs to the specific local scale of the observatory. Temperature is estimated via multiple linear regression on the analogue days, using predictors related to thermal forcings, while Precipitation is obtained by averaging the most similar analogue days, which also allows for the estimation of the probability of rain.

Within URBREATH, Machine Learning (ML) is foreseen to further enhance accuracy by optimizing these statistical models and transfer functions, using approaches like deep neural networks to learn complex non-linear patterns. This refined downscaling ensures that adaptation measures in the Living Labs are based on highly precise local climate conditions expected for the coming decades.

The climate projections for basic variables (temperature, precipitation, etc.) are available for cities and technical partners at MinIO platform. Climate data will be complemented through tailored indices for those cities who require it. In any case, it will be delivered once due to a difference with short- and medium-term forecasts; climate projections are static due to its timescale scope. The updated results are described in the deliverable D3.2 submitted at the end of December 2025.

3.1.3 Land Surface Temperature Model (Heat Stress)

The Land Surface Temperature (LST) model developed by Latitudo 40 (LAT) generates high-resolution (10 m) temperature maps by combining the thermal infrared information from Landsat-8/9 with the finer spatial detail provided by Sentinel-2 reflectance bands. This fusion is achieved through an AI-based downscaling framework specifically designed to bridge the gap between the coarse resolution of thermal sensors and the high resolution required for urban analyses.
This approach not only improves spatial precision but also increases the robustness and reliability of temperature estimates across complex urban landscapes, making the model particularly valuable for applications related to urban heat island analysis, climate adaptation planning, and environmental monitoring.

3.1.3.1 Data Acquisition and Pre-Processing

3.1.3.1.1 Satellite Constellations

The downscaling architecture relies on the synergistic use of two distinct satellite constellations:

- Thermal Data: Acquired from Landsat-8 and Landsat-9 (TIRS sensors), providing thermal information at a native resolution of 100 m (resampled to 30 m in standard Level-2 products).
- Predictor Data: Acquired from Sentinel-2 (MSI sensor), providing multispectral reflectance data in the visible, near-infrared (NIR), and shortwave infrared (SWIR) ranges at 10 m and 20 m resolutions.

3.1.3.1.2 Temporal Co-registration and Masking

An automated ingestion pipeline identifies and pairs Landsat and Sentinel-2 acquisitions within a strict temporal window (max ±5 days) to minimize surface change artifacts. Both datasets undergo rigorous quality control, including atmospheric correction and the application of cloud and shadow masking algorithms to ensure radiometric integrity.

3.1.3.1.3 Feature Engineering

Prior to model ingestion, Sentinel-2 bands are harmonized to a consistent geometry. To capture the biophysical properties driving surface temperature, the feature space is augmented with a proprietary selection of spectral indices. These indices serve as proxies for critical environmental variables, including:

- Vegetation density and health (e.g., NDVI).
- Surface moisture content.
- Impervious surface fraction and built-up density.

3.1.3.2 Methodological Framework

3.1.3.2.1 Theoretical Basis

The core premise of the downscaling algorithm is the assumption of **scale invariance**. It posits that the mathematical relationship established between spectral reflectance (predictors) and surface temperature (target) at a coarse resolution remains valid at a finer resolution.

3.1.3.2.2 Machine Learning Architecture

The system employs an **Extra Trees Regressor** (Extremely Randomized Trees), an ensemble learning method chosen for its computational efficiency and resistance to overfitting. The training process proceeds as follows:

1. **Upscaling:** Sentinel-2 multispectral features are aggregated (resampled) to match the coarser spatial resolution of the Landsat thermal band.
2. **Model Training (Scene-Specific):** A regression model is trained for each specific Area of Interest (AOI).
3. **Inference (Downscaling):** The trained function is applied to the original, native-resolution Sentinel-2 data, resulting in a predicted LST map at 10 m resolution.

This approach uses zonal training and stratified cross-validation (70/30 split) to account for local microclimatic variations and surface heterogeneity.

3.1.3.3 Validation and Performance Metrics

3.1.3.3.1 Validation Protocol

Due to the absence of widespread in-situ thermal ground-truth at 10 m resolution, validation is performed via Aggregation-Based Consistency Analysis.
The downscaled 10 m LST product is re-aggregated (upscaled) to the original Landsat resolution. These

values are then compared against the original Landsat LST observational data. This method ensures that the downscaling process preserves the radiometric energy balance of the original thermal acquisition.

3.1.3.3.2 Global Accuracy Metrics

Performance metrics are continuously updated post-inference. Current global performance indicators are as follows:

- **Root Mean Square Error (RMSE):** 1.25 °C
- **Mean Absolute Error (MAE):** 0.98 °C

3.1.3.4 Technical Implementation and Applications

The processing pipeline is fully automated and containerized, utilizing a Python-based stack (scikit-learn, GDAL, Rasterio, NumPy) for scalable deployment. The system is optimized for high-throughput processing, capable of resolving 100 km² areas in approximately 15 minutes.

Applications:

This high-resolution thermal data supports critical decision-making in:

- **Urban Climatology:** Micro-scale Urban Heat Island (UHI) identification and mitigation planning.
- **Precision Agriculture:** Crop water stress monitoring.
- **Hydrology:** Evapotranspiration modeling and water resource management.

3.1.4 Numerical models for Nature-based Solutions (general)

Several numerical simulation models are applied and developed in Task 3.4. Deliverables D3.4 and D3.5 describe these models in more detail. None of these models are currently based on AI techniques, as we apply white box models and routines to calculate specific indicators to report on the environmental quality in the URBREATH cities. To make the use of these simulations more user friendly and try to increase the adoption of the tools by the cities, we've started outlining an approach to use agentic AI. Agentic AI can assist the user at the input side to ease configuration of the model by translating the question of a user into model settings. At the output side agentic AI can assist users in understanding model outcomes. The latter could as first step be understood as agent that allows users to 'chat with maps'. As an example, a user could have an interest in learning from a heat stress map what the average heat stress is in a neighborhood, as well as the coolest and hottest locations. More in general, we can formulate the first goal as:

- Automated geospatial intelligence workflows, enabling natural-language queries to generate environmental insights. An end-to-end workflow can be seen as:
  - A user asks a question in a user interface (linked to the URBREATH toolbox)
  - An orchestrator interprets the question and identifies geographic and data context
  - Agents compute (zonal) statistics using standardized geo-services
  - A data lake / repository provides authoritative and versioned geospatial layers
  - Results are returned as clear, decision-ready insights

Depending on the achievements and the interest of the URBREATH users, next goals will be set.

3.1.5 Crime Statistics Tool

The crime statistics component of the URBREATH analytical toolkit incorporates AI-based methods designed to analyse crime dynamics, quantify potential relationships between crimes and NbS or land use, and support data-driven urban decision-making. In this first version, the tool applies classical time-series analysis techniques, including the decomposition of crime records into trend, seasonality, and residual (error) components. This enables the identification of long-term structural changes, seasonal patterns, anomalies, and data inconsistencies. Additionally, the tool integrates a Long Short-Term Memory (LSTM) neural network model trained on historical crime data provided by the Leuven municipality. The LSTM model aims to capture complex temporal dependencies and non-linear behaviors, allowing for improved prediction of future crime levels at the neighborhood scale. These AI-supported analytical processes seek to provide an efficient way to monitor urban safety conditions and potential emerging risks. A detailed description of the methodology, including extended model evaluation and results, is presented in Deliverable 3.8 “AI models for Socioeconomic, Community, Organization, and Citizen Well-being – Version 2”.

3.1.5.1 Time Series Analysis Tool

Time-series analysis plays a central role in the crime prediction and analysis framework developed within URBREATH. Crime data usually shows temporal patterns, such as seasonal fluctuations, weekly cycles, long-term trends, and abrupt changes, making time-series–based analytical tools particularly suitable for detecting irregularities and predicting future behavior. By analyzing historical crime records from Leuven, the tool identifies deviations from expected patterns, highlights anomalous events, and supports proactive safety planning. Outlier detection methods, such as **residual-based anomaly analysis and Isolation Forest**, are integrated to distinguish normal variability from unusual or potentially concerning shifts in crime activity.

Residual-Based Outlier Detection

Residual-based outlier detection is a classical statistical method that identifies unusual observations by examining the difference between predicted and observed values [12]. After fitting a model (e.g., ARIMA, LSTM, or linear regression), the residuals (the errors between expected and actual crime counts) are analysed to detect anomalies. Large residuals or sudden spikes often indicate: 1) data inconsistencies, 2) abrupt events affecting crime levels (e.g., festivals, policy changes), or 3) long-term trends. Figure 15 presents indicative results of the application of the residual outlier detection algorithm on the A43 Sector of Leuven municipality’s central region.

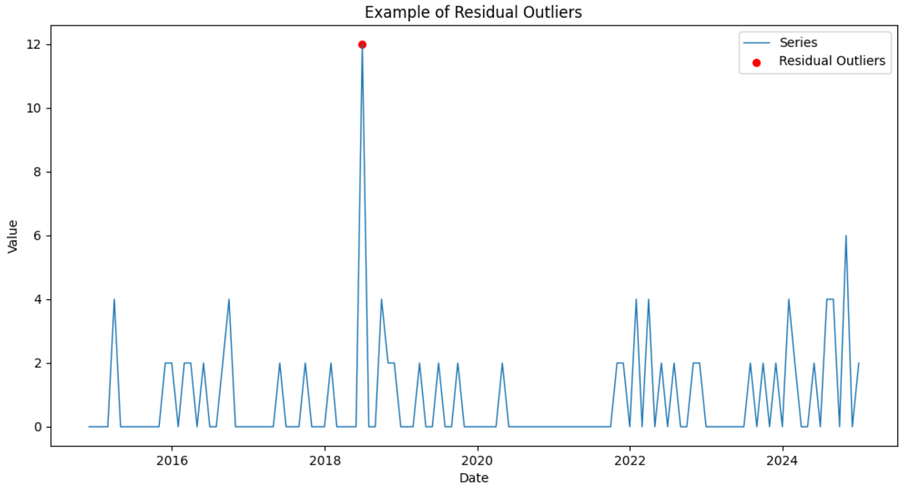


Figure 15: Example output of the residual anomalies method applied to a representative urban crime time series.

