AI Integrated Framework for Intelligent Geospatial Handling and Robust Operation in MultiGIS Applications (Ai4MultiGIS)

Topic: Multidimensional Geographic Information Systems (MultiGIS)

AI4MultiGIS

D2.2

AI4MultiGIS Reference Architecture and Technical specification

Work Package	2
Delivery Date	M6
Responsible Partner	UPPA
Authors	Akram Hakiri (UPPA) Prabhat Kumar (LUT) Ankit Agrawal (LUT)
Contributors	Akram Hakiri (UPPA) Prabhat Kumar (LUT) Ankit Agrawal (LUT) Richard Chbeir (UPPA)
Distribution	*Project Level*
Version	0.1

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:	D2.2
Related WP:	WP 2
Due Date:	M6

Editor:	Akram Hakiri
Contributor(s):	Akram Hakiri, Richard Chbeir, Prabhat Kumar, Ankit Agrawal
Reviewer(s):	UVT
Approved by:	All partners

Executive Summary

The Ai4MutiGIS project aims to offer an integrated framework for MultiGIS data generation and management, aiming to enhance the overall GIS capabilities for optimized processing chain of MultiGIS applications. This deliverable, D2.1, encompasses a comprehensive analysis of the state of the art with the areas that are within the scope of the project.

The deliverable outlines the current landscape of MultiGIS and tools that are used to enhance the overall GIS capabilities in terms of data collection, generation, storage, and real-time data processing from a set of diverse heterogeneous sources.

The deliverable also investigates the adoption of Artificial Intelligence (AI) and their potential for the MultiGIS application. Specifically, AI is already widely considered to support various capabilities of GIS.However, there are challenges involved in the responsible for the usage of AI enabled application for MultiGIS. Finally, the documents outline the challenges in the adoption of AI in MultiGIS domain and provides future research direction.

Table of Contents

List of Acronyms and Abbreviations

A2A	Agent-to-Agent
CNNs	Convolutional Neural Networks
DLT	Distributed Ledger Technology
DSS	Decision Support System
DT	Digital Twin
FCNs	Fully Convolutional Networks
FL	Federated Learning
FedMARL	Federated Multi-Agent Reinforcement Learning
GANs	Generative Adversarial Networks
GNNs	Graph Neural Networks
GeoAI	Geospatial Artificial Intelligence
KPIs	Key Performance Indicators
LLMs	Large Language Models
LRP	Layer-wise Relevance Propagation
LSTM	Long Short-Term Memory
MARL	Multi-Agent Reinforcement Learning
MAUP	Modifiable Areal Unit Problem
NeRFs	Neural Radiance Fields
OGC	Open Geospatial Consortium
QC	Quality Control
QGIS	Quantum GIS
QNNs	Quantum Neural Networks
RAG	Retrieval-Augmented Generation
RNNs	Recurrent Neural Networks
SAGIN	Space-Air-Ground Integrated Networks
SAR	Synthetic Aperture Radar
SSL	Self-Supervised Learning
ST-GCNs	Spatio-Temporal Graph Convolutional Networks
TCNs	Temporal Convolutional Networks
TSDBs	Time Series Databases
UI	User Interface
WSNs	Wireless Sensor Networks
XAI	Explainable AI

1 Introduction

1.1 Purpose and Scope of the project

The growing complexity of geospatial data demands an intelligent, secure, and decentralized infrastructure capable of effectively supporting synthetic data generation, real-time multimodal data collection, automated processing, and trustworthy data management. To manage MultiGIS data in a decentralized manner, sophisticated edge-cloud infrastructures are essential for handling synthetic data generation, real-time multimodal data collection, and automated processing pipelines, ensuring reliable geospatial analysis. Edge computing enables low-latency processing of diverse sensor data, while cloud platforms provide the scalability needed for AI-driven analytics, simulations, and large-scale data integration (Zhang et al. 2020). Automated pipelines facilitate the consolidation of data, the detection of anomalies, and the development of predictive models. To ensure trust and security, blockchain-based decentralization guarantees data integrity, transparency, and access control, strengthening MultiGIS resilience against cyber threats (Ahrens et al. 2021) (Wang et al. 2022). This approach provides scalable, secure, and real-time geospatial intelligence for critical applications such as smart cities, disaster response, and digital twin systems.

The AI4MultiGIS project has been designed to tackle these increasing challenges by implementing a cutting-edge, unified framework for data collection and processing, AI-powered analytics (Gama et al. 2019), blockchain-based decentralization (Dastjerdi et al. 2016), and automated data pipelines. The project introduces novel solutions to address the current challenges surrounding real-time multimodal geospatial data management, security, and scalability, while also preparing for future demands involving autonomous decision-making and advanced geospatial analysis. This is achieved through the seamless integration of blockchain technology for secure data exchange, smart contracts for automated trust management (Gama et al. 2019), multimodal large language models (LLMs) for synthetic data generation, and retrieval-augmented generation (RAG) prompting engineering to support complex decision-making (Qiu et al. 2021). Furthermore, knowledge representation techniques are utilized to enhance the contextual understanding of dynamic environments, ensuring that the MultiGIS ecosystem is more adaptive, resilient, and intelligent.

The AI4MultiGIS framework provides a foundation for future-proofing geospatial systems, addressing the critical need for innovation in both technical solutions and strategic approaches to data management. By fostering secure, scalable, and intelligent data pipelines, the project aims to advance the development of next-generation geospatial applications that benefit from real-time analytics, predictive capabilities, and seamless automation.

1.2 Contribution to other Deliverables

The deliverable D2.2 presents a comprehensive and in-depth review of the overall architecture and technical specifications of AI4MultiGIS, defining both functional and non-functional requirements. It lays out the high-level technical architecture, mapping key components and their interactions to ensure seamless alignment with project requirements and specifications. This architecture serves as the foundation for WP3, guiding the development of core components, while also facilitating their integration and interoperability within WP4, ensuring a robust, scalable, and efficient AI-driven MultiGIS ecosystem.

1.3 Structure of the Document

The structure of this deliverable is as follows:

- **Section 2** provides a high-level conceptual overview of the AI4MultiGIS architecture, detailing its core components, including data generation, management, integration, and spatiotemporal analysis tools.

- **Section 3** outlines the technical specifications of AI4MultiGIS, focusing on real-time data collection, synthetic data generation, automated processing, decentralized data management, and GIS integration powered by advanced computing and AI-driven analytics.
- **Section 4** defines the AI4MultiGIS Reference Architecture, covering key principles, high-level design, and component interactions, ensuring scalability, interoperability, and seamless system integration.
- **Finally**, the deliverable concludes with a summary and references.

2 AI4MultiGIS: Conceptual Overview

2.1 Overview

The development of the AI4MultiGIS framework aims to to deliver an integrated framework which aims to optimise the processing chain of MultiGIS data to support robust operation of GIS-enabled applications and services. Furthermore, AI4MultiGIS aims to ensure transparency, privacy, fairness, and relevant regulatory compliance for the AI-enabled applications. To achieve this, AI4MultiGIS develops novel methods, techniques, and tools for data collection, real-time spatiotemporal data processing, geostatistical analysis and cross-model/multi-modal data interpretation with plug and play capabilities, and policy to govern the assurance of responsible AI for AI-enabled MultiGIS applications.

2.2 key components

There are two main components in the AI4MultGIS:

2.2.1 Novel Customisable, and Trustworthy Data Generation and Management

This component is responsible for enhancing the collection, generation, processing, and management of geospatial data within the MultiGIS environment. The design of this component involves real-time data collection, processing and integration, synthetic data generation using either graph knowledge and multimodal LLMs to improve data representativity and edge-to-cloud geospatial Digital Twin (DT) to enhance predictive analysis. Besides, the framework involves automated outlier detection and data consolidation to enable reliable data processing. It will also involve blockchain, decentralized ledgers, and intelligent contracts (smart contracts) to enable data management.

The Intelligent real-time data collection role is to create an intelligent ecosystem that collects high-volume, multi-modal data in real-time using IoT sensors, edge-cloud devices, and adaptive AI-based scraping techniques. The goal is to automate the data collection process while optimizing its efficiency. It uses AI to process the data on the edge, refine it, and integrate it seamlessly into the MultiGIS application, allowing for continuous improvement based on real-time feedback. The event-driven architecture ensures that only relevant data is collected as the system evolves.

The synthetic data generation for improved MultiGIS representation focuses on generating synthetic datasets using advanced techniques like GANs (Generative Adversarial Networks) and multimodal LLMs (Large Language Models). The goal is to simulate space occupancy, movement patterns, and sensor data. The component will also use Neural Radiance Fields (NeRFs) to generate synthetic imagery. Additionally, an edge-to-cloud geospatial Digital Twin (DT) will be developed to augment real-time MultiGIS data with synthetic data, enhancing predictive analytics. This component enables simulation and scenario modelling to improve the accuracy and coverage of geospatial representations.

The purpose of the automated outlier detection and data consolidation for reliable data processing is to ensure that the data collected and generated maintains high quality. It involves multiple methods to assess data quality, detect outliers (using adaptive AI models), and consolidate the data into a unified, trustworthy MultiGIS dataset. The integration of Bayesian and Ensemble methods helps quantify uncertainty, while the human-in-the-loop approach allows users to refine AI models based on feedback. The outcome is a high-integrity dataset that can be used for more reliable analysis.

The component will also be creating a decentralized data management system using blockchain and Distributed Ledger Technology (DLT) to ensure the security, traceability, transparency, and auditability of geographic data, which is essential for maintaining trust in the MultiGIS ecosystem. The system uses blockchain, distributed storage, and interoperable frameworks to ensure secure, transparent, and efficient geospatial data management. By decentralizing the storage of data and enforcing data quality through the DLT, the system reduces transaction overhead, improves data integrity, and enhances the long-term interpretability of geospatial records. This system will be critical for supporting the data consolidation efforts.

2.2.2 Integrated Data Handling and Modular Tools for Spatiotemporal Analysis

This component is responsible for the design of Federated Multi-Agent Reinforcement Learning (FedMARL) model to enhance edge computing with spatiotemporal data processing locally, without compromising privacy and minimizing data transmission. Additionally, this component will use the services and datasets developed in the previous component to create a GIS-based system that integrates cross-model and multi-modal data sources, involving linking climate models, urban growth models, infrastructure models, and socio-economic models to provide a comprehensive, multidimensional view of an area's resilience, development, or sustainability. It will also be creating plug-and-play tools that users can easily plug into the GIS platform. To this end, the component will require the creation of robust API that allow users to focus on building a flexible, scalable architecture that facilitates easy integration of external models, data sources, and algorithms. There are several approaches to integrated APIs to MultiGIS, which should include key features to support either RESTful or GraphQL API. The former uses web services for providing stateless communication and easy integration with other web services. The latter provides more flexibility by allowing clients to request specific data and aggregate data from different sources in a single request.

3 AI4MultiGIS Individual Component Technical Specification

3.1 Component 1: Novel Customisable, and Trustworthy Data Generation and Management

Figure 1 present the main components that will be implemented in component 1. AI4MultiGIs will implement a comprehensive approach that combines enhanced geospatial data handling, edge-to-cloud digital twin (DT) integration, and blockchain-enabled data governance. It will support real-time collection, processing, and integration of geospatial data, enriched through synthetic data generation using graph knowledge and multimodal large language models (LLMs). By integrating geospatial digital twins across edge and cloud environments, AI4MultiGIs will enable predictive analytics, automated outlier detection, and reliable data consolidation. Furthermore, the project will adopt decentralized ledgers and smart contracts to establish secure, transparent, and intelligent governance of geospatial information.

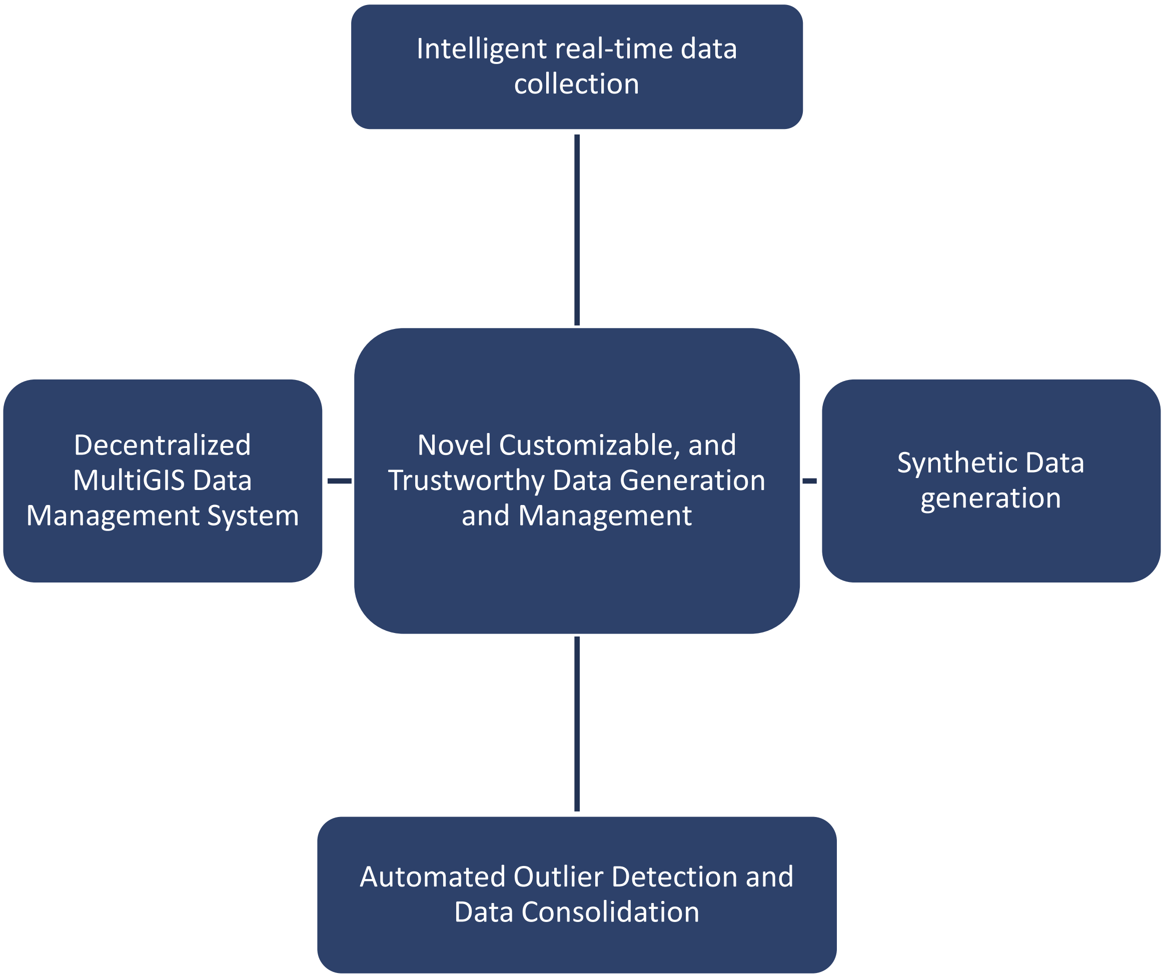

Figure 1: Overview of the Core Component for Customizable and Trustworthy Data Generation and Management, with Supporting Subcomponents.

3.1.1 Overview of the intelligent real-time data collection

MultiGIS integrates multiple data sources to provide comprehensive geospatial analysis. The advancement of intelligent, real-time data collection in GIS is driven by the increasing demand for high-precision, dynamic, and adaptive geospatial information (Zhang & Li, 2023). This progress involves leveraging AI, IoT, and real-time analytics to enhance decision-making across various applications, including urban planning, environmental monitoring, transportation, and disaster management (Chen et al., 2022).

To enable intelligent, real-time data collection, several technologies and methodologies must be integrated. In particular, MultiGIS should be capable of aggregating multimodal data sources from diverse origins to improve spatial analysis. These sources include remote sensing data, such as satellite imagery, UAV (drone) footage, and LiDAR scans, which provide high-resolution raster information for environmental monitoring (Smith & Kumar, 2023). Additionally, IoT and sensor networks (such as LoRaWAN, GPS trackers, and weather stations) generate real-time spatial data that must be processed efficiently. IoT-enabled edge devices perform local AI processing before transmitting key information to GIS databases (Chen et al., 2022). Open Data APIs from platforms such as OpenStreetMap, along with user-generated geospatial data, are also collected in real time, and multispectral sensors contribute to high-resolution mapping through Earth observation data from satellites like Sentinel, Landsat, or private satellite networks (Zhang & Li, 2023).

Furthermore, as geospatial data come in various formats, scales, and resolutions, robust data pipelines—such as those illustrated in Figure 2—are essential for the integration, processing, and analysis of these diverse datasets. These pipelines provide a holistic view of spatial phenomena by extracting, transforming, and charging different types of spatial data to deliver accurate insights, a challenge well documented in recent research on disaster management and urban analytics (Jones et al., 2022).

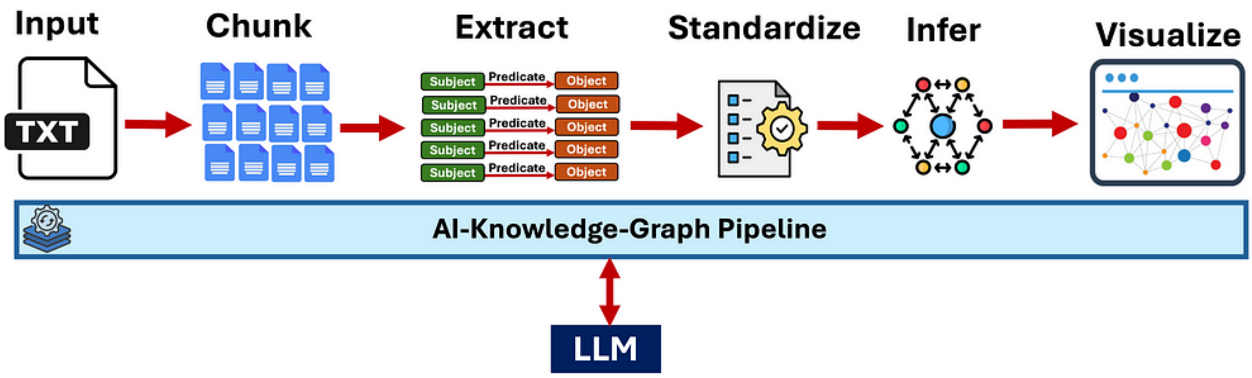

Figure 2: Data Pipeline Preprocessing

3.1.2 Synthetic Data generation for Improved MultiGIS Representation

Multimodal Large Language Model (LLM) can be leveraged to create high-fidelity, synthetic data, realistic, and context-aware synthetic datasets that improve the representation of MultiGIS systems. This requires AI-driven data augmentation, simulation-based synthetic data generation, and deep learning-based feature fusion to enhance geospatial applications in smart cities, environmental monitoring, and disaster management.

