chist-era

AI Integrated Framework for Intelligent Geospatial Handling and Robust Operation in MultiGIS Applications (AI4MultiGIS)

Topic: Multidimensional Geographic Information Systems (MultiGIS)

AI4MultiGIS

D3.1

Deliverable D.3.1 Intelligent Real-time Data Pipeline

Work Package	WP3
Delivery Date	M12
Responsible Partner	LIUPPA
Authors	Fatima Chahal (LIUPPA) Kahina Alitouche (LIUPPA) Akram HAKIRI (LIUPPA) Richard Chbeir (LIUPPA)
Contributors	All partners
Distribution	*Project Level*
Version	1.0

a.r.u. Anglia Ruskin University

West University of Timişoara

LUT University

UNIVERSITÉ DE PAU ET DES PAYS DE L'ADOUR

Deliverable Factsheet

Project Acronym:	Ai4MultiGIS
Project Title:	AI integrated framework for intelligent geospatial handling and robust operation in MultiGIS applications
Call:	Chist-Era 2023
Start date:	02/01/2025
Duration:	36 Months

Deliverable Name:	D3.1
Related WP:	WP 3
Due Date:	M12

Editor:	Akram Hakiri
Contributor(s):	All partners
Reviewer(s):	ARU
Approved by:	All partners

Executive Summary

The Ai4MultiGIS project is designed to deliver an integrated framework for the generation and management of MultiGIS data, with the objective of enhancing GIS capabilities and optimizing the processing chain of MultiGIS applications.

This deliverable, D3.1, focuses on the development of an intelligent real-time data collection pipeline designed to create a smart ecosystem. This ecosystem automatically gathers large volumes of multi-modal data in real time by leveraging AI, IoT, and edge–cloud devices to optimize and automate data acquisition. AI-based preprocessing at the edge refines the collected data before its seamless integration into GIS databases, ensuring both efficiency and scalability.

Given the diversity of input formats, resolutions, and scales, robust data pipelines are employed to integrate, transform, and process multimodal datasets. The adoption of an event-driven architecture ensures that only relevant data are collected, while continuous feedback loops enable ongoing refinement. Overall, this deliverable contributes to the development of a scalable, adaptive, and intelligent framework that advances the state of real-time geospatial information systems, supporting WP3 by providing the core data infrastructure needed for the AI4MultiGIS project's pilots and analyses.

Document History

Version	Date	Author(s)	Comments
0.1	05/09/2025	Fatima Chahal	Table of Content (ToC) is added with main sections and related subsections
0.2	24/09/2025	Ankit Agrawal	Integration of Methods for Outlier Detection and Management
0.3	25/09/2025	Zhuoqian WU	Adding techniques related to blockchain and other forms of DLT relevant for MultiGIS
0.4	08/10/2025	Fatima Chahal	Integration of the design and Implementation of a Distributed Data Workflow
0.5		Akram Hakiri	Integration of EdgeAI and TinyML that will be considered for EdgeAI Processing and Data Preprocessing
0.6	04/11/2025	Fatima CHAHAL	Integration of Pilot 1 use case: SuDS
0.7	25/11/2025	Fatima CHAHAL	Integration Real-time Data Pipeline Architecture
0.8	28/11/2025	Kahina ALITOUCHE	Integration Pilot 2
0.9	22/01/2026	Fatima CHAHAL	Do the modifications based on the review comments
1.0	24/01/2026	Fatima CHAHAL	Final version

Table of Contents

List of Figures

Acronyms and Abbreviations

Abbreviation	Full Form
AI	Artificial Intelligence
AI4MultiGIS	Artificial Intelligence for Multi-source Geospatial Information Systems
DAG	Directed Acyclic Graph
DEM	Digital Elevation Model
GIS	Geographic Information System
CRS	Coordinate Reference System
IoT	Internet of Things
NiFi	Apache NiFi
JDBC	Java Database Connectivity
SQL	Structured Query Language
PostgreSQL	PostgreSQL Database Management System
PostGIS	Spatial extension of PostgreSQL
ETL	Extract, Transform, Load
API	Application Programming Interface
WP	Work Package
D3.1	Deliverable 3.1
SuDS	Sustainable Drainage Systems
SatImg	Satellite Imagery
OS	Ordnance Survey
DEM	Digital Elevation Model
CPU	Central Processing Unit
RAM	Random Access Memory
CSV	Comma-Separated Values
JSON	JavaScript Object Notation
XML	eXtensible Markup Language
UTC	Coordinated Universal Time
RDBMS	Relational Database Management System
JVM	Java Virtual Machine

1. Introduction

1.1. Purpose and Scope

The purpose of this deliverable is to present how intelligent real-time data collection contributes to the AI4MultiGIS project, by both defining its role and implementing the necessary mechanisms. It establishes the foundations for a scalable and adaptive ecosystem capable of capturing, processing, and integrating large volumes of multi-modal geospatial data. By leveraging AI, IoT, edge–cloud computing, and event-driven architectures, this deliverable develops a real-time data collection pipeline that automates data acquisition, improves efficiency, and enables continuous refinement through real-time feedback.

The scope of this deliverable encompasses:

- Identification of relevant data sources (e.g., remote sensing imagery, UAV and LiDAR data, IoT sensor networks, Open Data APIs, and user-generated content).
- Definition of data processing mechanisms at the edge and in the cloud, including AI-driven preprocessing, filtering, and integration into GIS platforms.
- Design of robust data pipelines to handle heterogeneous formats, scales, and resolutions for reliable spatial analysis.
- Specification of the event-driven architecture to ensure contextual relevance and adaptability in data collection.
- Application domains for this intelligent data collection are illustrated through the AI4MultiGIS pilot cases, including real-time monitoring of urban infrastructure, environmental data acquisition for ecosystem analysis, transportation network optimization, and emergency response management.

This deliverable therefore provides the conceptual and technical framework that underpins the development of intelligent, real-time, and adaptive MultiGIS systems within the broader objectives of the project.

1.2. Contribution to other Deliverables

Deliverable D3.1 is the first outcome of WP3 and represents the operationalization of the architecture and requirements defined in D2.2. While D2.2 specifies the overall system design and technical specifications, D3.1 translates these into a functional pipeline capable of ingesting, preprocessing, and integrating heterogeneous geospatial datasets in real time. This work establishes the technological foundation on which subsequent developments depend.

Within WP3, D3.1 supports the creation of synthetic data models (D3.2) by ensuring the integration of real and artificial datasets in a consistent framework. It also provides the preprocessed data streams necessary for outlier detection in D3.3, and offers the event-driven, traceable infrastructure that underpins blockchain and distributed ledger approaches in D3.4. Together, these links demonstrate how D3.1 enables reliable, scalable, and secure data management across the work package.

Beyond WP3, D3.1 plays a crucial role in WP4. The pipeline ensures that continuous and harmonized data streams are available for real-time spatiotemporal computing (D4.1) and for the development of geostatistical cloud-based AI models (D4.2). Moreover, it informs the API and middleware design in D4.3, which builds on the interfaces and streaming mechanisms introduced in this deliverable. Finally, it supports the plug-and-play integration framework in D4.4, guaranteeing that real-time data pipelines align with the overall AI4MultiGIS architecture.

1.3. Structure of the Document

The development of the AI4MultiGIS intelligent real-time data pipeline is organized into five interrelated work areas, each addressing a critical aspect of geospatial data acquisition, processing, integration, and governance. The document is organised to move from contextual and methodological foundations to concrete architectural choices, pilot instantiations, and perspectives for further developments. The structure of this deliverable is as follows:

- **Section 2** introduces the AI4MultiGIS project, its overall objectives, and targeted application domains. It highlights the role of intelligent, multi-modal data pipelines in the global architecture and summarises the main scientific and technical challenges addressed in this deliverable, such as heterogeneity, real-time processing, scalability, and privacy-aware analytics.
- **Section 3** presents the pilot use cases that drive the requirements for the pipeline. It describes Pilot Use Case 1 on Sustainable Drainage Systems (SuDS) and Pilot Use Case 2, detailing their study areas, stakeholders, and data needs. The section summarises the main vector and raster datasets, associated preprocessing and harmonisation workflows, and the key outcomes obtained so far for both pilots.
- **Section 4** details the design of the intelligent real-time data pipeline. It covers event-driven pipeline principles, ingestion and consolidation of multimodal datasets, workflow orchestration and automation, and scalable Edge–Cloud deployment strategies enabling robust and efficient geospatial analytics.
- **Section 5** addresses governance, trust, and privacy aspects associated with multi-source geospatial data processing. It presents principles and mechanisms for protecting client-specific and sensitive datasets, regulating access and sharing, and ensuring secure, compliant data flows within the pipeline.
- **Section 6** describes how the pipeline is monitored and adapted over time. It introduces real-time metrics and performance monitoring, fault tolerance and resilience mechanisms, elasticity and scalability capabilities, and feedback loops for continuous improvement, illustrated with early implementations on the pilot use cases.
- **Section 7** synthesises the main achievements of the deliverable regarding the conception and early deployment of the AI4MultiGIS intelligent data pipeline. It outlines planned extensions, including more advanced AI-based analytics, broader pilot coverage, and further reinforcement of governance, monitoring, and adaptation capabilities in future project phases.

2. Design and Implementation of a Distributed Data Workflow

2.1. System Overview

The developed workflow constitutes a modular, containerized architecture designed to automate the ingestion, processing, and storage of diverse geospatial datasets within the AI4MultiGIS project. The system integrates five core components **NiFi** (1), **Spark–Sedona** (2), **PostGIS** (3), **Airflow** (4), and **Jupyter Notebook** (5) all deployed through a unified **Docker Compose** (6) environment. Each component carries out a clearly defined role in the end-to-end data pipeline while ensuring interoperability through shared volumes and a dedicated internal network.At a high level, the workflow operates according to the following stages:

2.1.1. Data Ingestion

Apache **NiFi** is responsible for collecting datasets originating from heterogeneous sources. It transfers the acquired raw files into a shared directory accessible to the subsequent components. This flexible ingestion mechanism supports multiple input formats and data acquisition patterns, including scheduled downloads and streaming connectors (7).

2.1.2. Pre-processing

The ingested data is processed using **Apache Spark**, extended with the **Sedona** geospatial library. Spark-Sedona reads the raw weather files, performs data cleaning, and extracts relevant spatial attributes (e.g., longitude, latitude, temperature, metadata). The processed data is then exported as structured, analysis-ready files. This step enables scalable geospatial manipulation and ensures consistency across datasets (8).

2.1.3. Storage

The structured output is then persisted in **PostGIS**, a spatially enabled PostgreSQL database. PostGIS supports efficient storage, indexing, and querying of large geospatial datasets, enabling downstream spatial analytics and AI-based modelling within WP3 and across the broader AI4MultiGIS framework (9).

2.1.4. Orchestration

Apache Airflow provides workflow automation and operational supervision. It coordinates interactions between NiFi, Spark-Sedona, and PostGIS by defining task dependencies, scheduling data processing jobs, and ensuring robustness and fault tolerance during pipeline execution. Airflow thus guarantees reproducible and traceable data workflows (10).

2.1.5. Development and Monitoring

A **Jupyter Notebook** environment is included to support interactive development, debugging, and visualization. Researchers can inspect intermediate data products, test processing logic, and perform exploratory spatial analysis directly from within the same containerized infrastructure (11).

Figure 1 presents the functional workflow of the Ai4MultiGIS intelligent data pipeline. It highlights the sequence of tools responsible for data ingestion, preprocessing, format-specific transformations, quality validation, and final integration into the system. Each module operates autonomously within the

orchestration environment, ensuring consistency, reproducibility, and real-time processing of both raster and vector datasets.

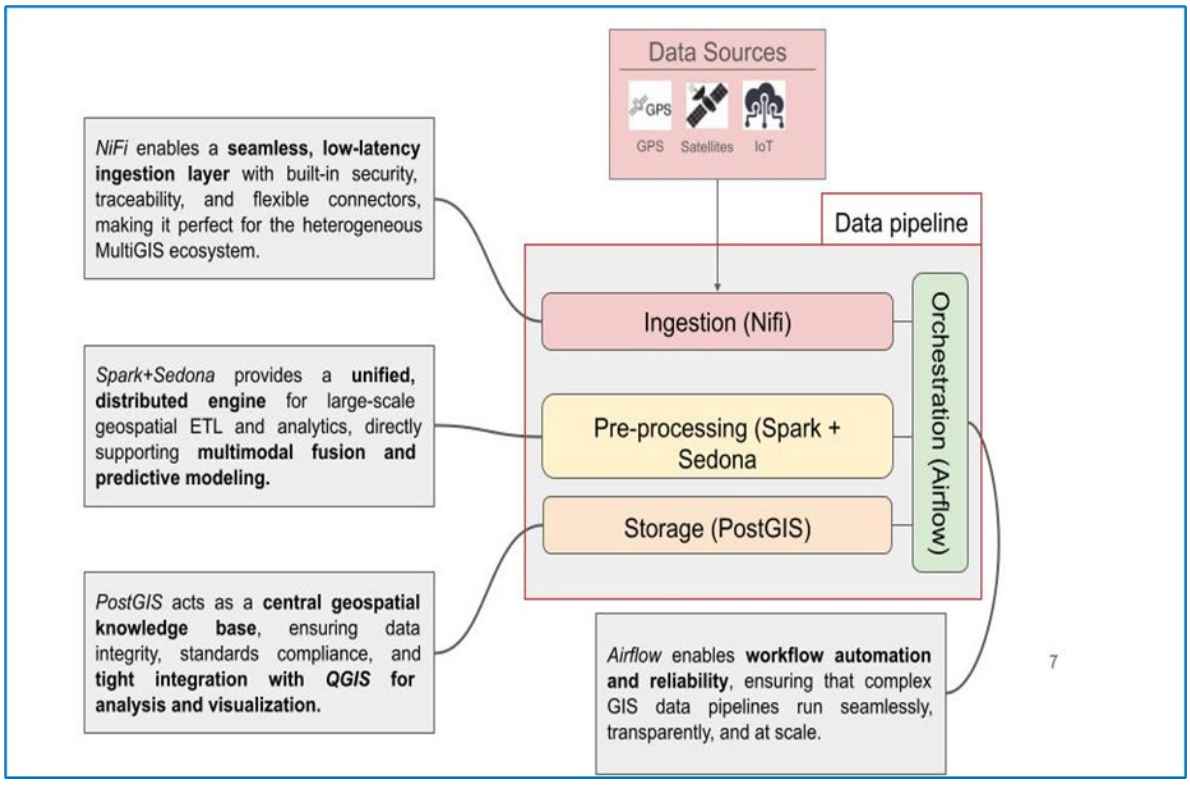

Figure 1: illustrating pipeline tools.

2.2. Containerized Deployment and Capabilities

The entire workflow is encapsulated within Docker containers, ensuring reproducibility, isolation of execution environments, and straightforward scalability. Shared directories, mounted as Docker volumes, allow seamless data exchange among services without relying on external storage or network dependencies. The workflow can be initialized or shut down through a single Docker Compose command, facilitating rapid deployment for experimentation, testing, or operational use.

In its current configuration, the workflow supports both batch and near-real-time geospatial data processing, providing a versatile foundation for producing harmonized datasets that serve as inputs for the AI-based analytical and predictive models of AI4MultiGIS. Its modular, containerized architecture also enables the integration of additional processing components, machine learning modules, or distributed data management layers developed later in WP3 (e.g., decentralized data governance mechanisms under T3.4), ensuring the workflow can evolve with project needs.

2.3. Technological Components

2.3.1. Apache NiFi

Apache NiFi is used as the dedicated ingestion layer of the workflow. It provides a configurable framework for collecting, routing, and transforming raw data from heterogeneous sources into the processing environment. In this project, NiFi serves as the entrypoint for all weather datasets in JSON format.

Using a set of processors (GetFile, InvokeHTTP, RouteOnAttribute, PutFile), data can be retrieved from APIs, local folders, or networked sources and deposited into the */data* shared directory. This directory is mounted as a Docker volume, ensuring accessibility by other services. NiFi's real-time monitoring, provenance tracking, and ease of configuration make it highly suitable for reproducible and traceable data ingestion workflows.

The illustration in Figure 2 presents an example of the data ingestion workflow realized with Apache NiFi. This diagram visually demonstrates how the tool is utilized to automate and manage the collection and real-time transport of data into the Ai4MultiGIS project pipeline.

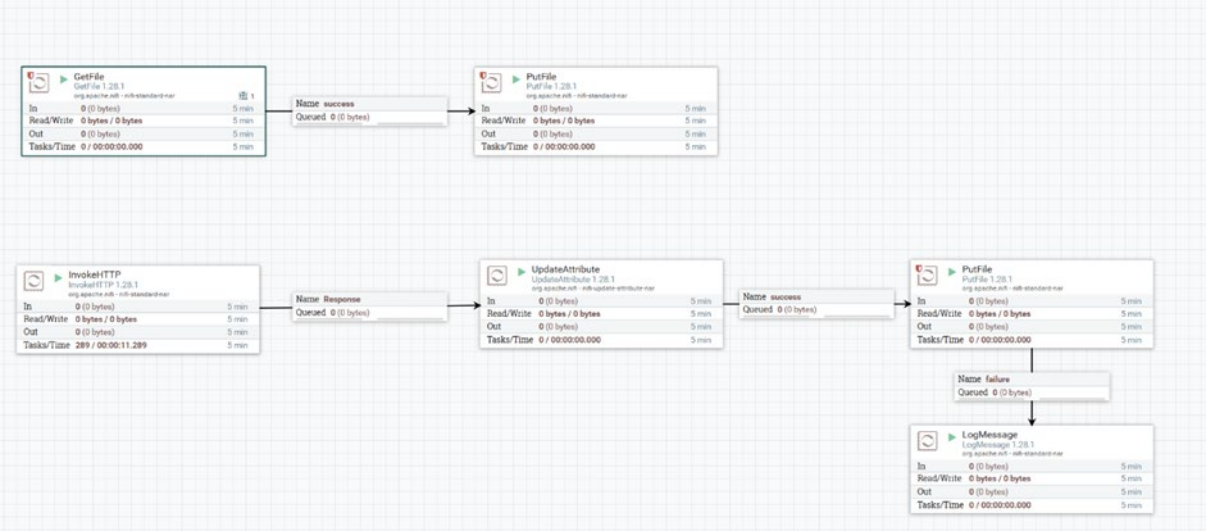

Figure 2: illustrating Nifi example.

2.3.2. Apache Spark with Sedona Extension

Apache Spark acts as the central processing engine, providing distributed computation capabilities for large-scale data. The integration of the Sedona extension enables Spark to natively handle geospatial operations.

The implemented Spark–Sedona workflow performs the following operations:

- Reads all JSON files generated by NiFi from the */data/processed/* directory.
- Extracts relevant spatial and climatic variables (longitude, latitude, ...).
- Normalizes and cleans the records, converting them into a structured Spark DataFrame.
- Exports harmonized outputs into CSV format for persistent storage.

This step ensures that all ingested data becomes consistent, structured, and ready for spatial indexing. Spark–Sedona can be further extended for advanced operations such as spatial joins, buffering, clustering, or converting data into different coordinate reference systems.

Figure 3 illustrates an example of how Apache Spark is used within the pipeline to execute distributed processing tasks on large geospatial datasets.

Figure 3: illustrating Spark example.

2.3.3. PostGIS

PostGIS is used as the spatially enabled storage backend of the workflow. Built on top of PostgreSQL, it allows efficient storage, indexing, and querying of georeferenced data. The processed outputs generated by Spark–Sedona are ingested into PostGIS using JDBC connectivity, ensuring a seamless transfer between the processing and storage layers.

The database schema includes fields describing the spatial coordinates and associated weather attributes. A geometry column is defined to represent each observation as a geospatial point, enabling spatial indexing and advanced geospatial querying. This structure makes it possible to perform proximity searches, spatial filtering, and integration with GIS tools or higher-level analytical modules in the AI4MultiGIS framework.

PostGIS thus provides a robust and scalable foundation for storing harmonized geospatial datasets and supports efficient retrieval for downstream analytics and modelling tasks.

2.3.4. Apache Airflow

Airflow provides orchestration and high-level supervision of the entire workflow. Each stage of the pipeline (ingestion, processing, loading) is represented as a task within a Directed Acyclic Graph (DAG).

In the current implementation, Airflow:

- Initializes its internal environment via a PostgreSQL metadata database.
- Schedules the execution of NiFi, Spark–Sedona, and PostGIS tasks.
- Automatically manages dependencies and retries.
- Offers a web-based UI for monitoring execution logs and task states.

This modular orchestration layer ensures reliability and makes it simple to integrate upcoming stages such as machine learning inference, validation pipelines, or distributed data publication systems.

Figure 4 presents an illustrative example of Apache Airflow usage for the orchestration of the Ai4MultiGIS data pipeline workflow. This diagram shows how Airflow ensures the scheduled execution and dependency management between the different process steps.

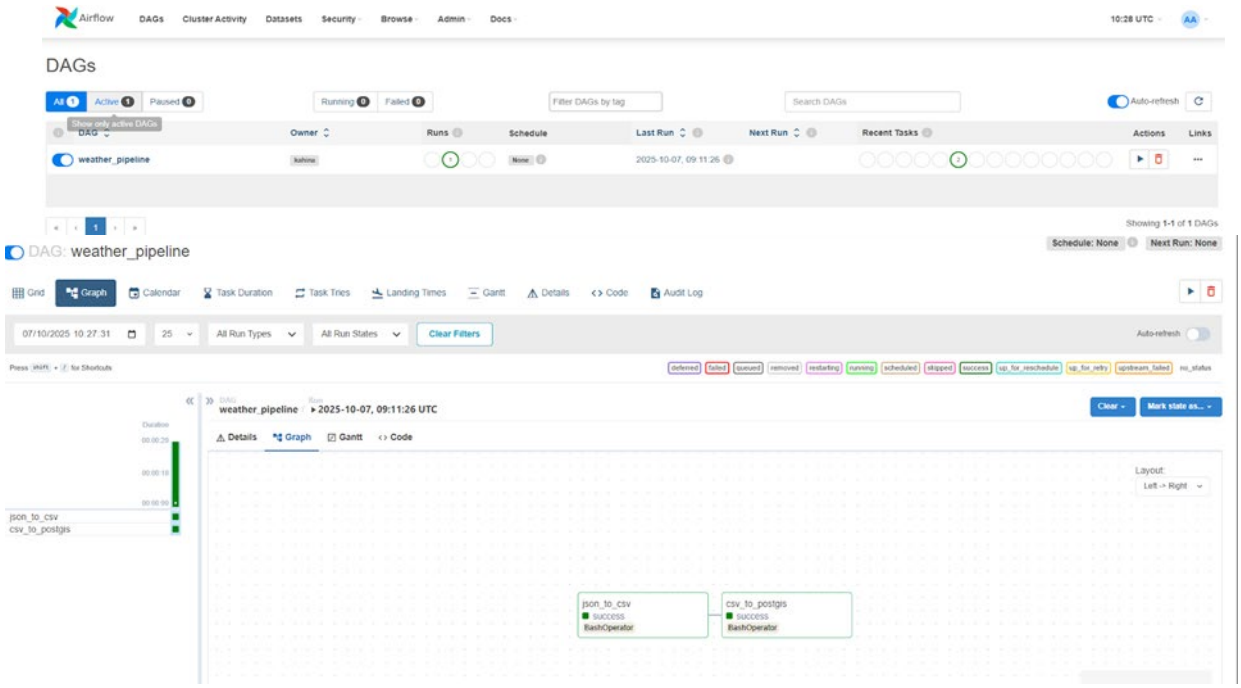

Figure 4: illustrating Airflow example.

2.5. Pipeline Implementation

The pipeline implementation describes how all components—NiFi, Spark–Sedona, PostGIS, Airflow, and Jupyter—are deployed and executed in an integrated workflow. While the previous section described the deployment and capabilities of each component, this section focuses on their concrete implementation within the distributed environment.