Isolation Forest

Isolation Forest is a machine learning-based technique for detecting anomalies, specifically designed to identify outliers in high-dimensional datasets **Errore. L'origine riferimento non è stata trovata.**,**Errore. L'origine riferimento non è stata trovata.**. Unlike methods that rely on distance or density, Isolation Forest isolates anomalies by recursively splitting the data space using random decision trees. Since anomalies are rare and differ from the majority of observations, they are typically isolated with fewer splits, resulting in shorter average path lengths within the forest.

In the context of crime analytics, which may have nonlinear dynamics, Isolation Forest can identify unusual temporal spikes, changes in variability, or deviations associated with extraordinary events (e.g., crime drops during lockdowns / extreme weather). A major advantage of the Isolation Forest method is that it is effective even on non-Gaussian distributions, and crime counts often fall into this category. Figure 16 presents indicative results of the application of the Isolation Forest outlier detection algorithm on the A15 Sector of Leuven municipality’s central region.

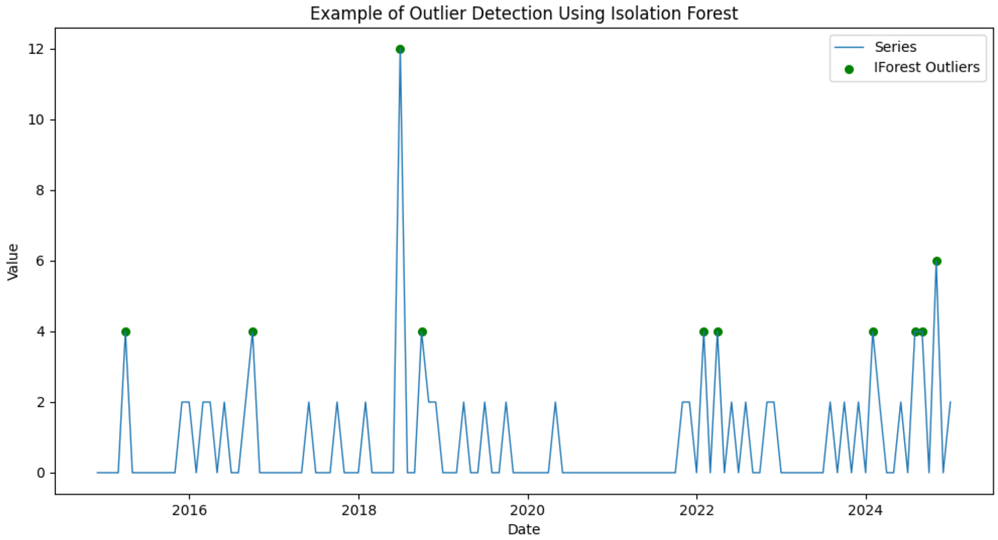


Figure 16: Example output of the Isolation Forest method applied to a representative urban crime time series.

3.1.5.2 AI crime level prediction tool

Within the URBREATH crime statistics tool, the LSTM architecture is applied to historical crime data from the Leuven municipality to uncover temporal patterns and generate predictions of future crime activity. Crime time series often contain seasonality, gradual structural changes, and abrupt deviations that cannot be effectively captured by simpler statistical models. By learning these temporal dependencies, the LSTM model supports improved forecasting, enhances the detection of emerging trends, and contributes to proactive decision-making for urban safety and citizen well-being.

Long Short-Term Memory (LSTM) networks

Long Short-Term Memory (LSTM) networks are a specialised type of recurrent neural network (RNN) designed to model sequential data by capturing dependencies that unfold over time **Errore. L'origine riferimento non è stata trovata.**, **Errore. L'origine riferimento non è stata trovata.**. Unlike traditional RNNs, LSTMs incorporate internal memory mechanisms that allow them to retain information over long sequences, making them particularly effective for time-series regression problems where recognising trends, seasonal patterns, and long-term relationships is essential.

An LSTM network is generally composed of the following key components:

- **Input Layer:** Processes the multivariate time-series data, such as socioeconomic indicators, environmental variables, or historical crime records.
- **LSTM Units:** Each unit contains a memory cell together with three gating mechanisms that regulate the flow of information:
    - **Input gate:** determines which new information is added to cell state.
    - **Forget gate:** identifies which past information should be discarded.
    - **Output gate:** controls what information is passed to the next layer.

These gates enable the model to preserve long-term dependencies and mitigate issues such as vanishing gradients.

- **Dense Layer:** Maps the temporal features learned by the LSTM units to a set of predictive representations.
- **Output Layer:** Produces the final regression output, typically representing the predicted value for each future time step.

LSTM networks excel at modelling nonlinear and long-range temporal behaviours, making them well suited for applications where sequential patterns and contextual dependencies strongly influence future outcomes. Figure 17 presents indicative preliminary loss function results for the LSTM model trained on vandalism incidents in Section A00, central Leuven.

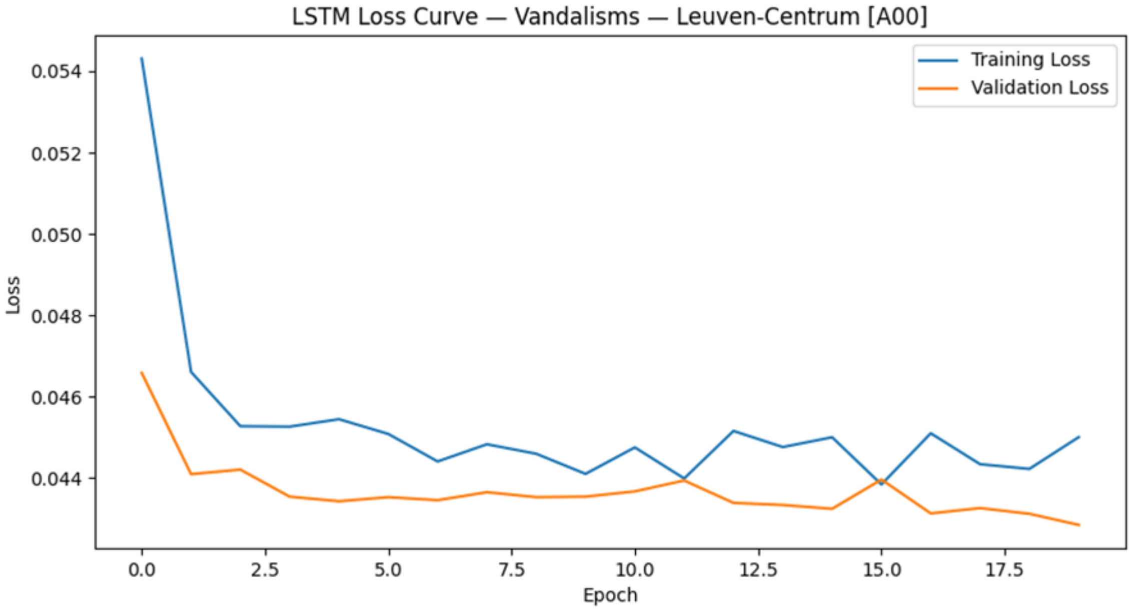


Figure 17: Indicative preliminary results of the loss function during model training for vandalism incidents in Section Aoo of central Leuven.

4 Explainability, Visualisation and interoperability

4.1 Explainability: the VIE-AI tool

VIE-AI (Visual Interpretable and Explanable AI) is a multi-model explainable AI dashboard system. The platform provides transparent, interpretable explanations for multiple machine learning models including flooding prediction, climate anomaly detection and infiltration modeling. The system integrates state-of-the-art XAI techniques including SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) to ensure model transparency and trustworthiness.

VIE-AI enables end-users to better understand the decision-making processes of complex systems. The tool focuses on visual interpretability for faster and more accessible communication to users, developing visualizations and interactive tools to assist data scientists and domain experts.

The primary function of VIE-AI is to take as input the AI models developed in the project, along with a representative subset of the datasets used to train those models. This approach allows VIE-AI to generate meaningful insights into how various input features affect models' predictions.

4.1.1 Core Capabilities

VIE-AI uses a range of explainability techniques designed for different user needs. Feature Importance shows how much each input variable contributes to a model's predictions-for instance, in an infiltration model, VIE-AI can reveal whether soil moisture, precipitation, or temperature has the greatest impact on predicted infiltration rates.

The tool implements SHAP (SHapley Additive exPlanations) values, a state-of-the-art method for attributing model predictions to individual input features in a mathematically sound and visually intuitive way. This allows users to assess not only which features are important, but also the direction and magnitude of their influence on specific predictions.

The tool applies advanced interpretation methods including SHAP and LIME (Local Interpretable Model-agnostic Explanations) directly to the trained models. SHAP provides both global feature importance rankings and individual prediction explanations using game-theoretic principles, while LIME generates local explanations by fitting interpretable surrogate models around specific predictions. These complementary techniques allow users to understand model behavior at both the global level (overall patterns across all predictions) and the local level (why a specific prediction was made).