Building a framework for syntactic data generation should include the following functionalities:

- **Multimodal Input Data Processing**

Multimodal input data processing involves integrating and analysing data from different modalities or sources, such as text, images, video, audio, and sensor data, to provide a more comprehensive understanding of a situation. This approach is often used in AI and machine learning tasks to enhance decision-making and predictions by leveraging the strengths of different data types. It involves the followings:

- **Data Fusion**: Combining heterogeneous data sources (e.g., satellite imagery, IoT sensors, social media) (Chen 2022) using AI/ML techniques to create comprehensive, standardized geospatial models. Automated feature extraction (roads, buildings, vegetation) supports urban planning and environmental monitoring. The framework will leverage convolutional neural networks (CNNs) to improve the processing and interpretation of satellite and drone imagery, allowing for precise land cover classification, infrastructure detection, and real-time disaster monitoring. This will enhance responsiveness to natural events like floods or deforestation through faster data analysis and actionable insights. By embedding AI and ML directly into GIS workflows, AI4MultiGIS will enable more accurate predictive modelling and anomaly detection, improving decision-making in urban planning, environmental monitoring, and beyond. Furthermore, AI4MultiGIS will also adopt state-of-the-art data fusion techniques at signal, feature, and decision levels to handle heterogeneous geospatial datasets. It will integrate multimodal remote sensing data (data from LiDAR, Synthetic Aperture Radar (SAR), satellite images, optical imagery, and spatiotemporal sources) into deep learning methods such as CNNs and fully convolutional networks (FCNs) to improve urban mapping, land use classification, and overall geospatial data integration (Salcedo-Sanz 2020). Besides, Feature-level fusion will be implemented for satellite data fusion along with aerial data fusion for land use and land cover mapping.

- **Representation Learning:** AI4MultiGIS will develop advanced representation learning techniques to create robust, shared embeddings that unify diverse geospatial and multimodal data within a common semantic space (Manzoor2023). It will implement state-of-the-art self-supervised learning (SSL) methods, leveraging data-driven supervision to learn powerful representations without relying on labelled datasets, thus enabling efficient knowledge extraction across various geospatial tasks. The framework will integrate generative models such as diffusion models as part of its self-supervised learning toolkit, enhancing the ability to capture complex data distributions across multiple modalities (e.g., imagery, text, sensor data). AI4MultiGIS will also incorporate multimodal and multilingual representation learning approaches to align and reason across heterogeneous data types, such as satellite images, socio-economic text, and audio, facilitating richer, cross-domain insights within the GIS context. Furthermore, to ensure robustness and privacy (Yaodong2024), AI4MultiGIS will embed adversarial robust and privacy-preserving learning techniques, safeguarding models against inference attacks and enabling trustworthy deployment in sensitive geospatial applications. A core innovation will be the fusion of Large Language Models (LLMs) with Graph Neural Networks (GNNs) (Wu 2024) to jointly model textual, structural, and multimodal geospatial data. AI4MultiGIS will implement architectures similar to GL-Fusion, LensGNN, and other cutting-edge models to embed graph structures directly into transformer layers or align graph embeddings with language tokens, achieving superior understanding and generalization in complex spatial networks. Moreover, AI4MultiGIS will pioneer methods to handle heterogeneous graphs by dynamically processing diverse node and edge types with LLM-GNN bootstrapping techniques, enhancing representation learning for varied geospatial entities and relationships without heavy prior assumptions. By adopting a principled taxonomy of LLM-GNN integration strategies, AI4MultiGIS will optimize model design to balance knowledge extraction, organization, and training, positioning itself at the forefront of multimodal, multilingual, and multiscale geospatial representation learning. This will empower the framework to reason across data modalities and scales, driving smarter, more interpretable, and resilient geospatial AI solutions.
- **Multimodal LLM-based Synthetic Data Generation**

AI4MultiGIS will implement cutting-edge multimodal large language model (LLM)-based synthetic data generation to transform geospatial analytics, addressing key challenges like data sparsity, privacy, and domain generalization. It will develop scalable, high-fidelity synthetic datasets that integrate diverse data types, e.g., text, images, spatial coordinates, and sensor streams, enabling richer, ethically sourced geospatial data for urban resilience, climate adaptation, smart cities, and logistics. AI4MultiGIS will generate synthetic spatial narratives, imagery, and sensor logs that augment training data for AI/ML tasks including land cover classification, traffic prediction, and disaster response. This approach enhances privacy protection and supports simulation of rare or under-observed events, such as floods, earthquakes, and urban fires, thereby improving model robustness and decision-making in sensitive contexts.

Besides, this module is connected to the previous data processing module, as it uses the upstream structured synthetic data generation pipeline to help in encoding multimodal inputs from GIS platforms, e.g., QGIS, allow generating synthetic datasets that should be validated downstream by the digital twins and stored/processed in interoperable data format like GeoJSON or CityGML. To this end, the AI4MultiGIS framework is expected to allow geaspatical data augmentation API, embedding learning tools like Earth2Vec and CLIP, spatial ML libraries such as PyTorch Geometric and DeepGIS, and immersive digital twin platforms integrating Unity3D and Cesium with LLM APIs.

- **Spatial-Temporal Data Fusion**

AI4MultiGIS will incorporate advanced spatial-temporal data fusion techniques to integrate and analyse multimodal geospatial data across both space and time. This capability is fundamental to supporting real-time decision-making in applications such as environmental monitoring, disaster management, smart cities, autonomous mobility, and climate adaptation. By fusing spatial data (e.g., satellite imagery, GPS coordinates) with temporal data (e.g., sensor logs, historical records), AI4MultiGIS enables the understanding and prediction of dynamic phenomena over both geographic and temporal dimensions. At its core, AI4MultiGIS will use geospatial databases, time-series analysis, and deep learning architectures, such as Spatio-Temporal Graph Convolutional Networks (ST-GCNs) introduced by Yu et al. (2018), to model traffic flow and mobility patterns from real-time sensors and historical data. Similarly, Temporal Convolutional Networks (TCNs), as studied by Hao et al. (2020), will be used for forecasting in sequential geospatial data, enabling accurate predictions of urban dynamics or environmental changes. Beyond structured sources, AI4MultiGIS will fuse heterogeneous data streams, including unstructured content like social media. For example, Zhang et al. (2023) show the integration of social media and satellite imagery for urban disaster response, illustrating how multimodal fusion improves crisis situational awareness. Evangelos et al. (2021) also demonstrate the combination of satellite imagery, in-situ sensor data, and meteorological forecasts to support precision agriculture, a method that AI4MultiGIS will extend for climate-sensitive crop planning.

- **Validation and Quality Control**

AI4MultiGIS will implement a robust validation and quality control (QC) framework to ensure the integrity, accuracy, and reliability of spatial-temporal data fusion. Given the heterogeneity of input sources. the project will develop advanced data cleaning techniques to detect and correct errors, handle missing values, and resolve inconsistencies across modalities. To guarantee data correctness, AI4MultiGIS will integrate automated validation procedures (Dmitry 2021) that assess data conformity with expected patterns and real-world benchmarks (e.g., verifying GPS data). It will also establish procedures for aligning and reconciling multi-source datasets, ensuring they accurately reflect the spatial and temporal phenomena being modelled.

AI4MultiGIS will also apply cross-validation techniques (e.g., k-fold and leave-one-out) (Meijier 2013) to assess the generalizability of fusion models across time and space. Furthermore, it will utilize statistical error metrics, such as RMSE, MAE, etc. to quantitatively evaluate the accuracy of fused outputs compared to ground-truth data. Finally, the project will implement spatial and temporal consistency checks to ensure coherent integration of heterogeneous data, preventing logical conflicts and supporting high-quality downstream analytics.

- **MultiGIS Representation and Integration**

AI4MultiGIS will integrate multiple layers of geographic information across spatial, temporal, thematic, socio-economic, and environmental domains to provide a comprehensive, multi-dimensional representation of geographic space. This integration will support advanced applications such as urban planning, environmental monitoring, and resource management.

To achieve this, AI4MultiGIS will:

- Develop a flexible data representation framework supporting diverse formats (vector, raster, hybrid) to encode geographic features and continuous phenomena.
- Implement advanced visualization tools to display dynamic changes over time using animated maps and interactive layers.
- Incorporate thematic mapping techniques (e.g., heatmaps) to represent and analyze attribute data such as land use, pollution, and population density.

AI4MultiGIS will bring cutting-edge innovation to geospatial data processing by implementing advanced spatial-temporal integration pipelines that align time-series data and unify coordinate systems. Through the use of sophisticated geospatial indexing methods such as R-trees and quad-trees (Lee 2024), the platform will enable efficient querying and retrieval of massive spatial datasets. To ensure semantic consistency across diverse sources, AI4MultiGIS will apply semantic data integration techniques, crafting ontologies and taxonomies that harmonize concepts and terminologies. Harnessing the power of cloud platforms, it will support scalable, real-time processing and collaborative analytics via Cloud GIS technologies. In parallel, the project tackles critical integration challenges: overcoming data heterogeneity through format and quality harmonization, enabling real-time data fusion for dynamic scenarios such as disaster response and traffic monitoring, and ensuring scalability via distributed computing infrastructures. Furthermore, AI4MultiGIS emphasizes interoperability, seamlessly bridging sensors, satellite data, and third-party services to deliver a robust, unified geospatial intelligence platform. Through these developments, AI4MultiGIS will provide a unified, scalable, and intelligent geospatial data infrastructure capable of supporting decision-making across multiple domains.

3.1.3 Automated Outlier Detection and Data Consolidation for Reliable Data Processing

The Automated Outlier Detection and Data Consolidation software component is designed to streamline data processing workflows by identifying anomalous data points (outliers) and ensuring that the data used for analysis and decision-making is consolidated, accurate, and reliable. This component plays a critical role in preparing raw data for further use in systems such as machine learning models, predictive analytics, and business intelligence. That is, to ensure the reliability and robustness of the data pipeline, this component focuses on the implementation of automated outlier detection mechanisms coupled with intelligent data consolidation techniques. The objective is to create a data pre-processing layer that ensures high data fidelity before any real-time analytics or optimization processes are applied. Hence, this component plays a critical role in preparing raw data for further use in systems such as machine learning models, predictive analytics, and business intelligence.

To implement the core features of this component within the AI4MultiGIS framework, we can leverage recent advancements in geospatial artificial intelligence (GeoAI). Key core features that should be implemented in this component are described in the remaining of this section.

Core Features:

- **Outlier Detection**

A robust anomaly detection engine will be implemented to automatically identify inconsistent or suspicious data points originating from faulty sensors, communication errors, or environmental interferences. The system may leverage geospatial AI (GeoAI) techniques to implement a robust anomaly detection engine to automatically identify inconsistent or suspicious data points originating from faulty sensors, communication errors, or environmental interferences. Recently, Self-Supervised Anomaly Detection (SeMAnD) transforms multimodal geospatial data into semantically meaningful tensors and uses self-supervised learning to detect geometric anomalies. Such an approach has seemed to be effective in identifying local defects in vector geometries such as roads and buildings. It allows identifying **geometric anomalies** in multimodal geospatial datasets.

AI4MultiGIS will develop semantically meaningful tensor representations from heterogeneous data sources (e.g., satellite imagery and vector layers) and use approache similar to SeMAnD to identify subtle and impactful anomalies. For example, it can integrate tailored data augmentation strategies, such a RandPolyAugment, to enrich training samples by synthetically altering vector geometries (e.g., shifting, distorting polygons) in order to enhance the model robustness. Besides, to tackle challenges such as inconsistent spatial resolutions, varying temporal frequencies, and data sparsity or redundancy, AI4MultiGIS transforms multimodal inputs into semantically aligned, image-like tensors, enabling effective cross-modality comparison and fusion. It further mitigates noise and uncertainty through self-supervised objectives that detect non-correlated local variations while filtering redundant information, ensuring high-fidelity anomaly detection and more reliable geospatial data fusion.

- **Data Consolidation**

The goal of the data consolidation phase is to create a unified, coherent, and high-quality geospatial dataset by integrating multiple heterogeneous data sources. This step ensures that the downstream analytical tasks, such as modelling, prediction, and decision-making, which are grounded in accurate and contextually consistent information. Specifically, AI4MultiGIS framework will integrate geospatial data from a variety of formats and modalities, such as satellite imagery (raster), IoT sensor outputs (time series), and vector maps (shapefiles, geoJSON), into a single and interoperable dataset. The framework will implement multimodal data fusion and consolidation techniques, such as those based on CNN and GNN, to extract relevant features from each data type, normalize them, and fuse them into a common representational space. Recent approaches in data consolidation implement either data-level fusion, feature-level fusion, or decision-level fusion. data-level fusion combines raw data sources where compatible (e.g., merging multiple raster layers with the same resolution). Conversely, feature-level fusion extracts and combines features from each modality to form a composite input for machine learning models. Meanwhile, decision-level fusion integrates outputs from independently trained models to produce robust, consensus-based results.

One interesting technique to enhance data consolidation is semantic alignment (Bordogna et aL 2023), which connects diverse geospatial data into a meaningful, interoperable, and trustworthy system. Semantic alignment (Foerster et al. 2019) enables more consistent querying and retrieval of geospatial information, regardless of the source or format. It significantly reduces ambiguity and inconsistency in the representation of features, which is particularly crucial in large-scale data fusion. Moreover, it facilitates interoperability across systems and domains, making it easier to integrate, compare, and analyse data from disparate sources, whether from different satellites, sensor networks, or national mapping agencies.

The goal of using semantic alignment in AI4MultiGIS framework is to preserve and harmonize the meaning and context of data elements across diverse sources. This is essential for enabling accurate, context-aware geospatial analysis, particularly in multimodal or multitemporal data fusion environments. It enables seamless integration, accurate analysis, and scalable decision-making across space, time, and domains. To achieve semantic consistency, several strategies can be employed. One key method is ontology-based alignment, which leverages established geospatial ontologies like GeoSPARQL or INSPIRE. These frameworks help define and align concepts across datasets, i.e., ensuring for example that vegetation cover extracted from satellite imagery semantically matches equivalent land use classes in vector-based maps. Another technique involves semantic embedding methods, such as Word2Vec or BERT, applied to geospatial metadata. These embeddings are used to recognize and reconcile synonyms, abbreviations, and conceptual variations in annotations, enabling more intelligent linking of similar features labeled differently across datasets. Additionally, AI-assisted schema mapping is employed to automate the process of detecting schema equivalences or required transformations. This includes identifying matching variables with different names, units, or structures using machine learning and logical inference.

- **Quality Control**

To ensure that AI4MultiGIS delivers actionable, reliable, and high-integrity geospatial intelligence, quality control is embedded into the data lifecycle (Zhao et al. 2020). This includes automated quality assessment powered by AI and metadata tagging for transparency and traceability of data reliability across the system. The AI4MultiGIS platform addresses a critical challenge in geospatial intelligence systems: ensuring the reliability and usability of heterogeneous spatial and sensor data before it feeds into analytical or decision-making pipelines. Two core functionalities have been developed to this end: automated quality assessment and metadata tagging. The automated quality assessment component integrates supervised and unsupervised AI models to evaluate incoming data streams. Supervised classifiers are trained on labelled datasets to verify structural and semantic consistency in data types such as satellite imagery, IoT sensor feeds, and vector layers. Complementarily, unsupervised techniques like clustering and autoencoders detect anomalies, including outliers, data dropouts, or unnatural spatial transitions, which may otherwise introduce bias or noise in real-time analytics. Furthermore, spatial-temporal validation mechanisms flag inconsistent sensor readings and suspicious patterns using rule-based and pattern-recognition systems. Fusion-based cross-validation and Bayesian error estimation methods, inspired by recent advances in uncertainty quantification in geospatial AI (Samuel et al., 2025), help quantify prediction confidence and maintain robustness in emergency scenarios like flood modelling.

- **Integration and Scalability**

AI4MultiGIS is designed with a modular and loosely coupled architecture, enabling seamless integration of heterogeneous data sources, ranging from satellite imagery and real-time IoT sensor feeds to socio-economic and urban infrastructure datasets. Each module (data ingestion, preprocessing, AI modelling, visualization) operates independently yet communicates via standardized APIs. This design follows the direction outlined by (Temitope 2024), who emphasized the role of autonomous and extensible GIS agents in enhancing spatial analysis workflows. The modularity supports the plug-and-play integration of emerging AI tools, domain-specific analytical components, and novel geospatial datasets without overhauling the core system. It also facilitates extensibility for diverse use cases, including emergency response, urban planning, environmental monitoring, and logistics.

To ensure scalability and high availability, AI4MultiGIS leverages cloud-native technologies (e.g., Kubernetes, Docker, and serverless functions) hosted on a distributed cloud infrastructure either on premise or in public cloud servers (e.g., AWS, GCP, or European sovereign cloud platforms). This setup enables elastic scaling in response to dynamic data volumes and computational demands, particularly during critical events like natural disasters or large-scale simulations. Moreover, cloud infrastructure supports distributed data processing frameworks (e.g., Apache Spark, Dask) and scalable storage systems (e.g., S3-compatible object storage, distributed databases), ensuring performance and responsiveness under load.