2.5.1. Docker Compose Infrastructure

The entire system is deployed using a Docker Compose configuration that specifies all services, their interdependencies, shared volumes, and networking rules. This approach ensures that each component of the Ai4MultiGIS pipeline operates within an isolated container while maintaining seamless interoperability with the rest of the architecture. The configuration includes several core services, such as the NiFi ingestion engine (exposed on port 8082), a distributed Spark cluster composed of master and worker nodes, a PostGIS database providing spatial storage and querying capabilities, and the Airflow components—webserver, scheduler, and initialization service—responsible for orchestrating the workflow. It also integrates a Jupyter Notebook environment used for development, monitoring, and exploratory validation. Shared volumes, including the ./data and ./app directories, enable efficient exchange of datasets, scripts, and configuration files across containers. Additionally, environment variables, persistent storage options, and port mappings are carefully configured to ensure stable communication between services and consistent runtime behavior. This containerized design guarantees reproducibility, simplifies maintenance, and supports future scaling of the system toward larger or distributed infrastructures.

2.5.2. Data Ingestion with NiFi

The ingestion phase is handled by NiFi, which collects raw JSON weather data and makes it available to the processing layer. The predefined dataflow monitors input directories or external data sources and routes incoming files to the shared volume /data/processed.

NiFi's flow-based interface allows flexible definition of ingestion logic—ranging from simple file collection to more advanced pipelines involving external APIs or streaming sources. This modularity ensures that additional data sources can be integrated into the workflow without structural changes.

2.5.3. Preprocessing with Spark–Sedona

The preprocessing stage is carried out using a Spark–Sedona script deployed within the Spark cluster, which ensures scalable and efficient handling of large geospatial datasets. This script begins by reading the JSON files that have been previously ingested and stored by NiFi in the shared directory. Once loaded, it systematically extracts the relevant variables, focusing on both geospatial coordinates and weather-related attributes, while standardizing the structure of the records to maintain consistency across the dataset. During this process, the data undergoes thorough cleaning and validation to remove inconsistencies, missing values, or any erroneous entries that could compromise downstream analysis. After the records are harmonized, the script exports the dataset into CSV format, producing a clean and structured output that is ready for spatial indexing. These output files are written to the /app/data/weather/ directory, ensuring that they are immediately accessible for subsequent loading into the spatial database, thus enabling seamless integration with the rest of the pipeline and supporting efficient query and analysis operations.

2.5.4 Loading Data into PostGIS

Processed records are integrated into PostGIS via Spark's JDBC connector, enabling seamless management of geospatial datasets. The workflow automatically structures tables and updates data continuously, while coordinate values are transformed into spatial geometries to support indexing and efficient geospatial queries. This design allows downstream modules to access data based on spatial relationships—such as proximity or boundaries—facilitating real-time analytics and informed decision-making within the AI4MultiGIS project.

2.5.5. Workflow Orchestration with Airflow

Airflow orchestrates the pipeline by defining dependencies between tasks in a Directed Acyclic Graph (DAG), ensuring that data ingestion, preprocessing, and storage are executed in the correct sequence. This orchestration guarantees data consistency, minimizes operational errors, and enables continuous, reliable execution of the workflow. Additionally, Airflow's monitoring and traceability features provide insights into pipeline performance, supporting maintainability and robustness for advanced geospatial analytics and AI-driven processes across AI4MultiGIS.

2.6. Data Management

This section describes how the system handles, structures, and stores geospatial data within the MultiGIS pipeline. The workflow is designed to support both **raster** and **vector** datasets, which were the two pilot use cases used to test and validate the pipeline.

Figure 5 illustrates the overall data management scheme within the distributed workflow of deliverable D3.1. This diagram represents the unified architecture for handling vector and raster data, from their ingestion to their storage within the PostGIS environment, ensuring the traceability and accessibility of the harmonized datasets.

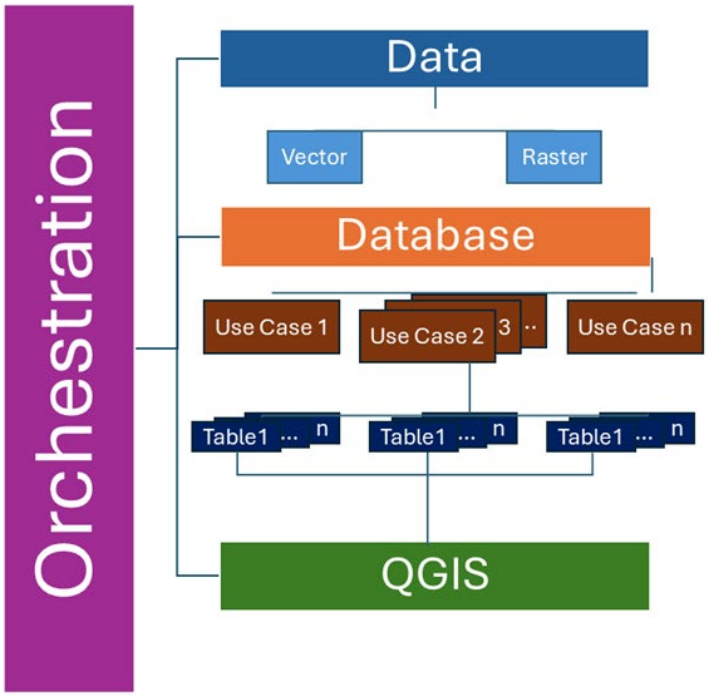

Figure 5: illustrating data management.

2.6.1. Data Formats and Structure

The pipeline is designed to ingest heterogeneous geospatial datasets and transform them into standardized formats suitable for downstream processing and analysis. Vector data—such as points, lines, and polygons—are parsed to extract both their geometric components and the associated attribute information. During preprocessing, each geometry undergoes a series of validation checks to ensure integrity, is reprojected into an appropriate coordinate reference system, and is converted into a structured tabular format that aligns with the requirements of spatial database storage. Raster datasets, including gridded weather surfaces or environmental layers, are processed to retrieve essential spatial metadata, pixel values, bounding coordinates, and any additional descriptors required for efficient spatial indexing and querying. Despite their differences, both vector and raster inputs pass through a harmonization phase that aims to guarantee spatial coherence, metadata completeness, and overall compatibility with the PostGIS environment. This unified preprocessing strategy ensures that all ingested datasets conform to consistent standards, thereby enabling reliable integration and supporting further analytical operations within the Ai4MultiGIS framework.

2.6.2. Automated Detection and Preprocessing

Using workflow orchestration tools, the system automatically determines whether the ingested dataset is raster or vector and triggers the corresponding preprocessing script based on its type. Once the appropriate branch is selected, the system applies the required format-specific transformations. For vector data, this includes geometry extraction, structural verification, and validation to ensure consistency and spatial correctness. For raster data, the system performs metadata extraction, reprojection, tiling, or other spatial adjustments when necessary. Through this automated branching mechanism, each dataset follows the correct preparation workflow before entering the storage layer, ensuring accuracy, coherence, and proper integration into subsequent processing stages.

2.6.3. Data Storage and Accessibility

All datasets, regardless of their format or origin, are ultimately managed within a unified spatial environment to ensure seamless access and efficient querying. Vector datasets are ingested into dedicated PostGIS tables, where their geometry fields are enriched with spatial indexes to optimize common geospatial operations such as proximity analysis, spatial filtering, and spatial joins. Raster datasets follow a complementary approach: when required by the use case, they are stored directly within PostGIS using its native raster support; otherwise, they remain as raster files within shared storage volumes. In both scenarios, relevant metadata—such as spatial extent, resolution, coordinate reference system, and file location—is recorded in PostGIS tables to facilitate rapid search, cataloguing, and retrieval. This hybrid storage strategy combines the performance benefits of spatially indexed database structures with the flexibility of accessing original source files when needed, creating a versatile and efficient data management environment within the Ai4MultiGIS framework.

2.6.4. Traceability and Data Lineage

The pipeline is organized into distinct phases **data ingestion, preprocessing, and workflow orchestration** each serving a specific purpose. The **ingestion phase**, managed by NiFi, collects and streams diverse geospatial datasets from multiple sources in real time. The **preprocessing phase**, carried out by Spark, cleans, validates, and transforms the data to ensure consistency, quality, and readiness for analysis. Finally, **Airflow orchestrates the workflow**, coordinating tasks and dependencies to maintain correct execution and data flow.

At each phase, metadata is generated to enable complete traceability of datasets from their initial ingestion to their final storage. NiFi records provenance information such as the original data source, timestamps, file characteristics, and ingestion events. Spark logs every transformation, schema validation result, and quality-control outcome. Airflow captures execution metadata for each task, ensuring reproducibility and providing a transparent view of the pipeline's operations. Together, these layers of structured provenance create a comprehensive audit trail, allowing any dataset to be traced from its origin, through intermediate processing, to its final representation essential for quality assurance, debugging, and long-term maintainability within the AI4MultiGIS framework.

2.6.5. Data Quality and Validation

Before datasets are loaded into PostGIS, a series of automated validation checks are performed to ensure that only complete and reliable information is integrated into the system. For vector data, geometric features are examined for validity and internal consistency, preventing issues such as self-intersections or malformed shapes. Raster datasets undergo verification of their metadata, including projection information, spatial extent, and resolution, to ensure they conform to the expected standards. Across both data types, the pipeline checks for the presence of mandatory fields and attributes required for downstream operations. Any records that fail these conditions, whether due to missing information or structural inconsistencies, are automatically flagged for further inspection and manual review. These validation mechanisms safeguard the integrity of the geospatial layers stored in PostGIS and establish a solid foundation for reliable spatial analysis and subsequent AI-driven workflows within Ai4MultiGIS.

3. Use Cases

3.1. Pilot Use Case 1: Sustainable Drainage Systems (SuDS)

3.1.1. Introduction

This use case demonstrates the application of the data collection pipeline developed in Deliverable D3.1. Its objective is to integrate heterogeneous geospatial datasets to support flood-risk assessment and Sustainable Drainage Systems (SuDS) analysis. By implementing the pipeline, the project builds a unified, standardized, and query-ready geospatial database that consolidates information from multiple public sources. This database enables spatial analytics, visualization, and AI-driven modeling, illustrating how the pipeline facilitates the automated ingestion, preprocessing, and integration of multi-source GIS data within AI4MultiGIS.

This pilot initiative is strategically designed around the utilization of two fundamental categories of geospatial data, each serving a distinct purpose in environmental and infrastructure analysis:

1. **Vector Datasets**: These datasets are inherently structured to describe discrete, clearly defined geographical features. They are composed of points, lines, and polygons that model real-world objects with specific boundaries and locations. Key examples relevant to this pilot include River Networks (Detailed linear features mapping the hydrographic system, crucial for flow modeling and flood prediction), Road Infrastructures (Polygonal and linear features representing transportation networks), Drainage Areas (Catchments) (Polygonal boundaries defining the area of land where all surface water converges to a single point), and Flood-Risk Polygons (Areas delineated based on historical data and hydraulic modeling that indicate varying levels of flood hazard, foundational for risk assessment and urban planning).
2. **Raster Datasets**: In contrast to vector data, raster datasets represent continuous spatial surfaces. They are structured as a grid of cells (pixels), with each cell containing a value that represents a specific attribute of the area it covers. This format is ideal for representing continuously varying phenomena. Relevant examples for this pilot are Satellite Imagery (High-resolution optical and synthetic aperture radar (SAR) images providing visual and spectral information about the Earth's surface, critical for monitoring land cover change and event damage), Terrain Elevation (Digital Elevation Models - DEMs) (Grids where the cell value represents the altitude above a reference datum, indispensable for slope analysis, flow direction modeling, and hydraulic simulations), Drainage Capacity Layers (Spatially explicit continuous layers that might represent the permeability or infiltration rate of the ground, influencing runoff and flood intensity), and Land-Use Layers (Gridded maps classifying the surface into various categories (e.g., forest, urban, agricultural).

Pilot 1 therefore tests the pipeline’s ability to ingest, preprocess, harmonize, and store multi-format, multi-source, and multi-resolution datasets within a single, coherent spatial database.

3.1.2. Data format and structure

Pilot 1 combines a diverse set of geospatial file formats. These include:

3.1.2.1. Vector formats

The pipeline supports a wide range of geospatial formats commonly used in environmental and infrastructural analyses. Many datasets are provided as Shapefile bundles, consisting of the core components—.shp, .shx, .dbf, and .prj—alongside provider-specific auxiliary files such as .cst, .shb, or .qpj. These Shapefile collections are frequently employed to represent flood-risk polygons, river network geometries, or detailed road topologies. Other inputs arrive as GeoPackage (.gpkg) files, which may contain both vector and raster layers within a single container, offering a compact and versatile alternative. Some administrative or thematic datasets are delivered in GeoJSON (.geojson) format, particularly when lightweight web-friendly structures are preferred. In addition to spatial layers, the system ingests a variety of metadata files, including JSON configuration documents that describe the characteristics of the input sources, specify preprocessing parameters, and define any dataset-specific rules required during ingestion. This diversity of formats reflects the heterogeneity of the data sources addressed by Ai4MultiGIS and underscores the need for a robust and flexible preprocessing pipeline.

3.1.2.2. Raster formats

Raster datasets ingested by the pipeline are typically provided in GeoTIFF (.tif) format, often accompanied by auxiliary files that supply additional spatial or descriptive information. These may include world files (.tfw) that define georeferencing parameters, .aux files containing extended metadata, or compressed archives bundling multi-band imagery required for spectral or environmental analyses. The raster inputs encompass a diverse set of geospatial products, ranging from satellite imagery and digital elevation models (DEMs) to land-use classifications and continuous flood-related surfaces. Their varying resolutions, band structures, and metadata profiles highlight the importance of a preprocessing workflow capable of handling heterogeneous raster sources while preserving the spatial fidelity necessary for downstream geospatial analytics within Ai4MultiGIS**.**

Compared with Pilot 2, which uses more uniform formats (GeoJSON, Excel, and vector rasters), Pilot 1 required **dataset-specific handling** due to the diversity and heterogeneity of its sources.

3.1.3. General Preprocessing Workflow

To address this heterogeneity, the preprocessing workflow was designed around a **generic logic** for vector and raster data, while keeping flexibility through dedicated scripts for each dataset.

3.1.3.1. Vector Data Workflow (General)

All vector datasets follow the same ETL pattern:

- ***Extraction***

During the extraction phase, vector files are loaded using GeoPandas or GDAL drivers, allowing the system to access both geometries and associated attributes. Once the data are ingested, several integrity checks are performed, including verifying geometry validity, assessing attribute completeness, and ensuring the presence of a defined Coordinate Reference System. These steps guarantee proper visual alignment and attribute integrity before the data progress to subsequent processing stages.

- ***Transformation***

The transformation phase ensures that all vector datasets adhere to a consistent and interoperable structure. Each dataset is first reprojected to a unified coordinate reference system, specifically EPSG:4326, to guarantee cross-dataset compatibility. The geometries are then normalized by converting them into standard MultiPolygon or MultiLineString formats and by removing unnecessary Z or M dimensions. Attribute fields are also standardized through the removal of unused columns, the renaming of relevant fields, and the cleaning or harmonization of attribute values. When geometric inconsistencies are detected, the system applies automated repair procedures, such as the use of buffer(0) operations or GDAL-based correction tools, to ensure topological validity before the data progresses to subsequent processing stages.

- ***Loading***

During the loading phase, the data are converted to WKT or WKB formats and inserted into the corresponding PostGIS tables. Each spatial field is stored using the expression `ST_Multi(ST_Force2D(ST_GeomFromText(..., 4326)))`, ensuring that geometries are consistently represented as two-dimensional multi-geometries in the standard EPSG:4326 coordinate system. Once the data are inserted, GIST spatial indexes are created to optimize spatial querying and significantly improve performance during subsequent analysis and retrieval operations.

- ***Validation***

During the validation phase, detailed logs capture all processing steps, including geometry counts, coordinate reference system consistency, and any warnings generated throughout the workflow. To ensure spatial and semantic accuracy, the resulting layers are then inspected in QGIS, where they are visually checked for proper alignment, coherence, and attribute integrity. This combined logging and visual validation approach guarantees that the processed datasets are reliable and ready for integration into the system.

The spatial overlay shown in Figure 6 captures the vulnerability of the road network: the 'Flood Risk from Surface Water' layer is combined with the 'Road Network' layer. This provides a clear visual assessment to prioritize areas for Sustainable Drainage Systems (SuDS) intervention.

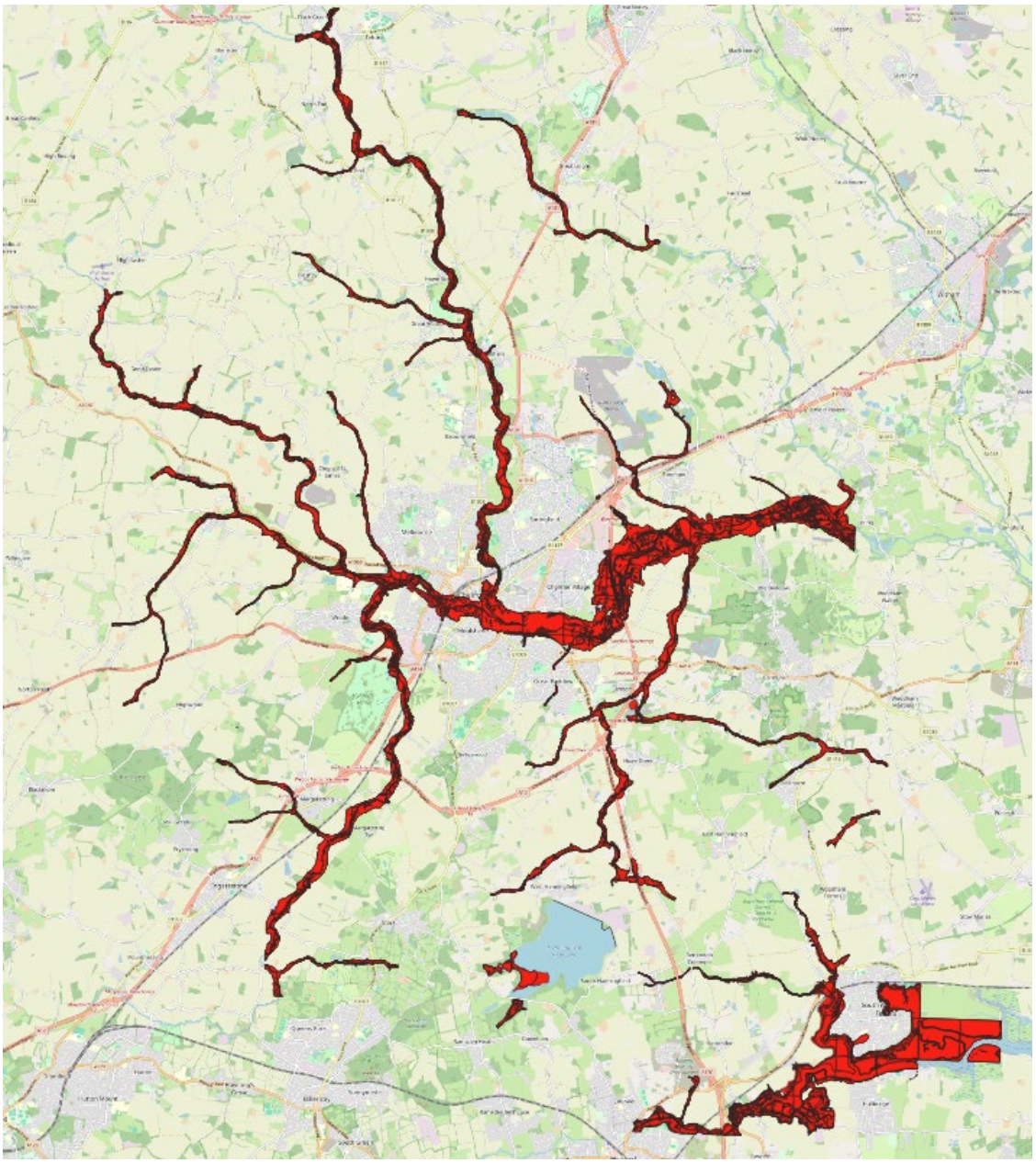

Figure 6: illustrates the result of the spatial integration stage of the D3.1 pipeline, where heterogeneous datasets are harmonized and combined for analysis. The overlay of the "Flood Risk from Surface Water" layer with the "Road Network" highlights road segments exposed to surface water flooding. This visualization demonstrates how the implemented pipeline enables spatial joins and risk-aware querying, supporting decision-making for prioritizing road sections requiring Sustainable Drainage Systems (SuDS) interventions.

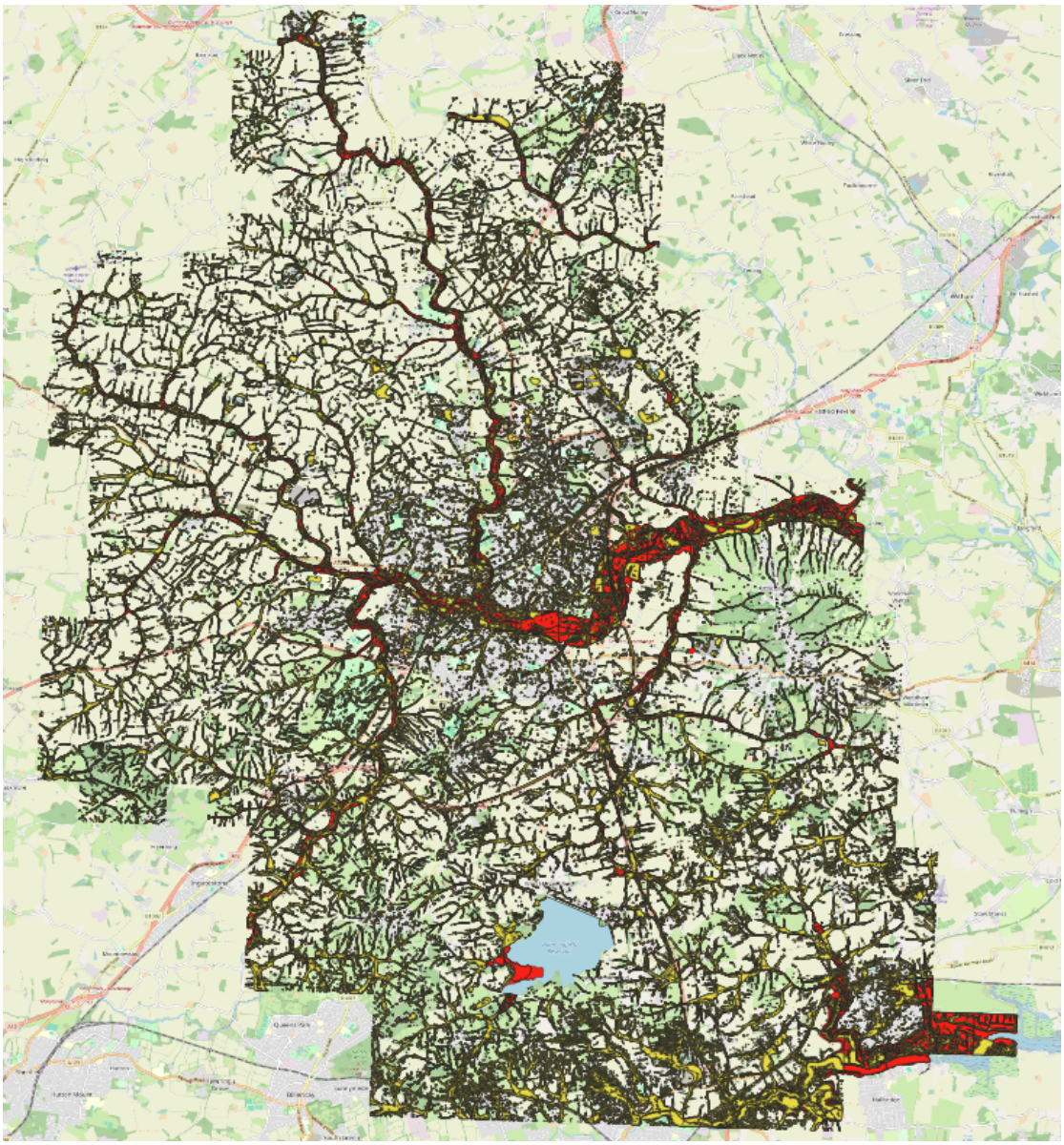

Fig 7: presents the integration of climate-related geospatial data with transportation infrastructure. By overlaying the "Climate Change" layer with the "Road Network," the pipeline reveals road segments potentially sensitive to long-term climate impacts. This figure exemplifies how the D3.1 workflow supports the incorporation of thematic environmental layers into a unified geospatial database, enabling planners to identify priority areas for adaptation measures such as SuDS.