VIE-AI addresses the needs of different URBREATH users through its interactive XAI features. Data scientists and domain experts can use What-If analysis to adjust feature values and observe how predictions respond in real-time. SHAP dependence plots and LIME explanations show how individual features influence model outputs, helping users identify patterns and dependencies that shape model behavior. For city managers and policymakers, the dashboard provides accessible visualizations -

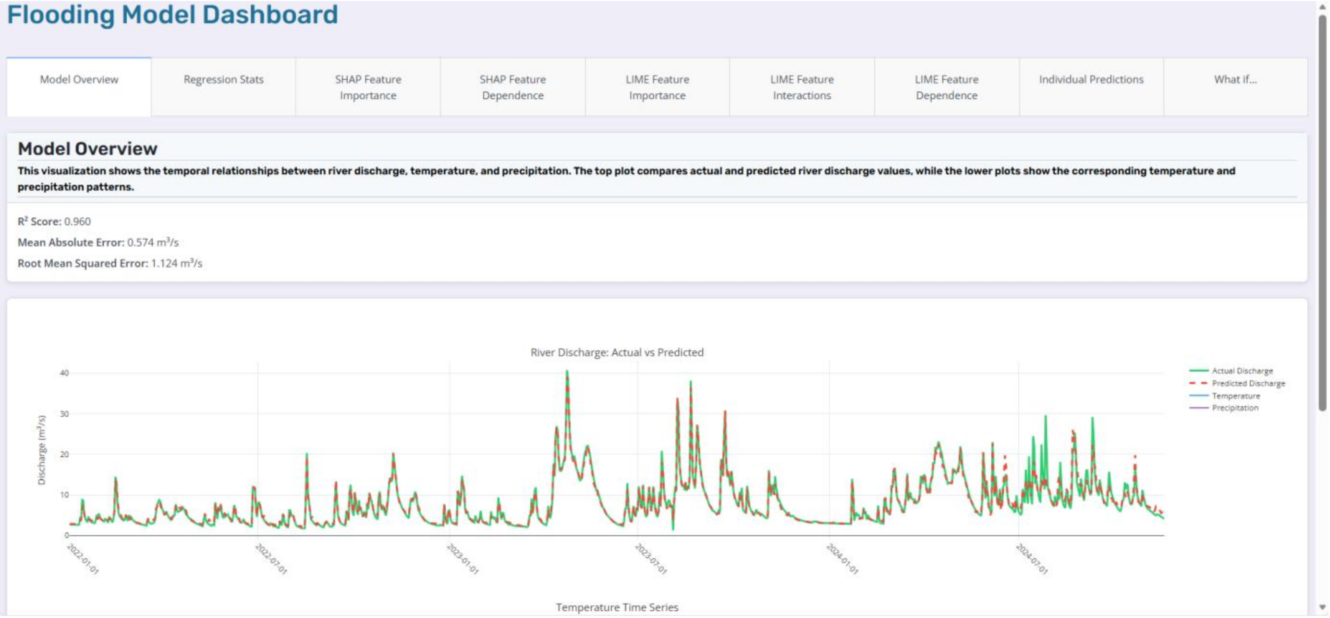
including performance metrics, prediction timelines, and color-coded feature contributions that support informed urban planning decisions.

4.1.2 Key Features and Functionality

The VIE-AI tool integrates an array of features designed to enhance model transparency and support deep interpretability of AI systems. The dashboard is organized into three main sections: Basic Model Understanding, SHAP-based Explanations, and LIME-based Explanations, each offering different perspectives on model interpretation.

4.1.2.1 Basic Model Understanding

The Basic Model Understanding section provides users - particularly data scientists and technical experts - with essential insights into the nature of the model being analyzed. This includes a summary of the model's type (e.g., regression or classification), key performance metrics, and visual diagnostics such as observed vs. predicted value plots and residual plots. These visualizations help users quickly assess the reliability and performance of the model by highlighting where and how predictions diverge from observed outcomes (Figure 18).



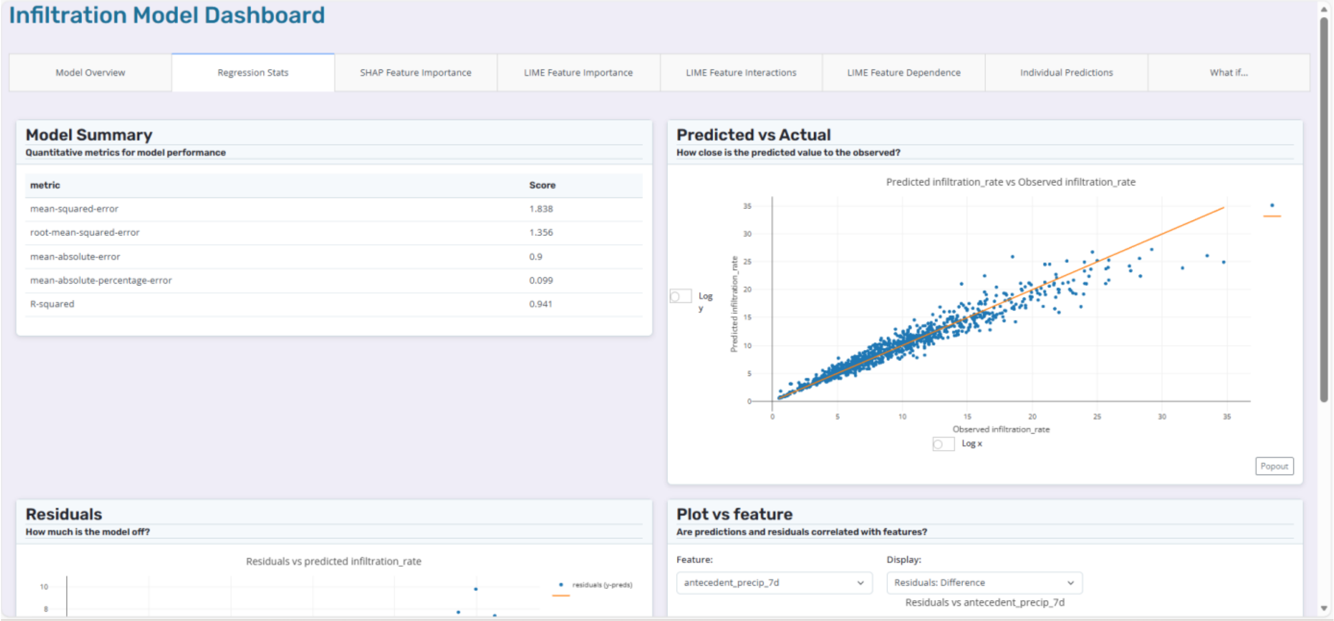
Model Overview

This visualization shows the temporal relationships between river discharge, temperature, and precipitation. The top plot compares actual and predicted river discharge values, while the lower plots show the corresponding temperature and precipitation patterns.

R^2 Score: 0.960
Mean Absolute Error: 0.574 m^3/s
Root Mean Squared Error: 1.124 m^3/s

Figure 18:Example of Model Overview Tab showing performance metrics and predicted vs. actual plot

The Regression/Classification Stats tab offers detailed statistical analysis useful for technical users who need deeper statistical insights. The Predicted vs Actual plot shows observed and predicted values together - a perfect model would have all points on the diagonal. The Residuals plot displays the difference between observed and predicted values, allowing users to check if residuals are higher or lower for different outcome ranges. The Plot vs Feature visualization shows residuals plotted against feature values, enabling inspection of whether the model performs worse for particular ranges of feature values (Figure 19).



Infiltration Model Dashboard

Model Summary

Quantitative metrics for model performance

metric	Score
mean-squared-error	1.838
root-mean-squared-error	1.356
mean-absolute-error	0.9
mean-absolute-percentage-error	0.099
R-squared	0.941

Predicted vs Actual

How close is the predicted value to the observed?

Predicted infiltration_rate vs Observed infiltration_rate
Predicted infiltration_rate
0
5
10
15
20
25
30
35
Observed infiltration_rate
Log y
Log x
Popout

Residuals

How much is the model off?

Plot vs feature

Are predictions and residuals correlated with features?

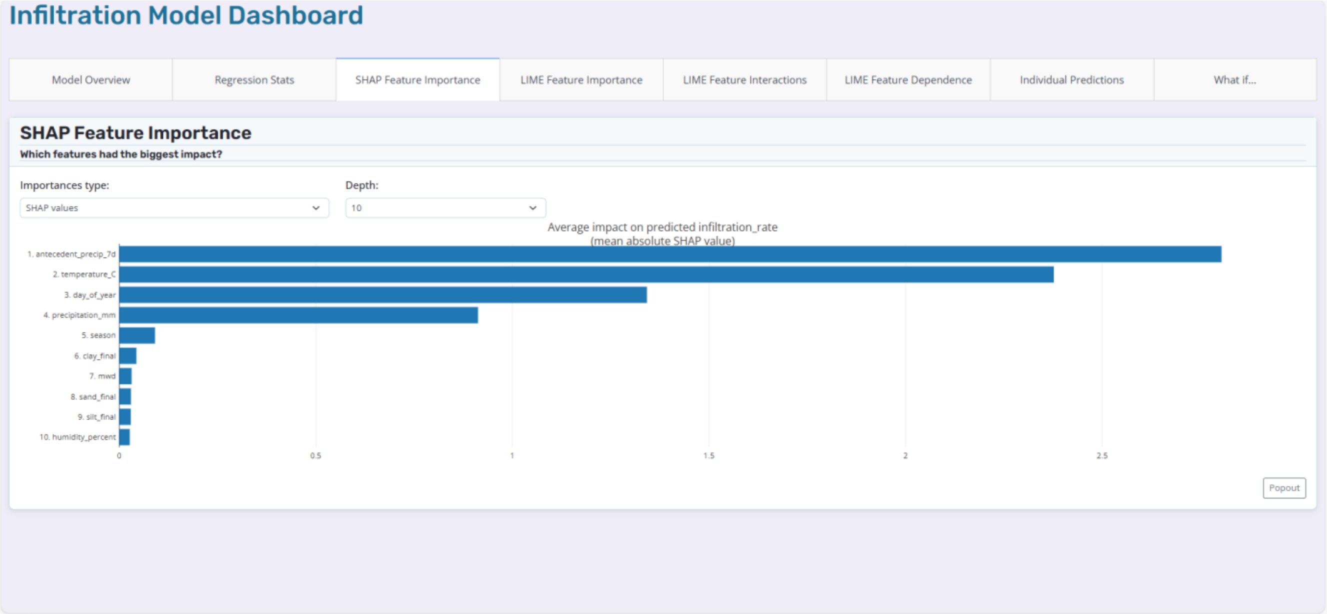
Feature: antecedent_precip_7d

Display: Residuals: Difference

Figure 19: Example of Regression Stats Tab

4.1.2.2 *SHAP Analysis*

A core pillar of VIE-AI is its integration of SHAP analysis. The tool offers different SHAP-based tabs, including Feature Importance plots where input variables are ranked according to their contribution to model predictions. Users can sort these visualizations based on absolute SHAP values (average absolute impact of the feature on final prediction) or permutation importance (how much the model degrades when a feature is shuffled) and customize the number of features displayed (Figure 20).