To ensure data interoperability and compliance with standards, AI4MultiGIS adheres to Open Geospatial Consortium (OGC) Standards (https://www.ogc.org/standardsstandards), such as WMS, WFS, and GeoJSON. By adopting semantic data models and geospatial metadata ontologies, AI4MultiGIS ensures interoperability and facilitates cross-border collaboration and integration with other platforms and services. This is especially important for Euro-Mediterranean data ecosystems and public-sector interoperability initiatives. Besides, to support real-time geospatial intelligence and reduce latency, AI4MultiGIS incorporates edge computing nodes where preliminary data filtering, aggregation, and anomaly detection are performed near the data sources (e.g., on drones or IoT gateways). The processed data is then sent to the cloud for deeper analytics and visualization. This hybrid approach balances responsiveness and computational efficiency, while ensuring scalability from local (city) to regional (Euro-Med) levels. Furthermore, a robust DevOps pipeline is implemented to automate the integration of new features, updates, and AI models. Through continuous testing, monitoring, and deployment, the platform evolves iteratively in response to user feedback and scientific advances

- **Advanced Analytics and Reporting**

AI4MultiGIS incorporates intelligent geospatial agents that can autonomously manage specific analytic tasks such as image segmentation from satellite data, traffic congestion prediction from sensor streams, or flood risk analysis based on historical rainfall patterns and topography. These agents collaborate in a distributed workflow to orchestrate end-to-end remote sensing and decision-support tasks. The AI4MultiGIS framework will be supporting at least one of the following integrated analytics:

- AI4MultiGIS employs advanced machine learning models, such as Long Short-Term Memory (LSTM) networks, to capture temporal dependencies in data. Additionally, integrating Explainable AI (XAI) techniques can improve the precision and transparency. Furthermore, AI4MultiGIS enhances time-series forecasting by integrating with foundation models like TimesFM (Das 2023) and potentially TimeGPT (Garza 2023), enabling advanced spatiotemporal predictions across urban, environmental, and infrastructure systems. Through ingestion of multi-source real-time geospatial data (e.g., IoT sensors, remote sensing (satellites), and urban monitoring platforms), AI4MultiGIS leverages TimesFM’s zero-shot and few-shot generalization capabilities, trained on a broad spectrum of time series, to predict complex phenomena like flood risks, energy demand surges, or urban heat islands with high accuracy. TimesFM’s ability to model multivariate and multi-resolution sequences makes it particularly effective in forecasting across heterogeneous GIS layers. Meanwhile, TimeGPT's API-centric architecture facilitates easy integration into AI4MultiGIS as a forecast-as-a-service solution, offering scalable, on-demand forecasting for planners and emergency response systems (Liao 2025). This combined framework empowers AI4MultiGIS to dynamically inform dashboards, trigger real-time alerts, and simulate interventions, ultimately improving urban resilience, environmental sustainability, and data-driven decision-making in smart territories.

- Anomaly detection in urban infrastructure (e.g., water leaks, traffic incidents): The framework includes real-time anomaly detection to identify irregularities in urban infrastructure systems. This includes detecting leaks in water distribution networks, spotting unexpected traffic patterns that could indicate accidents or road failures, and identifying abnormal energy consumption that could signify equipment issues or tampering. AI4MultiGIS implements unsupervised techniques such as Autoencoders and Isolation Forests to detect outliers without prior labelling. Moreover, Graph Neural Networks (GNNs) are utilized to model urban networks, like water pipes or road systems, as graph structures, enabling more contextualized and accurate anomaly detection. Detected anomalies are flagged in real-time and visualized on the interactive dashboard, with automated alerts sent to city operators and maintenance teams.

All these analytical components are integrated into AI4MultiGIS via a modular, microservices-based architecture. This design allows each analytics module to operate independently while being orchestrated as part of a larger pipeline. A central analytics engine intelligently sequences and executes the relevant models based on user queries, data availability, and geographic context. Real-time and batch data inputs are supported via APIs and streaming protocols, ensuring scalability and responsiveness. Insights are delivered through customizable dashboards and reports, enabling stakeholders, from city managers to scientists, to derive value from data without deep technical expertise.

- **User Interface (UI)**

Detailed scheme of these integrated features

The AI4MultiGIS framework will be designed to offer an intuitive and interactive user interface (UI) that facilitates seamless communication between users and connected components (e.g., databases, real-time data streams, AI models, IoT devices). The goal is to ensure that the system is accessible to a wide range of users, from non-experts to specialists, and allows for smooth interaction with both the system’s underlying AI models and external components (e.g., geospatial data sources, emergency networks, public APIs).

The AI4MultiGIS UI will consist of multiple components designed to provide users with easy access to functionalities, real-time data, and actionable insights:

- **Dashboard:** The AI4MultiGIS Dashboard serves as the central hub of the platform's user interface, offering a comprehensive and real-time overview of critical data and system status. It allows users to customize their experience with widgets that visualize key information, such as disaster zones, environmental data, and traffic flow, enabling a tailored layout that suits specific needs. The dashboard also provides live monitoring, showcasing real-time data on traffic conditions, air quality, and disaster alerts, sourced from a network of connected systems. To support quick decision-making, it highlights Key Performance Indicators (KPIs), offering intuitive visualizations of metrics like emergency resource availability, response times, and environmental health indicators. This dynamic tool empowers users with the insights needed to respond swiftly and efficiently in both urban planning and disaster management scenarios.

- **Map Interface:** The AI4MultiGIS Map Interface offers an interactive and intuitive platform for visualizing and analysing geospatial data, enabling users to interact with both static and dynamic data layers. Through the map, users can seamlessly zoom, pan, and click on specific regions to obtain in-depth details about locations, whether it's disaster zones, infrastructure status, or emergency routes. The interface features robust Layer Management, allowing users to toggle between various data layers, such as population density, flood-prone areas, and air quality, to personalize the information displayed. Additionally, the Geospatial Analytics tools empower users to perform advanced queries on geographic data, such as identifying the most vulnerable areas to flooding, and generate tailored reports or forecasts to inform decision-making. This feature transforms complex spatial data into actionable insights for effective emergency response and urban planning.
- **Alerts & Notifications System:** AI4MultiGIS Alerts & Notifications System ensures users stay informed with real-time updates about critical events, such as disasters, emergencies, or significant changes in environmental data. With Customizable Alerts, users can set personalized thresholds for receiving notifications—whether it's triggered by falling air quality levels, the issuance of a disaster alert, or other relevant criteria. These notifications are instantly delivered via Real-Time Push Notifications, accessible through in-app alerts, SMS, or email, depending on user preferences. The system also features an Event History Log, offering users a comprehensive record of past alerts and actions taken. This functionality allows for the tracking of historical data, providing valuable insights and the ability to analyse trends over time, which is crucial for future preparedness and decision-making.

3.1.4 Developing Decentralised MultiGIS Data Management System

This system ensures that GIS data can be managed in a decentralized manner, allowing for more collaborative, trustworthy, and efficient data handling without relying on a central authority. By using blockchain and DLT, it guarantees the integrity and authenticity of GIS data

3.2 Integrated Data Handling and Modular Tools for Spatiotemporal Analysis

The AI4MultiGIS framework leverages the modular architecture of QGIS to support robust and flexible handling of heterogeneous spatiotemporal data. By integrating custom Python-based plugins, especially through the QGIS Processing Framework, users can seamlessly process, analyse, and visualize dynamic geospatial datasets across time and space. Each plugin operates as a self-contained module that performs specific analytical or data transformation tasks, enabling streamlined workflows that are reusable, interoperable, and scalable. The system's plug-and-play design allows domain-specific tools, such as anomaly detection algorithms, synthetic data generators, or AI-driven inference engines, to be easily added or updated without impacting the core platform. This integration of modular tools within QGIS ensures consistent data handling practices while empowering users to perform complex, multi-source geospatial analysis in a unified, extensible environment.

3.2.1 Novel Computing Techniques for Enhanced Real-Time MultiGIS Spatiotemporal Data Processing

The integration of Multi-Agent Reinforcement Learning (MARL) with Federated Learning (FL) on edge devices offers a promising framework for energy-efficient, privacy-preserving, real-time data processing and analysis. This combination empowers distributed agents to collaboratively and adaptively make decisions based on localized data while preserving user privacy through federated learning techniques. The integration allows continuous system improvement and energy-efficient real-time responses, facilitated by synchronized updates between edge-based and cloud-based digital twins.

The AI4MultiGIS framework integrates FL with MARL to achieve scalable, privacy-preserving, and intelligent spatiotemporal data processing at the edge. This hybrid learning architecture supports low-latency decision-making, real-time analytics, and continuous learning in distributed MultiGIS environments.

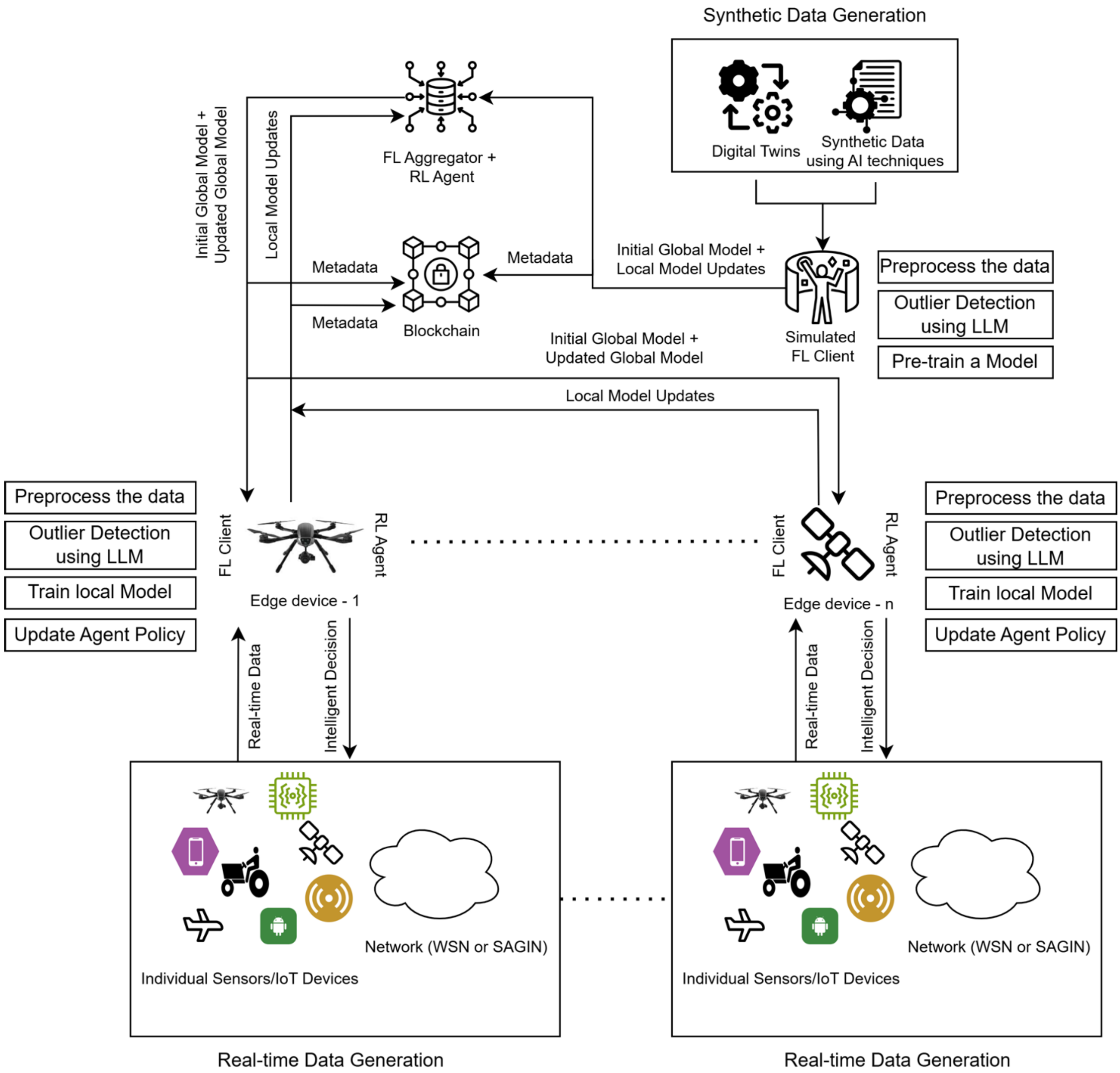

Figure 3: System Workflow of Novel Computing Techniques for Enhanced Real-Time Spatiotemporal Data Processing (A flow diagram showing the integration of FL clients and RL agents on edge devices, real-time data pipelines, synthetic data augmentation, model update exchanges, and blockchain-backed metadata integrity.)

The working flow of this component is illustrated in Figure 3 and is described as follows:

At the data generation layer, heterogeneous IoT and sensor networks (e.g., drones, weather sensors, GPS, mobile devices) collect real-time environmental data from geospatial regions. Each edge device (e.g., UAVs, sensors) functions as an autonomous FL client equipped with an embedded RL agent, responsible for making intelligent local decisions based on local data and pre-trained policies.

Each edge device performs:

- **Preprocessing of raw data**, including transformation and formatting,
- **Outlier detection** using Large Language Models (LLMs) to ensure data quality,
- **Local model training** with federated learning protocols, and
- **Policy updates** to improve the RL agent's decision-making logic.

Before federated training begins, a simulated FL client trains a model on synthetic data generated using Digital Twins, LLMs, or GAN-based simulations. This trained model is used to generate the initial global model, which is then shared with the FL aggregator. The aggregator distributes this initialized model to real FL clients (edge devices) to bootstrap their local training processes.

Following this, both real FL clients and the simulated client independently update their local models using their respective datasets. These local model parameters are then sent to the FL aggregator, which performs model aggregation (e.g., using federated averaging) to compute an updated global model. This updated model is redistributed to all clients, maintaining the iterative learning cycle.

A Digital Twin-based Synthetic Data Generator continues to provide high-fidelity synthetic datasets, supporting simulated clients and enabling robustness in underrepresented or rapidly changing scenarios. This synthetic stream enhances the adaptability of the overall system by maintaining consistent learning across diverse environmental conditions.

To ensure data integrity, traceability, and auditability, a blockchain-backed metadata management layer records model versioning, client participation, and training history. This supports secure, transparent exchanges and builds trust in the federated learning process.

By integrating FL and MARL:

- **FL** ensures data privacy by keeping raw data local and supporting secure model training,
- **MARL** enables agents to learn optimal strategies under dynamic, spatially distributed conditions, and
- Together, they enable distributed, adaptive, and secure real-time MultiGIS processing.

This architecture supports fault-tolerant, low-latency, and energy-efficient computing, making it ideal for critical applications such as flood risk prediction, invasive species monitoring, and urban infrastructure resilience.

3.2.2 Developing Interpretable Strategies for Cross-Model and Multi-Modal Data Integration in MultiGIS Applications

AI4MultiGIS will develop innovative and interpretable strategies to integrate and process cross-model and multi-modal data within cloud-based GIS applications. This task is crucial for supporting advanced geospatial decision-making in contexts where diverse data sources. The objective is to build robust AI models capable of handling complex spatial patterns while ensuring transparency and scalability.

To achieve this, AI4MultiGIS will implement advanced cross-model fusion architectures that integrate multiple types of neural networks. These will include Convolutional Neural Networks (CNNs) for image-based data, Recurrent Neural Networks (RNNs) for temporal geospatial sequences, and Graph Neural Networks (GNNs) for modeling spatial networks and relationships. Additionally, Quantum Neural Networks (QNNs) will be explored to address high-dimensional, multi-modal geospatial data by capturing intricate dependencies and enabling efficient computation.

Furthermore, AI4MultiGIS will address key geostatistical challenges that often affect spatial data interpretation. It will implement GNN-based models to mitigate spatial autocorrelation by explicitly learning topological dependencies between spatial entities. To counter the Modifiable Areal Unit Problem (MAUP), the project will develop adaptive spatial aggregation strategies that enable models to operate across varying scales and zoning configurations. Scalability issues will be addressed through the use of distributed cloud processing and hierarchical data partitioning techniques, ensuring that the system can manage and analyze large-scale datasets efficiently.

To ensure transparency and trust in these complex AI systems, AI4MultiGIS will also develop post-hoc interpretability mechanisms. These will include the integration of tools such as LIME, SHAP, and Layer-wise Relevance Propagation (LRP) to explain the decisions made by CNNs and GNNs. For GNNs specifically, AI4MultiGIS will implement graph-specific explainability methods such as node importance ranking and subgraph visualization. Additionally, for QNNs, the project will explore new approaches to interpret quantum decision paths and feature interactions, helping stakeholders understand how quantum-driven models reach conclusions.

AI4MultiGIS will integrate these models and interpretability tools into the MultiGIS cloud platform. It will package the models as reusable microservices, expose them through APIs, and develop an interactive dashboard that overlays model outputs and explanations directly onto GIS maps. This will allow users—such as urban planners, environmental analysts, and emergency responders—to explore not only the results of AI models but also the rationale behind those results in a spatially meaningful way.

Through this comprehensive and modular approach, AI4MultiGIS will lay the foundation for a new generation of intelligent, explainable, and scalable geospatial applications powered by advanced AI techniques.

3.2.3 GIS Platform Integration for Enhanced Spatial Analysis

We have considered multiple geospatial platforms, including QGIS, ArcGIS, SWMM, and Plaxis. However, we have chosen QGIS as the core integration platform for the AI4MultiGIS framework. The selection is based on several key factors: its open-source nature, cross-platform compatibility, plugin extensibility, and strong support for both academic and operational use cases. Additionally, QGIS provides seamless integration with Python-based machine learning tools and spatial analysis libraries, which aligns with the AI-driven objectives of our project.

QGIS (Quantum GIS) is a free and open-source Geographic Information System widely recognized for its robust capabilities, flexibility, extensibility, and strong community-driven ecosystem. As a feature-rich platform, QGIS enables users to visualize, edit, analyze, and publish geospatial data across a variety of vector, raster, and database formats. It supports both 2D and 3D rendering, integration with web mapping services, and advanced spatial analysis capabilities, making it a comprehensive GIS solution for both research and operational use cases. It plays a pivotal role in the AI4MultiGIS framework as a foundational platform for spatial data management, visualization, and analysis. Its extensible architecture allows seamless integration of domain-specific plugins, making it an ideal host environment for advanced geospatial intelligence tools.

The core functionalities of QGIS that make it an essential tool in this context include:

Multi-Format Data Support: QGIS supports a wide variety of spatial data formats including vector formats (e.g., Shapefile, GeoJSON), raster formats (e.g., GeoTIFF), and databases (e.g., PostGIS, SpatiaLite). It also connects to web mapping services like WMS, WMTS, and WFS, enabling dynamic access to external geospatial datasets.

Layer-Based Visualization and Map Rendering: Users can overlay and style multiple spatial datasets using advanced symbology, categorization, and labeling options. QGIS supports both 2D and 3D map views, enabling detailed exploration of terrain, infrastructure, and other physical features in both flat and volumetric space.

Interactive Editing and Attribute Management: The platform includes tools for digitizing geometry, editing attributes, creating topology rules, and validating spatial integrity. This makes it suitable for maintaining high-quality spatial databases with manual or semi-automated data curation workflows.

Geospatial Analysis and Processing: QGIS includes a robust Processing Toolbox, integrating native algorithms and tools from external providers like GRASS, SAGA, and GDAL. It supports spatial queries, overlays, proximity analysis, raster algebra, hydrological modeling, and more.

Automation and Scripting with PyQGIS: Using the Python-based PyQGIS API, developers can script custom analysis workflows, manipulate layers, and extend QGIS functionalities. This enables advanced automation, batch processing, and tight integration with AI-driven modules in the AI4MultiGIS framework.