The spatial overlay of the 'Climate Changer' layer with the 'Road Network' in Figure 7 visually represents the road segments potentially sensitive to the effects of climate change. This visual representation can inform the selection of priority areas for Sustainable Drainage Systems (SuDS) intervention.

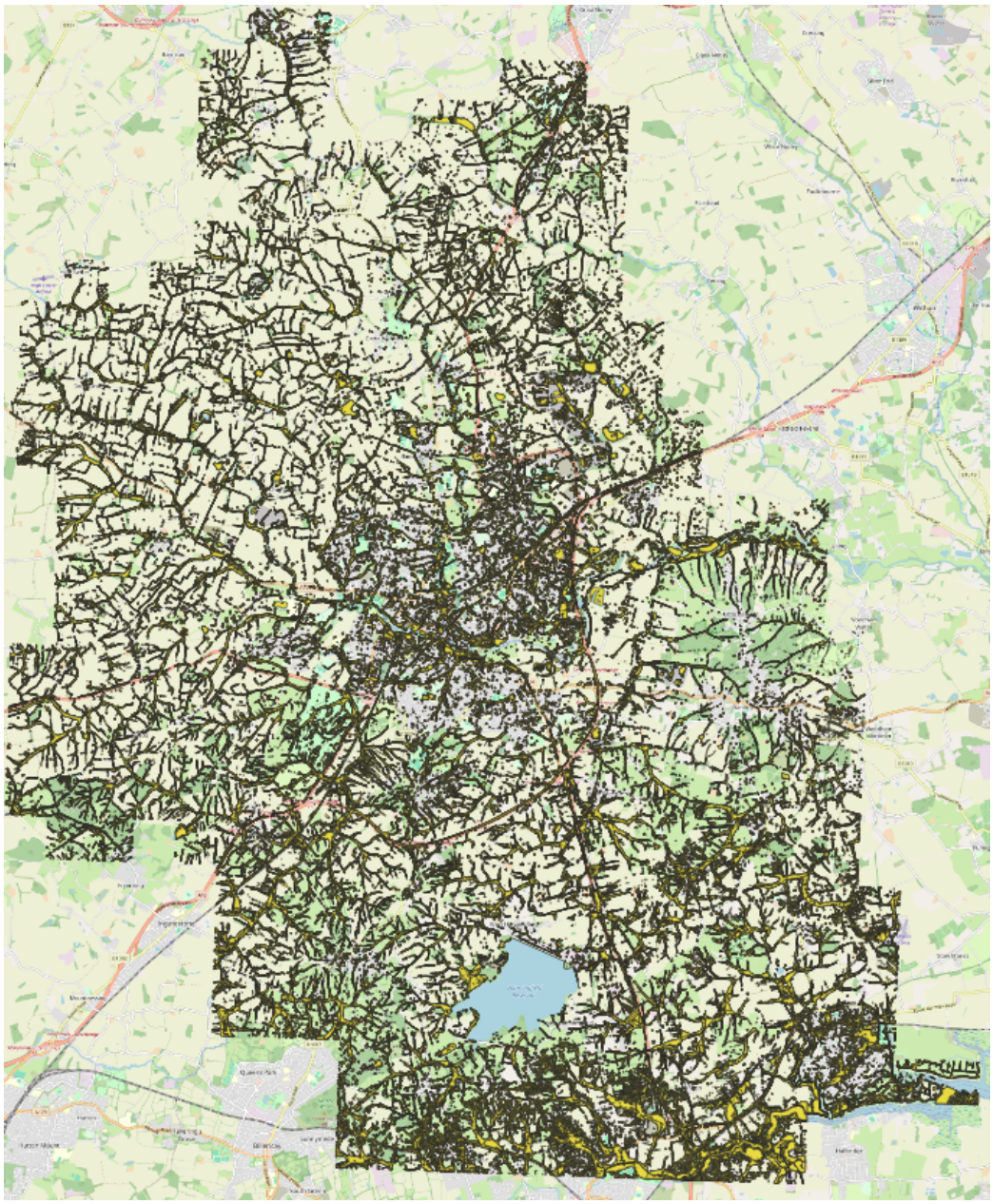

Fig 8: shows a combined risk perspective produced by the pipeline, where surface water flooding data influenced by climate change scenarios is overlaid with the road network. This visualization highlights locations where multiple vulnerabilities intersect, demonstrating the pipeline's capability to support multi-layer spatial analysis. Such combined views are essential for advanced risk assessment and illustrate the added value of automated data integration within AI4MultiGIS.

In Figure 8, the overlay of the 'Surface Water - Climate Changer' and 'Road Network' layers provides a combined view of surface water and climate change risk areas, highlighting the points where these vulnerabilities intersect along the road network.

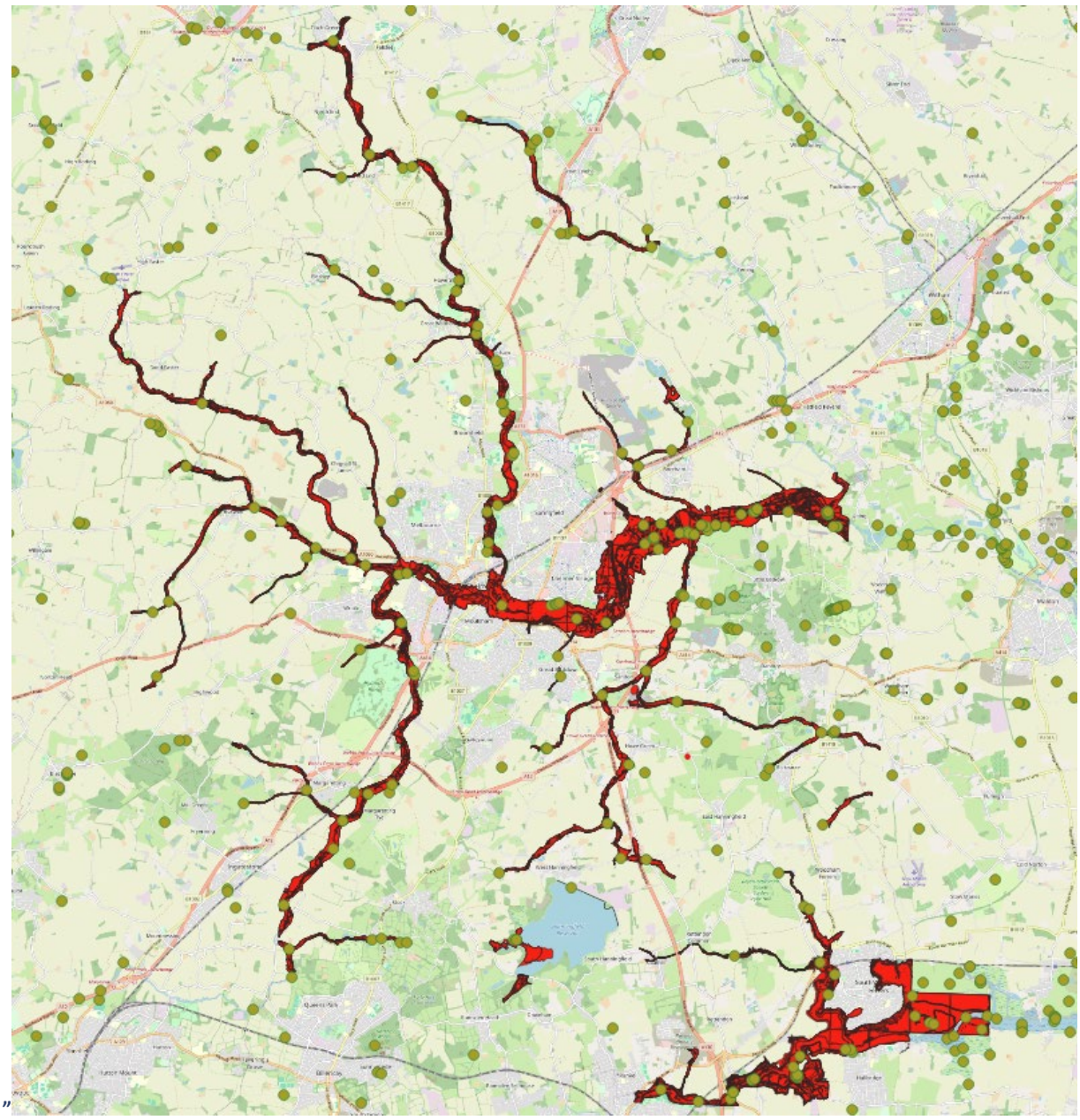

Fig 9: illustrates a key preprocessing step of the D3.1 pipeline, focusing on data harmonization and interoperability. The "OS Open Rivers" layer is visualized alongside flood-risk polygons, emphasizing the alignment of Coordinate Reference Systems (CRS) prior to spatial analysis. This figure directly relates to the pipeline's preprocessing phase, where data consistency and spatial correctness are ensured before integration into the geospatial database.

In Figure 9, the illustration shows the preparation of vector data for analysis, showcasing the 'OS Open Rivers' geographical layer. This layer is visualized together with informational polygons that delineate flood risk areas. The illustration also highlights the Coordinate Reference System (CRS) alignment

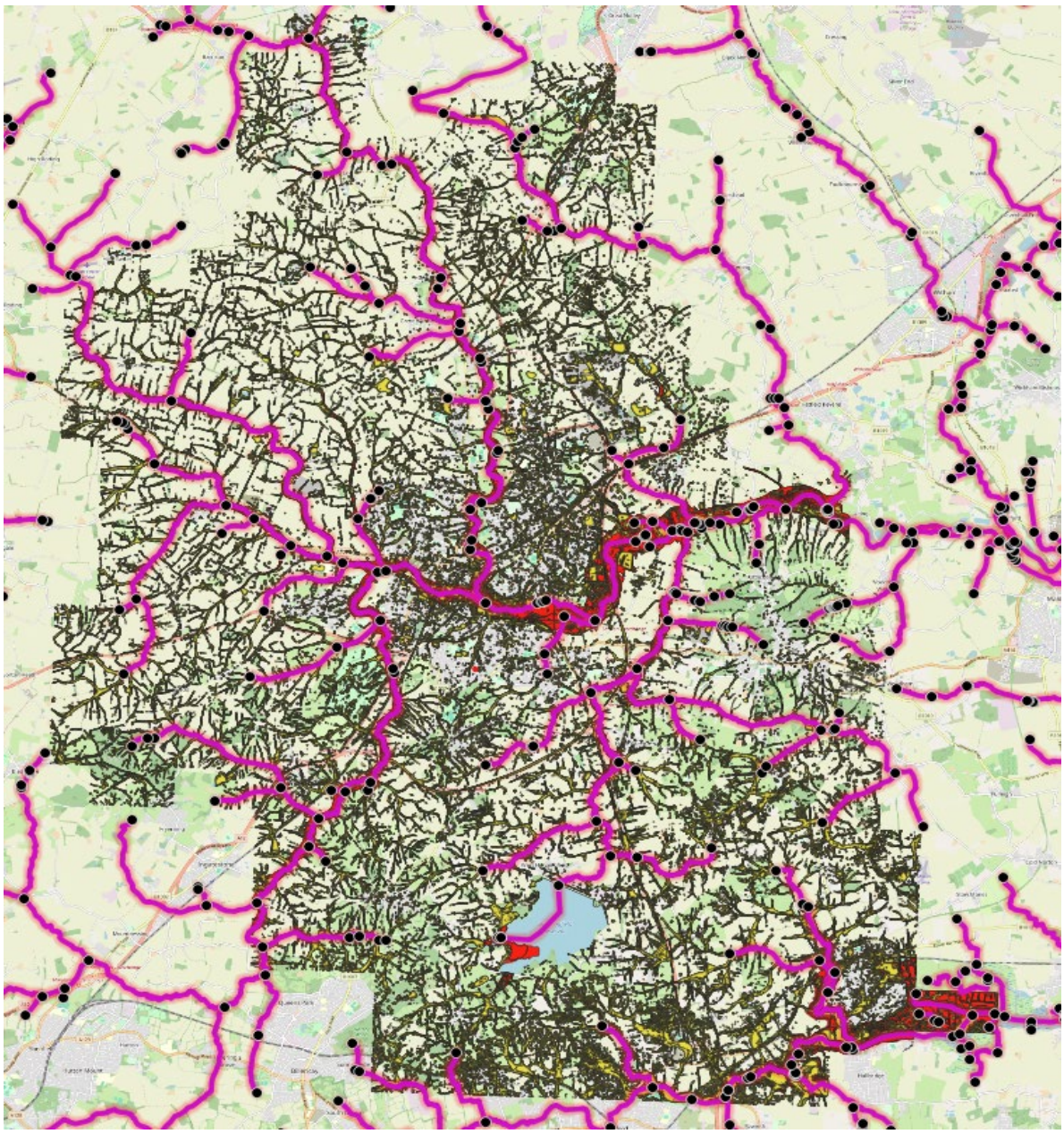

Fig 10: presents the final integrated view generated by the pipeline, combining flood-risk data with the road network. This overlay exemplifies the end-to-end functionality of the D3.1 deliverable from ingestion and preprocessing to spatial integration and visualization. It demonstrates how the resulting query-ready datasets can support operational analyses, risk evaluation, and AI-driven modeling within the AI4MultiGIS pilot use case.

This workflow ensures that all vector layers—regardless of their original provider or CRS—enter PostGIS in a harmonized form.

3.1.3.2. Raster Data Workflow (General)

- Raster datasets follow a similar but raster-specific workflow:

- ***Extraction***
  Raster files, typically provided in GeoTIFF format, are ingested and processed using the rasterio and GDAL libraries, which offer robust support for geospatial raster data. During this stage, essential

metadata is extracted from each file, including the spatial resolution, geographic extent, number of bands, and the coordinate reference system (CRS). This information is crucial for ensuring accurate alignment with other datasets, supporting subsequent analyses, and maintaining consistency across the spatial database.

- ***Transformation***

  Once the raster files and their metadata have been loaded, the data undergoes a series of preprocessing steps to ensure consistency and usability. First, all rasters are reprojected to the EPSG:4326 coordinate reference system, providing a standardized spatial framework compatible with other datasets. Next, the rasters are clipped to the SuDS study area, which not only reduces the data volume but also ensures spatial coherence within the area of interest. When multiple raster surfaces need to be combined, they are resampled to a unified resolution, facilitating accurate overlay and analysis. Finally, the associated metadata is normalized, enabling thorough documentation, efficient indexing, and reproducibility of the preprocessing workflow across different datasets and future analyses.

- ***Integration & Storage***

  After preprocessing, the raster datasets are saved in the project's designated processed directory, ensuring organized storage and easy access for subsequent operations. At the same time, the extracted metadata is registered in PostGIS tables, providing a comprehensive catalog that supports efficient retrieval and management of the raster files. This setup allows QGIS and other GIS tools to dynamically link the raster datasets with vector layers stored in PostGIS, enabling integrated spatial analyses and visualization within a unified geospatial framework.

- ***Validation***

  To ensure the accuracy and integrity of the spatial data, a **visual inspection is performed in QGIS**, verifying that the raster and vector layers are properly aligned and share the same coordinate reference system. This step provides a final quality check, confirming that the preprocessing and registration processes have been correctly applied and that the datasets are ready for further spatial analysis and integration within the project workflow.

Figure 11 illustrates the integration of geospatial datasets by overlaying the **satellite imagery raster (SatImg)** with the **road and river network layers**. The visualization highlights the spatial relationships between natural and man-made features, allowing for a clear assessment of connectivity, terrain interaction, and potential areas of interest. By combining raster and vector layers in a single view, this figure demonstrates the effectiveness of the preprocessing and registration pipeline in producing accurately aligned, ready-to-analyze geospatial data.

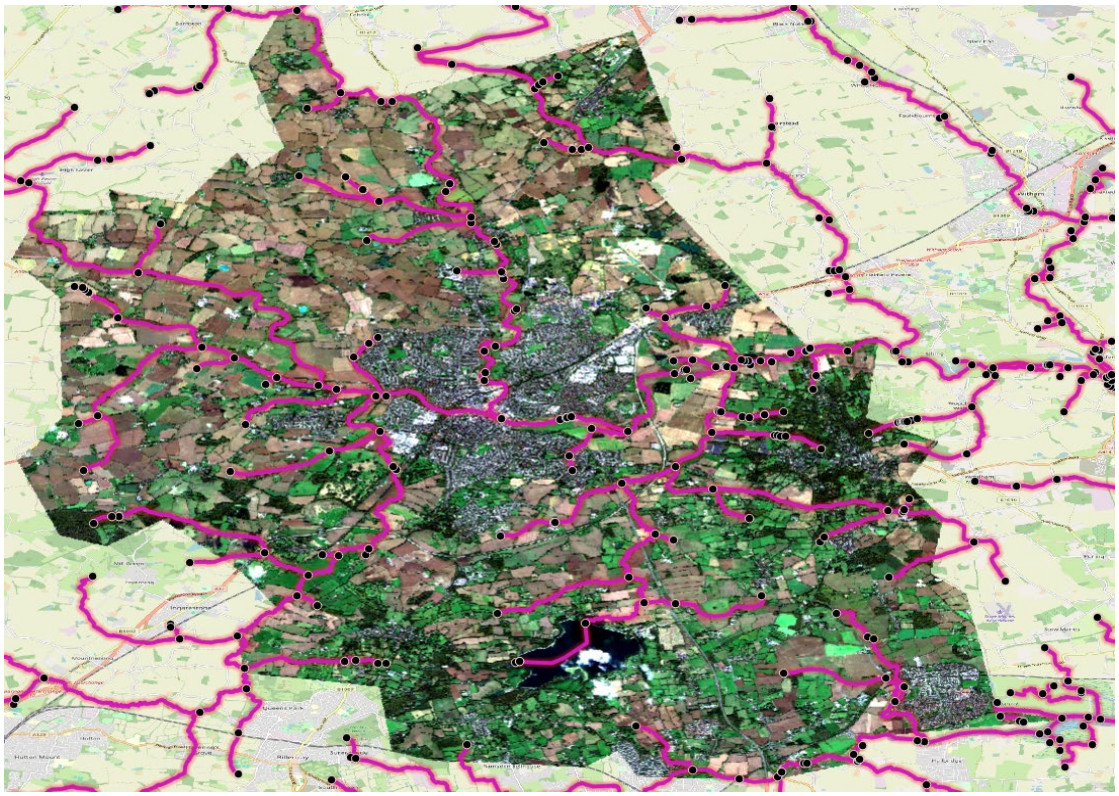

Fig 11: illustrates the application of the D3.1 data integration pipeline to heterogeneous raster and vector datasets. The satellite imagery raster (SatImg) is overlaid with road and river network layers after preprocessing, spatial registration, and alignment. This visualization highlights the spatial relationships between natural and man-made features and demonstrates the pipeline's ability to harmonize multi-format geospatial data into a consistent, analysis-ready representation within the AI4MultiGIS framework.

Figure 12 presents the **satellite imagery raster (SatImg)** overlaid with **the road and river network layers** alongside **climate risk data**. This visualization enables a comprehensive spatial analysis by combining natural features, infrastructure, and risk factors within the same geographic context. It illustrates how the integrated preprocessing and spatial indexing workflow allows for accurate alignment of multiple datasets, facilitating the assessment of potential vulnerabilities and supporting informed decision-making in environmental and urban planning applications.

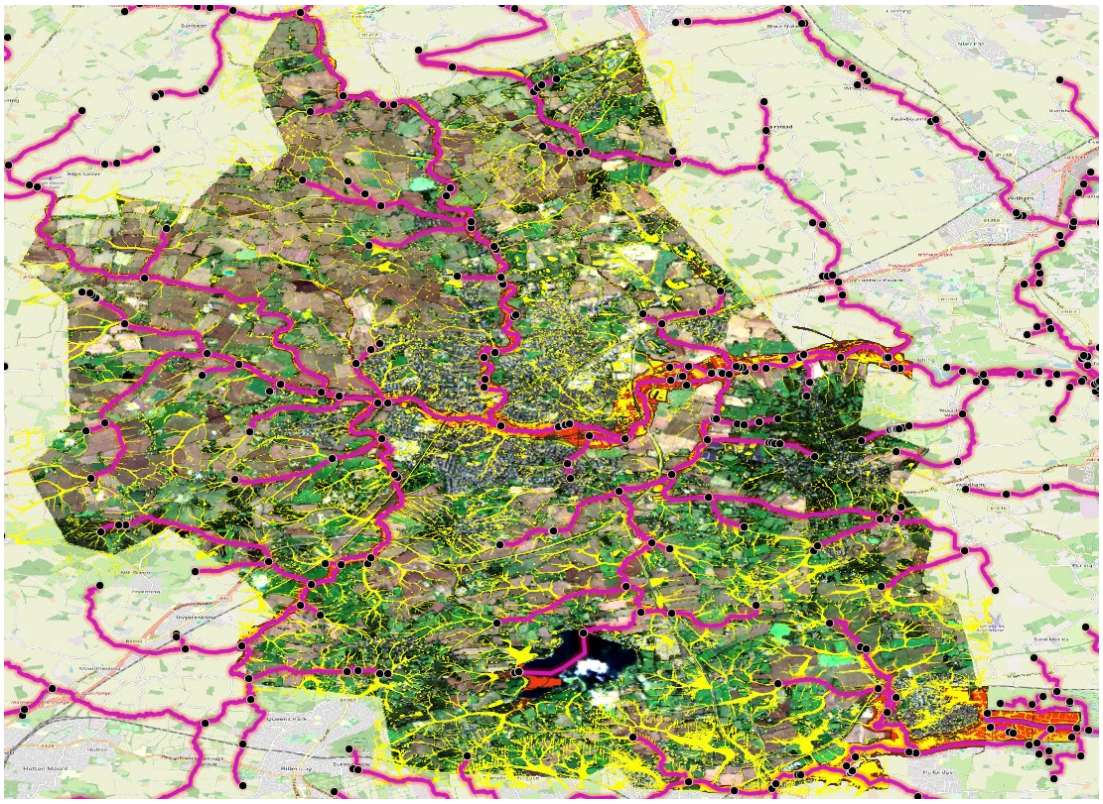

Fig 12: extends the integration process by incorporating climate risk data alongside satellite imagery and infrastructure layers. The overlay illustrates how the pipeline developed in Deliverable D3.1 enables the joint analysis of environmental factors, infrastructure, and risk indicators within a unified spatial context. This figure demonstrates the value of the integrated preprocessing, alignment, and indexing workflow in supporting vulnerability assessment and decision-making for environmental monitoring and urban planning use cases.

Figure 13 illustrates the **Digital Elevation Model (DEM)** raster overlaid with the **open river network**. This visualization **highlights the topographic features** of the study area in relation to its hydrological network, allowing for an assessment of elevation patterns, flow paths, and potential flood-prone zones. The figure demonstrates the effectiveness of the preprocessing and spatial integration workflow in producing accurately aligned raster and vector datasets for comprehensive geospatial analysis.