Infiltration Model Dashboard

SHAP Feature Importance

Which features had the biggest impact?

Importances type: SHAP values

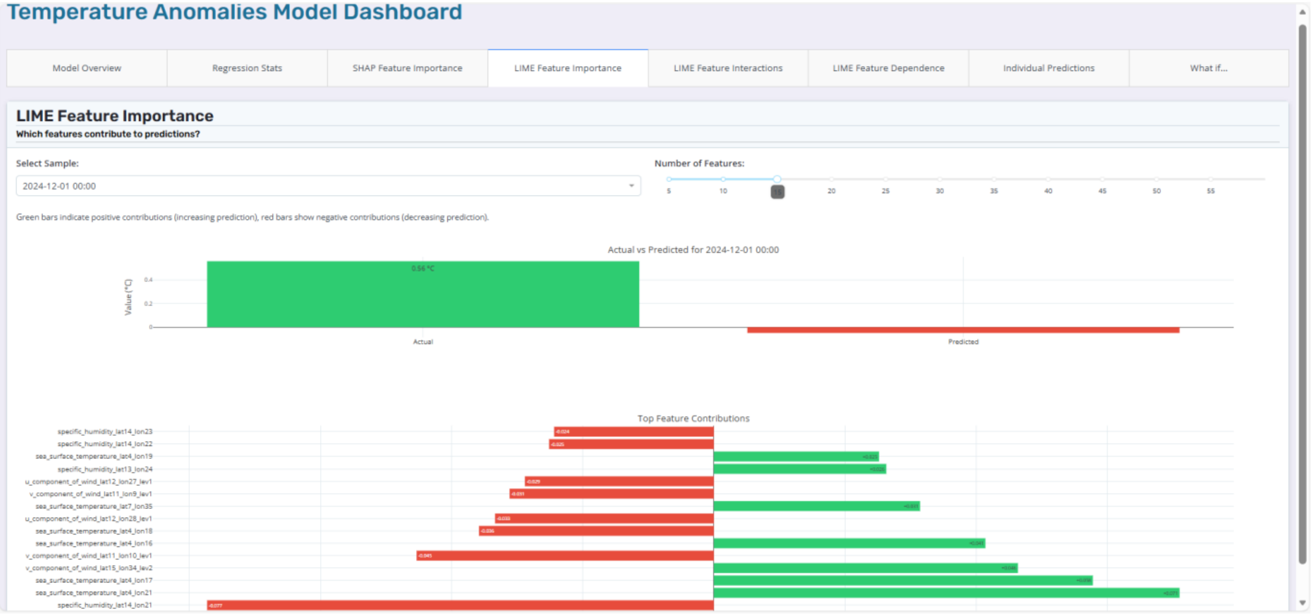
Depth: 10

Figure 20: Example of SHAP Feature Importance Tab showing ranked features with bar charts

4.1.2.3 LIME Analysis

Complementing the SHAP analysis is the inclusion of LIME (Local Interpretable Model-agnostic Explanations). This section mirrors SHAP's objectives but uses alternative visualization techniques, such as heatmaps and case-based analysis of prediction contributions. The LIME Feature Importance tab explains individual predictions with color-coded bars showing feature contributions - green bars indicate positive contributions while red bars show negative impact. These visualizations allow users to compare actual versus predicted outcomes on an individual basis and examine how specific features influence a single prediction.

The LIME Feature Interactions tab (Figure 21) offers local interaction analysis using heat maps for visualization, helping understand specific predictions, and supporting detailed investigation. The LIME Feature Dependence section (Figure 22) demonstrates local feature relationships with interactive plotting options, useful for validating model behavior and model refinement.



Temperature Anomalies Model Dashboard

Model Overview
Regression Stats
SHAP Feature Importance
LIME Feature Importance
LIME Feature Interactions
LIME Feature Dependence
Individual Predictions
What if...

LIME Feature Importance

Which features contribute to predictions?

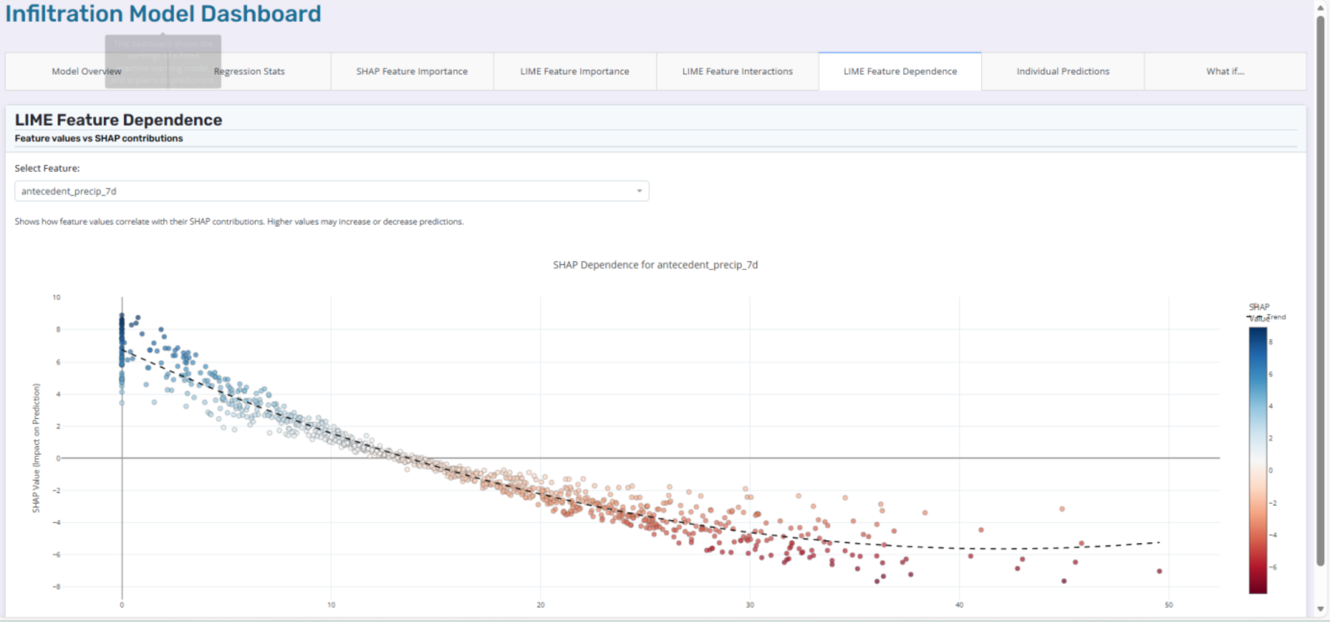
Select Sample: 2024-12-01 00:00

Number of Features: 15

Green bars indicate positive contributions (increasing prediction), red bars show negative contributions (decreasing prediction).

5
10
15
20
25
30
35
40
45
50
55
Actual vs Predicted for 2024-12-01 00:00
Value (°C)
0.4
0.2
0
0.56 °C
Actual
Predicted
Top Feature Contributions
specific_humidity_lat14_lon23
-0.024
specific_humidity_lat14_lon22
-0.025
sea_surface_temperature_lat4_lon19
+0.025
specific_humidity_lat13_lon24
+0.026
u_component_of_wind_lat12_lon27_lev1
-0.029
v_component_of_wind_lat11_lon9_lev1
-0.031
sea_surface_temperature_lat7_lon35
+0.031
u_component_of_wind_lat12_lon28_lev1
-0.033
sea_surface_temperature_lat4_lon18
-0.036
sea_surface_temperature_lat4_lon16
+0.041
v_component_of_wind_lat11_lon10_lev1
-0.045
v_component_of_wind_lat15_lon34_lev2
+0.046
sea_surface_temperature_lat4_lon17
+0.058
sea_surface_temperature_lat4_lon21
+0.071
specific_humidity_lat14_lon21
-0.077

Figure 21: Example of LIME Feature Importance with color-coded contribution bars



Infiltration Model Dashboard

This dashboard shows the workings of a fitted machine learning model and explains its predictions
Model Overview
Regression Stats
SHAP Feature Importance
LIME Feature Importance
LIME Feature Interactions
LIME Feature Dependence
Individual Predictions
What if...

LIME Feature Dependence

Feature values vs SHAP contributions

Select Feature: antecedent_precip_7d

Shows how feature values correlate with their SHAP contributions. Higher values may increase or decrease predictions.