Plugin Architecture and Extensibility: QGIS features a mature plugin ecosystem where developers can introduce custom tools via Python. Plugins can range from simple utilities to complex applications with graphical user interfaces, data pipelines, and model integration.

Processing Framework for Modular Algorithms: Through the Processing Framework, developers can register their own algorithms to appear in the toolbox. These can be integrated into graphical models or called programmatically, supporting both user-driven and automated workflows in spatial analysis.

QGIS and AI4MultiGIS: A Programmable Workbench for Intelligent Spatial Analysis:

QGIS's powerful support for custom plugin development using Python (via the PyQGIS API) enables seamless integration of external algorithms, AI models, and domain-specific workflows. Within the AI4MultiGIS project, QGIS has been selected as the primary GIS platform due to its modular architecture, active plugin ecosystem, and suitability for enhancing spatial analysis capabilities.

As part of the planned integration, the project envisions the development of custom plugins that will support AI-driven functionalities such as anomaly detection, federated analytics, synthetic data workflows, and real-time data integration. These potential extensions aim to leverage QGIS's flexibility to bring advanced MultiGIS capabilities into the geospatial analysis environment used by domain experts and decision-makers.

In addition, QGIS's Processing Framework provides a robust foundation for registering custom scripts and algorithms as reusable geoprocessing modules. These can be accessed through graphical models or executed programmatically, enabling both interactive and automated workflows.

Through this envisioned integration, QGIS is positioned to act not only as a visualization platform but also as a programmable geospatial workbench, bridging real-time data streams, AI-powered insights, and user interaction within a unified and intelligent spatial analysis environment tailored for the MultiGIS context.

3.2.4 Development of GIS 'Plug and Play' with Integration Capabilities and AI4MultiGIS framework

To enable scalable, modular, and domain-specific extensions within the AI4MultiGIS framework, a ‘plug and play’ architecture is envisioned using the QGIS plugin ecosystem. This architecture will allow spatial analysis tools, such as those leveraging AI models or real-time geospatial data streams, to be added, tested, and deployed without disrupting the core GIS platform.

The overall development pipeline designed for this integration is illustrated in Figure 4. It outlines the different paths available for developing QGIS plugins and highlights how these plugins can be seamlessly integrated and tested within the QGIS environment.

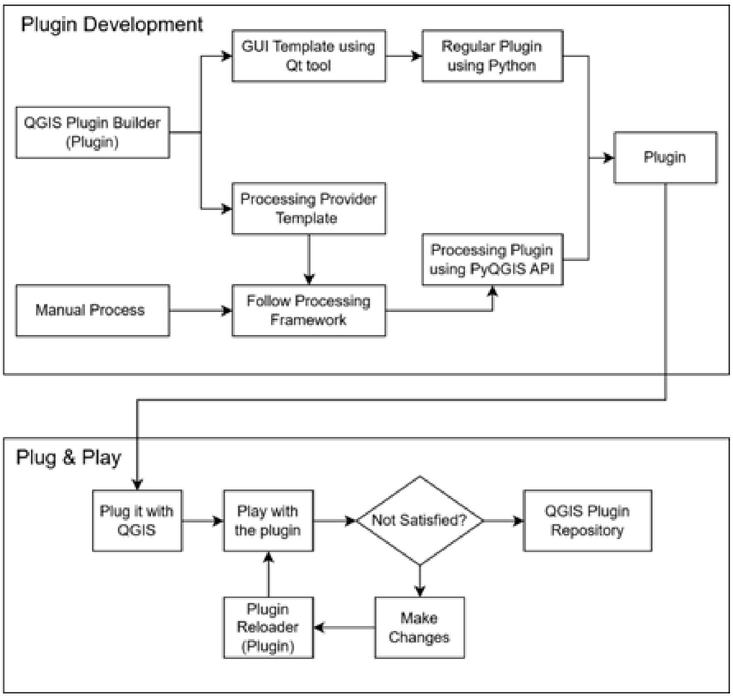

Figure 4: Plugin Development and Integration Pipeline

Plugin Development Approaches:

Plugins are much more integrated into the QGIS system than Python Scripts. They are managed by Plugin Manager and are initialized when QGIS starts. QGIS supports two primary types of plugins: Regular Plugins and Processing Plugins. These plugins can be developed either manually or by using tools like the QGIS Plugin Builder, which helps generate the boilerplate structure required for plugin development.

Using QGIS Plugin Builder:

The QGIS Plugin Builder is a development plugin that assists in generating the essential file structure and starter code for a new plugin. It allows developers to choose between different templates depending on the plugin type:

- The **"Text for Menu Item"** template is used to generate the basic structure for Regular Plugins, including standard plugin files and a Python script that links to a menu item in QGIS (*Gandhi, n.d.-a*). This builder plugin allows users to select the appropriate menu location (e.g., under Vector, Raster, or Plugins) where the plugin command will be added in the QGIS interface.
- The **"Processing Provider"** template is used for developing Processing Plugins. This template sets up the structure for a processing provider that can contain multiple algorithms integrated into the QGIS Processing Toolbox (*Gandhi, n.d.-b*).

Plugin Builder generates standard files such as __init__.py, metadata.txt, resources.qrc, and others, enabling developers to focus on functionality rather than on the initial setup of plugin infrastructure.

Regular Plugins: Regular Plugins are GUI-based tools designed for interactive use. They typically include custom toolbars, dockable panels, or dialogs that users interact with directly in the QGIS interface. These plugins are suitable for workflows that rely on user-driven inputs, visual manipulation, or step-by-step operations.

Using the Plugin Builder, developers can create a Regular Plugin scaffold and then use Qt Designer (*The Qt Company, n.d.*) to design .ui interface files, while implementing business logic in Python using the PyQGIS API (*QGIS Development Team, n.d.-a*).

While Regular Plugins are flexible and suitable for highly customized tools, they come with a few disadvantages:

- Developers are responsible for designing the entire user interface, which can lead to inconsistent user experiences across plugins.
- Regular Plugins are not integrated into QGIS's Processing Framework. As a result, their functionality cannot be reused within other tools like batch processing, the graphical modeler, or Python scripts.

Processing Plugins: Processing Plugins are non-GUI, algorithm-driven components that integrate directly into the QGIS Processing Framework (*QGIS Development Team, n.d.-b*). These are most appropriate for analytical operations such as spatial computation, AI-based inference, or data transformation tasks.

During setup with the Plugin Builder, developers can select the "Processing Provider" template, which automatically creates the required structure for a processing plugin. This includes a provider class for registering algorithms and an example algorithm that can be expanded or modified.

The Processing Framework offers a more standardized and streamlined development path when the plugin requires only minimal user interaction, such as selecting input layers and setting output parameters. QGIS automatically generates a consistent UI for processing algorithms, based on the parameters defined in the code.

These algorithms are implemented using the PyQGIS API, which provides programmatic access to QGIS's core functionalities, such as manipulating vector and raster layers, performing geoprocessing tasks, accessing attributes, and writing output datasets.

Key advantages of Processing Plugins include:

1. **No custom UI needed** – QGIS auto-generates a clean interface from your input/output parameter definitions.
2. **Built-in input validation** – Reduces errors and ensures type correctness.
3. **Multi-threading support** – Allows execution in the background without freezing the interface.
4. **Batch processing support** – Enables users to apply the same algorithm across multiple input datasets in a single operation.
5. **Standardized user feedback** – Includes built-in progress bars and status messaging.
6. **Modeler integration** – Custom algorithms can be incorporated into workflows using QGIS's **Graphical Modeler**.
7. **Scriptability** – Algorithms can be accessed and executed from the Python Console, enabling automation and integration with custom pipelines.

The Processing Framework API makes it easy to create reusable and interoperable geospatial tools with minimal code. It ensures compatibility with QGIS's internal tools and with external providers like GRASS and SAGA.

Given these strengths, Processing Plugins are the preferred method for integrating advanced geospatial logic and AI-driven analysis within QGIS. For the AI4MultiGIS framework, where tools are primarily algorithmic, data-driven, and need to support automation and interoperability, Processing Plugins offer the ideal path forward.

Manual Development of Processing Plugins:

Although tools like the QGIS Plugin Builder simplify plugin creation, Processing Plugins can also be developed manually, offering greater transparency and flexibility over the plugin architecture. This approach is especially valuable for advanced users who want to build minimal, purpose-driven plugins for spatial analysis tasks within the AI4MultiGIS framework.

To be recognized by QGIS, a plugin must include just two essential files:

- **__init__.py:** This file is the starting point of the plugin and contains the main logic of the plugin. It must have certain methods, such as __init__() method gives the plugin access to the QGIS Interface, initGui() method is called when the plugin is loaded and unload() method which is called when the plugin is unloaded. We will develop and integrate our AI models within this file to ensure direct linkage between the plugin interface and the underlying intelligence modules.
- **metadata.txt:** A configuration file containing the plugin name, description, version, author, and QGIS compatibility metadata used by plugins website and plugin manager.

This minimal structure is sufficient for a basic plugin to be listed and loaded in the QGIS Plugin Manager (*Dobias et al., n.d.*). However, we can keep multiple python file through separating the main logic. For instance, one python file contains the main logic of the plugin. The second file (__init__.py) is the starting point of the plugin. It imports the plugin class created in the main logic python file and creates an instance of it. For Processing Plugins, developers extend this base by manually implementing:

- A custom provider class (typically in a separate Python module), which registers one or more processing algorithms.
- One or more algorithm classes, each implementing a subclass of QgsProcessingAlgorithm from the PyQGIS API.

- An updated __init__.py to register the processing provider during plugin initialization.

Developers define algorithm inputs, outputs, and logic entirely using the PyQGIS API, without the need for designing a GUI. Once defined, the algorithm automatically appears in the QGIS Processing Toolbox with a generated interface that includes built-in parameter validation, batch execution, and progress reporting.

This manual approach allows:

- Fine-grained control over plugin structure and dependencies
- Lightweight deployments ideal for scripting and AI integration
- Full compatibility with QGIS's processing ecosystem, including modeler, batch processing, and console scripting

Although manual setup requires a deeper understanding of plugin architecture, it aligns well with the modular, algorithm-centric needs of the AI4MultiGIS framework, particularly for developing AI-enhanced geospatial analysis tools that must be efficient, interoperable, and maintainable.

Plug and Play Integration and Deployment:

A central advantage of using QGIS for the AI4MultiGIS framework is its ability to support modular plugin integration through a straightforward plug-and-play model. Once a plugin is developed, whether manually or via Plugin Builder, it can be integrated into QGIS and tested locally with minimal setup, facilitating rapid iteration and validation.

Local Deployment for Development and Testing

During development, plugins are typically deployed and tested locally. This can be achieved using two main methods:

- **Direct Installation**: Copy the plugin folder into the user's QGIS plugin directory (usually under C:\Users\<username>\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins\ on windows or equivalent paths on linux/Mac). The plugin will then appear in the QGIS Plugin Manager after a restart.
- **ZIP Installation**: Compress the plugin folder into a .zip file and use QGIS's "Install from ZIP" option to install it through the GUI (see Figure 5). This method is especially useful for sharing early-stage plugins within teams or across systems.

To speed up the development cycle, developers often use the Plugin Reloader plugin, which allows them to reload the current plugin without restarting QGIS. This accelerates testing and debugging by reducing turnaround time between code changes and runtime feedback.

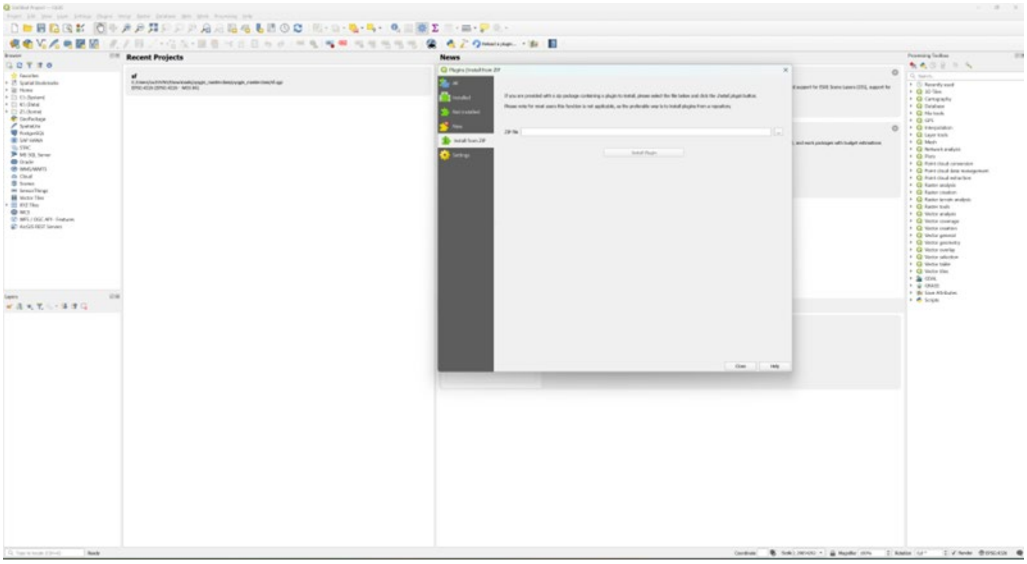

Figure 5: QGIS Plugin Manager – Installing a Plugin from ZIP File

Plugin Publishing and Distribution

Once the plugin is validated and reaches a stable version, it can be packaged and published to the official QGIS Plugin Repository. This makes it accessible to a global community of QGIS users and ensures streamlined deployment across multiple AI4MultiGIS nodes or partner environments.

Illustrative Example: Developing a Custom Processing Plugin in QGIS

To demonstrate the modular plugin architecture envisioned for AI4MultiGIS, we present a hands-on example of creating a custom Processing Plugin in QGIS. This plugin, titled Save Attributes, enables users to export attribute tables from vector layers into CSV files—a foundational task that can be extended for more complex analytical pipelines such as data pre-processing for AI models.

Step 1: Plugin Generation Using QGIS Plugin Builder

The development of the Save Attributes Processing Plugin begins with the use of the QGIS Plugin Builder. This tool streamlines the initial setup by generating the boilerplate files and folder structure necessary for plugin development.

Step 1.1: Launching Plugin Builder

From the QGIS main menu, as shown in Figure 6, navigate to: Plugins → Plugin Builder → Plugin Builder. This will launch a setup wizard that guides the user through a series of input fields.

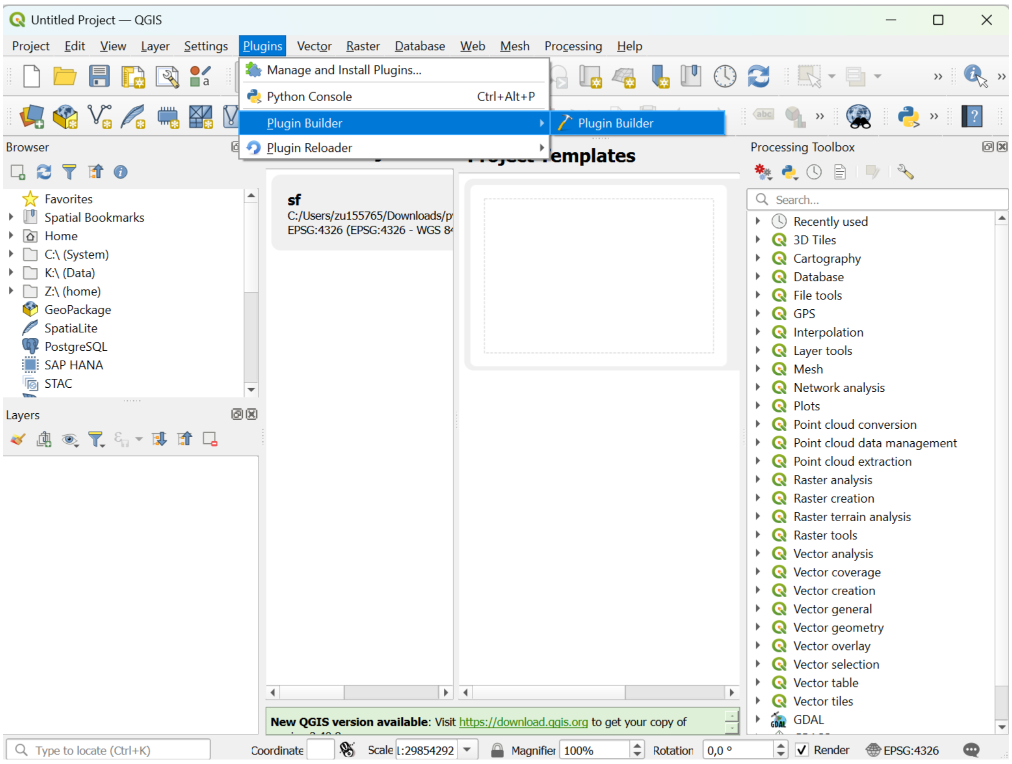

Figure 6: Accessing Plugin Builder from the QGIS main menu.

Step 1.2: Filling Plugin Metadata

The Plugin Builder prompts the user to enter basic metadata such as the class name, plugin name, and module name. These values define how the plugin will be referenced internally in QGIS, how it appears in the Plugin Manager, and how its files are organized. In this example, appropriate values were entered in each field to reflect the plugin's function. These selections are shown in Figure 7.

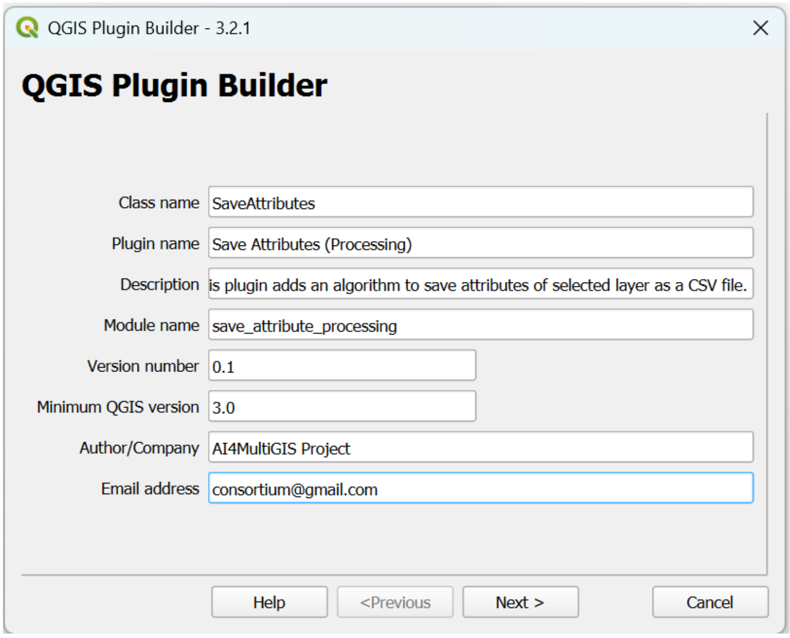

Figure 7: Metadata entry fields in the QGIS Plugin Builder wizard.