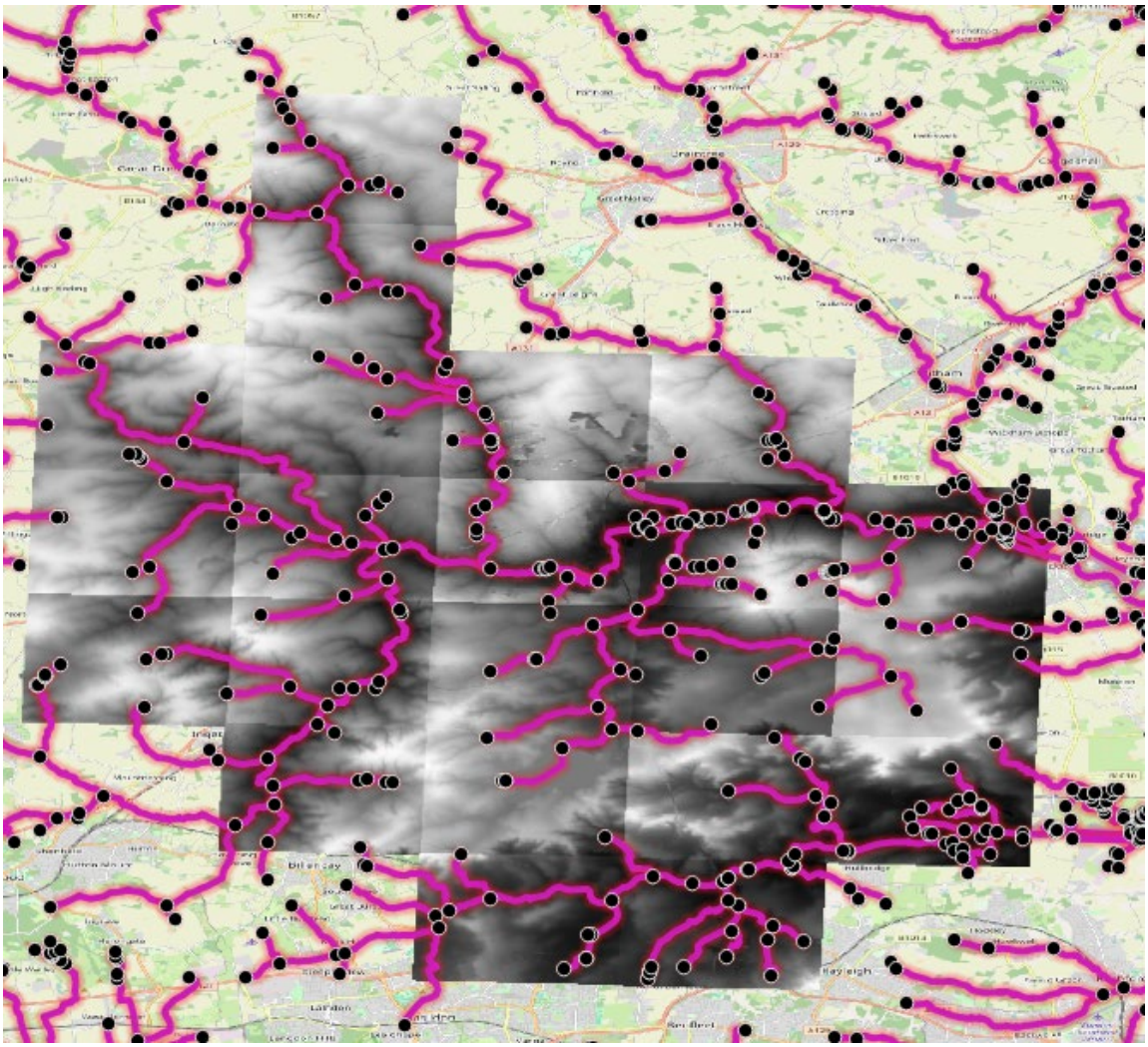

Fig 13: presents the integration of elevation data with hydrological features, combining the Digital Elevation Model (DEM) raster and the open river network. This visualization supports the analysis of topography in relation to water flow and potential flood-prone areas. It illustrates how the D3.1 pipeline enables consistent alignment and integration of raster and vector datasets, providing a reliable foundation for terrain-based spatial analysis and hydrological modeling within AI4MultiGIS.

3.1.4. Through Dedicated Scripts

Although the workflow is designed to be generic, each dataset in Pilot 1 has its own dedicated preprocessing script to account for the diversity of file formats and schemas. These scripts are stored in the /app/pilot1/scripts/ directory and include, for example, Raster-script.py and Vector-script.py. Despite differences in the specific data handled, all scripts adhere to a standardized structure. They begin by loading the raw data, followed by the application of general vector or raster transformations. Throughout the process, all steps are logged to ensure transparency and reproducibility. The scripts then store the processed outputs either in PostGIS for vector data or in the processed rasters directory, and register metadata to support traceability and future reference. This consistent design ensures readability, maintainability, and scalability, facilitating ongoing management and updates of the preprocessing workflow.

3.1.5. Infrastructure

3.1.5.1. Dockerized Pipeline

The entire workflow is executed within a containerized environment, which guarantees reproducibility and consistency across different computing platforms. The environment integrates several key components, each

serving a specific role in the pipeline. PostGIS provides centralized spatial data storage for both vector and raster datasets. Spark with Sedona offers a scalable environment for preprocessing large geospatial datasets efficiently. NiFi manages the ingestion of raw files into the system, ensuring a reliable and structured flow of data. Airflow orchestrates the execution of the pipelines, monitoring ETL processes and managing dependencies between tasks. Data exchange between these components is facilitated through shared volumes, allowing smooth and seamless communication across the containerized services while maintaining data integrity.

3.1.5.2. Orchestration and Automation

The orchestration layer of the workflow is designed to automatically detect whether a newly ingested dataset is vector or raster, ensuring that the appropriate processing steps are applied without manual intervention. Once the data type is determined, Airflow triggers the corresponding preprocessing script, coordinating execution and managing dependencies within the pipeline. Throughout this process, all logs and metadata are systematically recorded, providing full traceability and supporting reproducibility, so that every step of the workflow can be audited and repeated reliably across different datasets and project iterations.

3.1.6. Summary

Pilot 1 demonstrates the successful integration of heterogeneous geospatial datasets into a unified spatial database. The use of a generalized preprocessing workflow—applied separately to vector and raster inputs combined with containerized execution (Docker, Spark, Airflow, PostGIS) ensures reproducibility, scalability, and robustness.

The harmonized datasets now provide a solid foundation for advanced hydrological modelling, flood-risk assessment, and future AI-based analytics within the AI4MultiGIS framework.

3.2. Pilot Use Case 2: Invasive Crayfish Monitoring

3.2.1. Introduction

Pilot 2 represents a more standardized and uniform data environment compared to Pilot 1. The objective of this use case is to validate the pipeline's ability to ingest, classify, and preprocess heterogeneous—but well-structured—datasets, including raster hazard maps, vector administrative layers, and tabular socioeconomic indicators. Unlike Pilot 1, which required specific handling for mixed and inconsistent input formats, Pilot 2 is processed using a fully generic workflow without additional customization.

3.2.2. Input Data Characteristics

The dataset provided for Pilot 2 comprises three main categories of data. First, raster GeoTIFF files (.tif) represent hazard maps or environmental indicators, providing spatially continuous information. Second, vector GeoJSON files (.geojson) contain geometries such as administrative boundaries, infrastructure elements, or thematic layers, enabling detailed spatial analysis. Third, tabular data (.xlsx or .csv) describe socioeconomic characteristics, exposure values, or aggregated indicators, offering essential context for risk assessment and decision-making. Because these formats are well-structured and adhere to widely accepted standards, the pipeline is able to process them efficiently using its default ingestion and preprocessing logic, minimizing the need for dataset-specific customization.

3.2.3. Automated Data Classification

During the ingestion stage, the orchestration system, composed of Airflow and NiFi, automatically inspects each incoming file to determine its type, whether it is a raster (.tif), vector (.geojson), or tabular (.xlsx or .csv) dataset. Based on this classification, the system activates the corresponding preprocessing path, ensuring that each dataset is handled with the appropriate transformations and validations. This mechanism allows the pipeline to remain generic and robust, unaffected by pilot-specific variations, which is a critical requirement for maintaining interoperability across the MultiGIS framework.

3.2.4. Workflow

Once the data type is identified, the pipeline applies the dedicated transformation logic:

3.2.4.1. Vector Data

For vector datasets, the preprocessing workflow begins with parsing the GeoJSON structure to extract the spatial features and associated attributes. Each geometry is then validated to ensure topological correctness and data integrity. Subsequently, the vector data is reprojected into the EPSG:4326 coordinate reference system, providing a standardized spatial framework compatible with other datasets. Attribute names are then harmonized to maintain consistency across the project and facilitate integration. Finally, the data is transformed into a PostGIS-compatible structure, allowing seamless storage, indexing, and further spatial analysis within the centralized geospatial database.

3.2.4.2. Raster Data

For raster datasets, the preprocessing workflow begins with the extraction of essential metadata, including spatial extent, resolution, and coordinate reference system (CRS). If necessary, the rasters are reprojected to EPSG:4326, ensuring spatial consistency across all datasets. The data is then converted into a standardized format that is fully compatible with PostGIS raster storage, enabling efficient indexing and retrieval. Depending on the requirements of the analysis, an optional raster-to-vector sampling step can be performed to generate vector representations from the raster surfaces, facilitating specific types of spatial analysis or integration with vector layers.

3.2.4.3. Tabular Data

- For tabular datasets, the preprocessing workflow begins with the cleaning of rows and handling of missing values to ensure data integrity and reliability. Column names are then normalized to maintain consistency and facilitate integration with other datasets. When longitude and latitude fields are present, a geometry column is created, allowing the tabular data to be spatially represented and analyzed. Finally, the dataset is aligned with the data model established across WP3, ensuring compatibility and seamless integration with the broader MultiGIS processing framework.

3.2.5. Storage in PostGIS

Once the preprocessing is complete, all datasets—raster, vector, and tabular—are loaded into the PostGIS database for centralized storage and management. Vector layers are stored as geometry tables, while raster layers are saved using PostGIS raster types, preserving spatial resolution and metadata. Tabular datasets are inserted with their attributes, and where applicable, include Point geometries to enable spatial analysis. All layers are maintained in a unified coordinate reference system (EPSG:4326), ensuring seamless integration across different data types and supporting robust cross-layer analyses within the geospatial environment.

3.2.6. Validation and Outcomes

The execution of **Pilot 2** highlights the robustness and generality of the WP3 pipeline. During this pilot, several key capabilities were successfully validated. The pipeline **automatically and accurately detects the type of each incoming dataset**, whether raster, vector, or tabular, enabling the correct preprocessing path to be applied without manual intervention. Standard modules are sufficient for **generic preprocessing**, eliminating the need for additional, pilot-specific scripts. The system also ensures the **reliable storage of multiple spatial data types in PostGIS**, maintaining data integrity and consistency. Full traceability is provided through the combination of **Airflow orchestration, Spark processing, and detailed system logs**, supporting reproducibility and auditability. Finally, the use of **standardized formats** allows for high compatibility across different data providers, demonstrating the pipeline's flexibility and interoperability in diverse geospatial contexts.

Figure 14 illustrates the **contours of Romania** derived from a **Digital Elevation Model (DEM) with 30-meter resolution**, overlaid with various status layers representing key geographic or thematic features. This visualization highlights the topographic variation across the country, while simultaneously providing contextual information from additional status layers. By combining elevation data with thematic overlays, the figure demonstrates the pipeline's ability to integrate multiple data types, ensuring accurate alignment, consistent spatial referencing, and readiness for detailed geospatial analysis.

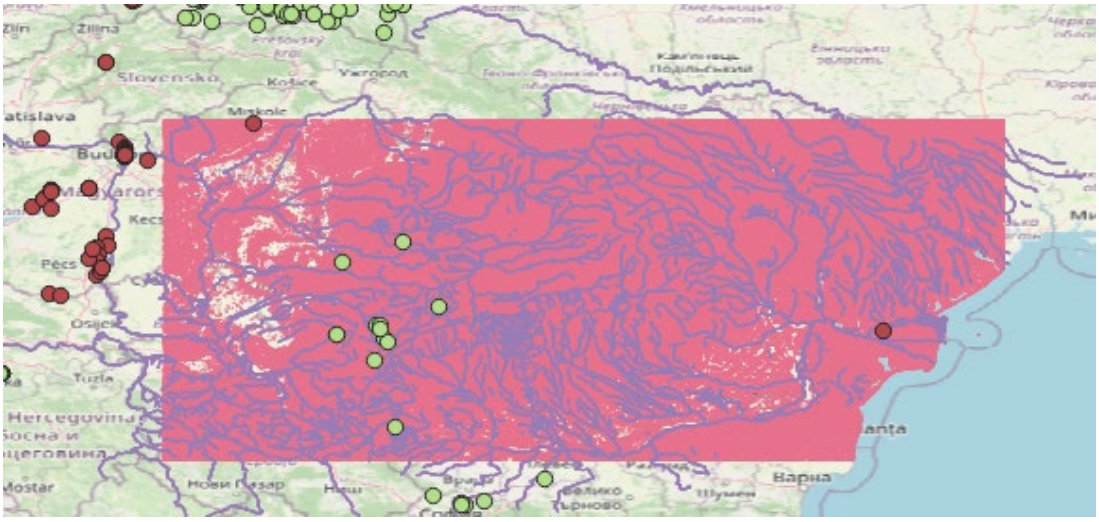

Figure 14: illustrates the application of the D3.1 pipeline to large-scale elevation data, using a 30-meter resolution Digital Elevation Model (DEM) of Romania combined with multiple thematic status layers. This visualization demonstrates the pipeline's ability

Figure 15 presents the **30-meter resolution elevation data** of the study area overlaid with the river network. This visualization highlights the relationship between topography and hydrology, illustrating how elevation influences river courses and potential water flow patterns. The figure demonstrates the effective integration of raster and vector datasets within the pipeline, showcasing accurate alignment, consistent coordinate reference systems, and readiness for further spatial analysis and environmental assessment.

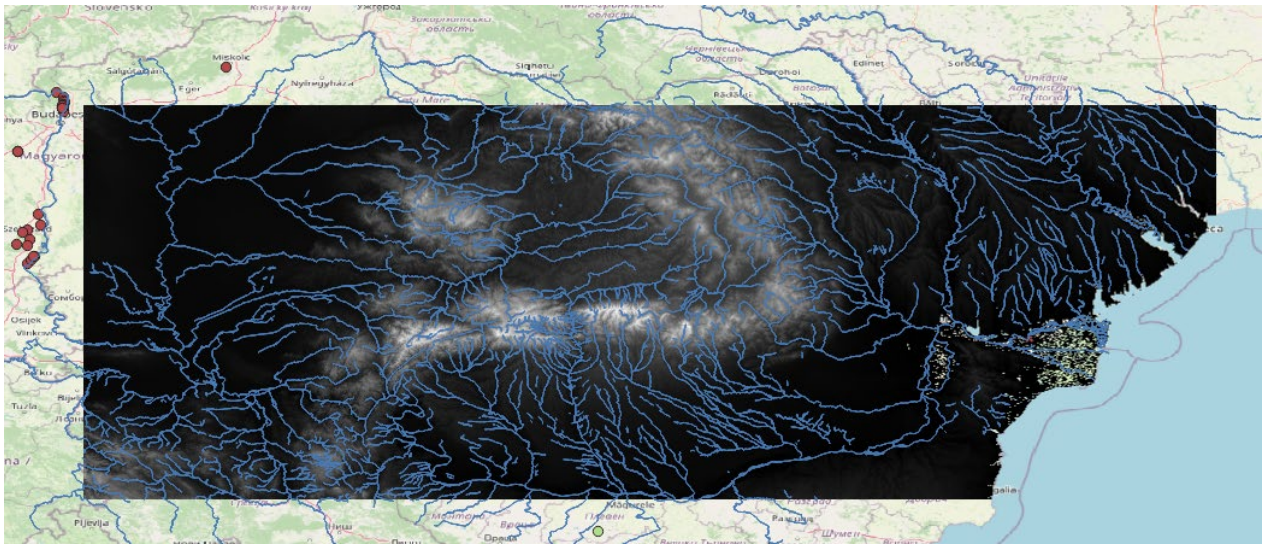

Fig 15: presents the integration of elevation data and hydrological networks produced by the D3.1 pipeline. The overlay of the 30-meter resolution DEM with river vectors highlights the relationship between terrain morphology and water flow patterns. This figure demonstrates the effectiveness of the preprocessing and spatial alignment steps in enabling combined raster–vector analysis, which is essential for hydrological modeling, flood assessment, and environmental monitoring use cases in AI4MultiGIS.

Figure 16 depicts the **distribution of the "Native" crayfish** within the study area. The map highlights observed locations and habitat zones, providing insight into species presence and spatial patterns. This figure demonstrates the pipeline's ability to integrate **biological observations with geospatial layers**, enabling ecological analysis and supporting habitat management and conservation planning.

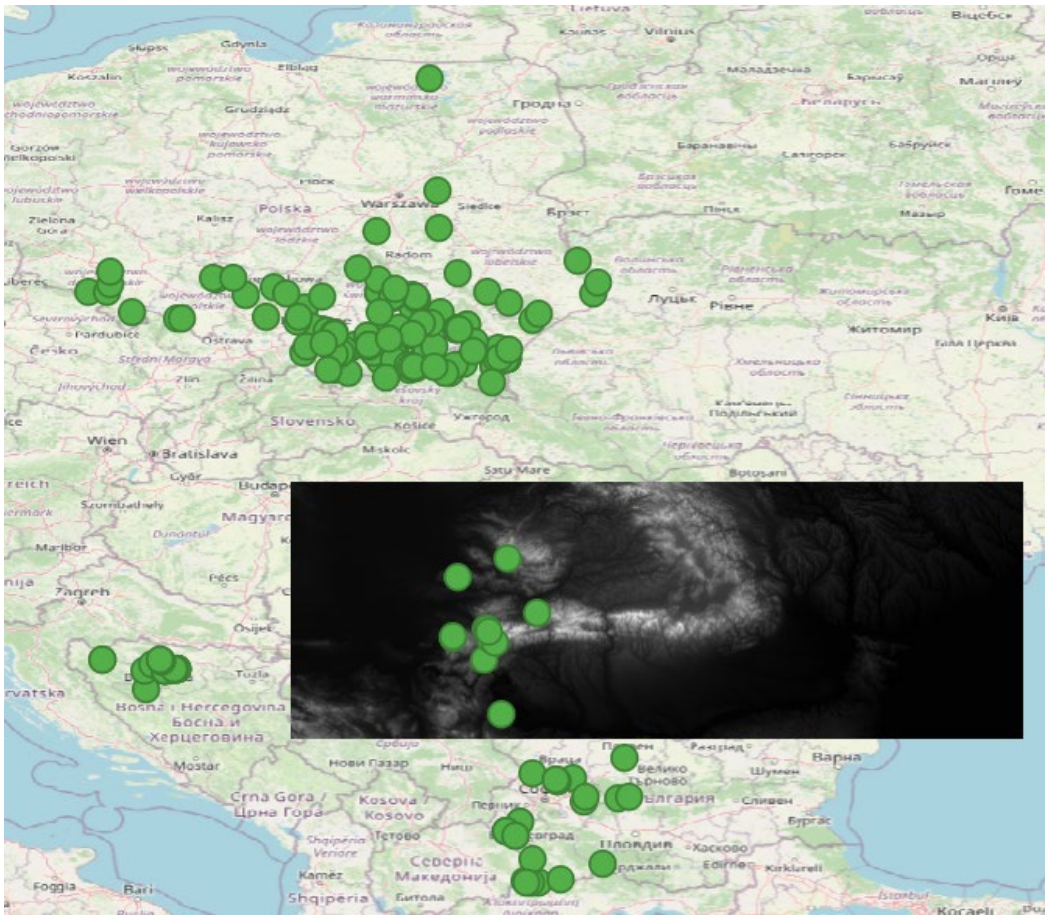

Fig 16: shows the spatial distribution of native crayfish occurrences integrated into the geospatial pipeline. By incorporating biological observation data alongside existing geographic layers, this visualization demonstrates the pipeline's capability to handle ecological datasets in addition to physical and environmental data. The resulting map supports habitat analysis and illustrates how D3.1 enables biodiversity-related spatial analytics within the project's environmental monitoring pilots.

Figure 17 illustrates the **spatial distribution of the "Alien" crayfish** across the study area. The mapped points indicate the locations where this non-native species has been observed, highlighting its spread and areas of potential ecological impact. This visualization demonstrates the integration of species occurrence data into the geospatial pipeline, supporting environmental monitoring and enabling comparative analysis with native crayfish populations for conservation and management purposes.

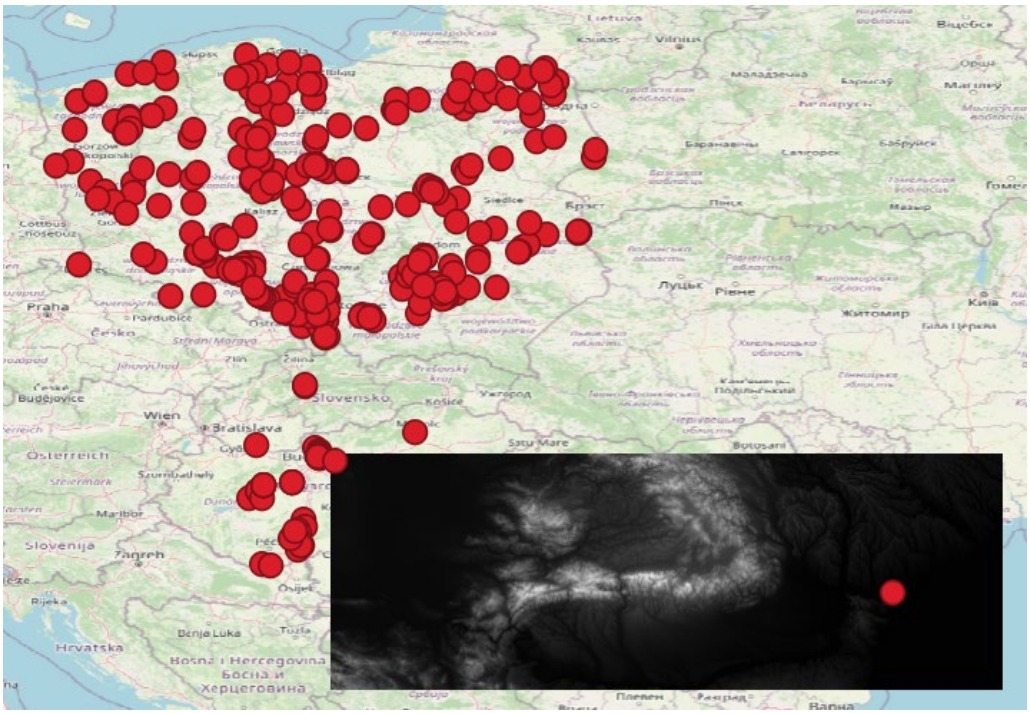

Fig 17: illustrates the integration of non-native (alien) crayfish occurrence data into the unified geospatial framework. The visualization highlights the spatial extent and distribution patterns of the invasive species, demonstrating how the D3.1 pipeline supports environmental monitoring by combining species occurrence data with geospatial context. This enables comparative analysis and early identification of areas potentially affected by biological invasions.

Figure 18 presents the combined spatial distribution of both **"Native"** and **"Alien"** crayfish within the study area. By displaying the two species on the same map, the figure highlights areas of overlap, potential interaction zones, and regions where invasive presence may pose ecological pressure on native populations. This integrated visualization demonstrates the pipeline's capacity to merge multiple biological datasets into a unified geospatial layer, supporting comparative ecological analysis and informing biodiversity monitoring and management strategies.