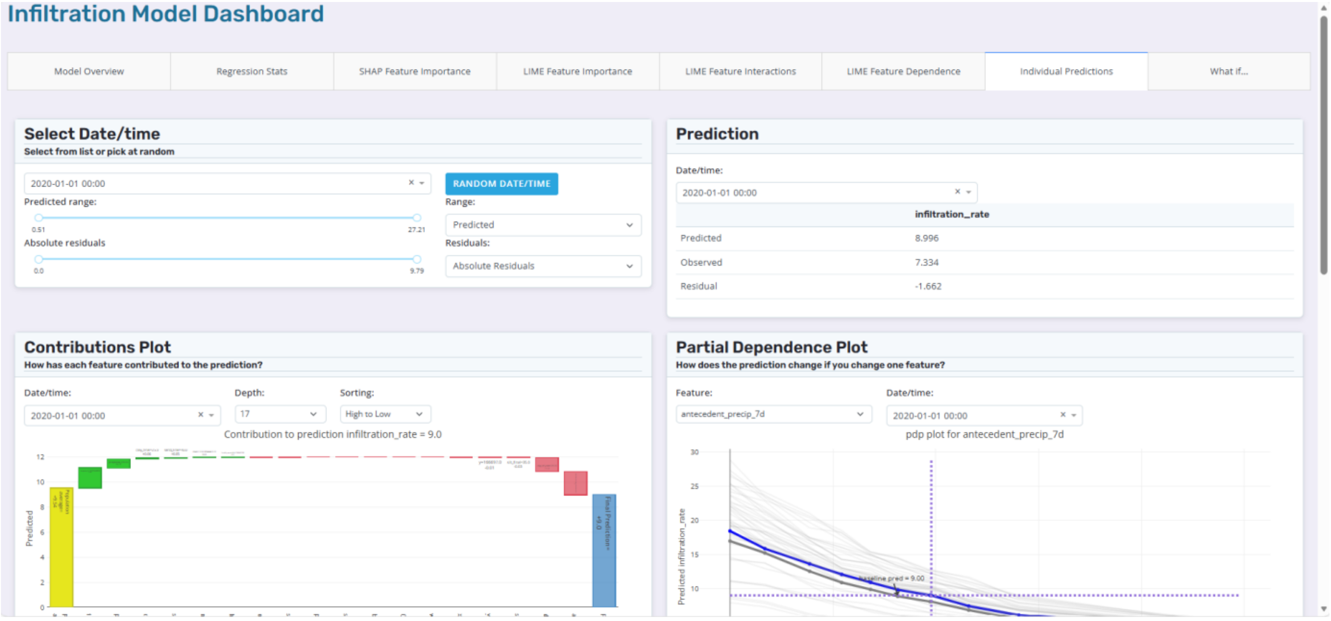
SHAP Dependence for antecedent_precip_7d
SHAP Value (Impact on Prediction)
10
8
6
4
2
0
-2
-4
-6
-8
0
10
20
30
40
50
SHAP Value
Trend
8
6
4
2
0
-2
-4
-6

Figure 22: Example of LIME Feature Dependence Tab

4.1.2.4 *Interactive Capabilities*

The Individual Predictions tab enables users to select specific instances or time points to examine in detail. This includes viewing prediction errors and the cumulative contribution of features toward a specific output. Users can compare different instances and export detailed reports.

The Contributions Plot shows how each individual feature has contributed to the prediction for a specific observation, with contributions starting from the population average and adding up to the final prediction. The Partial Dependence Plot (PDP) shows how the model prediction would change if one particular feature is altered. Users can adjust sampling rates and visual parameters to analyze how changes in one input feature affect model predictions on average.


Infiltration Model Dashboard

Select Date/time

Select from list or pick at random

2020-01-01 00:00

Predicted range: 0.51 – 27.21

Absolute residuals: 0.0 – 9.79

Range: Predicted

Residuals: Absolute Residuals

Prediction

Date/time: 2020-01-01 00:00

	infiltration_rate
Predicted	8.996
Observed	7.334
Residual	-1.662

Contributions Plot

How has each feature contributed to the prediction?

Date/time: 2020-01-01 00:00

Depth: 17

Sorting: High to Low

Partial Dependence Plot

How does the prediction change if you change one feature?

Feature: antecedent_precip_7d

Date/time: 2020-01-01 00:00

pdp plot for antecedent_precip_7d
Predicted infiltration_rate
30
25
20
15
10
baseline pred = 9.00

Figure 23: Example of Individual Predictions tab showing contribution plots and partial dependence

Furthermore, the tool features a What If analysis capability, which allows users to manipulate feature values in real-time and observe corresponding changes in predictions. This offers a powerful means of testing hypotheses and exploring model sensitivity, making it perfect for scenario exploration. Another feature is the Contribution Table that lists features from most to least impactful for a selected instance, further supporting localized explanation.

Together, these features make VIE-AI a robust, model-agnostic solution for explainable AI. It accommodates various model types and supports both high-level summaries and granular explorations. Designed primarily for domain experts and technical users, it offers both the flexibility and depth required to understand and communicate complex model behaviors effectively.

4.1.3 Technical Implementation

The VIE-AI tool implements a model-agnostic explainability framework designed to work with pre-trained machine learning models in pickle format. The system takes as input trained ML models and representative testing datasets, then provides comprehensive explainable AI capabilities without requiring access to the original training pipeline or data sources.

System Components:

The architecture consists of four primary layers:

1. Model Loading and Validation Layer - Handles ingestion of pre-trained ML models from pickle/joblib files, validates model compatibility, and extracts model metadata and feature requirements.
2. Data Processing Layer - Processes testing datasets to match model input requirements, handles feature alignment, and prepares data for explainability analysis.
3. Explainability Engine - Provides core XAI functionality through model-agnostic RegressionExplainer/ClassificationExplainer, SHAP/LIME integration for feature attribution, and explanation generation without requiring original training data.
4. Visualization Layer - Contains custom Plotly components for model performance analysis, prediction explanations, and interactive dashboards with specialized tabs (feature importance, interactions, dependence plots, what-if analysis).

Key Technical Features:

- Model Agnostic Design: Works with any scikit-learn compatible model or custom models, without requiring access to training pipelines or original datasets.
- SHAP/LIME Integration: Model-agnostic explainability using representative background samples from testing data, supporting various model types through unified SHAP explainer interface and custom LIME explainer interface.
- Containerized Deployment: Standalone Docker deployment requiring only model files and test datasets, ensuring portability across different environments without dependency on training infrastructure.

Component Communication:

The system follows a streamlined workflow as depicted in Figure 24:

pre-trained model loading → test dataset ingestion → model-data compatibility validation → explainer initialization with background sampling → interactive dashboard generation → real-time explanation computation for user interactions

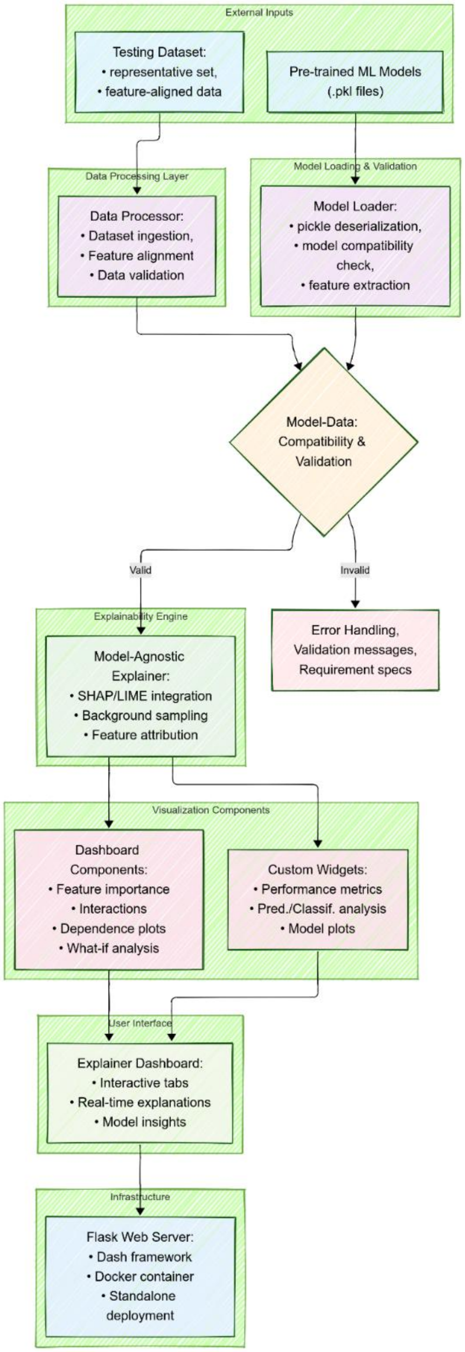


Figure 24: VIE-AI System Architecture showing the four-layer architecture and data flow between components