Step 1.3: Selecting the Plugin Template and File Types

The Plugin Builder offers a choice of templates for generating different types of plugins. In this example, the Processing Provider template was selected as shown in the Figure 8, which sets up the necessary structure for registering algorithms in the QGIS Processing Toolbox.

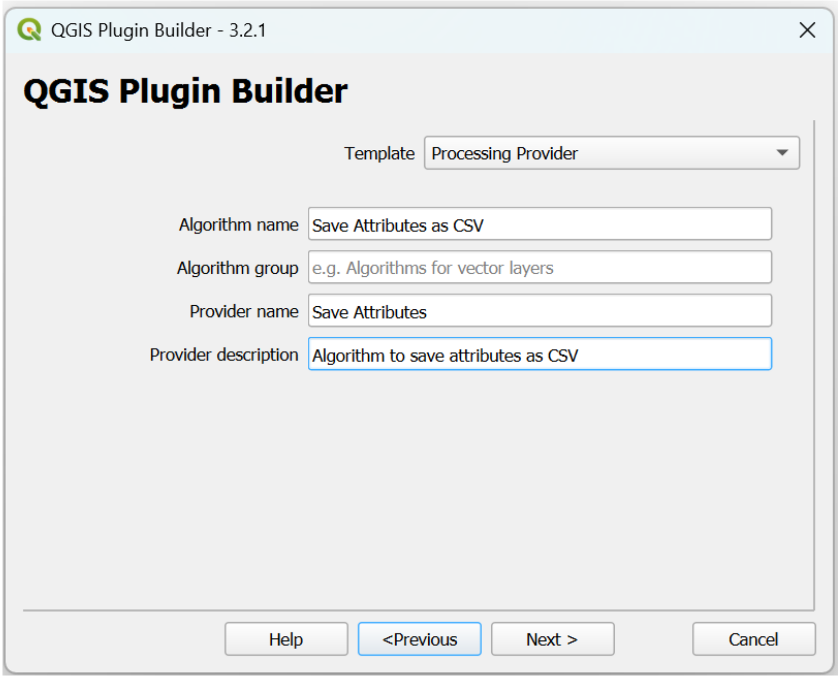

Figure 8: Template selection screen in Plugin Builder

After selecting the template, Plugin Builder prompts for the types of files to generate. These typically include a provider class, algorithm class, and supporting files. We retained the default selection and proceeded by clicking Next.

Step 1.4: Setting Plugin Metadata for Publication

In this step, the Plugin Builder requests optional publication-related metadata such as Bug Tracker, Repository, and Home Page URLs, as shown in Figure 9. These fields are mainly relevant when publishing the plugin to the QGIS Plugin Repository. Since this plugin is intended for internal demonstration and not for public distribution, we retained the default placeholder values in these fields.

To indicate the plugin's experimental nature, we enabled the checkbox labeled "Flag the plugin as experimental".

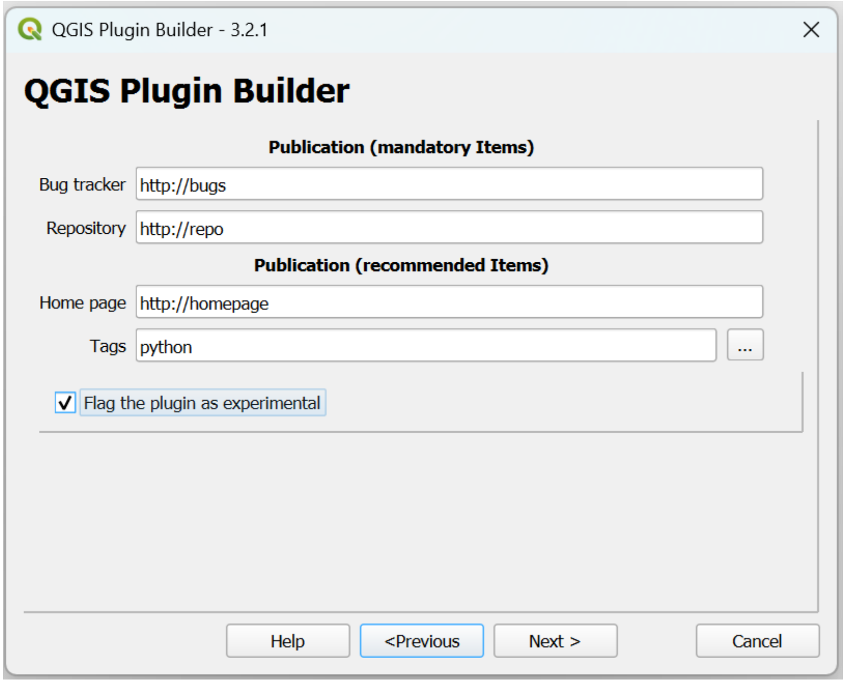

Figure 9: Optional publication metadata fields with experimental flag selected

Step 1.5: Choosing the Output Directory

After the metadata entry is complete, Plugin Builder prompts the user to select a directory where the generated plugin should be saved. Once the target path was selected, we clicked Generate to complete the process.

Step 1.6: Completion Confirmation

Once the plugin is successfully generated, a confirmation dialog is displayed, as shown in Figure 10. It shows the location where the plugin has been saved and confirms that the template was created without issues.

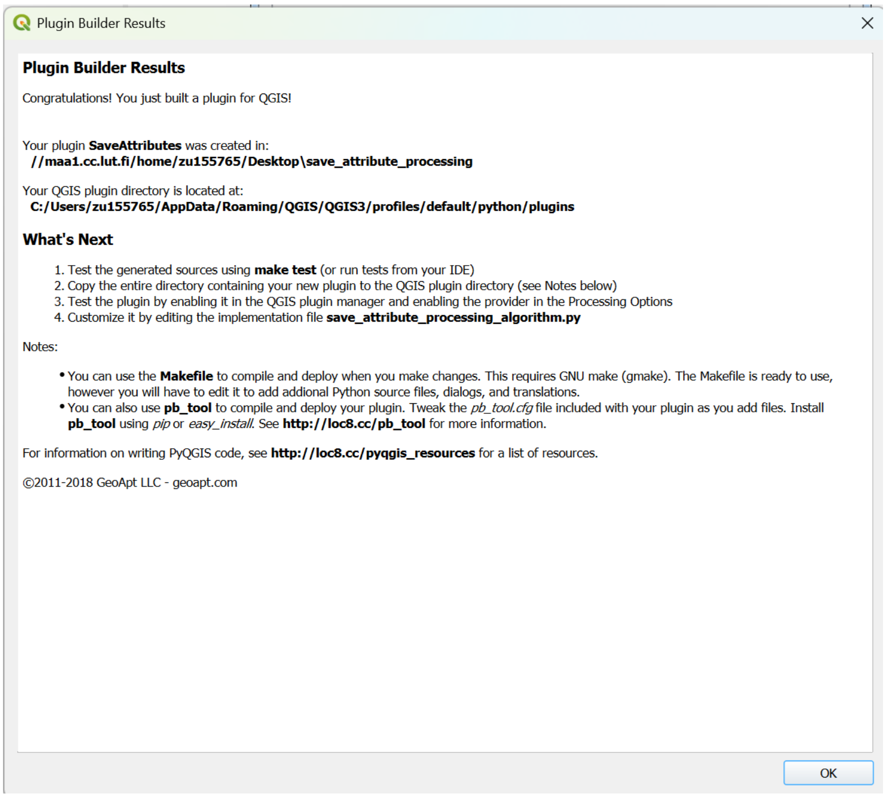

Figure 10: Confirmation dialog displayed after plugin generation

Step 2: Plugin Installation and Initial Activation

After generating the plugin using Plugin Builder, the next step is to install it in QGIS for testing and further development.

Step 2.1: Placing the Plugin Folder in the Correct Directory

QGIS stores plugins within each user profile's directory. To access this location, we open QGIS and navigate to Settings → User Profiles → Open Active Profile Folder. This opens the active user profile's directory on the system. Within the profile folder, navigate to the python/plugins subdirectory. If it does not exist, create it manually. Next, copy the newly generated plugin folder (in this example, save_attribute_processing) into this location, as shown in Figure 11. QGIS will recognize any valid plugin placed in this folder at the next startup.

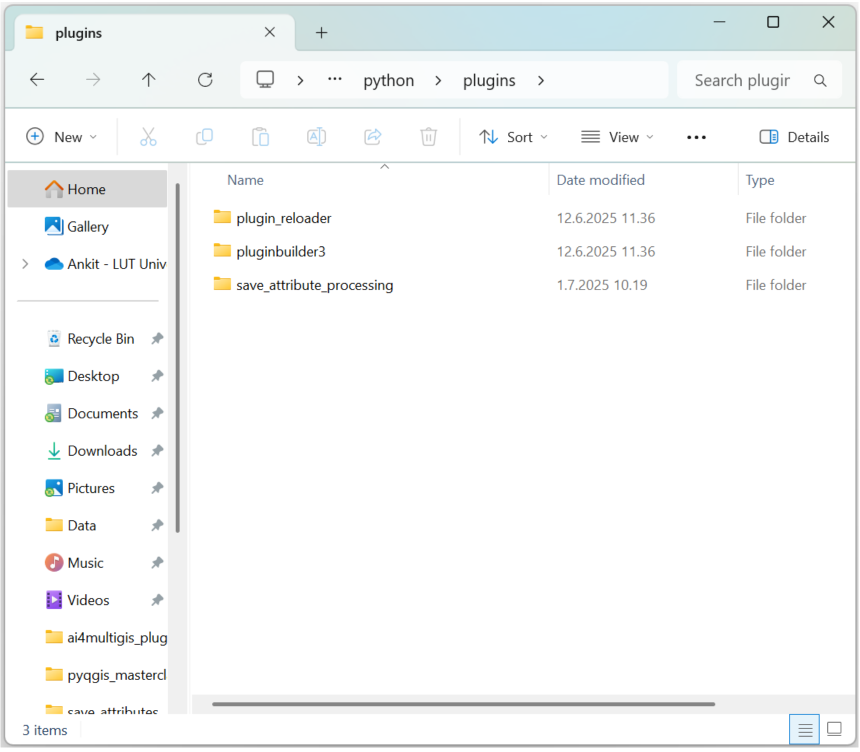

Figure 11: Plugin folder "*save_attribute_processing*" placed in the QGIS user plugins directory

Step 2.2: Enabling the Plugin in QGIS

After restarting QGIS, open the Manage and Install Plugins window from the Plugins menu. Under the Installed tab, find the plugin named Save Attributes (Processing) and enable it by checking the box next to it, as shown in Figure 12.

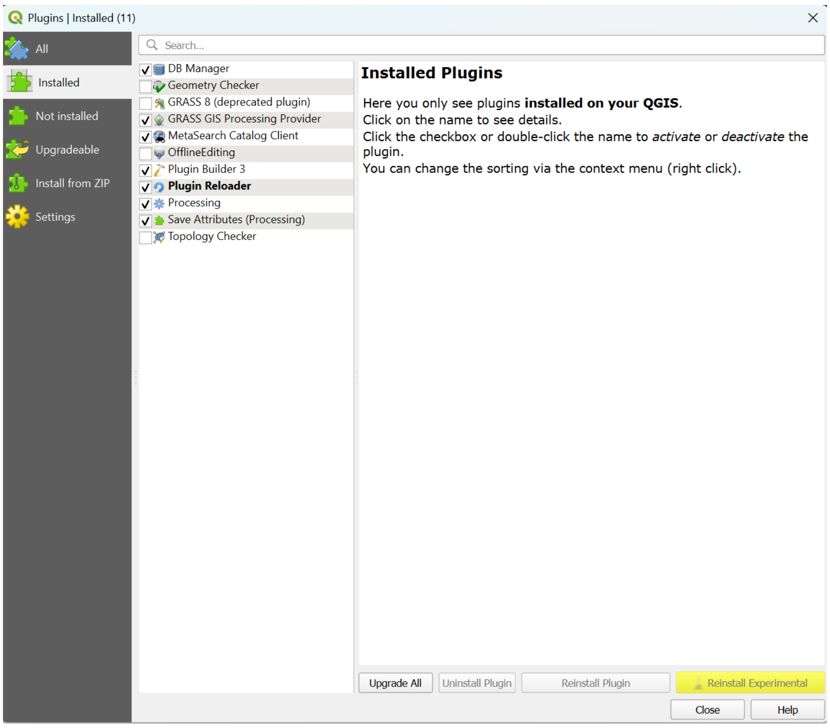

Figure 12: Enabling the newly created plugin from the list of installed plugins

Step 2.3: Verifying Plugin Availability in the Processing Toolbox

Once the plugin is activated, open the Processing Toolbox from the Processing menu. At the bottom of the list, a new provider named *Save Attributes* will appear, as shown in Figure 13. Expand this entry to view the included algorithm — *Save Attributes as CSV*.

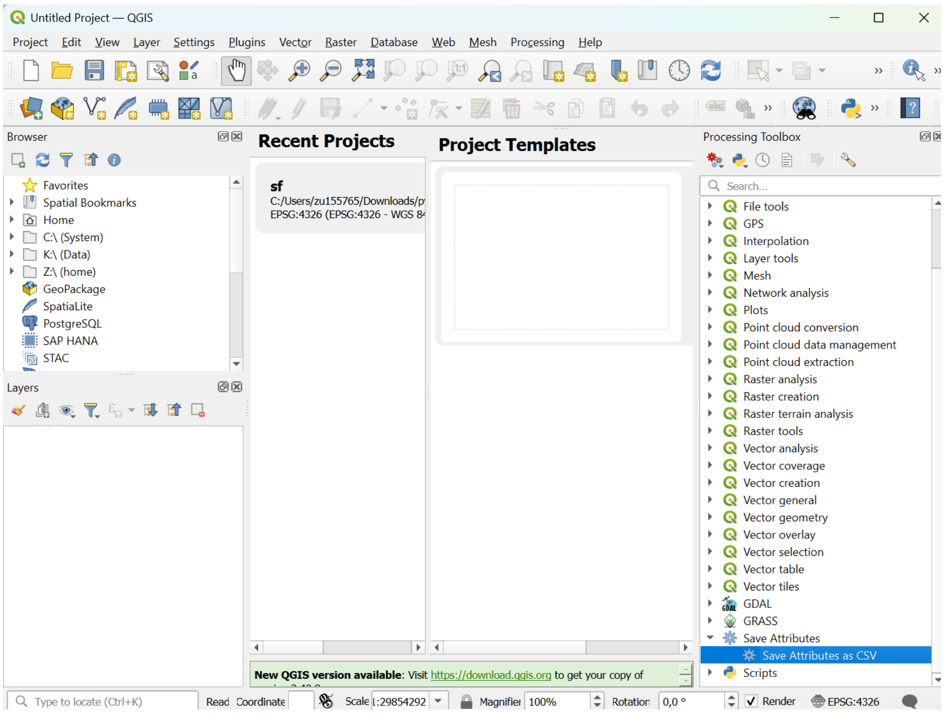

Figure 13: The plugin provider and its algorithm listed in the Processing Toolbox

Step 2.4: Viewing the Algorithm Dialog

To view the interface of the newly added algorithm, double-click on Save Attributes as CSV listed under the Save Attributes provider in the Processing Toolbox. This opens the standard QGIS Processing dialog as shown in Figure 14, which currently provides basic options such as selecting an input vector layer and specifying the destination for the output—either as a temporary layer or saved file. There is also an option to automatically open the output file after the process is complete. Since we have not yet added any custom logic, the dialog is still in its default state. In the next steps, we will customize it to implement the required functionality.

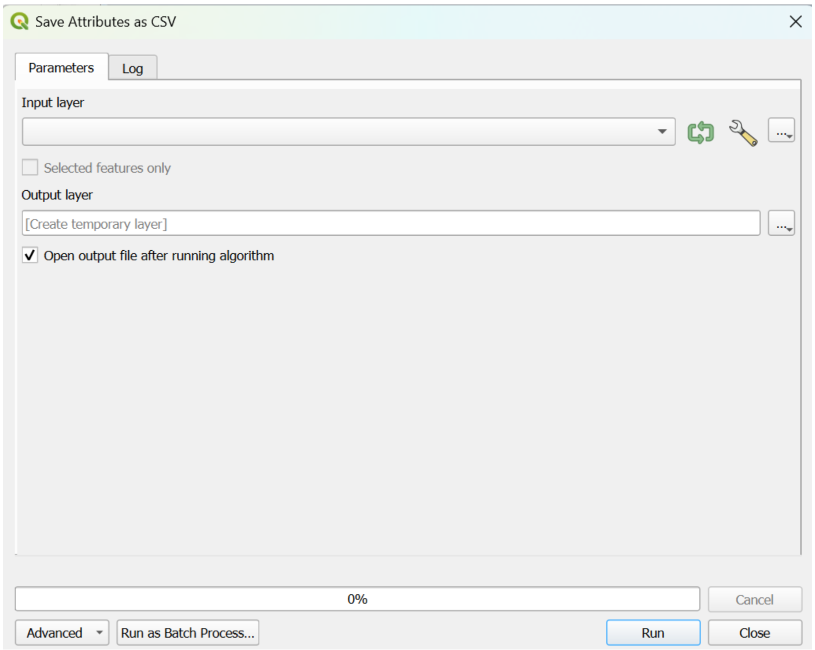

Figure 14: Default interface of the plugin's algorithm “Save Attributes as CSV”

Step 3: Reloading the Plugin Using Plugin Reloader

During development, it is common to make frequent changes to the plugin's source code. Restarting QGIS after every modification can be time-consuming, so QGIS offers a useful tool called Plugin Reloader to streamline this process. If not already installed, the Plugin Reloader plugin can be added via the Plugin Manager. Once installed, navigate to Plugin → Plugin Reloader → Reload a plugin…, and select the save_attributes_processing plugin from the list in the configuration dialog, as shown in Figure 15. This setup allows you to reload your plugin instantly with a single click—without restarting QGIS—making it highly efficient for iterative testing and debugging.