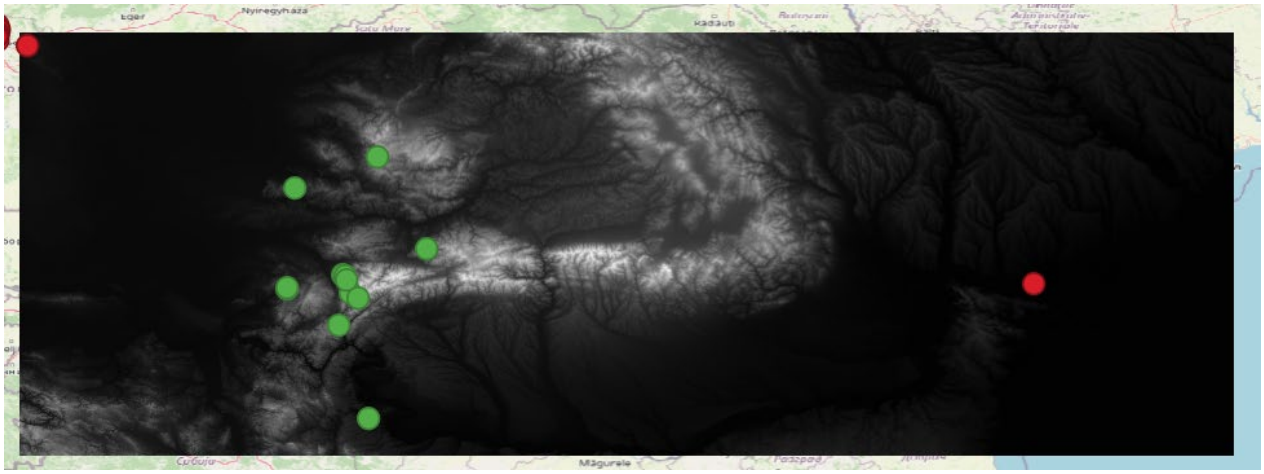

Fig 18: presents a combined view of native and alien crayfish distributions, generated through the integration capabilities of the D3.1 pipeline. By displaying both datasets within a single spatial context, the figure highlights zones of overlap and potential ecological interaction. This visualization demonstrates the pipeline's capacity to merge multiple biological datasets into a coherent geospatial representation, supporting comparative ecological analysis and informed biodiversity management.

Figure 19 provides an overall overview of the integrated geospatial datasets used in the pilot. The figure brings together the key raster and vector layers—such as elevation surfaces, hydrological networks, administrative boundaries, and species distribution points—into a single composite map. This comprehensive visualization illustrates the coherence of the preprocessing pipeline, the consistency of the coordinate reference system, and the successful alignment of diverse data sources. It offers a global perspective on the study area, demonstrating how the harmonized datasets can be jointly analyzed to support environmental assessment, spatial modelling, and decision-making activities.

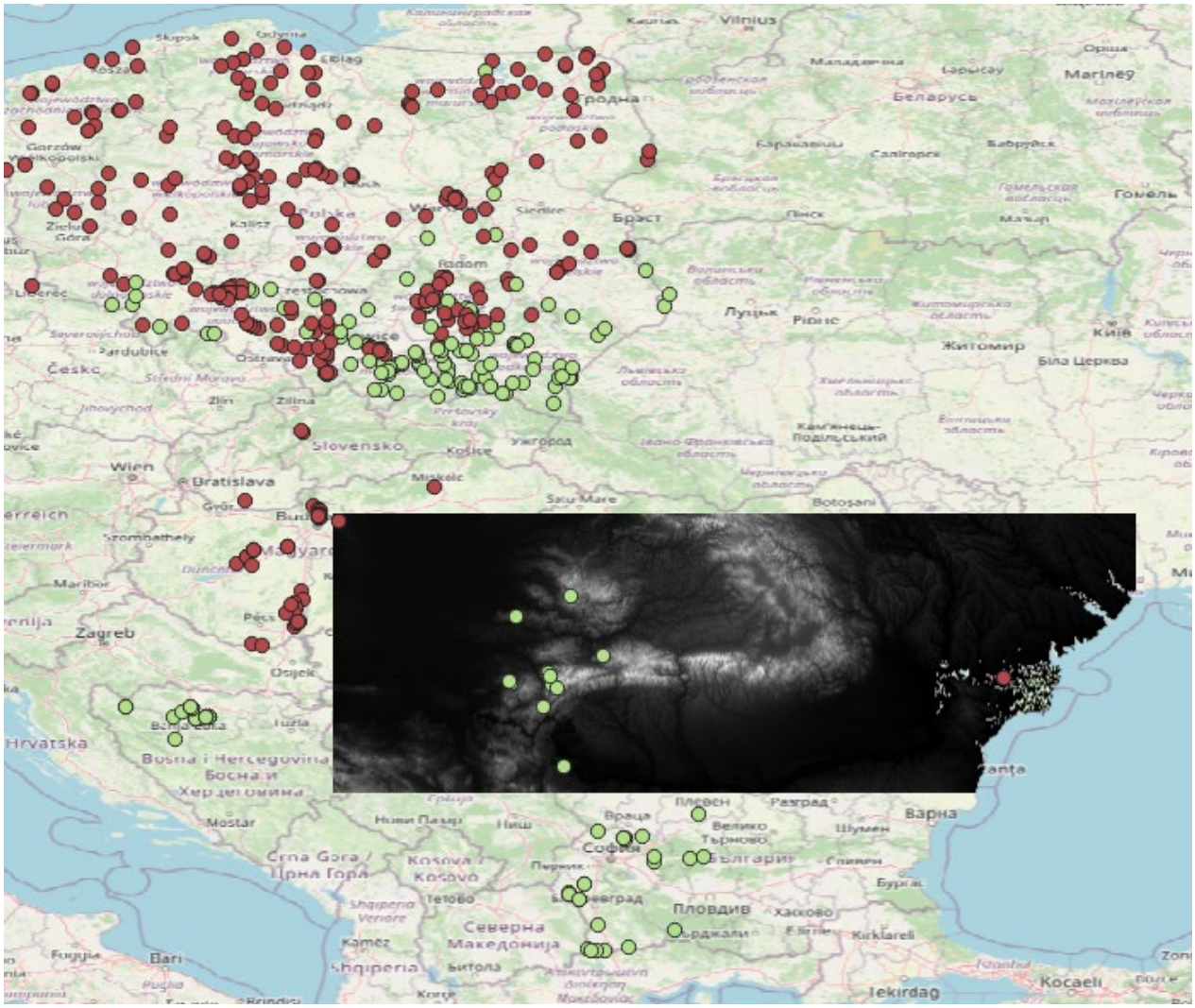

Fig 19: provides a global overview of the harmonized datasets produced by the D3.1 pipeline. It combines key raster and vector layers, including elevation data, hydrological networks, administrative boundaries, and species distribution points, into a single composite map. This figure summarizes the end-to-end effectiveness of the preprocessing, alignment, and integration workflow, illustrating how diverse data sources are transformed into a coherent, query-ready geospatial framework supporting environmental assessment, spatial modeling, and decision-making within AI4MultiGIS.

Pilot 2 confirms the pipeline's ability to process operational, multi-format datasets in a fully automated and harmonized way, supporting the broader MultiGIS objective of interoperable and scalable geospatial data management.

4. Data Pipeline Architecture

4.1. Principles of Event-Driven Pipeline Design

The pipeline is organized into three main phases as presetnded in figure 20, **monitoring and ingestion**, **data categorization**, and **processing and storage**, orchestrated through a time-triggered and event-driven workflow.

The process starts at a scheduled time (02:00 AM), when the system begins monitoring a designated raw data directory associated with a pilot use case. This monitoring phase runs for a fixed duration and acts as the entry point of the pipeline. If no data are detected within the allocated time window, the workflow follows a timeout branch and triggers a notification mechanism to report the absence of incoming data, ensuring operational transparency.

When new data are detected, the ingestion phase is initiated through Apache NiFi. NiFi is responsible for automatically handling incoming files, including detecting compressed archives, extracting their contents, and organizing the data based on their type. At this stage, datasets are classified into distinct categories—raster, vector, tabular, or other formats—and moved into corresponding structured directories. This classification step enables early harmonization of heterogeneous data sources and prepares them for downstream processing.

Once ingestion and sorting are completed, NiFi is stopped, and the pipeline transitions to the processing and storage phase. A set of generic processing scripts is then executed, each tailored to a specific data type. Vector, raster, and tabular datasets are processed independently while following a common logic for validation, transformation, and metadata extraction. During this phase, relevant metadata, structural information, and identifiers are inserted into the PostgreSQL/PostGIS database, enabling consistent storage, indexing, and future querying of the integrated datasets.

After successful processing, all datasets are moved to a dedicated “processed” directory, marking the completion of the workflow. This final step ensures traceability between raw and processed data and prevents duplicate processing in subsequent pipeline executions.

Overall, the figure provides a consolidated view of the real-time data pipeline architecture, highlighting the interaction between scheduling, event-driven ingestion, data categorization, automated processing, and structured storage. It demonstrates how Deliverable D3.1 implements a modular, reproducible, and scalable pipeline capable of handling heterogeneous geospatial datasets in support of the AI4MultiGIS pilot use cases.

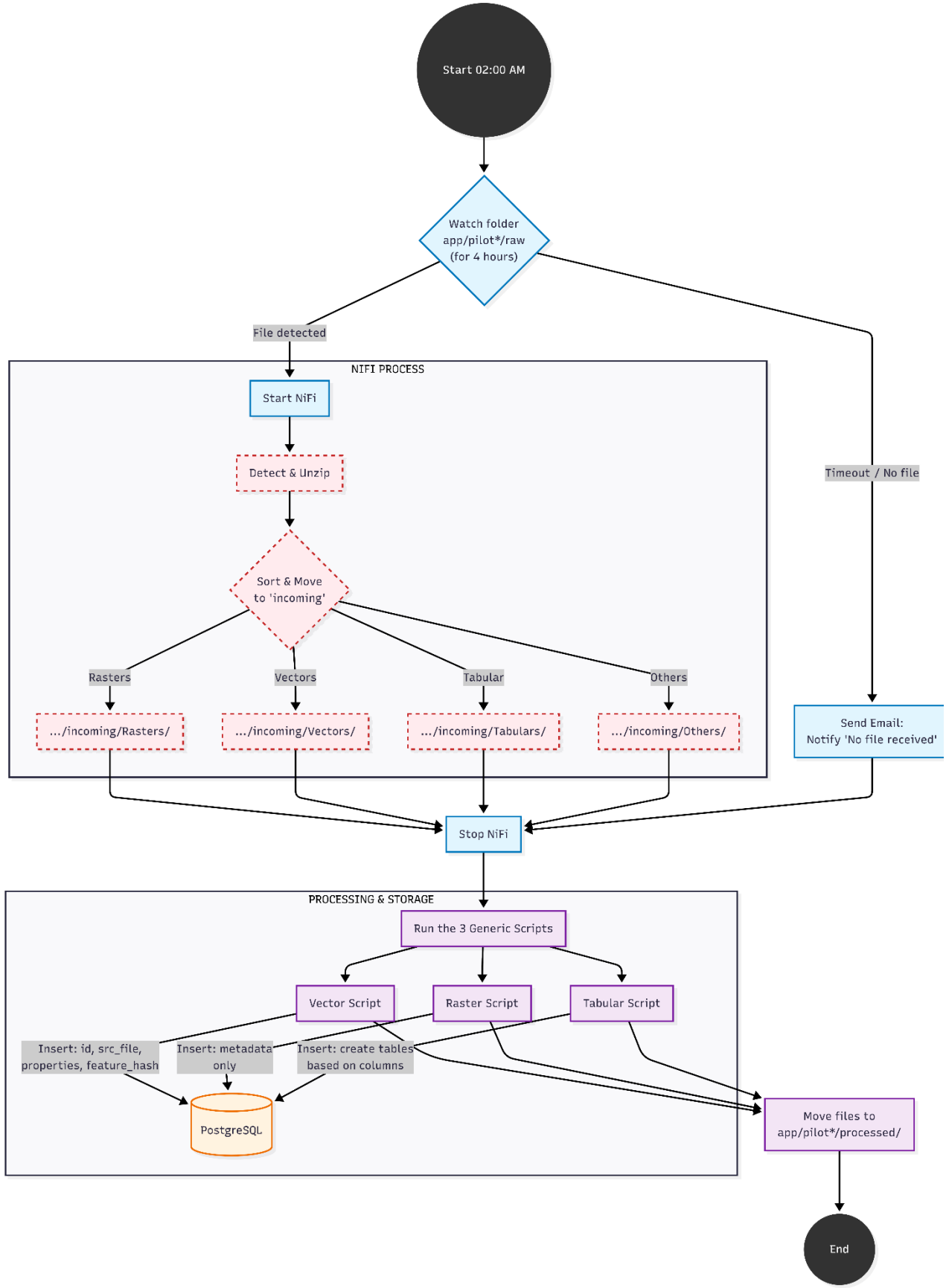

Figure 20: illustrates the overall architecture and execution flow of the real-time data pipeline implemented in Deliverable D3.1 within the AI4MultiGIS project.

The event-driven architecture of the system provides a high degree of modularity and adaptivity, allowing each component of the pipeline ranging from data ingestion and preprocessing to validation, storage, and analytical processing to operate as an independent, loosely coupled service. Instead of relying on rigid, sequential workflows, these modules communicate through event notifications, which enables the pipeline to adjust dynamically to variations in data volume, the integration of new data sources, or shifting processing requirements. This design choice ensures that the system remains flexible and capable of evolving alongside the project's growing complexity.

A second major advantage of this architecture lies in its resource efficiency and inherent scalability. Pipeline components are activated only when relevant events occur for example, when a new satellite tile becomes available, when an IoT stream publishes an updated measurement, or when a user requests a new computation. This "on-demand" activation avoids unnecessary resource consumption during idle periods and makes it possible to scale seamlessly across both edge and cloud infrastructures. As data volumes increase or new sensors are deployed, the system can naturally scale horizontally without requiring major structural changes.

The architecture also enhances real-time responsiveness. Event-driven triggers allow the system to react immediately to time-sensitive operations such as validating and storing flood-risk sensor data, launching automated analysis workflows for incoming imagery, or updating derived products like risk maps or accessibility indices. By reducing latency between data arrival and processing, the pipeline ensures that analytical outputs remain as current and reliable as possible, which is especially critical in dynamic environmental contexts.

In addition, the event-based structure supports robust error handling and continuous monitoring. When a task encounters a failure, the system emits a dedicated error event, enabling rapid intervention through automated retries, targeted alerts to system operators, or the activation of fallback mechanisms. This approach prevents failures from propagating silently through the pipeline and contributes to a more resilient and maintainable system.

Finally, the architecture is inherently extensible. New sensors, additional data modalities, or advanced analytical modules can be introduced simply by defining new event types and the handlers associated with them. This extensibility allows the pipeline to grow steadily over time, integrating new capabilities without disrupting existing workflows and ensuring long-term sustainability as project requirements evolve.

By leveraging event-driven principles, the Ai4MultiGIS pipeline achieves a balance of flexibility, operational efficiency, and real-time performance crucial for intelligent geospatial operations across multi-source, multi-modal scenarios.

4.2. Ingestion and Consolidation of Multimodal Datasets

The Ai4MultiGIS pipeline is engineered to ingest and consolidate a diverse spectrum of geospatial data types, encompassing satellite imagery, UAV data, IoT sensor streams, publicly available open datasets, and synthetically generated sources. This multidimensional approach enables comprehensive situational awareness and robust geospatial analysis.

The pipeline begins with a robust multimodal data acquisition process designed to handle highly heterogeneous datasets that differ in format, spatial resolution, update frequency, and origin. Through a set

of automated connectors, the system interfaces seamlessly with external APIs, sensor networks, satellite providers, UAV data streams, and batch file repositories. This automation ensures that all relevant data is ingested in a timely and consistent manner, eliminating manual intervention and preserving the continuity of monitoring operations.

Once acquired, every dataset enters a unified preprocessing and harmonization workflow. Here, the pipeline applies a series of standardized transformations such as coordinate reference system harmonization typically involving reprojection to a common spatial reference like EPSG:4326—along with format conversion, spatial subsetting to the project's area of interest, and the enrichment of attributes through metadata augmentation. These steps establish a coherent and compatible foundation, allowing otherwise disparate datasets to flow through the analytical chain without friction.

To safeguard data reliability, the pipeline incorporates an automated layer of quality control and standardization. Validation routines detect missing or anomalous values, structural inconsistencies, and temporal or spatial outliers. These checks occur both at the edge, where raw data is first captured, and in the cloud, where aggregated datasets are verified before deeper analysis. The system assigns quality flags and generates anomaly logs, storing them as metadata to ensure full transparency and enabling targeted remediation whenever necessary.

Processed datasets are then stored and indexed within a centralized PostGIS environment, allowing efficient spatial querying and seamless cross-referencing with vector and raster layers. This database-centric approach also supports advanced visualization workflows, including direct integration with platforms such as QGIS. Complementing the storage system is a metadata repository that documents every dataset's provenance, source characteristics, and processing status, ensuring complete traceability and accountability throughout the pipeline.

Finally, the architecture supports dynamic consolidation and integration, allowing newly ingested datasets to be merged and analyzed in real time. This capability makes it possible to fuse high-resolution UAV imagery with sensor-derived flood analytics, or to combine open geographic information with proprietary sources to enhance decision-making. Such multimodal integration strengthens situational awareness, enriches analytical outputs, and supports the development of comprehensive, data-informed strategies tailored to complex environmental and urban challenges.

Through efficient ingestion and rigorous consolidation mechanisms, Ai4MultiGIS ensures that multi-modal geospatial data is consistently transformed, harmonized, and made immediately actionable for end-users and analytical modules.

4.3. Workflow Orchestration and Automation

A central feature of the Ai4MultiGIS pipeline is automated workflow orchestration, enabling streamlined, reliable, and scalable execution of complex geospatial data processing tasks. The pipeline leverages advanced orchestration tools—such as Apache Airflow and containerized microservices—to manage and automate every step from ingestion through analytics and storage.

Within the pipeline, all tasks are structured as Directed Acyclic Graphs (DAGs), which explicitly define dependencies and execution order. This DAG-based organization enables parallel processing where possible, optimizes resource allocation, and simplifies modifications or extensions of workflows as the project requirements evolve. Complementing this structure, the pipeline employs automated task scheduling that combines periodic routines with event-based triggers. These mechanisms initiate workflows for data

ingestion, preprocessing, validation, and integration without manual intervention, ensuring that incoming geospatial datasets are processed promptly and consistently.

The system also incorporates continuous monitoring and failure recovery capabilities. Integrated monitoring modules track the health, progress, and resource utilization of all running tasks, automatically detecting failures or performance bottlenecks. When a task fails, orchestration tools trigger predefined recovery procedures, such as automated retries, fallback processes, or alerts to system operators, maintaining operational continuity and minimizing disruptions.

Scalability and modularity are further reinforced through containerization, which allows each pipeline stage to be deployed independently across edge or cloud environments and scaled according to processing demands. This modular architecture ensures that new data types, analytical modules, or processing tasks can be integrated seamlessly into the existing workflow without interrupting ongoing operations.

Finally, the pipeline emphasizes auditability and traceability. All workflow executions are meticulously logged and versioned, providing transparency, supporting reproducibility, and facilitating debugging. Each task is associated with comprehensive metadata, including timestamps, input sources, processing parameters, and execution outcomes, ensuring that the entire processing chain can be reviewed, validated, and replicated when necessary.

By combining robust orchestration, automation, and monitoring, Ai4MultiGIS delivers a resilient and extensible system for real-time geospatial data processing that minimizes operational overhead and maximizes analytical productivity.

4.4. Scalable EdgeCloud Pipeline Architectures

The Ai4MultiGIS pipeline is architected for high scalability and robustness, leveraging hybrid edge-cloud infrastructure to optimize geospatial data processing and analytics.

The pipeline is built on a hybrid architecture that integrates both edge computing, located close to data sources, and centralized cloud resources with high computational capacity. At the edge, lightweight AI and preprocessing modules perform initial filtering, format harmonization, and real-time anomaly detection directly on sensor nodes or field devices. This approach reduces data transmission overhead and enhances privacy by processing sensitive information locally, before transferring only the essential outputs to central systems. In parallel, the cloud infrastructure handles more computationally intensive tasks, including large-scale analytics, long-term storage, complex spatial modeling, and advanced visualization, leveraging scalable resources to support high-demand operations.

All components of the pipeline are deployed as containerized microservices, using technologies such as Docker, which ensures portability, rapid scaling, and simplified maintenance across heterogeneous environments. The system incorporates dynamic resource allocation, whereby orchestration tools like Apache Airflow, NiFi, and Kubernetes automatically adjust computational and storage resources in response to fluctuations in data volume, network conditions, and analytical demands. This ensures optimal performance while minimizing resource waste.

The architecture also emphasizes interoperability and flexibility. Its modular design supports a wide variety of data formats and processing frameworks, including Spark-Sedona for distributed spatial analytics, PostGIS for spatial storage, Airflow for workflow management, and QGIS for visualization. This modularity allows for the seamless integration of new data sources, analytical tools, and services, future-proofing the pipeline against evolving project requirements.

Finally, end-to-end monitoring and adaptive mechanisms are embedded at both the edge and cloud levels. Continuous monitoring detects process drift, resource bottlenecks, or anomalous events, and automated feedback loops enable the system to respond dynamically. Resources can be reallocated, new tasks triggered, or human operators alerted as needed, ensuring that the pipeline remains resilient, efficient, and responsive to changing operational conditions.

By seamlessly integrating edge and cloud infrastructures, the Ai4MultiGIS pipeline ensures scalable, resilient, and performant handling of large and diverse geospatial datasets—supporting advanced analytics and real-time applications across multiple domains.

5. Trustworthy Data Governance and Security

5.1. Blockchain/DLT for Data Security

Before exploring the technical details of distributed ledger technology, it is important to note why blockchain and other forms of DLT are considered relevant for MultiGIS. MultiGIS systems require coordination between multiple organisations, each of which must trust that the shared records of data collection, storage, and sharing are accurate and tamper evident. Distributed Ledger Technology (DLT) provides a common foundation for this trust, enabling different stakeholders to collaborate securely without relying on a single central authority.

To build on this context, the following subsection introduces the basic concept of distributed ledger technology and explains how it applies to MultiGIS environments.

5.1.1 Distributed Ledger Technology (DLT) for MultiGIS

Distributed ledger technology (DLT) is a way to keep the same record across many computers run by different parties. Each update is added to the end of the record, time-stamped, and digitally signed by whoever made it. All authorised participants can check the history and confirm that nothing has been changed silently. On the other hand, Blockchain is one family of DLT. In a blockchain, updates are grouped into “blocks,” and each block includes a cryptographic link (a hash) to the previous one. This makes the history tamper-evident: if someone changes past data, the hashes no longer match. Not all DLTs use blocks; some use other structures (for example, directed acyclic graphs or replicated logs). In practice, people often say “blockchain” when they mean “DLT,” but blockchain is a specific design within the broader DLT space.

This overview naturally leads to a discussion of the different types of blockchain networks that can be employed in practice.

Types of Blockchain Networks

- **Public, permissionless:** Anyone can join, read, and write. These networks are open and resistant to censorship. However, they often suffer from variable transaction fees, lower throughput, and limited privacy.
- **Public, permissioned (read-open / write-restricted):** Anyone can read, but only approved entities can write. This model is used in some civic or registry settings.

- **Consortium / permissioned:** A known group of organisations runs the network. Membership is controlled, and read/write access follows agreed policies. Consensus mechanisms are usually fast and final (e.g., Raft/BFT-style).
- **Private / permissioned (single organisation):** One organisation operates all nodes. This model offers high performance but relies on centralised governance.