4.2 Monitoring, Visualisation and Interfaces with WP4 Digital Twin and City Dashboards

4.2.1 Simulator

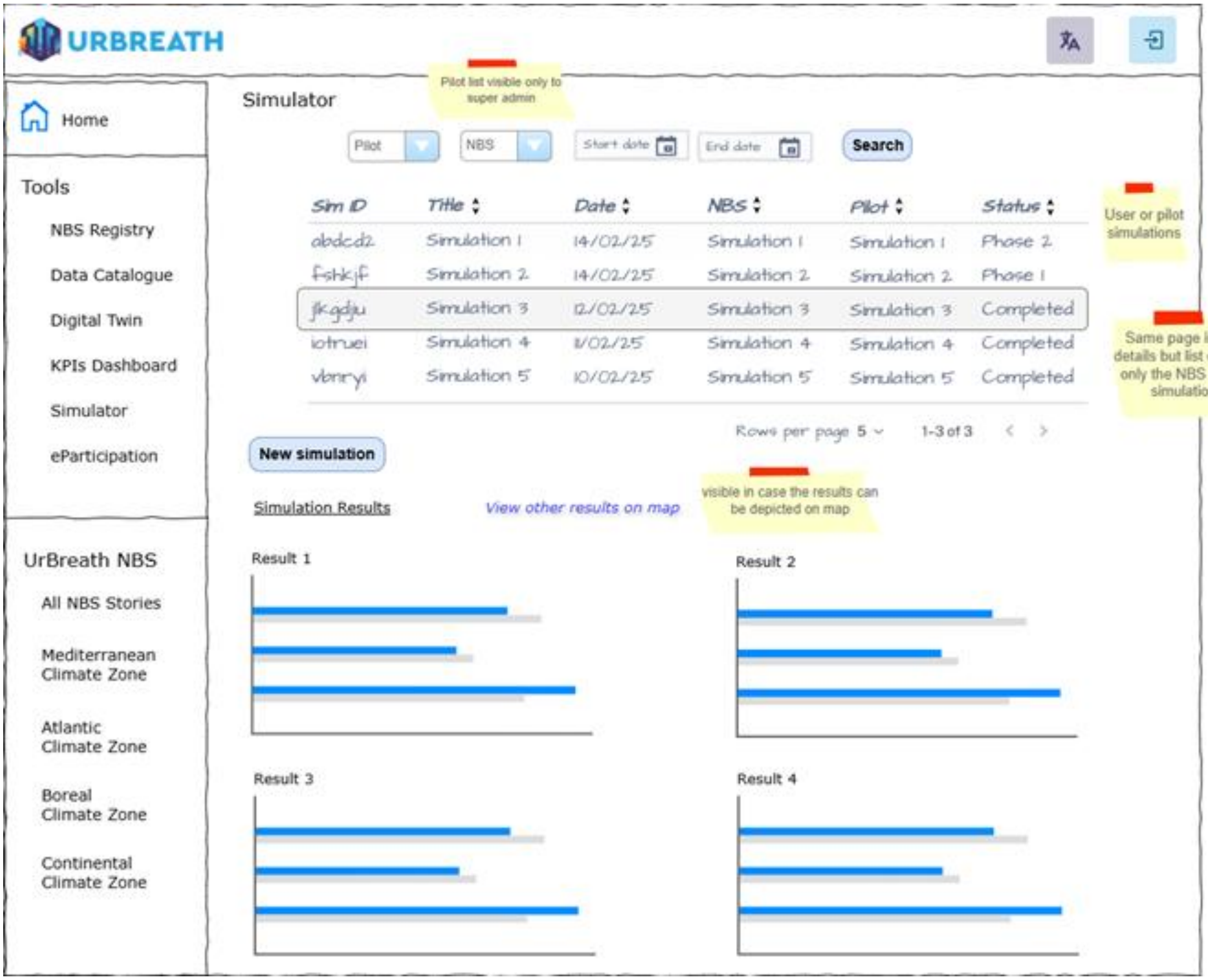
The Simulator component is currently at design phase, and it will allow the users to perform what if scenarios are, utilizing the AI models of the system. These scenarios (simulation runs) will be correlated to specific NBSs to add more value to the user's evaluations.

More precisely, a user will be able to perform the following operations:

- Selection of models and initialization of its parameters
- Initiation of a new simulation
- View of a list of simulation runs
- Search for simulation runs based on certain criteria like date range, NBS etc.
- Sort simulation runs based on criteria like date, NBS etc.
- View of simulation results for selected simulation runs

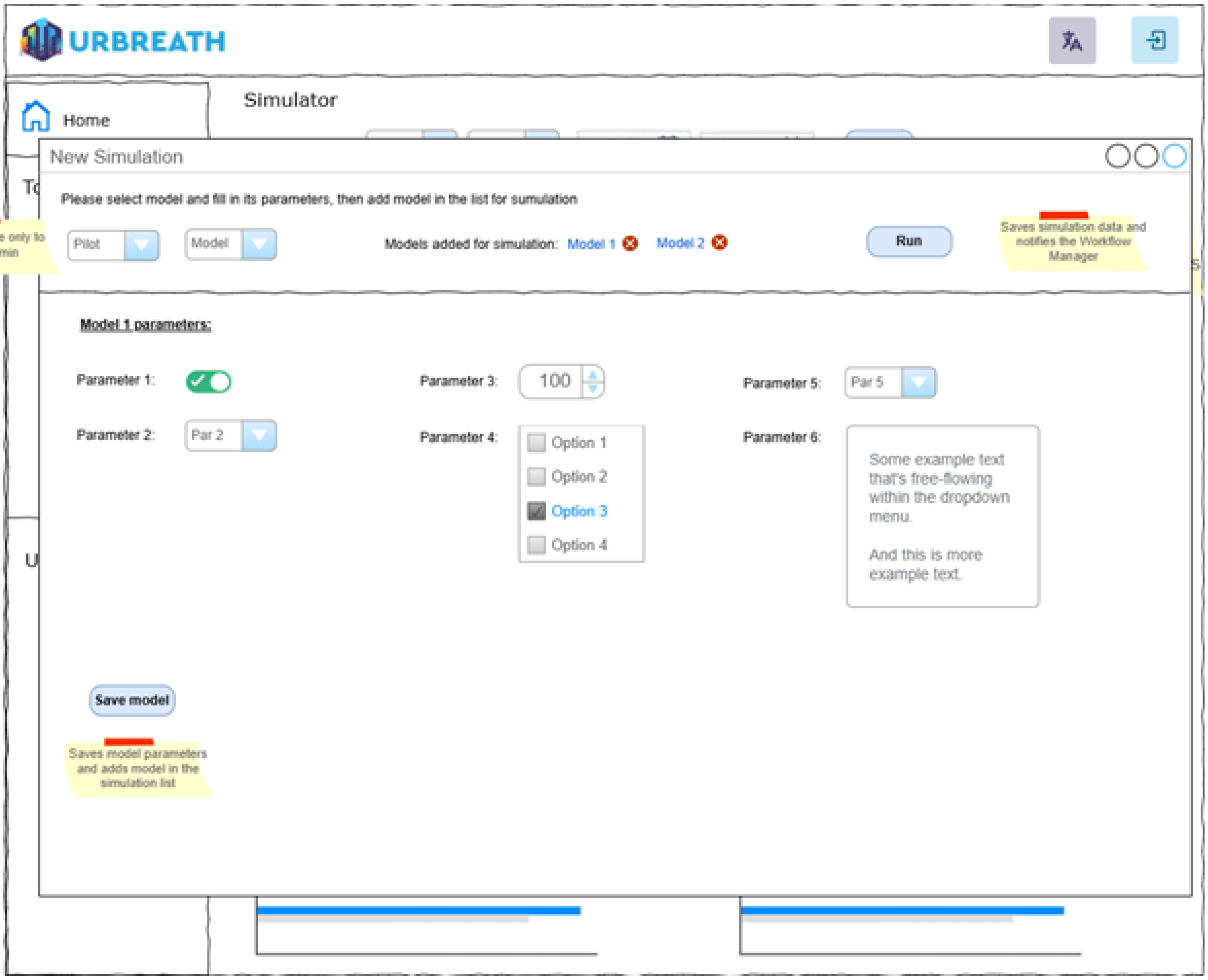
The Simulator will consist of a user interface developed in React JS and backend services and APIs developed in Java Spring Boot, backed by a PostgreSQL for data storage, and it will be integrated with the system's Message Bus in order to initiate the execution of the models (through the Workflow Engine) and to retrieve the results of the models.

Below we present the initial indicative wireframes presenting the key functionalities of the Simulator and the initial design of the database. Next steps involve the creation of design mockups to be validated by the pilot users and the finalization of the database and the Simulator UIs and services.



Sim ID	Title	Date	NBS	Pilot	Status
abdcd2	Simulation 1	14/02/25	Simulation 1	Simulation 1	Phase 2
fshkjf	Simulation 2	14/02/25	Simulation 2	Simulation 2	Phase 1
jkgdju	Simulation 3	12/02/25	Simulation 3	Simulation 3	Completed
iotruei	Simulation 4	11/02/25	Simulation 4	Simulation 4	Completed
vbnryi	Simulation 5	10/02/25	Simulation 5	Simulation 5	Completed

Figure 25: Simulator - Simulation runs view



Please select model and fill in its parameters, then add model in the list for sumulation

Pilot list visible only to super admin

Models added for simulation: Model 1 ☒ Model 2 ☒

Saves simulation data and notifies the Workflow Manager

Parameter 1: ✓

Parameter 2: Par 2

Parameter 3: 100

Parameter 4: □ Option 1 □ Option 2 ☑ Option 3 □ Option 4

Parameter 5: Par 5

Parameter 6: Some example text that's free-flowing within the dropdown menu.

And this is more example text.