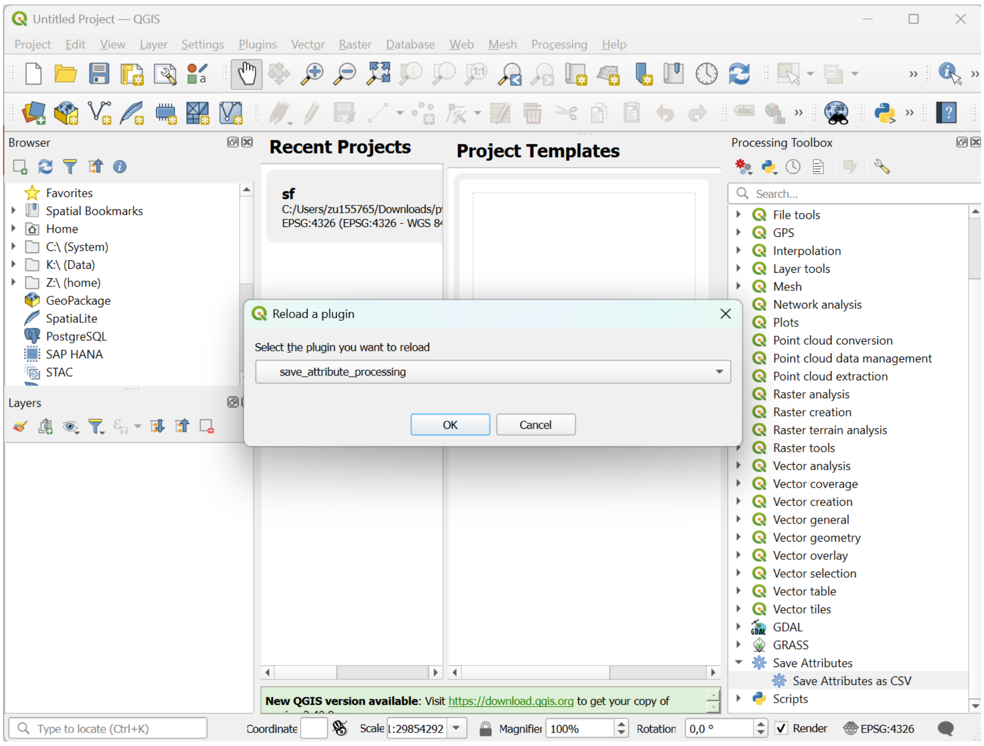

Figure 15: Selecting the plugin in Plugin Reloader for dynamic reloading during development

Step 4 – Customization and Testing

After the initial plugin structure is generated, we proceed to customize the plugin's core logic. This is done by modifying the save_attributes_processing_algorithm.py file to implement the desired functionality—in this case, exporting vector layer attributes to a CSV file.

Once the changes are made, we reload the plugin using the Plugin Reloader to avoid restarting QGIS. To test the updated functionality, we first load vector layers into QGIS. An example map view with multiple loaded layers (e.g., parcels, zoning, streets) is shown in Figure 16.

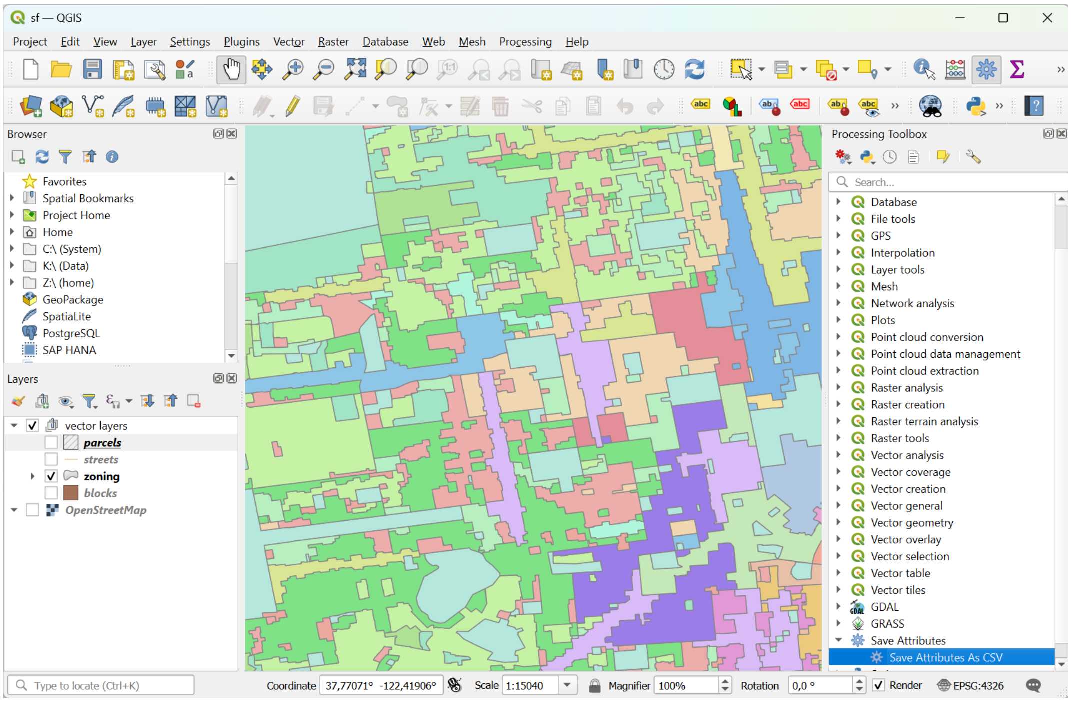

Figure 16: Map view with multiple loaded vector layers

We then open the Processing Toolbox and launch the algorithm by navigating to: Processing → Toolbox → Save Attributes → Save Attributes as CSV

In the dialog, we choose an input vector layer and specify the output location and filename (e.g., output1.csv) for the resulting CSV file. The process is illustrated in Figure 17. Clicking Run executes the plugin, and the CSV file is created at the selected location containing the attributes of the input vector layer.

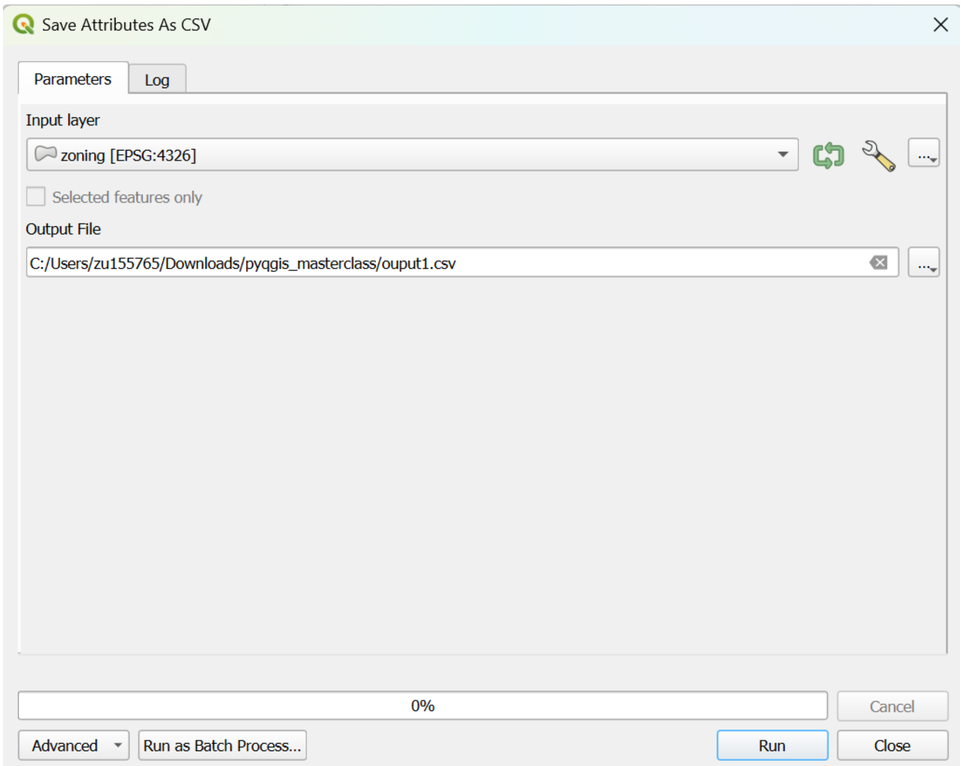

Figure 17: Updated interface of the plugin's algorithm “Save Attributes as CSV”

This step validates that the plugin operates correctly and performs the intended geospatial processing task.

Step 5 – Batch Processing and Integration

Although the "Save Attributes as CSV" algorithm is part of a custom plugin, it is fully integrated with QGIS’s native Processing Framework. This allows the plugin to take advantage of built-in tools such as batch processing, graphical modeler integration, and scripting—key benefits of developing as a Processing Plugin.

To demonstrate its interoperability with the QGIS environment, we can run this algorithm through the batch processing interface:

Step 5.1: In the Processing Toolbox, right-click the Save Attributes as CSV algorithm under the Save Attributes provider.

Step 5.2: Select Execute as Batch Process... from the context menu.

Step 5.3: In the Batch Processing dialog that appears, we can add multiple input layers—for instance, blocks and parcels—and specify separate output paths for each result, as shown in Figure 18. This allows the same processing logic to be applied across multiple datasets simultaneously, producing individual CSV outputs in a single run.

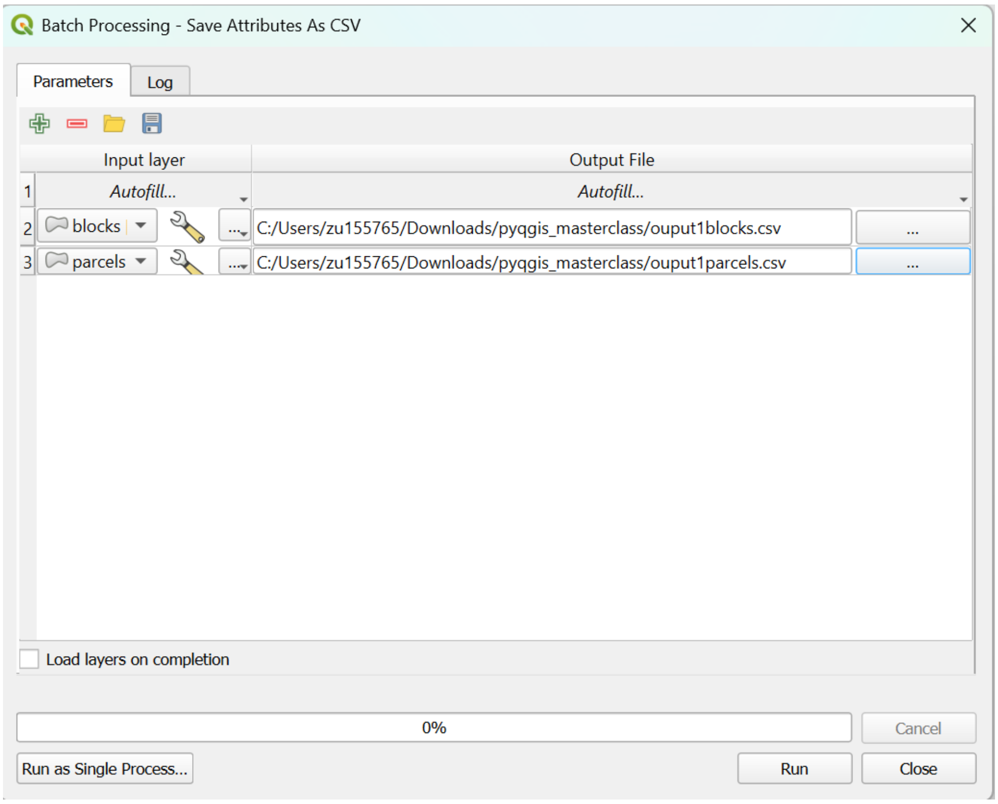

Figure 18: Adding multiple input layers in Batch Processing dialog

This capability highlights the advantage of using the QGIS Processing Framework, where custom algorithms written in Python integrate smoothly with QGIS's powerful spatial processing infrastructure. Once this step is completed, the plugin is fully functional and can be used as-is.

Step 6 – Toolbar and Menu Integration

To further enhance the user experience and improve the discoverability of this plugin, we can use a hybrid approach to make our processing plugin behave more like a regular GUI-based plugin. This involves integrating it into the QGIS interface by adding a dedicated menu entry and toolbar button.

To achieve this, we include the appropriate code in this plugin's main files and provide an icon for the plugin, which is placed in the generated plugin directory.

Once these modifications are in place, reload the plugin by clicking the Reload plugin button from the Plugin Reloader. We will now observe a new toolbar icon and a corresponding menu item under: Plugins → SaveAttributes → Save Attributes as CSV, as shown in Figure 19.

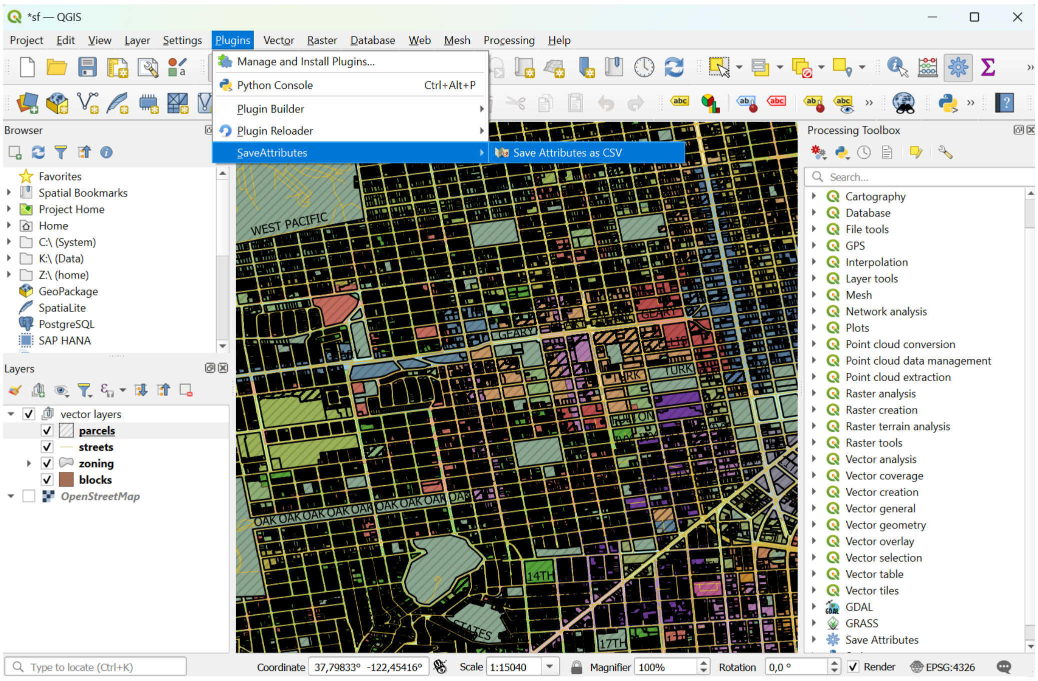

Figure 19: The Save Attributes plugin integrated with a menu entry under Plugins → SaveAttributes

Clicking either of these will launch the same Save Attributes as CSV algorithm, providing users with an intuitive and direct way to access the functionality without navigating through the Processing Toolbox.

Initially, both the processing provider and the algorithm might appear with default icons. To provide a cohesive visual identity, we modify the save_attributes_processing_provider.py and save_attributes_processing_algorithm.py files to associate custom icons with them. After making these changes and reloading the plugin, Figure 20 now reflects the customized iconography for both the provider and algorithm, resulting in a more polished and professional presentation within QGIS.

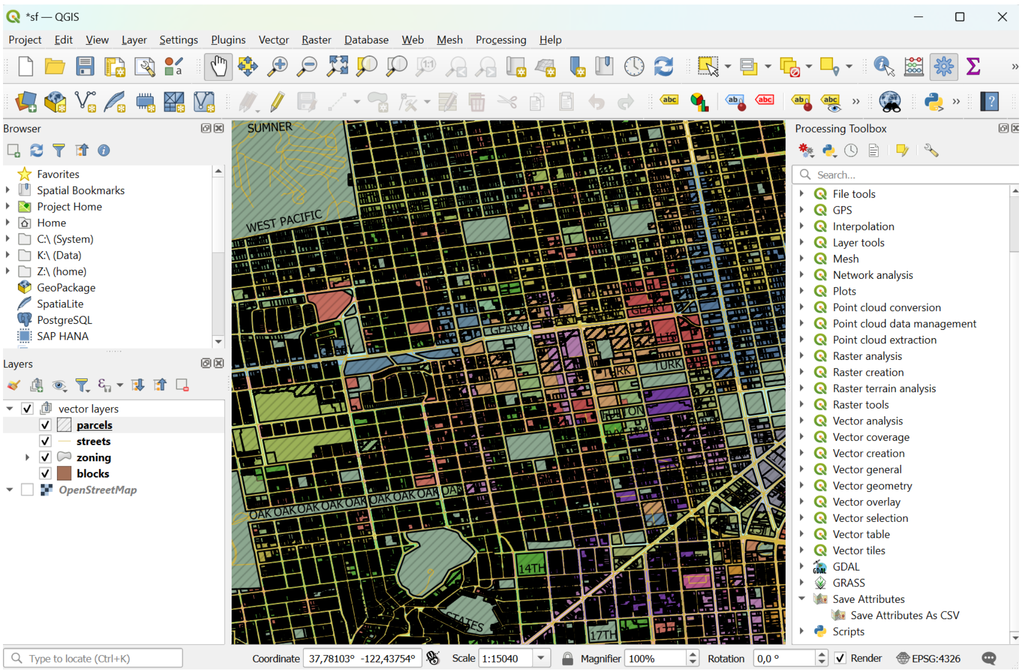

Figure 20: Custom icons applied to the Save Attributes provider and algorithm within the Processing Toolbox

4 AI4MultiGIS Reference Architecture

This section provides the high-level description of the AI4MultiGIS reference architecture blueprint.

4.1 Overview of the Reference Architectural Principles

4.2 High-Level Architecture

The reference architecture of the AI4MultiGIs platform is shown in Figure 21 and highlights the high-level design of the layered architecture, which is structured into five key layers to ensure modularity, scalability, and seamless functionality. These layers allow supporting real-time ingestion, processing, and fusion of heterogeneous geospatial data from diverse sources. It leverages advanced AI and machine learning models to deliver predictive analytics, optimization, and anomaly detection capabilities. The AI4MultiGIs platform is built with scalability, resilience, and interoperability in mind, for these reasons the system makes use of cloud-native, microservice-based components compliant with geospatial standards to allow dynamic and interoperable communication among different components. Besides, the architecture integrates decision support systems featuring customizable dashboards, robust APIs, and automated notification services for timely insights and actions.