Having outlined the different network types, it is important to specify which model best aligns with the needs of MultiGIS.

Which Model for AI4MultiGIS and Why?

For AI4MultiGIS we select a consortium, permissioned DLT. This choice fits research–public sector partnership that needs trustworthy data governance and predictable operations:

- **Governance and privacy:** We must control who can see metadata and who can write records. A permissioned network lets us enforce roles and policies and support GDPR-aligned selective disclosure.
- **Performance and finality:** The real-time pipeline benefits from low-latency confirmation and deterministic costs (no "gas" fees). Permissioned consensus gives fast, final commits suitable for operational events.
- **Integrity with practicality:** We keep large geospatial files off-chain (in object stores) and anchor their hashes and lifecycle events on-chain. This balances auditability with performance.
- **Accountability:** Every transaction is signed by a known organisation or service account, giving non-repudiation and a clear audit trail.
- **Operational control:** Membership, node placement (edge or cloud), and data residency can follow consortium and legal requirements. Keys can be rotated and access can be revoked.
- **Sustainability:** Permissioned networks avoid energy-intensive mining and have a smaller operational footprint.

A permissioned network assumes governance among known parties and does not offer the same level of openness as public chains. This is acceptable—and desirable—for our use case, where datasets, models, and policies are shared within a controlled consortium rather than the open internet. We therefore use a permissioned consortium blockchain (DLT) as the governance backbone. It records dataset and model identifiers, hashes, timestamps, policy states, and quality flags, while large files stay off-chain. This design supports trustworthy data governance in WP3 and prepares the dedicated DLT system in D3.4.

The choice of model provides the basis for considering which specific platforms are most suitable in real deployments.

Examples of Permissioned DLT Platforms

Several well-known platforms support permissioned (restricted membership) deployments that are relevant for MultiGIS applications:

1. **Hyperledger Fabric:** A mature permissioned platform designed for consortium use. It provides private channels, role-based access control, and flexible consensus options such as Raft. Fabric also supports private data collections and a robust identity model through Membership Service Providers (MSPs). Smart contracts ("chaincode") can be developed in multiple languages including Go, Java,

and JavaScript. Importantly, Fabric has no mandatory cryptocurrency, making it particularly well-suited for data-governance use cases rather than financial transactions.

2. **R3 Corda:** A distributed ledger platform designed for point-to-point privacy. Instead of broadcasting all transactions to the entire network, Corda ensures that only the participants involved in a transaction can see the relevant data. Uniqueness and finality are guaranteed through a notary service. Written primarily in Kotlin/Java, Corda is often positioned as "DLT, not classic blockchain," making it suitable for workflows where bilateral privacy is more important than global visibility.
3. **ConsenSys Quorum:** An enterprise-focused Ethereum variant that supports permissioned deployments. It is fully EVM-compatible, allowing the use of Solidity smart contracts and Ethereum development tools. Quorum also supports private transactions through privacy managers such as Tessera, and provides consensus mechanisms like IBFT and Raft. This makes it attractive when Ethereum ecosystem compatibility is important in a permissioned context.

Among these platforms, Hyperledger Fabric could be chosen as the focus for our MultiGIS work. The reasons include its proven maturity for consortium deployments, strong modularity, and absence of a built-in cryptocurrency, which avoids regulatory complexity while keeping operational costs predictable. Fabric's channel-based architecture allows different agencies to share only the data relevant to them, preserving confidentiality while maintaining overall trust and provenance. Its support for fine-grained identity management and flexible smart contract development makes it particularly suitable for the governance backbone of AI4MultiGIS.

Building on the overview of permissioned DLT platforms, we now turn to the major security issues that exist in MultiGIS environments, which form the foundation for understanding where blockchain can contribute.

5.1.2 Security Issues in MultiGIS Data

MultiGIS environments, which integrate multiple geospatial repositories and thematic layers, face several unique security challenges across the phases of data collection, storage, and sharing. Unlike traditional relational data, geospatial data is organized into complex thematic layers and may represent the same spatial region in multiple ways (e.g., vector data, raster imagery).

Unauthorized access during data acquisition occurs when attackers intercept or manipulate input data collected from drones, mobile sensors, or volunteered geographic information (VGI) platforms (12), (13). Spoofing and tampering at collection points arise when malicious actors manipulate GPS signals, sensors, or crowdsourced inputs, thereby compromising the accuracy and reliability of collected data (14). False identification risks in collection anonymization appear when geomasking applied at the point of data capture accidentally assigns sensitive attributes to the wrong location, creating ethical and legal risks (15). Consent and ethical collection practices are often neglected when individuals are not informed that their location or movements are being recorded (16).

Beyond the collection phase, repository and storage threats emerge when GIS repositories suffer from unauthorized access, dataset tampering, or piracy. These risks are especially significant in outsourced or cloud-based storage, where blockchain provenance and append-only ledgers can mitigate them by ensuring tamper-evidence (17). Ransomware and malware attacks on GIS stores also threaten availability when spatial databases and file stores are encrypted or exfiltrated by malicious actors (14). Heterogeneous inter-agency access policies during sharing create further vulnerabilities, as mismatched rules may lead to either

overexposure or underexposure of geospatial layers and attributes. Blockchain-based smart contracts can help unify such policies (17), (18).

Secure transmission and provenance during exchange remain critical because data shared across organizations must travel over authenticated channels. Blockchain can provide immutable provenance that prevents undetected alteration in transit (19), (18). Data integrity in storage can also be compromised by silent tampering or corruption, but blockchain mitigates this risk by ensuring immutable provenance and tamper-evidence (19), (18). Finally, accountability of data sharing is often lacking, as it is difficult to trace who shared which datasets and when. Blockchain addresses this by recording non-repudiable, traceable histories of sharing events (19), (18).

5.1.3 Techniques to Solve the Security Issues

To address the diverse security risks across the phases of data collection, storage, and sharing, researchers have proposed a combination of traditional safeguards and emerging distributed ledger mechanisms. One important approach is data anonymization at the point of collection, which employs techniques such as geomasking, k-anonymity, and geo-indistinguishability to obscure sensitive details before the information is stored or shared (20) (15). Complementing this, encryption of raw sensor streams ensures that video, audio, and GPS feeds remain protected through end-to-end encryption while they are being collected and transmitted (21).

Ensuring the trustworthiness of the data sources themselves is equally vital. Authentication of collection devices verifies that drones, sensors, and crowdsourcing applications originate from authorized sources, preventing the insertion of malicious or falsified data (17), (17). Alongside this, consent-aware collection frameworks enable the acquisition of data in a manner that respects user rights, obtaining informed consent at the time of capture and applying context-aware privacy defaults to minimize risks (16).

To reduce privacy exposure while still enabling analysis, federated learning at the edge keeps raw data on local devices and transmits only model updates, thereby minimizing the centralization of sensitive information (17). In addition to these measures, blockchain technologies are increasingly recognized as valuable complements to traditional security methods. Blockchain-based provenance and integrity mechanisms record hashes and metadata on a consortium ledger, ensuring that any alteration of datasets during collection, storage, or transfer can be detected (19), (14). Similarly, smart-contract access control encodes and enforces inter-agency rules directly on the ledger, supporting secure sharing while reducing the risks caused by inconsistent policies (22). Finally, tamper-evidence from the point of capture can be achieved by registering sensor or device outputs directly on the blockchain, thereby providing traceability and accountability from the earliest stages of the data lifecycle (21).

Together, these measures form a layered defense strategy, combining cryptography, authentication, consent management, privacy-preserving analytics, and distributed ledger technologies. Such an integrated approach strengthens the resilience of MultiGIS infrastructures and promotes trust among the various stakeholders engaged in collecting, storing, and sharing geospatial information.

5.1.4 Privacy Issues in MultiGIS Data

Privacy concerns in MultiGIS stem primarily from the ability to link spatial data with personal identities. Location traces, geodemographics, or household-level mapping can inadvertently disclose private attributes

such as health status, income, or vulnerabilities (16). Historical cases highlight the problem: during Hurricane Katrina, GIS maps displaying body recovery sites, even when symbolized imprecisely, allowed outsiders to identify individual houses, illustrating the danger of inadequate spatial confidentiality (22).

Another important issue is the re-contextualization of data, where spatial information collected for public purposes (e.g., urban planning) is reused for commercial profiling or surveillance without consent (16). The rise of IoT sensors, drones, volunteered geographic information (VGI), and cloud GIS introduces new vulnerabilities. Continuous georeferenced streaming data can expose mobility patterns, while unverified volunteered data may compromise both privacy and accuracy (14). These risks are compounded by legal and ethical concerns, as regulations such as the GDPR and CCPA impose obligations on location data protection, but inconsistent enforcement and cross-border data sharing leave significant gaps (16), (14).

Moreover, geomasking ethics present further challenges. Masked points snapped onto the wrong home or business can imply sensitive attributes (e.g., disease, crime) for uninvolved people, creating an ethical and legal risk that raises questions of fairness and accountability (15). While masking techniques may preserve global trends, they can also disrupt neighborhood-level analyses (e.g., walkability, amenities for chronic disease studies), creating uneven impacts that should be assessed at multiple scales (23). In addition, privacy leakage during capture is a major concern because high-resolution imagery, video, or audio collected by drones or sensors can reveal individuals or private property without consent (21).

Collectively, these issues underscore the importance of balancing data utility with respect for individual rights, equity, and trust in the collection, storage, and sharing of MultiGIS data.

5.1.5 Privacy-Preserving Techniques to Solve Privacy Issues

A wide range of privacy-preserving strategies has been developed to address risks in data collection, storage, and sharing within MultiGIS environments. One fundamental approach is geomasking and data perturbation, where geographic coordinates are altered to reduce the likelihood of identifying individuals while retaining analytic value (15). In a similar vein, aggregation and generalization publish information at coarser scales—such as census tracts rather than household-level data—thereby protecting sensitive details while still supporting meaningful analysis. Building on these techniques, anonymization models like k-anonymity, l-diversity, and t-closeness ensure that individuals remain indistinguishable within groups, thereby reducing the risk of re-identification (24), (20).

More advanced methods such as differential privacy introduce carefully calibrated noise into datasets, making it mathematically provable that the presence or absence of a single individual cannot be inferred. Its extension, geo-indistinguishability, adapts these principles to location data, preserving spatial utility while safeguarding privacy (14). Other techniques, such as data masking, replace sensitive attributes with substitutes, ensuring that statistical patterns are preserved while personal identifiers are hidden (24).

Recent advances in cryptography have added further options. Homomorphic encryption allows computations to be performed on encrypted geospatial data without the need for decryption, thereby preserving confidentiality throughout the analytic process (24). Secure Multi-Party Computation (SMPC) enables multiple stakeholders to jointly analyze datasets without exposing their raw inputs, while federated learning supports collaborative geospatial AI by keeping raw data at the edge and sharing only model updates (24). Beyond cryptography, consent-driven policies and privacy firewalls empower individuals and communities to exert greater control over how their spatial data is accessed and reused (16).

In addition, digital watermarking can secure copyright and authenticate the ownership of geospatial datasets, while ethical and educational frameworks emphasize the need for contextual integrity, transparency, and adherence to professional codes of conduct (16), (14). Case studies demonstrate that when these techniques are applied across sectors such as urban planning, healthcare, transportation, and disaster management, GIS-related data breaches can be reduced by more than 50% (14). Collectively, these strategies provide a robust foundation for protecting privacy without undermining the utility of geospatial information.

Before moving forward, we now discuss how blockchain can help to resolve some of the issues identified above, ensuring trust, accountability, and other critical benefits in MultiGIS environments (see Figure 20).

The figure illustrates how distributed ledger features (e.g., tamper-evident ledger, permissioned access, smart contracts) mitigate critical security risks in MultiGIS and translate into benefits such as integrity, accountability, resilience, and transparency.

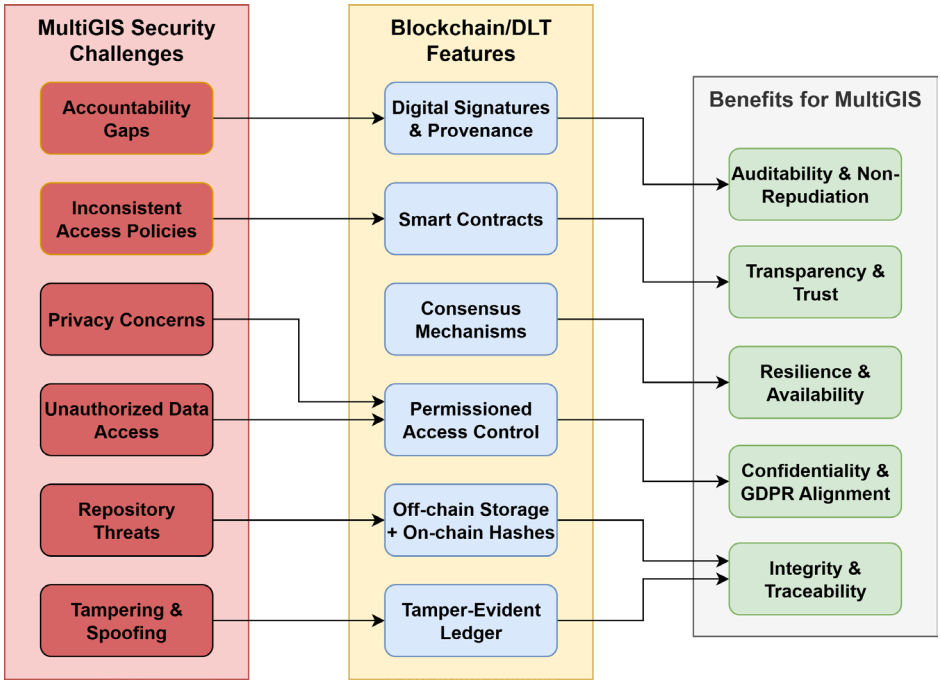

Figure 20: Role of Blockchain/DLT in addressing MultiGIS security challenges.

5.1.6 Consortium Permissioned Blockchain in MultiGIS

MultiGIS environments involve multiple agencies and organizations collaborating to collect, store, and share geospatial data. Such collaboration requires not only technical interoperability but also trust, accountability, and verifiable integrity across parties. Traditional security measures provide important safeguards, but they are often limited when data must move between administrative boundaries. Distributed Ledger Technology (DLT) , most commonly implemented as blockchain, offers a complementary solution by providing a tamper-evident, transparent, and shared record of geospatial data operations. The following points outline how blockchain contributes to security in MultiGIS contexts.

- **Integrity and Tamper-Evidence:** Blockchain keeps an append-only history of events. Each new record (a “transaction”) is grouped into a block that stores a cryptographic hash of its contents and a hash of the previous block, creating a chain where any change to past data becomes immediately detectable. In MultiGIS settings, this means that geospatial datasets, sensor readings, or imagery

registered to the ledger cannot be silently altered. Merkle roots can summarise many datasets or tiles, allowing efficient verification of whether a particular item belongs to a block without downloading everything. Every transaction is digitally signed by the submitting party, proving authorship and preventing forgery. Consensus among nodes fixes the order of blocks, preventing any single agency from rewriting the record of data collection or updates once confirmed.

- **Confidentiality in Permissioned Environments:** In a consortium blockchain, only authorised organisations and roles can read or write particular records. Access is governed by identities and policies, such as role-based rules or access control lists. Sensitive files do not need to be stored on-chain; instead, the ledger can hold only minimal metadata (identifiers, versions, hashes), while the full content remains in external GIS repositories under existing access controls. This on-chain minimality with off-chain content ensures that sensitive imagery or personal location traces remain protected, while integrity can still be verified.
- **Availability and Resilience:** The ledger is replicated across multiple nodes. Even if one agency's node fails, others continue to serve read requests and accept new confirmed entries. Consensus ensures that all participants converge on a single version of history, supporting service continuity for verification and audit. In MultiGIS disaster response scenarios, this resilience helps guarantee that critical provenance and access records remain available even under load spikes or partial outages.
- **Accountability and Governance:** Every ledger entry is digitally signed, providing non-repudiation. With timestamps and deterministic ordering, blockchain creates a clear sequence of events—who collected which data, when it was registered, and how it was shared. This accountability is essential for inter-agency coordination, compliance, and public transparency in geospatial data management.
- **Transparency and Verifiability:** A shared, append-only record lets any participant independently check claims about dataset versions, lineage, and timing. Compact manifests on the ledger (IDs, hashes, parent links) allow agencies to recompute a file's hash and confirm it matches the registered version without transferring large geospatial datasets. When transformations occur—such as aggregation, masking, or resampling—new entries can link to their sources, forming verifiable lineage that improves trust, reproducibility, and communication across institutions. This strengthens collaboration in MultiGIS by ensuring that data provenance is transparent and verifiable across different organisations.

In short, DLT augments MultiGIS infrastructures by offering tamper-evidence, resilient availability, fine-grained confidentiality, and accountable governance. While it does not remove the need for privacy-preserving methods such as geomasking or differential privacy, it provides the backbone for trustworthy multi-agency data collection, storage, and sharing.

5.2. Transparent and Verifiable Data Management-LUT

This subsection explains how a permissioned distributed ledger makes data use transparent (who used what, when, and under which terms) and verifiable (any participant can check the exact versions used) by recording compact manifests, lineage links, and policy states without copying large geospatial files (25)– (26). We list practical methods that a permissioned blockchain uses to deliver transparency and verifiability in geospatial workflows, then end with the concrete outcome.

- **Hash-anchored manifests (on-chain proofs, off-chain files):** In this approach, large files remain in conventional storage systems. For each version, a short manifest is recorded on the ledger with an identifier, a digital fingerprint (hash, or a summary hash for a group, called a Merkle root, a standard blockchain structure), a timestamp, links to input datasets (parents), key quality flags, and the

licence/consent/retention state. Later, any user can recompute the file's hash and compare it with the manifest; a difference indicates the file has changed. Where many files are registered together, a Merkle proof can be used to show that a file belongs to the recorded group without exposing every item. This pattern notarises geospatial content without exposing raw data (25)-(29).

- **Signed, time-stamped entries (accountability):** Each manifest and event is digitally signed by the writer's identity and time-stamped on the ledger. This results in a shared, append-only log of who did what, and when, which strengthens trust in multi-party work (29), (30).
- **Lineage links:** When a new dataset or model version is created, the manifest records which inputs were used—its parents—and the processing step that produced it. Here, a parent means any input artefact directly used to produce the new result (for example, input tiles or grids, a training dataset, a model version, or a code/parameter set). Over time, these links form a simple chain that readers can follow to see sources and methods. Because each parent and child has its own hash and timestamp on the ledger, others can check the exact inputs and reproduce the result (27), (26).
- **Verify-before-use:** Before a dataset is used in a model, map, or report, integrity is checked by computing its hash and comparing it with the hash stored in the ledger manifest. When many files were registered together, membership in the group's summary (the Merkle root) is verified. A manifest ID is the unique identifier of a version's manifest on the ledger; citing this ID in a report tells others exactly which version was used and allows them to fetch the same manifest, confirm the hashes, and repeat the run with the same inputs (25), (27).
- **Policy-aware transparency:** The manifest records the policy for that version, meaning the rules that govern how the data may be used and shared. Typical elements include the licence (e.g., CC BY 4.0), consent status, retention period, access scope, attribution requirements, purpose limitations, and any embargo dates. Access requests and approvals are logged as signed events. Where appropriate, small smart contracts (short programs on the ledger) enforce simple rules such as embargo windows or role-based gates (26), (30).
- **Minimal on-chain, permissioned visibility:** Only essential metadata are recorded on-chain; files remain off-chain under standard access control; ledger read/write access is limited to authorised participants. To support interoperability, records are aligned with practices from the Open Geospatial Consortium (OGC) and from Spatial Data Infrastructure (SDI) cataloguing, (26)-(28).

Participants reference artefacts by ID + hash (not copies); analyses cite manifest IDs; audits confirm versions and timing from the shared history. This gives a lightweight way to make data use both transparent and independently verifiable in MultiGIS-like settings (25)– (26).

5.3. Provenance and Integrity of Real-time Streams

Real-time spatial data streams in MultiGIS (e.g. sensor networks, UAV telemetry, satellite feeds) differ from stored or batch-collected data: they are continuous, high-velocity, often processed at the edge, and can arrive out of order or in bursts. Thus, provenance and integrity mechanisms must be lightweight, scalable, and sensitive to temporal ordering. While Sections 5.1–5.2 address integrity of stored datasets, Section 5.3 focuses on how to maintain temporal provenance and integrity of streaming spatial data.

5.3.1 Challenges Unique to Streaming MultiGIS Data

- Unlike stored datasets, where provenance can be established once data is archived and remains relatively fixed, real-time streams introduce a dynamic set of challenges. In streaming scenarios, provenance must be captured continuously and in sequence, while data is still in motion. This creates additional issues—such as ordering, replay attacks, aggregation at the edge, and clock synchronization—that are far less pronounced or even absent in static data collections. The main challenges include:
- **Ordering and continuity:** Streams may face network delays, packet reordering, or intermittent connectivity across nodes, complicating the guarantee of correct temporal sequence. Preserving order in distributed real-time data has been highlighted as a key challenge in real-time GIS and IoT systems (31), (32).
- **In-transit tampering, injection, or replay attacks:** An adversary might inject spoofed sensor readings, drop or replay windows, or reorder packets midstream to corrupt downstream analysis. In geospatial contexts, tampering during sensor data transmission directly affects the reliability of environmental monitoring and urban planning systems (33).
- **High throughput and scalability:** Recording provenance at per-event granularity on-chain is impractical under high event rates. Prior works note that large-scale real-time geospatial data requires compression or aggregation to remain computationally feasible (34).
- **Edge aggregation or filtering:** Many pipelines aggregate or compress data at edge gateways; provenance must account for those transformations (i.e., how to trace an aggregated window back to original events). This challenge is especially relevant in environmental sensor networks and UAV platforms (35).
- **Clock skew and synchronization:** Disparate nodes may have unsynchronized clocks; drift or misalignment must be tolerated or corrected to maintain coherent sequencing. Synchronization problems are well-known in distributed spatiotemporal data collection and can undermine provenance accuracy (36).

5.3.2 Provenance & Integrity Mechanisms with Blockchain Support

- To mitigate these challenges, the literature proposes hybrid schemes combining edge provenance capture and blockchain anchoring. The following mechanisms have been proposed to address them:
- **Window-based anchoring via Merkle roots**: Instead of anchoring every event, events are grouped into fixed or sliding time windows. A Merkle tree is built over events in the window, and the root hash is anchored onto the ledger. This enables lightweight proof of inclusion using Merkle proofs (37).
- **Chained window hashes & signatures**: Each anchored window manifest includes a hash pointer to the previous window's manifest. Combined with the source or gateway's digital signature, this chaining provides ordering assurance and prevents undetected omission or reordering (38).
- **Smart contract validation**: Smart contracts deployed on a consortium blockchain can enforce sequence continuity, validate device identities, and detect missing or duplicate anchors. Prior works highlight the role of smart contracts in automating provenance validation and accountability (39).
- **Hybrid on-chain / off-chain storage**: Bulk raw data remains stored off-chain in spatial databases, while the blockchain records only compact provenance manifests (window hashes, timestamps, signatures). This trade-off preserves scalability while anchoring trust (37), (26).

- **Edge / gateway provenance capture**: Trusted gateways preprocess events, derive hashes, sign manifests, and manage anchor submission. Recent blockchain–IoT frameworks emphasize combining edge provenance with blockchain anchoring to achieve both timeliness and trustworthiness (40).