Saves model parameters and adds model in the simulation list

Figure 26: Simulator - Model setup view

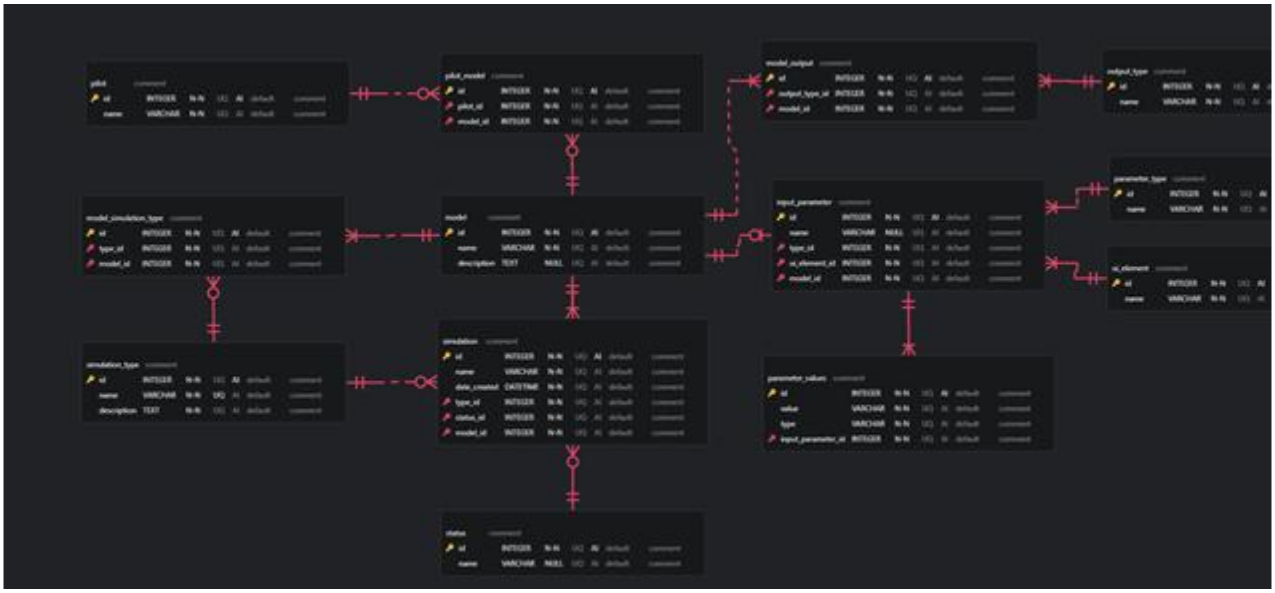

pilot comment
id INTEGER N-N UQ AI default comment
name VARCHAR N-N UQ AI default comment
pilot_model comment
id INTEGER N-N UQ AI default comment
pilot_id INTEGER N-N UQ AI default comment
model_id INTEGER N-N UQ AI default comment
model_output comment
id INTEGER N-N UQ AI default comment
output_type_id INTEGER N-N UQ AI default comment
model_id INTEGER N-N UQ AI default comment
output_type comment
id INTEGER N-N UQ AI
name VARCHAR N-N UQ AI
model_simulation_type comment
id INTEGER N-N UQ AI default comment
type_id INTEGER N-N UQ AI default comment
model_id INTEGER N-N UQ AI default comment
model comment
id INTEGER N-N UQ AI default comment
name VARCHAR N-N UQ AI default comment
description TEXT NULL UQ AI default comment
input_parameter comment
id INTEGER N-N UQ AI default comment
name VARCHAR NULL UQ AI default comment
type_id INTEGER N-N UQ AI default comment
ui_element_id INTEGER N-N UQ AI default comment
model_id INTEGER N-N UQ AI default comment
parameter_type comment
id INTEGER N-N UQ AI
name VARCHAR N-N UQ AI
ui_element comment
id INTEGER N-N UQ AI
name VARCHAR N-N UQ AI
simulation_type comment
id INTEGER N-N UQ AI default comment
name VARCHAR N-N UQ AI default comment
description TEXT N-N UQ AI default comment
simulation comment
id INTEGER N-N UQ AI default comment
name VARCHAR N-N UQ AI default comment
date_created DATETIME N-N UQ AI default comment
type_id INTEGER N-N UQ AI default comment
status_id INTEGER N-N UQ AI default comment
model_id INTEGER N-N UQ AI default comment
parameter_values comment
id INTEGER N-N UQ AI default comment
value VARCHAR N-N UQ AI default comment
type VARCHAR N-N UQ AI default comment
input_parameter_id INTEGER N-N UQ AI default comment
status comment
id INTEGER N-N UQ AI default comment
name VARCHAR NULL UQ AI default comment

Figure 27: Simulator DB

5 Integration and Implementation

5.1 Preliminary Results, Training sessions and Feedback

5.1.1 Infiltration Prediction Model

During demonstration session with Leuven stakeholders, several enhancement suggestions were identified. Leuven expressed specific interest in vegetation type effects on infiltration performance, noting the potential importance of distinguishing between shrubs, trees, and different plant species in NBS design. Additional feedback included requests for polygon-based spatial analysis to assess entire pilot areas rather than individual coordinate points, and soil pollution risk assessment to evaluate potential groundwater contamination from urban runoff.

Stakeholders also requested the ability to specify custom soil parameters reflecting local construction standards and highly disturbed urban soil conditions typical in redevelopment sites. The need for clearer distinctions between hydrological metrics (runoff, overflow, infiltration) and alignment with Flemish regulatory standards was emphasized. Integration possibilities with existing platforms (VCS, Sirio modeling, waterinfo.be flooding maps) were discussed as potential pathways for operational deployment.

Several feedback items align with ongoing research directions and will be investigated in upcoming project months, within the constraints of a research tool developed for demonstration and proof-of-concept purposes.

In the coming months, we will investigate whether vegetation type significantly affects soil permeability beyond current organic content parameterization. This includes evaluating whether local data availability (species inventories, root structure measurements) can support implementation of vegetation-specific infiltration factors. Literature indicates substantial differences between lawns, meadows, and tree cover in terms of soil structure maintenance and infiltration capacity.

We will explore methods for polygon-based analysis enabling assessment of heterogeneous pilot areas with mixed NBS interventions. This requires developing aggregation methods for spatially-variable predictions while maintaining computational efficiency within the research tool framework.

It is important to note that the infiltration model is at this stage a research tool designed for scenario exploration and proof-of-concept demonstration, not a fully operational decision-support system. Significant data limitations (synthetic soil-weather scenarios, literature-based NBS parameters, limited spatial resolution) constrain their application for site-specific engineering design. Full operational deployment would require extensive field validation, integration with certified hydrological models, and compliance with regulatory frameworks.

The feedback demonstrates strong interest while highlighting the gap between research tool capabilities and operational requirements for urban planning applications.

5.1.2 Satellite and Geospatial Data

During the initial engagement phase, Latitudo 40 conducted demonstration sessions with stakeholders from Parma, Aarhus, and Leuven to present satellite-derived geospatial data for urban climate monitoring. The demonstrations focused on two primary analytical layers generated from Sentinel-2 satellite imagery: Surface Urban Heat Islands (SUHI) identification and Heat Stress analysis. These layers provide spatially explicit information on temperature variations and thermal risk areas at 10-meter resolution.

Stakeholders were also introduced to a set of Key Performance Indicators (KPIs) developed to quantify environmental performance and evaluate the effectiveness of Nature-Based Solutions, including metrics related to cooling capacity, albedo variation, and vegetation cover density.

Following these demonstrations, collaborative activities were initiated to support the application of the Urban Simulation tool for pilot intervention area analysis. This tool enables predictive scenario generation by simulating the impact of proposed urban planning interventions on land surface temperature through synthetic satellite imagery. The next phase aims to establish standardized workflows for integrating local masterplans into the simulation environment, requiring training of pilot city stakeholders on spatial dataset preparation and formatting.

Concurrently, further refinement and expansion of the KPI framework is planned, in coordination with operational needs and validation objectives, to enhance decision-support capabilities within the URBREATH Toolbox.

During the demonstration sessions, stakeholders expressed interest in geospatial data layers and simulation capabilities. However, feedback indicated that the current KPI framework requires refinement to better align with local monitoring protocols and operational decision-making processes. Pilot cities highlighted the need for clearer threshold definitions, more granular temporal resolution, and enhanced integration with existing urban planning indicators. In response to this feedback, a structured review process will be initiated to refine and expand the KPI catalogue, ensuring that metrics are both scientifically robust and operationally relevant for urban climate adaptation planning.

5.1.3 Crime Statistic Tool

The initial implementation of the crime prediction module within URBREATH was tested on historical vandalism and assault records provided by the Leuven municipality. Using the LSTM-based forecasting approach, several model training sessions were conducted across both centrally located and peripheral sections of Leuven. Preliminary training results indicate that the model is able to capture key temporal dynamics in the crime data, such as recurring seasonal patterns and gradual long-term trends. The loss function curves show stable convergence behaviour across multiple training runs, suggesting that the selected hyperparameters and preprocessing steps provide a solid foundation for more advanced model tuning. These early outcomes demonstrate the feasibility of applying deep learning methods to urban crime time series and highlight the potential for improved predictive performance as additional data sources and features are incorporated.

Alongside the LSTM forecasting model, complementary analytical tools such as time-series decomposition and outlier detection (based on residual analysis and Isolation Forest) were evaluated to support the interpretation of the results. These methods proved valuable in identifying anomalies, inconsistent reporting periods, and intervals of unusual activity that may influence model performance. The preliminary results generated in Version 1 of the tool will guide the next development phase, where extended datasets, enriched feature sets, and cross-sectional validation will be employed to further strengthen the predictive capabilities of the system.

6 Roadmap Toward D3.10 - V2 (M48)

The evolution from D3.10 -V1 to its final version at M48 follows a structured and progressive roadmap that reflects both the maturity of the current implementation and the forward-looking requirements of WP3. The next phase aims to consolidate the foundational elements already developed (data acquisition, harmonization, AI-based modelling, and explainability) while gradually expanding their robustness, integration, and applicability to the Pilots. As the URBREATH technical framework continues to evolve, the roadmap prioritizes consistency across WP3 tasks, alignment with WP4's Digital Twin services, and validation activities carried out in WP5.

During the upcoming months, developments will focus on strengthening the existing AI tools, improving their scalability, and extending their operational readiness. EXUS will lead the enhancement of the VIE-AI tool, expanding its ability to support additional models and deepening its integration within the URBREATH data ecosystem.