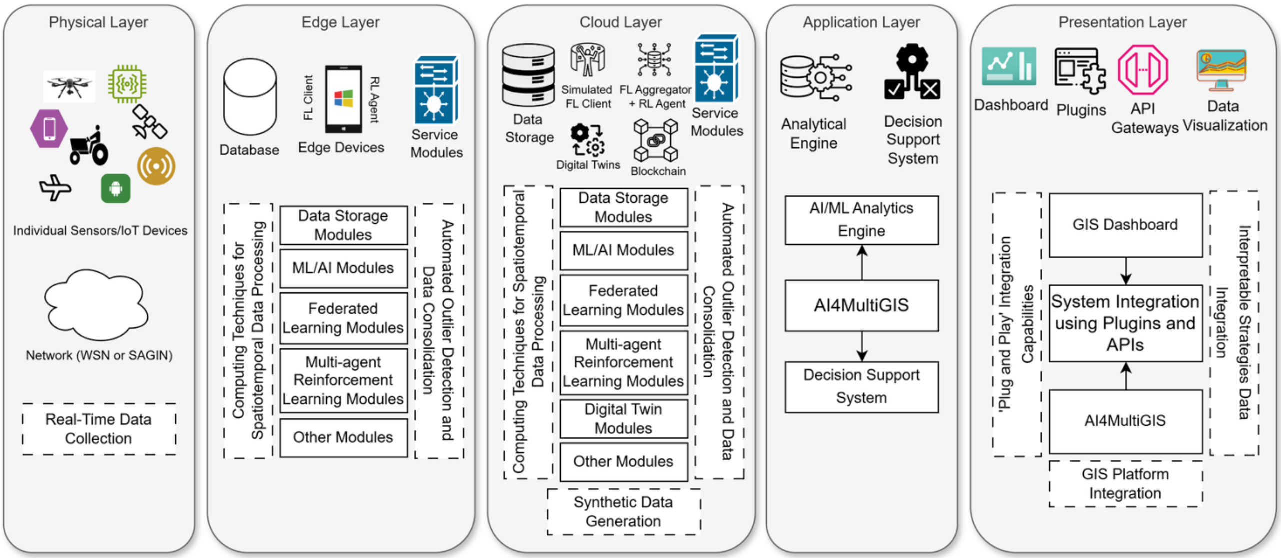

Figure 21: AI4MultiGIS High Level Referenced Architecture

Physical layer: The physical layer encompasses a range of technologies and methodologies designed to enable real-time data collection from diverse sources. Within the AI4MultiGIS framework, this layer should support the aggregation of multimodal data from various origins to enhance spatial analysis capabilities. Key data sources include remote sensing technologies such as satellite imagery, UAV (drone) footage, and LiDAR scans, all of which provide high-resolution raster data crucial for environmental monitoring. In addition, real-time spatial data is gathered through individual sensors, IoT devices, wireless sensor networks (WSNs), and Space-Air-Ground Integrated Networks (SAGIN). These systems leverage technologies such as LoRaWAN, GPS trackers, and weather stations to collect heterogeneous spatial information. To increase processing efficiency, many of these edge devices can be equipped with AI hardware accelerators, enabling localized AI inference before transmitting essential insights to GIS databases.

Edge layer: The Edge Layer of AI4MultiGIs plays a critical role in decentralized intelligence and low-latency processing. It encompasses local databases and federated learning clients embedded within edge devices, enabling privacy-preserving, on-site model training and inference. This layer is equipped with advanced computing capabilities tailored for spatiotemporal data processing, ensuring timely and context-aware analytics close to the data source. A diverse set of service modules operates at the edge, including AI/ML inference engines, federated learning modules, multi-agent reinforcement learning components, and real-time data storage solutions. These modules collaboratively support intelligent decision-making at the network's periphery, reducing reliance on centralized infrastructure. Moreover, the edge layer integrates automated outlier detection and data consolidation mechanisms to enhance data reliability and integrity before transmission to higher layers. By offloading computation and intelligence to the edge, AI4MultiGIs ensures scalable, efficient, and responsive geospatial data processing across distributed environments.

Cloud layer: The Cloud Layer of AI4MultiGIs serves as the backbone for high-reliability computing, persistent data storage, and centralized intelligence coordination. It hosts scalable and fault-tolerant storage systems capable of managing massive volumes of heterogeneous geospatial and temporal data. At this layer, a multi-source geospatial database, including technologies such as PostGIS and GeoMesa, is employed to store and manage high-resolution spatial data across multiple sources and formats. Complementing this, time series databases like InfluxDB and TimescaleDB handle continuous data streams from IoT sensors, enabling efficient storage, querying, and analysis of temporal patterns.

In addition, knowledge graphs built on geospatial ontologies using standards such as RDF and platforms like Neo4J offer a semantic layer for enhanced data interoperability, relationship modeling, and context-aware queries. These structures allow for deeper reasoning and automated data integration across domains. Furthermore, the data layer incorporates robust data storage modules that are optimized for spatiotemporal indexing, geo-referencing, and semantic harmonization. This provides a unified, queryable foundation that supports AI/ML analytics, digital twin modeling, and decision-support systems. By integrating both structured and semantic data stores, the Data Layer ensures consistency, accessibility, and intelligence throughout the AI4MultiGIs platform.

At this layer, federated learning (FL) aggregators receive and merge model updates from distributed edge clients, supporting privacy-aware, decentralized training of global AI models. Additionally, simulated FL clients are deployed to evaluate and optimize model performance under diverse scenarios. The cloud also executes advanced multi-agent reinforcement learning (MARL) agents, coordinating with their edge counterparts for continuous policy refinement across dynamic urban environments.

A key innovation at the cloud level is the Digital Twin infrastructure, which is a detailed virtual replica of the physical systems that mirror their real-world counterparts in near real-time. These twins support simulation, predictive analytics, and decision-making, and are deeply integrated with AI/ML engines to enable closed-loop learning from digital-physical interactions. Blockchain and decentralized ledger technologies are also embedded within the cloud layer, ensuring secure, auditable, and tamper-proof data exchange across the ecosystem. These decentralized technologies coordinate with modules such as FL aggregators, MARL agents, and digital twins via various secure communication patterns and smart contracts, enforcing transparency and trust in data handling. Moreover, the cloud layer accommodates compute-intensive services, including synthetic data generation for model training, blockchain mining operations, and decentralized orchestration mechanisms. Altogether, this layer forms a high-performance, intelligent, and secure environment that orchestrates the global behavior of the AI4MultiGIs system across all other architectural layers.

The Application Layer in the AI4MultiGIs architecture is the core intelligence hub that orchestrates analytics, decision-making, and user interaction. This layer hosts the AI/ML analytics engine, which is responsible for advanced prediction, classification, anomaly detection, and optimization tasks. These models leverage spatiotemporal data and continuously adapt through feedback from both the digital twin and real-world interactions. In parallel, the workflow orchestration engine uses event-driven triggers to coordinate data pipelines, model execution, and system responses in real time. A central component of this layer is the Decision Support System (DSS), which is responsible for generating actionable insights, summarizing critical events, prioritizing alerts, and conducting scenario-based simulations to support strategic and operational decisions. The DSS integrates closely with both the analytics engine and the user interface, ensuring that recommendations are not only data-driven but also context-aware and tailored to stakeholder needs. Additionally, the Application Layer includes the other modules for providing role-based access to dashboards, interactive maps, and analytical tools. It acts as the bridge between raw data and human decision-making, transforming complex analytics into intuitive visualizations and reports. Altogether, this layer brings together multiple modules—including analytics, orchestration, decision support, and user interaction, into a cohesive environment that empowers end-users with intelligent, timely, and interpretable geospatial insights.

The Presentation Layer of the AI4MultiGIs framework serves as the primary interface between users and the system, offering seamless, intuitive, and interoperable access to geospatial intelligence. At its core, this layer features web-based GIS dashboards, a mobile application interface, and a robust API gateway that facilitates integration with external platforms and microservices. These components ensure versatile access across devices and support both human interaction and machine-to-machine communication. This layer also includes data visualization modules that translate complex analytics and real-time data streams into interactive, multi-layered maps, charts, and reports. Integrated GIS plugins interact dynamically with the core framework, enabling plug-and-play capabilities for extended functionality and domain-specific tools. The presentation layer's modular design allows for easy customization and the addition of new components without disrupting the system, making it adaptable to evolving user needs and technological advancements.

A key feature of this layer is the interpretable strategies and data integration module, which ensures that outputs from AI/ML models are transparent, explainable, and aligned with the decision-making context. This fosters trust and accountability among stakeholders. Ultimately, the Presentation Layer is anchored by the GIS dashboard, the GIS platform integration components, and the system integration plugins and APIs, forming the user-centric core of the AI4MultiGIs framework. These tools collectively empower planners, decision-makers, responders, and citizens with real-time insights, actionable intelligence, and collaborative tools for smarter urban management. The Presentation Layer includes a web-based GIS dashboard, a mobile app interface, and an API gateway for integration with external systems, providing versatile user access and interoperability. The Application Layer hosts the AI/ML analytics engine responsible for prediction, classification, and anomaly detection, along with workflow orchestration through event-driven triggers and a comprehensive decision support system. The Processing Layer manages real-time data streams, batch processing tasks, and data fusion with semantic harmonization modules to unify heterogeneous data sources.

Finally, the Infrastructure Layer comprises IoT edge devices and sensors, cloud and edge computing platforms leveraging Kubernetes, and messaging systems like MQTT and Kafka to support robust, scalable data flows and computation.

4.3 Dependencies and Interaction among key components

4.3.1 Component 1: Novel Customisable, and Trustworthy Data Generation and Management

AI4MultiGIS – Functional Requirements

AI4MultiGIs is designed to support comprehensive geospatial intelligence through a set of integrated functional requirements. It enables real-time acquisition and integration of diverse data sources, including IoT sensors, satellite and drone imagery, and open geospatial datasets, consolidating this information into a unified spatiotemporal database with automated cleaning and semantic enrichment. The platform processes streaming data in real time using protocols such as MQTT and LoRaWAN, with AI-enhanced event detection and automated alerting. Central to AI4MultiGIs is a sophisticated Digital Twin Infrastructure that synchronizes physical urban layers with virtual models at multiple scales, supporting scenario simulations and continuous model learning. Advanced AI-based analytics power predictive modelling for disaster and resource demand forecasting, incorporating privacy-preserving federated learning and embedding intelligence within the digital twin. The Decision Support System provides actionable insights, event prioritization, and scenario planning with automated reporting. User interfaces are role-based and interactive, offering customized dashboards, multi-layer maps, and decision tracing within the digital twin. The system employs distributed multi-agent frameworks for sensor and model management, fostering collaborative decision-making with traceability and conflict resolution. Architecturally, AI4MultiGIs is built on scalable, cloud-native microservices compliant with geospatial standards, facilitating integration via APIs. Finally, blockchain-enabled governance ensures secure, transparent, and auditable data management through decentralized ledgers and smart contracts, guaranteeing data integrity throughout the platform.

As part of its core architecture, AI4MultiGIs will implement an advanced Digital Twin Infrastructure (shown in Figure 22) and an AI-powered Decision Support System to enable intelligent, real-time urban governance. The Digital Twin component will provide a virtual representation of physical city layers, ensuring real-time synchronization between actual infrastructure and its digital counterpart. This system will support multi-scale views—ranging from individual buildings to neighborhoods and the entire city—allowing for granular and holistic monitoring. AI4MultiGIs will enable scenario testing, such as flood mitigation or dynamic traffic rerouting, and incorporate feedback loops for continuous learning and model calibration, ensuring the twin evolves with the city.

The integrated Decision Support System will leverage the digital twin and multimodal AI capabilities to provide contextual, data-driven insights. AI4MultiGIs will utilize real-time and historical geospatial data to generate actionable recommendations, prioritize alerts, and summarize critical events for city operators. What-if simulations embedded in the twin will support strategic planning and emergency preparedness. Automated report generation tools will streamline the dissemination of insights to decision-makers, enhancing transparency and efficiency in urban management.

By combining these components, AI4MultiGIs will create a responsive, intelligent platform for urban analytics—bridging physical infrastructure, digital simulations, and decision intelligence to support sustainable and resilient city development.

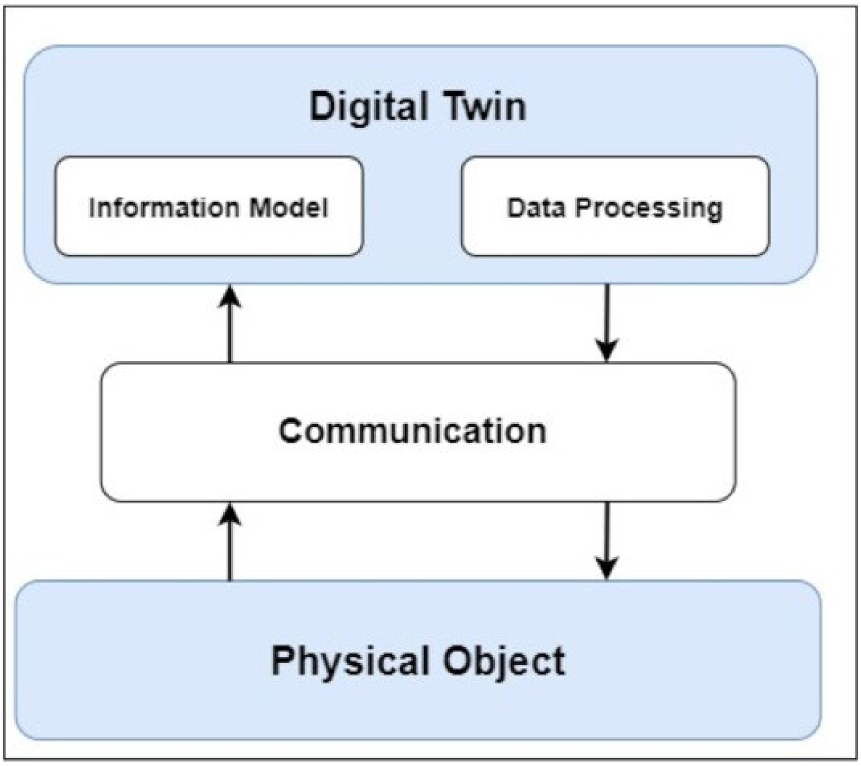

Figure 22: Basic Architecture of a Digital Twin System

AI4MultiGIs will incorporate a robust framework for Multi-Agent and Collaborative Intelligence (shown in Figure 23), enabling decentralized, adaptive, and human-aligned geospatial decision-making. The system will deploy distributed multi-agent systems responsible for managing sensors, data models, and specific decision zones. These agents will interact through localized negotiation and collaboration protocols, supporting autonomous decision-making while preserving global coordination. Furthermore, the platform will facilitate stakeholder collaboration by enabling multi-user coordination with full traceability, shared situational awareness, and conflict resolution mechanisms, fostering trust and transparency among public agencies, private actors, and citizens.

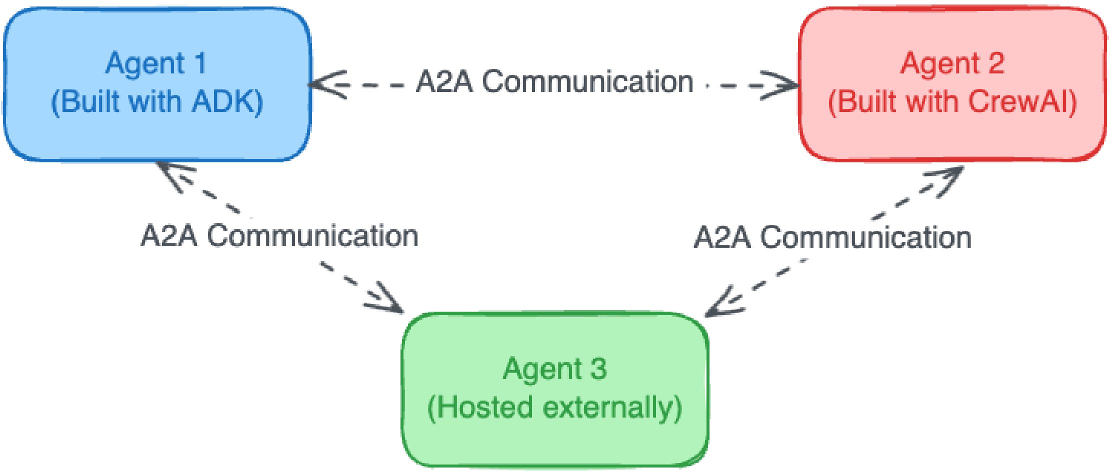

Figure 23: Agent-to-Agent (A2A) Communication Across Heterogeneous Agent Frameworks

To ensure flexibility and robustness, AI4MultiGIs will adopt a Scalable and Interoperable architecture based on cloud-native and edge-compatible microservices. Leveraging technologies like Kubernetes and Docker, the platform will support elastic scaling and dynamic resource allocation across distributed environments. Full compliance with geospatial interoperability standards—including OGC services (WMS/WFS), GeoJSON, and CityGML—will ensure seamless integration with existing systems and datasets, as shown in Figure 24. An API-first design will enable third-party services and developers to extend and interact with the platform, ensuring that AI4MultiGIs remains modular, future-proof, and widely adoptable across diverse urban and regional contexts.