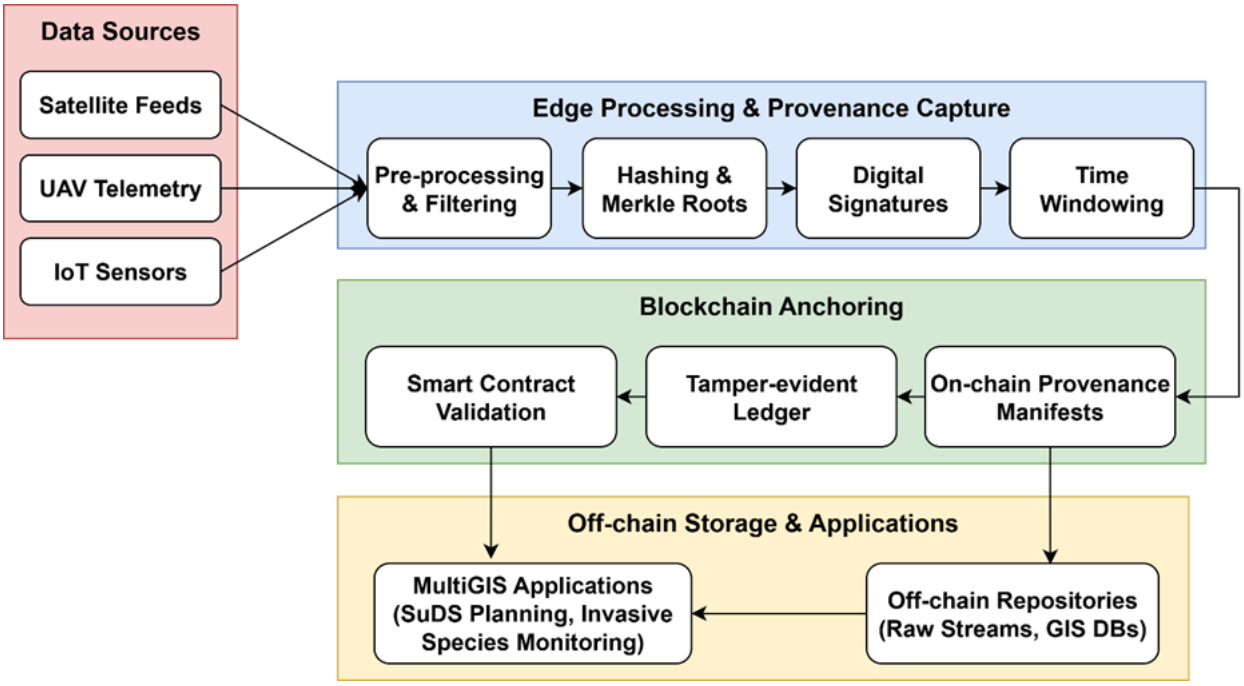

Figure 2: Blockchain-Enabled Provenance and Integrity in Real-Time MultiGIS Data Pipelines

5.3.3 Integration with AI4MultiGIS Use Cases

In MultiGIS scenarios, continuous spatial streams (e.g., UAV telemetry, environmental IoT feeds, satellite streams) can be anchored in windows via edge-level hashing and chained signatures. The consortium blockchain ensures tamper-evident anchoring and policy enforcement, while spatial analytics systems verify individual event integrity via Merkle proofs against the anchored manifest.

The architecture shown in Figure 2 captures this hybrid flow: data sources → provenance capture → blockchain anchoring → off-chain storage → MultiGIS applications. This ensures that real-time streams remain verifiable and accountable without sacrificing scalability.

5.4. Outlier Detection and Anomaly Management

5.4.1. An overview of Outliner

In geospatial and sensor analytics, an outlier is an observation that deviates markedly from expected behavior under a chosen data/feature model (e.g., neighborhood statistics, low-rank structure, or model residuals) (12) (16). An anomaly is broader: it includes outliers but also structural inconsistencies that arise from sensor faults, misregistration, or systematic process changes (e.g., telemetry integrity loss, clock drift, or geometry – radiometry mismatch) (25). In practice, the same datum may be an outlier under one context (local neighborhood) yet not globally extreme, hence anomaly detection must be context-aware (12) (16).

This distinction leads to different detection strategies depending on temporal behavior. Static anomalies are detectable from a single frame or batch (e.g., spectral spikes, isolated LiDAR noise points, or a corrupted packet (12), (13). Dynamic anomalies unfold over time or across streams—e.g., gradual sensor drift, temporal flatlines, inter-sample spikes, and timestamp/order inconsistencies in IoT feeds, or transient attitude/heading spikes in UAV telemetry (22), (24). Dynamic anomalies demand online or incremental methods that adapt to non-stationarity (e.g., recursive PCA, forgetting factors) (15). In addition to temporal patterns, anomalies can be classified spatially. Local anomalies violate spatial coherence or neighborhood consensus without necessarily being globally extreme (typical in satellite rasters near coasts/cloud edges, or in spatially irregular samples) (12), (15), (23). Global anomalies are extreme with respect to an overall distribution (e.g., very high Mahalanobis distance) but may still be legitimate rare events if supported by multi-sensor context (13). Distinguishing local from global is central to spatial outlier detection and to robust geospatial fusion (12).

These different anomaly types pose serious risks if left unaddressed. Undetected outliers bias gridded products and reconstructions (e.g., pseudo change from spectral spikes or cloud contamination), degrade multi-sensor fusion, and mislead downstream classification and forecasting (12), (13). In 3D mapping, tie-point and LiDAR outliers reduce planarity/inlier ratios, warp DSM/mesh geometry, and cause false planes or missed facets, harming building/asset models and linear-asset analytics (37), (40). For IoT, drift/flatlines/packet errors contaminate training sets and can mis-trigger real-time controls unless faults are separated from genuine events (14), (27). In GPS-denied or dynamic UAV scenarios, outliers in image matches or telemetry accumulate localization drift, increasing mission risk (17) , (25), (28). Moreover, early-stage anomalies can propagate across system layers. In event-driven, edge–cloud pipelines, early outliers propagate through resampling, aggregation, and learning loops, producing cascading error propagation and bias amplification that inflate storage and bandwidth while degrading model generalization and alert reliability (16), (26), (31). Robust, context-aware detection (spatial coherence, neighborhood consensus, recursive multivariate residuals) and traceable handling (flagging, quarantine, repair) are therefore essential to maintain integrity across acquisition, fusion, and analytics (15), (18), (23),. To make this discussion more concrete, the next sections will analyze outliers in the context of four representative data types: satellite imagery, UAV mapping, LiDAR scanning, and IoT sensor streams. Figure 1 summarizes the main sources and corresponding impacts of outliers in each modality.

5.4.2. Sources and Impacts of Outliers in MultiGIS Data Pipelines

5.4.2.1. Satellite-Based Data

Sub-visible or edge clouds, rain streaks, and land–water adjacency introduce local radiometric outliers such as anomalous brightness, temperature, or chlorophyll, and create discontinuous textures. Band striping, path-radiance error, and sensor noise yield pixel-level extremes that may pass per-pixel quality control yet violate neighborhood smoothness (16). These artefacts bias reconstructed fields and multi-sensor composites, generate pseudo-change near strong gradients, and degrade classification, front detection, and subsequent data assimilation, which in turn weakens downstream modeling and decision support (12), (13). Multi-source and multi-temporal rasters exhibit sub-pixel to multi-pixel shifts, most apparent along coastlines, oceanic fronts, and urban edges, producing spatial inconsistency and apparent changes unrelated to the underlying process (12), (13). Misregistration propagates systematic error into cross-sensor fusion, distorts boundaries and statistics, and contaminates training data used for analytics and simulation, reducing reliability of the integrated workflow (12), (13).

As noted above, radiometric anomalies that escape pixel-level checks contaminate reconstructions and multi-sensor fusions. This undermines product comparability and weakens data assimilation, and the resulting bias propagates into spatiotemporal prediction and anomaly monitoring within WP4 (16). Spectral noise and radiometric spikes around sharp boundaries further induce pseudo change, which degrades land-cover classification and the detection of geophysical fronts such as sea surface temperature and chlorophyll fronts (12), (13). Geolocation error introduces systematic registration bias into satellite, unmanned aerial vehicle, and LiDAR fusion, distorting boundaries and statistics and lowering the reliability of synthetic data in D3.2 and the training sets used in D4.2 (16), (33). At the governance layer, encoding structural anomalies as relational or constraint rules, such as land–water adjacency conditions, enables traceable quality control and targeted human review across the lifecycle (19).

5.4.2.2. UAV-Based Mapping

Short spikes in heading or attitude, wrap-around discontinuities on cyclic angles such as 359° to 0°, step-like position jumps, and brief integrity loss that later materialize as georegistration errors. Causes include sensor overload, electromagnetic interference, vibration, packet loss, and transient Global Navigation Satellite System degradation; cyclic variables invalidate naïve linear thresholds (17). Navigation drift corrupts exterior-orientation priors and degrades aerial triangulation and alignment, leading to mis-georeferenced imagery, unstable bundle adjustment, and increased mission risk in real-time operations (17). Motion blur and low texture reduce reliable tie points; repetitive patterns in canopies or façades increase false matches; parallax and occlusion introduce inconsistent correspondences that propagate into warped digital surface models and blunders in dense point clouds (21). The mapping products exhibit surface distortions, local collapses, and gaps; asset-level analytics become unreliable. In inspections and localization, these outliers reduce retrieval accuracy and robustness under GPS-denied conditions, and raise the cost of manual correction (21).

Outliers in feature matching reduce inlier ratios and destabilize bundle adjustment, which leads to surface deformation, mesh warping, and scaling errors, especially in scenes with repetitive spectral or textural patterns such as forests (21). Around wires and pylons, residual blunders translate into missed clearance alarms and false vegetation encroachment warnings, weakening the reliability of inspection workflows (29). Training data for WP4 are affected when noisy matches and mislabeled regions enter point-cloud datasets, reducing the generalization of three-dimensional and segmentation models under aerial viewpoints (27). Telemetry outliers in coordinates, velocities, and angles compromise the integrity of guidance loops and raise mission risk when they remain undetected (17). In visual localization, insufficient landmark curation and the absence of outlier-aware scale recovery cause the accumulation of drift and incorrect associations in dynamic environments.

5.4.2.3. LiDAR Scanning

Airborne or UAV LiDAR contains isolated high noise from birds or particulates, low noise from ranging or trajectory errors, and clustered outliers in dense urban or vegetated scenes. Density varies with platform motion, scan geometry, canopy penetration, strip overlaps, and linear assets such as wires, producing anisotropic sampling that defeats global thresholds (17). Neighborhood statistics and planarity assumptions are biased, surface and roof segmentation quality declines, and digital elevation or surface products inherit artefacts that affect building and infrastructure modelling. Multipath and mixed returns create points below the true surface and distort edges near strong height discontinuities such as eaves and ridges, inducing spurious planes and warped roof patches during plane fitting and region growing. Small or thin roof facets are missed, boundaries are over-segmented, and additional refinement is required to remove false planes, lowering completeness and correctness of the extracted structures (22).

As summarized earlier, LiDAR outliers depress planarity and inlier ratios and trigger false planes and missed small facets during roof extraction, while ridge and edge zones accumulate artefacts that reduce the completeness and correctness of building models (22). Negative returns below the true surface confuse terrain filters that assume the lowest surface is ground, which introduces digital elevation artefacts and propagates into clearance and encroachment analyses for linear infrastructure (17). Noise driven by density variation and edge distortions also contaminates the labels used for three-dimensional segmentation and classification, weakening model generalization and increasing the cost of subsequent cleaning (17). In practice, common filters such as statistical outlier removal still involve significant trade-offs and require manual rework, indicating that full automation is not yet achievable in many operational settings (17).

5.4.2.4. IoT Sensor Streams

Gradual drift due to thermal or humidity sensitivity and component aging shifts baselines. Wireless interference, congestion, and duty cycling cause missing or duplicated packets and desynchronization, visible as inconsistent values, gaps, and jittered time stamps (29), (30). Faulty streams trigger false alarms or mask true events, bias forecasting and anomaly models, and inflate bandwidth and storage through retries and duplication, thereby degrading service quality in real-time applications (29), (38). Flatlines indicate stuck sensors or frozen interfaces; spikes arise from electromagnetic interference or transient link errors; out-of-order or duplicated time stamps and drifting clocks result from intermittent connectivity and unsynchronized nodes (29), (30). Windowed analytics and cross-sensor correlation become unreliable; event ordering and latency guarantees are broken, which undermines alerting, control policies, and the integrity of datasets used for training and evaluation (29), (30).

Faulty streams mis-trigger actuators and alerts in domains such as environmental monitoring and traffic control, which degrades service quality and safety when detection is not conducted online (24). Persistent drift and flatlines contaminate training and evaluation data, harming the generalization of forecasting and anomaly models in WP4; distinguishing sensor fault from genuine event is essential to avoid discarding true signals (24). Spikes and duplications inflate bandwidth and storage and increase the cost of downstream cleaning, while pipelines that lack edge-side filtering amplify error propagation across stages. Traceability also suffers when timestamps are missing or incorrect; recording anomaly flags and repair actions is required to sustain auditability and reproducibility across acquisition, fusion, and analytics (14).

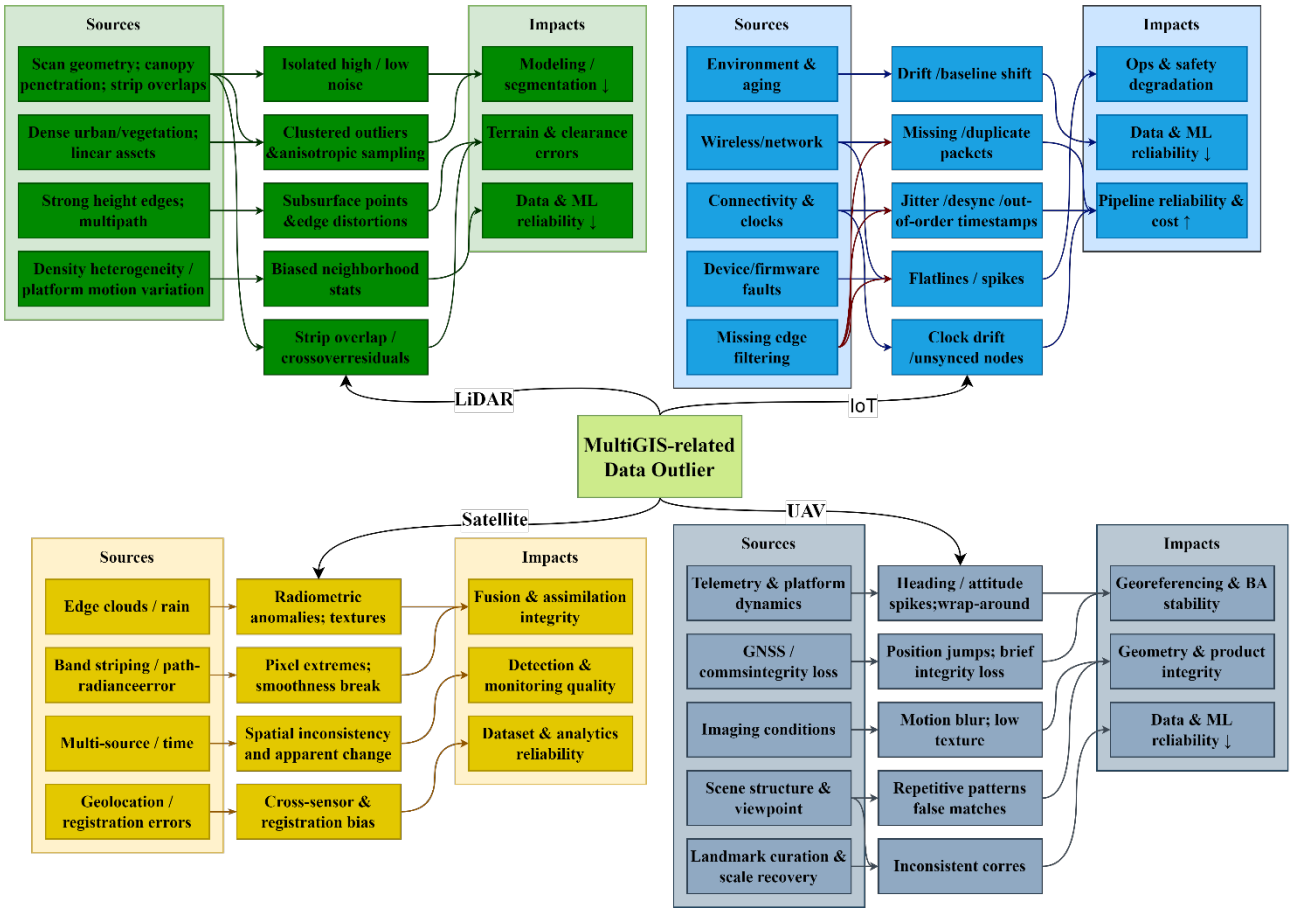

Figure 22: Sources and impacts of outliers in MultiGIS data.

5.4.3. Challenges in Outlier Detection

5.4.3.1. Outlier Detection Challenges in Satellite Imagery

Traditional detection methods suffer from modality-specific limitations. Statistical filters such as z-score and MAD (23) rely on intensity distribution assumptions that do not hold in scenes with overlapping histograms, such as between shadows and dark targets. Mis-scaled analysis windows easily smooth out small but real objects, resulting in false positives. Model-fitting techniques like RANSAC and robust regression assume local planarity and often favour dominant planes, discarding small, non-coplanar structures as outliers.Transform-based methods, including PCA and robust low-rank decompositions (28), treat background deviations as anomalies, but cloud shadows often share spectral features with valid ground objects. In time-series data, such models absorb genuine variation into the low-rank background, flagging meaningful change as noise. Density-based and clustering methods like DBSCAN and LOF (31) rely on stable local density metrics that break down in texture-poor regions such as water or asphalt. Their parameters are sensitive to land cover and sensor characteristics, making them hard to generalize. These limitations show that traditional methods require constant tuning of thresholds and analysis scales for each scene. Illumination instability and texture sparsity remain fundamental bottlenecks that limit transferability and automation (23).

5.4.3.2. Outlier Detection Challenges in UAV

UAV outlier rejection still depends on handcrafted thresholds and geometric consensus. Methods like robust estimators and RANSAC remain standard, but in high-resolution, repetitive scenes (e.g., tiled roofs, forest canopies), false consensus is common, causing true structures to be discarded (21). Advanced variants (e.g.,

MAGSAC, DEGENSAC) improve robustness but still assume dominant planes or motions, often missing thin structures and requiring flight-specific tuning. Linear infrastructure tasks, such as powerline extraction, rely on height thresholds, PCA, RANSAC, and Hough transforms (39), which fail under clutter, reflective surfaces, or anisotropic viewpoints. Shifting overlap and terrain further destabilize local statistics, making threshold-based rejection brittle. Georeferencing errors from POS/GNSS introduce spatially coherent artefacts mistaken as outliers. Rolling shutter, blur, and radiometric inconsistency confuse geometric filters. Even learning-based depth predictions suffer under oblique angles, occlusion, and lighting variation, shifting outlier patterns from sparse matching to dense prediction errors.

5.4.3.3. Outlier Detection Challenges in LiDAR

LiDAR filtering faces structural issues that persist across platforms and terrains. Global elevation rules and histogram-based thresholds often fail in complex relief or urban areas, especially when point distributions become multimodal due to slope, vegetation, or buildings. Local surface fitting assumes geometric regularity, but point density variations and anisotropy degrade neighborhood statistics, corrupt normals, and increase false detections near edges. Plane fitting methods like PCA and RANSAC are sensitive to noise and biased toward dominant surfaces. As a result, small yet valid structures are discarded as outliers, while near-coplanar clutter is retained, creating false splits and loss of details. Morphological filters and segmentation-first strategies struggle on curved or intricate roofs, where point density drops and geometry becomes irregular. In addition, upstream imperfections such as strip adjustment errors, miscalibration, and inconsistent intensity values introduce residuals that are hard to distinguish from true outliers. Multi-temporal acquisitions further complicate filtering, as real change interacts with registration noise, making it difficult to separate anomaly from evolution. These structural limitations explain the lack of generalizability across sites and the continued reliance on manual correction.

5.4.3.4. Outlier Detection Challenges in IoT

Outlier detection in IoT environments is constrained by the nature of sensor data, limited computation, and evolving context. Many methods assume univariate or stationary data, but real-world sensor streams are multivariate, noisy, and affected by temperature, humidity, and device degradation. Traditional approaches often ignore spatial and temporal correlations, leading to poor discrimination between true anomalies and environmental effects. Resource constraints on memory, power, and bandwidth prevent deployment of complex models at the edge. Lightweight methods must balance detection accuracy with computational feasibility. Furthermore, IoT deployments are highly heterogeneous and dynamic, so models trained offline often fail to adapt. Across common method families, statistical, clustering, proximity-based, neural/fuzzy, and classification, each shows limitations. Statistical techniques depend on known distributions, LOF and DBSCAN are sensitive to density assumptions, fuzzy systems scale poorly, and classifiers need labeled data and constant updates (14).

5.4.5. Methods for Outlier Detection and Management

5.4.5.1. Goals and Scope

The framework covers four data layers in a unified way. Raster includes satellite imagery, orthomosaics, hyperspectral data and terrain or intensity products. Three-dimensional point clouds include LiDAR and structure from motion. Vector and node data include fixed and mobile sensors, control points and network graphs. Cross modal and relational checks focus on consistency across sensors, time and spatial relations. The goal is traceable outlier labels and clear handling actions before any domain modelling. Emphasis is on online and incremental processing, representation learning, graph based spatiotemporal fusion and

calibration. Figure 2 illustrates the proposed outlier detection and management pipeline, along with its key outputs.

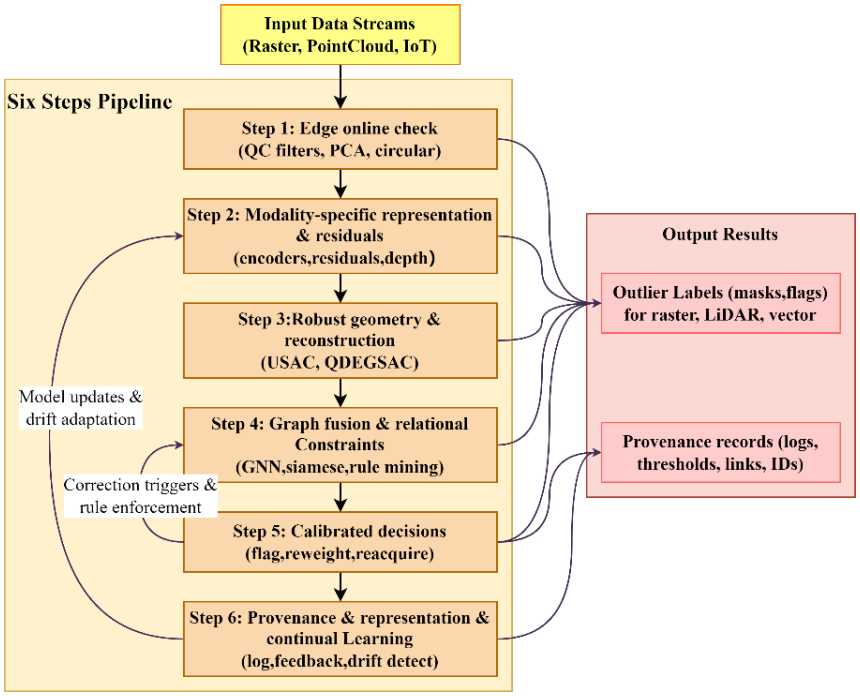

Figure 23: Outlier detection and management pipeline.

5.4.5.2 Pipeline with Six Steps

Step 1. Edge online screening: Basic checks run at the source. Range, rate of change, flatlines, spikes, missing records and timestamp order are verified. Circular statistics with a k sigma rule handle variables from 0 to 360 degrees. Recursive PCA updates online at cluster level and yields squared prediction error residuals for fast flags with short context windows.