In parallel, EXUS will also advance the Infiltration Prediction Model (already documented in D3.10 V1, Section 3) by validating it with new datasets, refining key parameters, and incorporating feedback from pilot use cases to ensure applicability across diverse urban contexts. All upgrades will follow the methodological principles established in the first version, ensuring continuity, reliability, and transparency in the resulting model outcomes.

In parallel with the development of the infiltration-related models, WP3 will continue advancing the numerical models for Nature-Based Solutions, with a particular focus on improving usability and expanding their adoption among practitioners. During 2026, VITO's team see initial experimentation with agentic AI to support a broader and more intuitive uptake of modelling tools and outputs. This exploratory work aims to allow users to interact with complex models through natural-language or task-oriented prompts, thereby reducing technical barriers and improving operational accessibility. As a first milestone in this direction, WP3 will work toward enabling AI-assisted geospatial data processing, allowing an AI agent to guide users through dataset preparation, spatial analysis steps, and model configuration workflows. This will set the foundation for more advanced forms of AI-supported modelling in future iterations.

Alongside the consolidation of existing tools, the next phase also includes the introduction of new analytical capabilities. Notably, EXUS will develop the Water Discharge and Flooding Prediction Model, which will extend the hydrological modelling capacity of WP3. This model will be designed to process multi-source hydrological, meteorological, and topographic inputs (such as river discharge measurements, precipitation data, catchment and land-surface characteristics, drainage system information, and detailed elevation models) to infer water level dynamics, discharge evolution, and potential flooding hotspots. Its outputs will include discharge time-series, flood probability indicators, threshold-based alerts, predicted water levels at critical points, and flood risk maps. These results will be compatible with the explainability and visualization framework already supported by VIE-AI, ensuring that new modelling functionalities remain aligned with the URBREATH transparency objectives.

The roadmap also envisions gradual refinement of the integration logic connecting AI models, harmonized datasets, and simulation workflows. Finally, the pathway to D3.10 V2 will include a strengthened emphasis on validation, performance assessment, and usability. Iterative testing cycles involving the Front Runner Cities will help evaluate the behavior of the models under real-world data conditions, while feedback from planners and practitioners will inform improvements in interface design, system interpretability, and integration with city dashboards. By M48, the deliverable will incorporate these refinements, providing a consolidated specification of AI-based tools, data management workflows, and operational mechanisms that are fully aligned with URBREATH's overarching objectives and prepared for long-term adoption.

The next phase of development will focus on expanding the analytical framework to incorporate additional features that may influence crime dynamics across Leuven. Preliminary results indicated limited correlation between crime levels and the presence of Nature-based Solutions (NbS), suggesting that a broader set of explanatory variables is required. As part of the roadmap toward the following version of the tool, we will enrich the model with supplementary spatial and socioeconomic factors, such as land-use patterns, population density, demographic composition, and urban mobility indicators, to improve predictive performance and enhance interpretability. A comprehensive statistical analysis will be conducted across all urban sections, and where appropriate, sections may be grouped according to criteria provided by the municipality or local stakeholders. This grouping will support more robust comparisons between areas with similar profiles and will help refine the identification of crime drivers. These enhancements will guide the development of a more accurate and actionable crime analytics tool for urban planning and community well-being in M48.

7 Conclusions

Deliverable D3.10 presents the first consolidated version of the URBREATH data management and AI framework, bringing together the architectural foundations, technical implementations, and functional capabilities developed across WP3 during the first half of the project. This initial release demonstrates significant progress in establishing a robust, scalable, and standards-based ecosystem capable of supporting advanced AI-driven analysis, data harmonization, and decision-support services for European cities.

The deliverable shows how the data acquisition, aggregation, scheduling, synchronization, and harmonization layer has matured into a coherent infrastructure that integrates heterogeneous data sources (from IoT networks and open data portals to geospatial repositories and environmental sensors) through interoperable standards such as NGSI-LD, SensorThings API, DCAT-AP, and OGC services. These components now provide the essential backbone on which AI models and analytical tools can operate reliably and consistently across multiple cities and domains.

On the AI side, D3.10 documents the first suite of AI-based tools implemented within WP3, including the infiltration prediction model, weather and seasonal forecasting components, land surface temperature downscaling, and the initial structure for numerical NBS models. These tools demonstrate the project's capacity to translate complex environmental and urban dynamics into actionable insights. Their integration with the VIE-AI explainability tool ensures transparency, interpretability, and user trust, aligning with the project's commitment to responsible AI.

The deliverable also outlines the initial steps toward deeper integration with WP4 and WP5, providing early mechanisms to connect AI outputs, harmonized datasets, and visualization dashboards. The ongoing development of the Simulator, workflow orchestration, and harmonized publication services further strengthens these cross-WP interactions and positions the project for more advanced scenario-based analysis in the next phase.

At the same time, this first version acknowledges the areas where further development is needed. Several models will undergo additional validation, refinement, and testing using new real-world datasets from the Front Runner Cities. The upcoming months will also bring new capabilities, such as EXUS's development of the Water Discharge and Flooding Prediction Model and early experimentation with agentic AI to enhance the usability of numerical models for NBS. These additions will expand the analytical depth of the URBREATH ecosystem while improving accessibility and decision support for municipal stakeholders.

Overall, D3.10 -V1 establishes a solid technical foundation that meets the project's current milestones and provides a clear path for continuous improvement. The next iteration (D3.10 -V2, M48) will consolidate the advancements made during the deployment and validation phases, integrate new models and functionalities, and further strengthen the interoperability and operational readiness of the URBREATH Toolbox. Through this progressive and collaborative process, WP3 will continue to contribute essential capabilities to support climate-neutral urban transformation and Nature-Based Solutions across the project's Living Labs and beyond.

8 References

[1] Abbott-Halpin, E. and Rankin, C., 2020. Introduction: Public Library Governance and Wicked Problems. In Public Library Governance (pp. 5-16). De Gruyter Saur.

[2] Armour-Gemmen, M.G., 2020. Innovation for the Engaged Librarian, American Society for Engineering Education, 2020, Faculty & Staff Scholarship. 2945. https://peer.asee.org/34831.pdf.

[3] Barras, R. (1990). Interactive innovation in financial and business services: the vanguard of the service revolution. Research Policy, 19, 215 - 237.

[4] Sit, M., Demiray, B. Z., Xiang, Z., Ewing, G. J., Sermet, Y., & Demir, I. (2020). A comprehensive review of deep learning applications in hydrology and water resources. *Water Science and Technology*, 82(12), 2635-2670.

[5] Li, J., & Heap, A. D. (2014). Spatial interpolation methods applied in the environmental sciences: A review. *Environmental Modelling & Software*, 53, 173-189.

[6] Saxton, K. E., & Rawls, W. J. (2006). Soil water characteristic estimates by texture and organic matter for hydrologic solutions. *Soil Science Society of America Journal*, 70(5), 1569-1578.

[7] Rawls, W. J., Brakensiek, D. L., & Miller, N. (1983). Green-Ampt infiltration parameters from soils data. *Journal of Hydraulic Engineering*, 109(1), 62-70.

[8] Oral, H. V., et al. (2020). A review of nature-based solutions for urban water management in European circular cities. *Blue-Green Systems*, 2(1), 112-136.

[9] Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.

[10] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD*, 1135-1144.

[11] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

[12] Hyndman, R.J., Athanasopoulos, G., 2018. Forecasting: principles and practice. OTexts.

[13] Liu, F.T., Ting, K.M. and Zhou, Z.H., 2008, December. Isolation forest. In 2008 eighth ieee international conference on data mining (pp. 413-422). IEEE.

[14] Chandola, V., Banerjee, A. and Kumar, V., 2009. Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3), pp.1-58.

[15] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.

[16] Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R. and Schmidhuber, J., 2016. LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, 28(10), pp.2222-2232.

9 Annexes

9.1 Annex A: CSW Connector

The following tables summarize how the CSW Connector extracts key fields from ISO 19139 metadata (from CSW 2.0.2) and maps them to the corresponding DCAT-AP (v 1.1) properties used by the portal.

DCAT-AP dataset mapping:

DCAT-AP Field	ISO 19139 Source	Notes
dct:identifier	gmd:fileIdentifier / gmd:identifier	-
dct:title	gmd:identificationInfo/.../gmd:title	Fallback: identifier
dct:description	gmd:abstract	-
dcat:keyword	gmd:descriptiveKeywords/.../gmd:keyword	Deduplicated strings
dct:spatial	EX_GeographicBoundingBox	-
dct:issued	CI_Date[dateType=publication]	Default if missing
dct:accessRights	MD_LegalConstraints	Mapped to short label
dcat:theme	topicCategory	If present

DCAT-AP Distribution mapping:

DCAT-AP Field	ISO 19139 Source	Notes
dcat:accessURL/ dcat:downloadURL	gmd:linkage/gmd:URL + function/protocol	keyword-based logic to decide which URLs are direct downloads and which are just access links
dct:title	gmd:name → description → protocol → host	Always non-empty
dct:format	URL extension / protocol / distributionFormat	Fallback: 'UNKNOWN'

9.2 Annex B: Waze/Orion entity mapping

Following tables summarize the mapping between Waze data and NGSI-LD entities “PointOfInterest” and “TrafficFlowObserved” to be published on Orion-LD Context Broker.

Waze JSONto PointOfInterest smart data model:

Waze Json	PointOfInterest entity
street	name
country city	address: addressCountry addressLocality
location	location

Waze JSON to TrafficFlowObserved smart data model:

Waze Json	TrafficFlowObserved
street	name
country city	address: addressCountry addressLocality
location	location
pubmillis	dateModified
speedKMH	averageVehicleSpeed
id	laneID