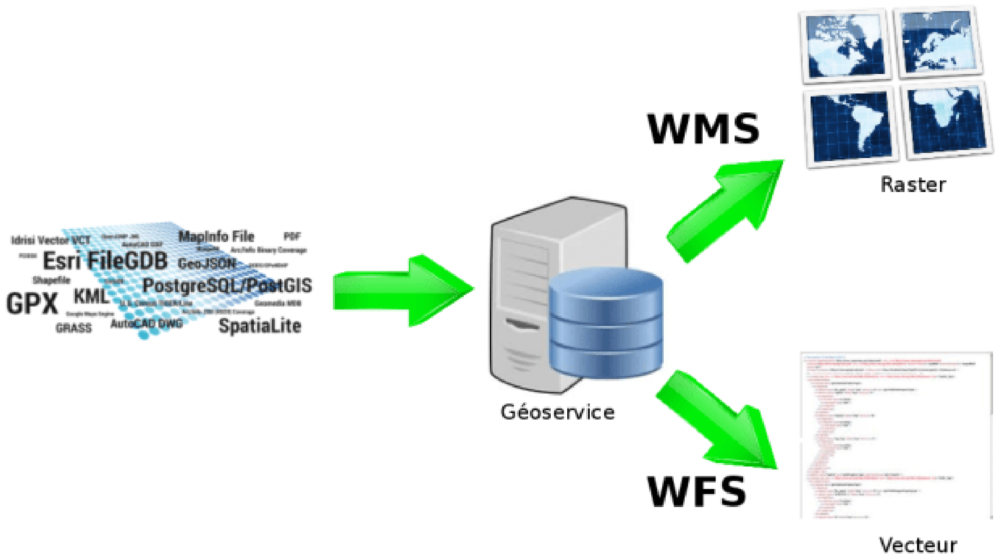

Figure 24: Integration of Multi-format Geospatial Data via Web Services

As a core part of its architecture, AI4MultiGIs will establish an end-to-end pipeline for Data Acquisition and Integration, leveraging diverse sources such as IoT sensors (environmental, infrastructure, mobility), satellite and drone imagery, and open geospatial/administrative datasets like OpenStreetMap and cadastral layers. These heterogeneous data streams will be fused into a unified spatio-temporal database, supported by robust processes for data cleaning, semantic enrichment, and geo-referencing, laying a solid foundation for downstream analytics.

To handle high-frequency updates from urban environments, AI4MultiGIs will implement Real-Time Data Processing through continuous streaming protocols (e.g., MQTT, HTTP, LoRaWAN). Ingested data will be indexed in Time Series Databases (TSDBs) to enable efficient querying and analysis. AI4MultiGIs will combine rule-based and machine learning techniques to detect anomalies and incidents, with automatic alert generation that includes severity classification to support rapid response and prioritization.

Advanced AI-Based Analytics will form the brain of the system, powering predictive models for disaster anticipation, urban risk assessment, and demand forecasting across domains such as mobility, energy, and water. The platform will utilize state-of-the-art learning techniques—including LSTM, Graph Neural Networks (GNN), and hybrid AI models—while supporting federated learning to ensure privacy-preserving distributed model training. These models will be tightly integrated within the Digital Twin, enabling continuous learning from digital-physical interactions and allowing the twin to adapt dynamically to evolving city dynamics.

Finally, AI4MultiGIs will feature Intelligent User Interfaces to democratize access to geospatial intelligence. Custom, role-based dashboards will serve different user groups, including urban planners, emergency responders, and citizens, with region-specific insights. Interactive maps equipped with multi-layer views, time sliders, and forecast overlays will enhance situational awareness. Additionally, user input and decision-making trails will be captured within the digital twin, enabling traceability, feedback incorporation, and improved human-AI collaboration.

4.3.2 Component 2: Integrated Data Handling and Modular Tools for Spatiotemporal Analysis

This component focuses on enabling flexible and modular tools for handling and analysing different types of spatiotemporal data within the AI4MultiGIS framework. These tools are intended to work within the QGIS environment and support key tasks such as data integration, real-time analysis, and plugin-based extensions. The goal is to make it easier for users to process, combine, and understand spatial data from multiple heterogeneous sources.

The tools will be designed and developed using Python and the PyQGIS API. They are planned as standalone QGIS Processing Plugins that can function independently or as part of larger workflows. These plugins will allow users to ingest, clean, transform, and export geospatial data through simple interfaces, while also supporting advanced features such as batch processing, scripting, and automation.

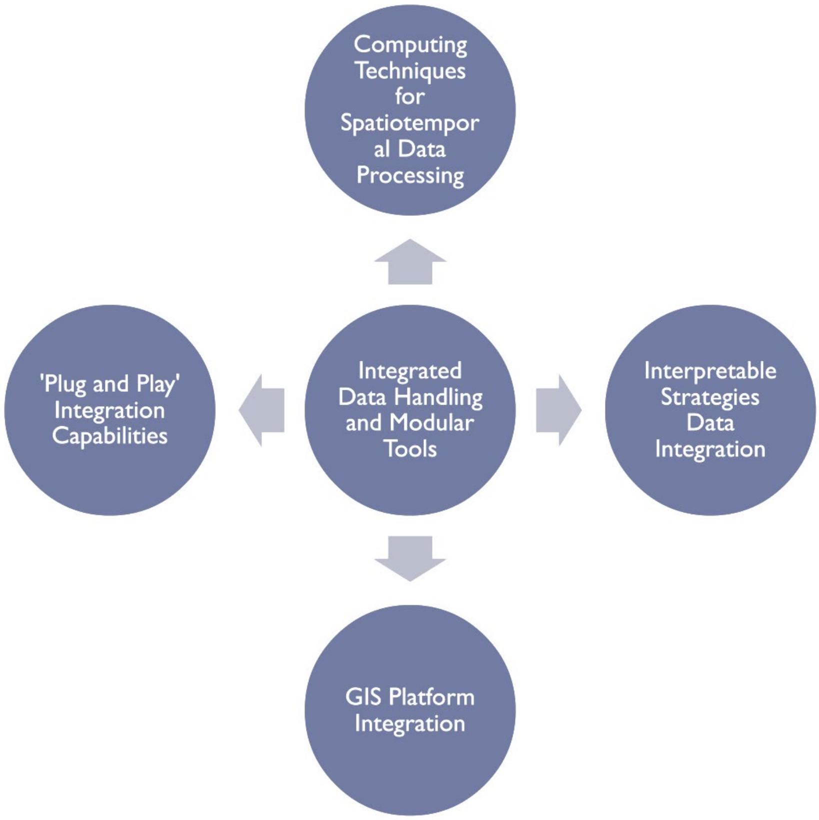

Figure 25: Key components of Integrated Data Handling and Modular Tools in AI4MultiGIS

As illustrated in the Figure 25, this component brings together four key elements:

1. Computing Techniques for Spatiotemporal Data Processing

To support the processing of real-time data from diverse sources, advanced techniques such as Federated Learning (FL) and Multi-Agent Reinforcement Learning (MARL) will be employed. These approaches allow for decentralized, privacy-preserving data processing on edge devices (e.g., sensors, drones), enabling low-latency responses and reducing the need to transmit raw data. Local models can be trained on-device and then aggregated centrally, supporting adaptive and efficient decision-making across distributed environments.

2. Interpretable Strategies for Data Integration

The system architecture is designed to integrate data from different models and modalities, such as climate datasets, satellite imagery, and urban infrastructure maps. To handle this complexity, deep learning models like Convolutional Neural Networks (CNNs), Graph Neural Networks (GNNs), and Recurrent Neural Networks (RNNs) will be used. In addition, interpretability tools such as SHAP and LIME will be incorporated to help explain model outputs, enabling users to understand and trust the results. These strategies will also address challenges like spatial autocorrelation and varying data scales.

3. GIS Platform Integration

QGIS has been selected as the core GIS platform due to its open-source nature, plugin extensibility, and wide adoption. It enables the visualization, editing, and analysis of spatial data using both built-in and custom tools. The system will integrate with the QGIS Processing Framework, allowing seamless use of the developed plugins in both interactive and automated workflows. QGIS also supports batch operations, model building, and scripting, making it a strong foundation for advanced geospatial analysis.

4. ‘Plug and Play’ Integration Capabilities

The architecture is being designed to support modular, plug-and-play integration of tools. This allows new plugins to be added or updated without affecting the rest of the system. These plugins may include components for AI model inference, data export, or real-time sensor integration. Following a consistent development structure, the plugins will be easy to install, manage, and reuse. Both graphical interfaces and automated scripts will be supported, ensuring accessibility for users with varying levels of technical expertise.

Together, these elements provide a flexible and scalable foundation for managing and analysing complex geospatial data. They will support real-time processing, interpretable AI-driven modelling, and seamless integration with external systems—enabling the development of next-generation spatial intelligence applications within the AI4MultiGIS framework.

5 Conclusion

Deliverable D2.2 lays the strategic and technical foundation for the AI4MultiGIS project, serving as the architectural compass that aligns all future developments across work packages. By defining a coherent, modular, and interoperable system architecture, it ensures that the components for data generation, handling, analysis, and integration can evolve harmoniously within a unified framework. This deliverable not only captures the project's vision of a trustworthy, customisable, and intelligent geospatial ecosystem, but also translates it into actionable guidelines and specifications that will drive innovation in subsequent deliverables. As a result, D2.2 empowers the consortium to build a scalable and future-proof AI4MultiGIS platform, ready to tackle the challenges of real-time, AI-driven spatiotemporal data analytics with transparency, adaptability, and resilience at its core.

6 References

(Chen 2022) Chen L, Wang J, Wang H, Jin T. Urban Air Quality Assessment by Fusing Spatial and Temporal Data from Multiple Study Sources Using Refined Estimation Methods. ISPRS International Journal of Geo-Information. 2022; 11(6):330. https://doi.org/10.3390/ijgi11060330

(Salcedo-Sanz 2020) S. Salcedo-Sanz, P. Ghamisi, M. Piles, M. Werner, L. Cuadra, A. Moreno-Martínez, E. Izquierdo-Verdiguier, J. Muñoz-Marí, Amirhosein Mosavi, G. Camps-Valls, Machine learning information fusion in Earth observation: A comprehensive review of methods, applications and data sources, Information Fusion, Volume 63, 2020, Pages 256-272, ISSN 1566-2535, https://doi.org/10.1016/j.inffus.2020.07.004.

(Manzoor 2023) Muhammad Arslan Manzoor, Sarah Albarri, Ziting Xian, Zaiqiao Meng, Preslav Nakov, and Shangsong Liang. 2023. Multimodality Representation Learning: A Survey on Evolution, Pretraining and Its Applications. ACM Trans. Multimedia Comput. Commun. Appl. 20, 3, Article 74 (March 2024), 34 pages. https://doi.org/10.1145/3617833

(Uelwer 2025) Uelwer, T., Robine, J., Wagner, S.S. et al. A survey on self-supervised methods for visual representation learning. Mach Learn 114, 111 (2025). https://doi.org/10.1007/s10994-024-06708-7

(Yaodong 2024) Yu, Yaodong. 'Reliable Representation Learning: Theory and Practice'. EECS Department, University of California, Berkeley, 2024. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-75.html.

(Wu 2024) Di Wu, Yibin Lei, Andrew Yates, and Christof Monz. 2024. Representational Isomorphism and Alignment of Multilingual Large Language Models. In Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024), pages 293–297, Miami, Florida, USA. Association for Computational Linguistics.

(Yu 2028) Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18). AAAI Press, 3634–3640.

(Hao 2020) Hao, Hongyan, Yan Wang, Yudi Xia, Jian Zhao and Shen Furao. "Temporal Convolutional Attention-based Network For Sequence Modeling." ArXiv abs/2002.12530 (2020): n. pag.

(Anastasiou 2019) Evangelos Anastasiou, Annamaria Castrignanò, Konstantinos Arvanitis, Spyros Fountas, A multi-source data fusion approach to assess spatial-temporal variability and delineate homogeneous zones: A use case in a table grape vineyard in Greece, Science of The Total Environment, Volume 684, 2019, Pages 155-163, ISSN 0048-9697,

(Pavlyuk 2021) Dmitry Pavlyuk, Spatiotemporal cross-validation of urban traffic forecasting models, Transportation Research Procedia, Volume 52, 2021, Pages 179-186, ISSN 2352-1465,

(Meijer 2013) Meijer, R.J. and Goeman, J.J. (2013), Efficient approximate k-fold and leave-one-out cross-validation for ridge regression. Biom. J., 55: 141-155. https://doi.org/10.1002/bimj.201200088

(Lee 2024) Lee, J., Kwon, T. & Jung, S. An effective spatial join method for blockchain-based geospatial data using hierarchical quadrant spatial LSM+ tree. J Supercomput 80, 17492–17523 (2024). https://doi.org/10.1007/s11227-024-06134-5

(Daria 2023) Reshetova, Daria, Swetava Ganguli, C. V. Krishnakumar Iyer and Vipul Pandey. "SeMAnD: Self-Supervised Anomaly Detection in Multimodal Geospatial Datasets." Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems (2023): n. pag.

(Bordogna 2023) Bordogna, G., & Fugazza, C. (Eds.). (2023). Artificial Intelligence for Multisource Geospatial Information. MDPI - Multidisciplinary Digital Publishing Institute.

(Foerster 2019) Foerster, T., Ostermann, F., & Kessler, C. (2019). Geospatial Semantic Web for data quality-aware GIS applications. International Journal of Digital Earth, 12(2), 153–171. https://doi.org/10.1080/17538947.2017.1391344

(Prudhomme 2019) Claire Prudhomme, Timo Homburg, Jean-Jacques Ponciano, Frank Boochs, Christophe Cruz, Ana-Maria Roxin, Interpretation and automatic integration of geospatial data into the Semantic Web, Towards a process of automatic geospatial data interpretation, classification and integration using semantic technologies, Computing (2019), Springer, pp 1–27, 13 February 2019, https://doi.org/10.1007/s00607-019-00701-y (IF 1.654, 2019)

(Prudhomme 2017) C. Prudhomme, T. Homburg, F. Boochs, A. Roxin, C. Cruz, Integration, quality assurance and usage of geospatial data with semantic tools, gis.Science, p.91-96, ISSN 1869-9391, 2017.

(De Paoli 2024) Flavio De Paoli, Michele Ciavotta, Roberto Avogadro, Emil Hristov, Milena Borukova, Dessislava Petrova-Antonova, Iva Krasteva, An interactive approach to semantic enrichment with geospatial data, Data & Knowledge Engineering, Volume 153, 2024, 102341, ISSN 0169-023X,

(Bilson 2025) Bilson, Samuel and Anna Pustogvar. “Uncertainty-aware Bayesian machine learning modelling of land cover classification.” ArXiv abs/2503.21510 (2025): n. pag.

(Temitope 2024) Akinboyewa, Temitope, Zhenlong Li, Huan Ning and Mohammad Naser Lessani. “GIS Copilot: Towards an Autonomous GIS Agent for Spatial Analysis.” ArXiv abs/2411.03205 (2024): n. pag.

(Das 2023) Das, Abhimanyu, Weihao Kong, Rajat Sen and Yichen Zhou. “A decoder-only foundation model for time-series forecasting.” ArXiv abs/2310.10688 (2023): n. pag.

(Garza 2023) Garza, Azul, Cristian Challu and Max Mergenthaler-Canseco. “TimeGPT-1.” (2023).

(Liao 2025) Wenlong Liao, Shouxiang Wang, Dechang Yang, Zhe Yang, Jiannong Fang, Christian Rehtanz, Fernando Porté-Agel, TimeGPT in load forecasting: A large time series model perspective, Applied Energy, Volume 379, 2025, 124973, ISSN 0306-2619,

(Zhang & Li, 2023) Zhang, Y., & Li, X. (2023). Advances in Intelligent Real-Time GIS Data Integration. ISPRS Journal of Photogrammetry and Remote Sensing, 189, 178-190.

(Chen et al., 2022) Chen, Y., et al. (2022). IoT-Enabled Edge Computing for Real-Time Geospatial Analysis. IEEE Internet of Things Journal, 9(8), 5123-5134.

(Smith & Kumar, 2023) Smith, J., & Kumar, P. (2023). Multimodal Data Integration for Enhanced Urban Planning: A MultiGIS Approach. Computers, Environment and Urban Systems, 95, 101740.

(Jones et al., 2022) Jones, A. B., et al. (2022). Data Pipelines and Analytics for Real-Time Disaster Management: A GIS Perspective. International Journal of Digital Earth, 15(4), 437-455.

(Zhang et al 2020) Zhang, J., & Zhang, X. (2020). "Edge Computing for the Internet of Things: A Survey." *IEEE Access*, 8, 132-144. https://doi.org/10.1109/ACCESS.2019.2965951

(Ahrens et al 2021) Ahrens, M., & Müller, H. (2021). "Blockchain and Decentralized Data Management: A Survey of Applications in Geospatial Data." *Computers, Environment and Urban Systems*, 84, 101542. https://doi.org/10.1016/j.compenvurbsys.2020.101542

(Gama et al. 2019) Gama, J., & Pinto, F. (2019). "Artificial Intelligence for Smart Cities: A Survey." *Journal of Artificial Intelligence Research*, 65, 567-602. https://doi.org/10.1613/jair.1.11587

(Dastjerdi et al. 2016) Dastjerdi, A. V., & Buyya, R. (2016). "A Survey of Edge Computing: Research, Opportunities, and Challenges." *ACM Computing Surveys*, 49(1), 1-39. https://doi.org/10.1145/2815600

(Wang et al. 2022) Wang, L., & Li, X. (2022). "Secure and Scalable Blockchain Technology in Geospatial Applications: A Review." *IEEE Transactions on Computational Social Systems*, 9(1), 54-67. https://doi.org/10.1109/TCSS.2021.3057065

(Qiu et al. 2021) Qiu, T., & Liu, X. (2021). "Retrieval-Augmented Generation for Real-Time Geospatial Intelligence." *Proceedings of the 2021 International Conference on Artificial Intelligence and Big Data*, 284-290. https://doi.org/10.1109/AIBD52883.2021.00054

(Gandhi, n.d.-a) Gandhi, U. (n.d.-a). *Building a Python Plugin (QGIS 3)*. QGIS Tutorials and Tips, [Online]. Available: https://www.qgistutorials.com/en/docs/3/building_a_python_plugin.html. [Accessed: Jun. 16, 2025]

(Gandhi, n.d.-b) Gandhi, U. (n.d.-b). *Creating a Processing Plugin (QGIS 3)*. QGIS Tutorials and Tips, [Online]. Available: https://www.qgistutorials.com/en/docs/3/processing_python_plugin.html. [Accessed: Jun. 16, 2025]

(The Qt Company, n.d.) The Qt Company, "*Qt Designer Manual (Qt 5.15),*" [Online]. Available: https://doc.qt.io/archives/qt-5.15/qtdesigner-manual.html. [Accessed: Jun. 16, 2025]

(QGIS Development Team, n.d.-a) QGIS Development Team. (n.d.-a). PyQGIS API documentation (master branch), [Online]. Available: https://qgis.org/pyqgis/master/. [Accessed: Jun. 16, 2025].

(QGIS Development Team, n.d.-b) QGIS Development Team. (n.d.-b). *QGIS User Manual: Processing Framework (v3.40), [Online]. Available: https://docs.qgis.org/3.40/en/docs/user_manual/processing/index.html. [Accessed: Jun. 16, 2025]*

(Dobias et al., n.d.) Dobias, M., Nijssen, R., Trimaille, É., Liu, B., & Barascu, T. (n.d.). *qgis-minimal-plugin* [Source code]. GitHub. https://github.com/wonder-sk/qgis-minimal-plugin