Step 2. Modality adaptive representation and residuals: Self-supervised encoders build robust features per modality. Raster uses masked or contrastive encoders. Point clouds use point level and block level encoders. Node time series use a lightweight Transformer or an RNN. A monocular depth prior is added for oblique UAV video using depth and pose networks with two image encoders and a 3D CNN decoder trained by view reconstruction and contrastive loss. Depth consistency residuals are computed against SfM or MVS and against LiDAR or DSM where available. Hyperspectral scenes use low rank and sparse matrix decomposition with cluster weighting. The low rank term models background and the weighted sparse term gives a stable spectral spatial score. Physically consistent residuals are prepared for fusion using phase correlation, mutual information and structural similarity for registration, point to plane distance with normal and curvature mismatch for geometry, and node to node plus node to raster residuals for networks.

Step 3. Robust geometry and reconstruction with a unified USAC core: A learning based wide baseline matcher reduces false correspondences before estimation. USAC unifies sampling, pre scoring, adaptive thresholds, degeneracy checks and early stopping. MAGSAC is the default scorer and termination rule so noise scale is marginalized and accuracy improves. QDEGSAC handles planar or quasi degenerate scenes with layered constraints and automatic model selection. Rolling shutter sequences use normalized weight rolling shutter bundle adjustment to refine pose and structure. Point cloud neighborhoods use an anisotropic search

by projecting neighbours to a Gaussian sphere and clustering. The largest single surface cluster supports normals, denoising and fitting.

Step 4. Spatiotemporal graph fusion with relational and semantic constraints: A spatiotemporal graph neural network aggregates messages across neighbours and time to produce consensus and propagation consistency scores. Deep feature outlier anchors come from intermediate embeddings of a pretrained segmentation backbone using Isolation Forest and receive higher weights in cross time and cross sensor checks. Siamese or contrastive change encoders classify new, missing, deformed and displaced objects and fuse with residual evidence from previous steps. Spatial and relational rules such as adjacency, containment and distance thresholds are mined and violations are stored as structural anomaly candidates.

Step 5. Calibrated decisions: Thresholds follow target quantiles per region and season. Conformal prediction produces p values and intervals for transfer across domains. Actions are flag and isolate, re register or reconstruct when geometry residuals are high, re acquire when integrity is low, down weight uncertain samples, and impute from models or neighbours for short gaps. All decisions are logged with confidence and links to affected objects.

Step 6. Provenance and continual learning: Every decision is recorded with anomaly type, threshold, parameters, masks, identifiers, time and notes. Input links and model versions are stored for full reproducibility. A feedback loop updates encoders and small classifiers with semi supervised and weakly supervised learning. Incremental training tracks season and sensor drift. The backbone stays stable when needed and only heads are fine tuned. Edge modules can be replaced without changing the core. Drift metrics trigger recalibration.

5.4.5.3 Minimal modality configuration with equal weight and plug in design

Raster including hyperspectral: Apply quality masks. Run self-supervised encoders and compute registration residuals. Use low rank and sparse decomposition with cluster weighting for a spectral spatial anomaly score. Add robust statistics in boundary buffers. Use EOF or DINEOF as a background prior when texture is weak. **Three-dimensional point clouds:** Build point and block embeddings and geometric residuals. Reduce mismatches with a learning based wide baseline matcher. Estimate models with USAC and MAGSAC. Enable QDEGSAC for quasi degenerate cases. Use anisotropic neighborhoods to improve normals, denoising and fitting while keeping sharp edges. **Vector and node networks:** Run recursive PCA for online aggregation and detection with squared prediction error residuals. Check neighbour correlation and upstream downstream lag correlation. Use circular statistics for heading and other angular variables. **Cross modal, relational and semantic checks:** Use self-supervised monocular depth for oblique UAV video to compute depth consistency residuals. Mine deep feature anchors and re score with a spatiotemporal graph neural network. Learn spatial and relational rules and verify them together with the graph score.

Raster including hyperspectral: Apply quality masks. Run self-supervised encoders and compute registration residuals. Use low rank and sparse decomposition with cluster weighting for a spectral spatial anomaly score. Add robust statistics in boundary buffers. Use EOF or DINEOF as a background prior when texture is weak. **Three-dimensional point clouds:** Build point and block embeddings and geometric residuals. Reduce mismatches with a learning based wide baseline matcher. Estimate models with USAC and MAGSAC. Enable QDEGSAC for quasi degenerate cases. Use anisotropic neighborhoods to improve normals, denoising and fitting while keeping sharp edges. **Vector and node networks:** Run recursive PCA for online aggregation and detection with squared prediction error residuals. Check neighbour correlation and upstream downstream

lag correlation. Use circular statistics for heading and other angular variables. **Cross modal, relational and semantic checks:** Use self-supervised monocular depth for oblique UAV video to compute depth consistency residuals. Mine deep feature anchors and re score with a spatiotemporal graph neural network. Learn spatial and relational rules and verify them together with the graph score.

5.4.5.4. Outputs

Outlier masks and labels. Pixel masks for raster data. Sparse anomaly matrices and cluster weights for hyperspectral scenes. LAS or LAZ with keep, remove and uncertain flags for point clouds. Strip and crossover quality reports and anisotropic neighbourhood logs. Geometric suspicious blocks. R PCA squared prediction error labels, down weight factors and missing value masks for nodes. Graph consistency scores, embeddings from the learning-based change encoder and a list of violated rules. Provenance and versions. A full record is kept for each run. The record includes configuration and version of encoders, graph model, USAC core, MAGSAC scorer, QDEGSAC module, self-supervised monocular depth model and LRaSMD settings. Thresholds, calibration targets, time span and identifiers of all affected tiles, blocks, and nodes are included. This enables complete traceability and audit.

6. Monitoring, Feedback, and System Adaptation

The AI4MultiGIS pipeline integrates continuous monitoring and adaptive feedback mechanisms to ensure reliable execution of ETL workflows, high-quality data integration, and consistent system behavior across heterogeneous geospatial sources. Although the current project phase primarily relies on static datasets (vector, raster, and tabular), these mechanisms build upon the ingestion, orchestration, and traceability components previously described to track workflow health, validate data integrity, and use execution feedback to improve robustness, data quality, and long-term adaptability across Pilots 1 and 2.

6.1. Continuous Monitoring of ETL Processes

All ingestion and preprocessing operations—whether carried out through dedicated Python scripts, Airflow operators, or NiFi flows—are subject to comprehensive monitoring to ensure reliability and data integrity. Each script generates detailed execution logs, capturing every step of the workflow, including data reading, transformation, and loading, as well as key metrics such as row counts, detected geometry types, missing fields, and coordinate reference systems. In parallel, workflow supervision dashboards provided by Airflow and NiFi offer real-time visibility into task performance, including execution duration, failures, retries, and scheduling behavior, enabling operators to quickly assess the health of the pipeline. Additionally, container health monitoring through Docker ensures that services such as PostGIS, Airflow, NiFi, and processing modules run in isolated and reproducible environments, while providing continuous tracking of computational and memory resource usage. This multi-layered monitoring framework allows for the rapid detection and diagnosis of issues, such as corrupted files, schema mismatches, missing metadata, or formatting deviations, thereby supporting robust, reliable, and auditable data processing across the pipeline.

6.2. Feedback Loops for Data Quality Assurance

The pipeline incorporates iterative feedback loops to maintain data quality during ingestion and harmonization. These loops rely on:

6.2.1. Automated Data Checks

To ensure the integrity and reliability of spatial and tabular data, the pipeline incorporates a comprehensive validation framework. For vector and raster datasets, the system verifies coordinate reference system (CRS) consistency, ensuring that all layers are spatially compatible for integration and analysis. Vector geometries undergo validity checks across types such as MultiPolygon, MultiLineString, and Point, with automated repair procedures such as applying a zero width buffer to correct invalid geometries when necessary. Raster datasets are subject to metadata validation, including assessments of resolution, band count, spatial extent, and alignment with other layers. For tabular data, the pipeline performs schema checks to confirm the presence of mandatory fields and automatically resolves inconsistencies in column naming. This rigorous validation process guarantees that all datasets entering the PostGIS database are accurate, consistent, and ready for downstream analysis, minimizing errors and ensuring the reproducibility of results.

6.2.2. Ingestion Outcome Reporting

Following each ETL execution, the pipeline generates **comprehensive summary reports** that document key processing outcomes. These reports include metrics such as the number of processed features or raster tiles, any coordinate reference system (CRS) transformations applied, detected anomalies, and warnings regarding missing or non-standard attributes. Additionally, the reports record the corrective actions automatically executed by the preprocessing scripts. By providing this detailed feedback, the reports serve as a practical tool for project partners, helping them to refine data delivery formats, address inconsistencies in upstream sources, and ensure that subsequent datasets are better aligned with the pipeline's standards. This feedback loop enhances data quality, streamlines collaboration, and supports continuous improvement of the overall ETL workflow.

6.2.3. Human-in-the-Loop Review

In cases where anomalies cannot be automatically resolved—such as incomplete attribute schemas, ambiguous field names, or data gaps—expert review is required.
This validation step is especially important for Pilot 1, where provider-specific Shapefile formats exhibit greater heterogeneity.

6.3. Performance Monitoring and System Visibility

The operational status of the pipeline is continuously monitored through a combination of tools and techniques to ensure reliability and efficiency. **Airflow monitoring utilities**, such as Gantt charts, task logs, and failure alerts, provide visibility into workflow execution, enabling the rapid identification of bottlenecks or failed tasks. **PostGIS performance metrics** track critical indicators including database storage growth, index utilization, and query performance, ensuring that spatial data operations remain efficient even as dataset volumes increase. **QGIS visual verification** allows partners to inspect the spatial alignment of layers, validate geometry correctness, and review the integration of merged outputs, providing a practical means of assessing data quality. Additionally, **Docker resource monitoring** maintains oversight of CPU, memory, and disk usage across the containerized processing environment, ensuring stability and optimal resource allocation. Collectively, alerts and logs produced by these monitoring systems promote **transparency, reproducibility, and rapid response**, allowing the pipeline to handle integration challenges effectively and maintain high operational standards.

6.4. System Robustness and Fault Tolerance

Although the current datasets are delivered in batches rather than continuous streams, the pipeline architecture is specifically designed to handle the types of failures that commonly occur in large-scale geospatial processing. **Automatic task retry and fallback routines within Airflow** ensure that transient errors or failed executions do not disrupt overall workflow progress. The system employs **checksum-based file verification** to detect and flag corrupted data deliveries before they propagate through the pipeline. By running each processing task within **isolated containers**, the architecture prevents failures in one component from cascading and affecting other services. Additionally, **redundant storage of intermediate files** allows the pipeline to recover from partial failures without requiring the reprocessing of entire datasets, limiting computational overhead and time delays. Together, these mechanisms provide robust fault tolerance, ensuring that integration continues uninterrupted even when input data are irregular, incomplete, or temporarily unavailable.

6.5. Pilot-Based Illustrations of Monitoring and Feedback

6.5.1. Pilot 1 — Sustainable Drainage Systems (SuDS)

Pilot 1 deals with highly heterogeneous datasets, including Shapefiles, GeoTIFFs, and provider-specific formats, which necessitate careful monitoring throughout the ingestion and preprocessing stages. During processing, the system identified several key issues that required attention. These included **coordinate reference system (CRS) inconsistencies** between different Shapefile packages, as well as **invalid geometries** in flood-risk polygon layers, which were automatically corrected during the ETL workflow. **Raster datasets** exhibited variations in metadata, such as differences in resolution and spatial extent, which were normalized to ensure compatibility across layers. Tabular and vector datasets also presented **schema irregularities**, including missing fields and non-standard naming conventions, which were addressed during preprocessing. **Visual validation in QGIS** confirmed that after transformation, the layers—including rivers, flood-risk zones, road networks, and raster elevation models—overlaid correctly, demonstrating the effectiveness of the preprocessing, harmonization, and validation processes in creating an integrated and reliable geospatial dataset for further analysis.

6.5.2. Pilot 2 — Multi-Indicator Hazard and Socioeconomic Data

Pilot 2 utilizes more uniform datasets, including GeoTIFFs, GeoJSON files, and Excel or CSV tables, which simplifies monitoring and enhances the clarity of feedback. During processing, the system detected several minor issues that were addressed automatically. **Excel and CSV files** occasionally contained missing fields or exhibited inconsistent column naming, which were standardized during preprocessing. **GeoJSON layers** required geometry normalization to ensure consistency between feature types, such as converting MultiPolygons to Polygons when necessary. Additionally, **raster datasets** displayed slight mismatches in spatial extent across different indicators, which were resolved through clipping to the project area. Thanks to the standardized nature of these data formats, the ingestion and preprocessing scripts could operate more generically, allowing anomalies to be detected and corrected automatically with minimal manual intervention. This streamlined workflow demonstrates the advantages of using well-structured and consistent datasets for efficient and reliable geospatial integration.

6.6. Continuous Improvement and System Adaptation

The **Ai4MultiGIS architecture** has been intentionally designed to accommodate evolution and expansion as additional datasets, use cases, and partner requirements emerge. To support this adaptability, the system incorporates **regular audit reviews** of execution logs, ETL performance metrics, and PostGIS indexing behavior, ensuring that performance bottlenecks or inefficiencies are identified and addressed proactively. **Preprocessing scripts are refined iteratively** whenever new datasets are integrated, allowing the pipeline to maintain compatibility and robustness across diverse data sources. The architecture also supports the **modular onboarding of new ingestion workflows**, enabling the integration of additional data streams without disrupting existing pipelines or analytical operations. Comprehensive **documentation-driven adaptation** ensures that all project partners can understand, reuse, and extend the unified processing environment with minimal effort. Together, these mechanisms guarantee the long-term sustainability, scalability, and flexibility of the MultiGIS data integration ecosystem, allowing it to evolve in line with project growth and emerging analytical needs.

7. Conclusion

This deliverable has presented the conception, design, and implementation of the Ai4MultiGIS data-processing pipeline, establishing a unified and reproducible framework for integrating heterogeneous geospatial datasets across multiple project pilots. The work undertaken addresses a central challenge in the MultiGIS initiative: enabling partners to ingest, validate, harmonize, and store diverse raster, vector, and tabular datasets within a consistent, transparent, and automated workflow.

Throughout the document, several key contributions have been achieved:

- **A Modular and Reproducible Pipeline Architecture**

  The pipeline has been designed around containerized services (PostGIS, Airflow, NiFi, and dedicated preprocessing scripts), ensuring consistent deployment across environments and simplifying collaboration among project partners. The modular structure enables each step—from ingestion to storage—to evolve independently while maintaining overall coherence.

- **Standardized Processing of Heterogeneous Geospatial Data**

  One of the major accomplishments is the ability to process a wide variety of inputs, including Shapefiles, GeoJSON, GeoTIFFs, and Excel/CSV tables. Through unified preprocessing strategies, the pipeline transforms pilot-specific data into harmonized spatial layers ready for analysis and visualization. This capability is essential for building a robust MultiGIS backbone, given the variability of data formats across providers.

- **Automated Workflow Management and Traceability**

  Using Airflow and structured logging, the system ensures transparent and traceable ETL processes. Each transformation step—including schema harmonization, CRS verification, geometry cleaning, raster validation, and database loading—is recorded and monitored. This contributes to reliability, reproducibility, and easier debugging throughout the integration lifecycle.

- **Continuous Monitoring and Quality Feedback**

  The pipeline incorporates monitoring mechanisms that detect anomalies such as CRS mismatches, invalid geometries, missing fields, and inconsistent metadata. Feedback loops—both automated and expert-driven—support iterative refinement of scripts, improved data preparation by partners, and overall enhancement of dataset quality. These mechanisms were particularly valuable in Pilot 1, where data heterogeneity was more pronounced.

- **Demonstrated Applicability Across Two Pilot Scenarios**

  The successful application of the pipeline to Pilot 1 (SuDS) and Pilot 2 (Hazard and Socioeconomic Indicators) confirms the architecture's flexibility.

  - In Pilot 1, the pipeline handled diverse and inconsistent Shapefile and raster collections, correcting numerous structural and geometric anomalies.
  - In Pilot 2, the more standardized datasets allowed a generic, reusable processing approach, showcasing the pipeline's capacity to scale efficiently to well-organized data environments.

- **Foundations for Future Expansion**

  The pipeline developed in D3.1 provides the core infrastructure upon which future stages of Ai4MultiGIS will build. Thanks to its modularity, new preprocessing modules, validation routines, analytical components, or data connectors can be integrated without disrupting existing workflows. This creates a solid foundation for the upcoming phases focused on advanced analytics, trust and security, and integrated multi-pilot operations.

In summary, this deliverable establishes the first functional layer of the Ai4MultiGIS ecosystem: a robust, transparent, and interoperable data pipeline capable of supporting multi-domain geospatial integration. Its implementation across two pilots demonstrates both the maturity of the architecture and its potential for expansion. The structured workflows, quality controls, and harmonized data outputs produced here will serve as the basis for the advanced AI-driven tools, predictive models, and cross-pilot analyses developed in the next stages of the project.

References

1. **apache.** https://nifi.apache.org/https://nifi.apache.org/. [En ligne]

2. ***https://sedona.apache.org/latest/.*** **[En ligne]**

3. ***https://postgis.net/.*** **[En ligne]**

4. ***https://airflow.apache.org/.*** **[En ligne]**

5. ***https://jupyter.org/.*** **[En ligne]**

6. ***https://docs.docker.com/compose/.*** **[En ligne]**

7. ***https://nifi.apache.org/docs/nifi-docs/*****. [En ligne]**

8. ***https://sedona.apache.org/latest/tutorial/files/geoparquet-sedona-spark/*****. [En ligne]**

9. ***https://postgis.net/documentation/*****. [En ligne]**

10. ***https://airflow.apache.org/docs/*****. [En ligne]**

11. ***https://docs.jupyter.org/en/latest/*****. [En ligne]**

12. ***McKenzie, G., Keßler, C., & Andris, C. (2019). Geospatial privacy and security. Journal of spatial information science, (19), 53-55.***

13. ***Zope-Chaudhari, S., & Venkatachalam, P. (2013). Conceptual framework for geospatial data security. International Journal of Database Management Systems, 5(5), 29.***

14. ***Marri, R., Varanasi, S., Chaitanya, S. V. K., & Marri, S. K. (2024). Strengthening GIS security: Anonymization and differential privacy for safeguarding sensitive geospatial data. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023,.***

15. ***Seidl, D. E., Jankowski, P., & Clarke, K. C. (2018). Privacy and false identification risk in geomasking techniques. Geographical Analysis, 50(3), 280-297.***

16. ***Fernholz, Y., Freidank, J., Zhu, J., Ivanov, I., & Kox, T. Ethics in GIS: A Systematic Analysis focusing on Privacy and Surveillance. GI_Forum 2024,, 12, 1-14.***

17. ***Atluri, V., & Chun, S. A. (2004). An authorization model for geospatial data. IEEE Transactions on Dependable and Secure Computing, 1(4), 238-254.***

18. ***Rawal, D., Amaduzzi, S., & Seedorf, J. (2024). Crypto-Spatial: A New Direction in Geospatial Data. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 48, 89-96.***

19. ***Bertino, E., Thuraisingham, B., Gertz, M., & Damiani, M. L. (2008, November). Security and privacy for geospatial data: concepts and research directions. In Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS a.***

20. ***Marri, R., Varanasi, S., Chaitanya, S. V. K., & Marri, S. K. (2024). Enhancing security in geographic information systems: Anonymization and differential privacy techniques for protecting sensitive geospatial data. Journal of Artificial Intelligence Gener.***

21. ***Qiu, Y., Long, J., Ma, C., Luo, J., & Xiao, Q. (2024). Security protection of video-GIS data based on data encryption and digital watermarking. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 48, 681-688.***

22. ***Curtis, A., Mills, J. W., & Leitner, M. (2006). Keeping an eye on privacy issues with geospatial data. Nature, 441(7090), 150-150.***

23. ***Ahmad, S. (2023). Exploring geomasking methods for geoprivacy: a pilot study in an environment with built features. Geospatial health.***

24. ***Marri, R., Varanasi, S., Chaitanya, S. V. K., & Marri, S. K. (2024). Enhancing security in geographic information systems: Anonymization and differential privacy techniques for protecting sensitive geospatial data. Journal of Artificial Intelligence Gener.***

25. ***Sharma, P., Martin, M., & Swanlund, D. (2023). MapSafe: A complete tool for achieving geospatial data sovereignty. Transactions in GIS, 27(6), 1680-1698.***

26. ***CHAFIQ, T., Rida, A. Z. M. I., Fadil, A., & Mohammed, O. (2024). Investigating the potential of blockchain technology for geospatial data sharing: Opportunities, challenges, and solutions. Geomatica, 76(2), 100026.***

27. ***Zhao, P., Cedeno Jimenez, J. R., Brovelli, M. A., & Mansourian, A. (2022). Towards geospatial blockchain: A review of research on blockchain technology applied to geospatial data. AGILE: GIScience Series, 3, 71.***

28. ***Tahar, A., Mendy, G., & Ouya, S. (2024, February). Efficient and Optimized Geospatial Data Representation in Blockchain-Based Land Administration. In International Congress on Information and Communication Technology (pp. 519-535). Singapore: Springer Nat.***

29. ***Farnaghi, M., & Mansourian, A. (2020). Blockchain, an enabling technology for transparent and accountable decentralized public participatory GIS. Cities, 105, 102850.***

30. ***Kumar, N., Chohan, D. K., Maurya, V., Singh, N., Kumar, I., & Agrawal, K. K. (2024, September). Managing the Geo-Spatial Data using Block chain. In 2024 International Conference on Communication, Computing and Energy Efficient Technologies (I3CEET) (pp. 1.***

31. ***Li, W., Batty, M., & Goodchild, M. F. (2020). Real-time GIS for smart cities. International Journal of Geographical Information Science, 34(2), 311-324.***

32. ***Roostaei, J., Wager, Y. Z., Shi, W., Dittrich, T., Miller, C., & Gopalakrishnan, K. (2023). IoT-based edge computing (IoTEC) for improved environmental monitoring. Sustainable computing: informatics and systems, 38, 100870.***

33. ***Neisse, R., Steri, G., & Nai-Fovino, I. (2017, August). A blockchain-based approach for data accountability and provenance tracking. In Proceedings of the 12th international conference on availability, reliability and security (pp. 1-10).***

34. ***Hamdi, A., Shaban, K., Erradi, A., Mohamed, A., Rumi, S. K., & Salim, F. D. (2022). Spatiotemporal data mining: a survey on challenges and open problems. Artificial Intelligence Review, 55(2), 1441-1488.***

35. ***Frincu, M., Penteliuc, M., & Spataru, A. (2022). A solar radiation forecast platform spanning over the edge-cloud continuum. Electronics, 11(17), 2756.***

36. ***Franchi, F., Graziosi, F., Di Fina, E., & Galassi, A. (2023). A survey of cloud-enabled gis solutions toward edge computing: Challenges and perspectives. IEEE Open Journal of the Communications Society, 5, 312-331.***

37. ***Honar Pajooh, H., Rashid, M. A., Alam, F., & Demidenko, S. (2021). IoT Big Data provenance scheme using blockchain on Hadoop ecosystem. Journal of Big Data, 8(1), 114.***

38. ***Liang, X., Shetty, S., Tosh, D., Kamhoua, C., Kwiat, K., & Njilla, L. (2017, May). Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability. In 2017 17th IEEE/ACM International Symposium on Clu.***

39. ***Zou, Q., Yu, W., & Bao, Z. (2023). A Blockchain solution for remote sensing data management model. Applied Sciences, 13(17), 9609.***

40. ***Gao, Z., & Yan, W. (2025). The real-time data processing framework for blockchain and edge computing. Alexandria Engineering Journal, 120, 50-61.***

41. ***Seidl, D. E., Jankowski, P., & Clarke, K. C. (2018). Privacy and false identification risk in geomasking techniques. Geographical Analysis, 50(3), 280-297.***