D4.4 Advances in HPDA and AI for Global Challenges

Date: October 31, 2025

Document Identification			
Status	Final	Due Date	31/10/2025
Version	1.0	Submission Date	31/10/2025

Related WP	WP4	Document Reference	D4.4
Related Deliverable(s)	D4.3, D4.5	Dissemination Level (*)	PU
Lead Participant	ICCS	Lead Author	Nikolaos Chalvantzis
Contributors	ICCS, PSNC, ATOS, FAU	Reviewers	László Környei, SZE
			Harald Koestler, FAU

Keywords:
High-Performance Data Analytics (HPDA), Artificial Intelligence (AI), Surrogate Modelling, Apache Spark, Computational Fluid Dynamics (CFD), Urban Air Pollution (UAP), Renewable Energy Forecasting, Wildfire Simulation, Graph Neural Networks (GNN), Material Transport in Water (MTW), Distributed Computing, Environmental Resilience, HPC Workflows, Lattice Boltzmann Method (LBM), Uncertainty Quantification

Document Information

List of Contributors	
Name	Partner
Nikolaos Chalvantzis	ICCS
Vasiliki Kostoula	ICCS
Angeliki Dimitriou	ICCS
George Filandrianos	ICCS
Georgios Stamou	ICCS
Dimitrios Tsoumakos	ICCS
Jesús Gorroñogoitia	ATOS
Michał Kulczewski	PSNC
Wojciech Stefaniak	PSNC
Filip Depta	PSNC
Aleksandra Krasicka	PSNC
Marcin Lawenda	PSNC
Shubham Kavane	FAU

Document History			
Version	Date	Change editors	Changes
0.1	02/09/2025	Nikolaos Chalvantzis (ICCS)	First draft with TOC, bullets explaining content, responsibilities and timeline
0.2	09/09/2025	Marcin Lawenda (PSNC), Rahil Doshi (FAU)	ToC, timeline and responsibilities approved
0.3	30/09/2025	Jesús Gorroñogoitia (ATOS), Vasiliki Kostoula, Angeliki Dimitriou, George Filandrianos (ICCS), Michał Kulczewski (PSNC)	Content contributions for Chapters 2 and 3

0.4	02/10/2025	Shubham Kavane (FAU), Aleksandra Krasicka, Marcin Lawenda (PSNC)	Content contributions for Chapter 3
0.5	06/10/2025	Michał Kulczewski, Wojciech Stefaniak, Filip Depta (PSNC)	Updated contribution for Sub-Chapter 3.3
0.7	07/10/2025	Nikolaos Chalvantzis (ICCS)	Integration and consolidation of authors' contributions; Document Information provided; Executive Summary, Introduction, Conclusions and References finalized
0.85	28/10/2025	Nikolaos Chalvantzis (ICCS)	Reviewers' comments addressed and consolidated
0.9	30/10/2025	Nikolaos Chalvantzis (ICCS)	Version ready for quality check
0.95	31/10/2025	Rahil Doshi	Quality assurance check
1.0	31/10/2025	Marcin Lawenda	Final check and improvements

Quality Control		
Role	Who (Partner short name)	Approval Date
Deliverable Leader	Nikolaos Chalvantzis (ICCS)	30/10/2025
Quality Manager	Rahil Doshi (FAU)	31/10/2025
Project Coordinator	Marcin Lawenda (PSNC)	31/10/2025

Table of Contents

List of Tables

List of Figures

List of Acronyms

Insert here all the acronyms appearing along the deliverable in alphabetical order.

Abbreviation / acronym	Description
EC	European Commission
Dx.y	Deliverable number y belonging to WP x
Tx.y	Task number y belonging to WP x
WP	Work Package
AI	Artificial Intelligence
AUC	Area Under the Curve
BCE	Binary Cross-Entropy
BP	Burn Probability
BRIEF	Binary Robust Independent Elementary Features
CBAM	Convolutional Block Attention Module
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
CV	Computer Vision
DDP	Distributed Data Parallel

EDA	Exploratory Data Analysis
ETRS89/UTM31N	European Terrestrial Reference System 1989, Universal Transverse Mercator zone 31N
FAST	Features from Accelerated Segment Test
GCN	Graph Convolutional Network
GNN	Graph Neural Network
GPU	Graphics Processing Unit
HOG	Histogram of Oriented Gradients
HPC	High-Performance Computing
HPDA	High-Performance Data Analytics
LBM	Lattice Boltzmann Method
LES	Large-Eddy Simulation
LS	Level-Set (in Signed Distance Field context)
LSTM	Long Short-Term Memory
LU	Lattice Units
MCC	Matthews Correlation Coefficient
ML	Machine Learning
MTW	Material Transport in Water (pilot)
NetCDF	Network Common Data Form
NOx	Nitrogen Oxides
ORB	Oriented FAST and Rotated BRIEF
ORC	Optimized Row Columnar
PCA	Principal Component Analysis
PV	Photovoltaic
RES	Renewable Energy Sources (pilot)
SDF	Signed Distance Field
SIFT	Scale-Invariant Feature Transform
SLURM	Simple Linux Utility for Resource Management
SSM	State Space Model
SURF	Speeded-Up Robust Features
UAP	Urban Air Pollution (pilot)
UB	Urban Buildings (pilot)
VTK	Visualization Toolkit
VTK format	A legacy VTK file format

VTU format	A VTK Unstructured Grid file format
WF	Wildfires (pilot)
WS	Wind Speed
WD	Wind Direction
YARN	Yet Another Resource Negotiator
bio3x3n, bio5x5n	Normalized biomass (0-1) at 3×3 and 5×5 cell kernels representing burnable fuel
cont6m, cont10m, CONT_NORM_2_20	Vegetation continuity (0-100) from WUIX index at 6, 10, and 20 m resolution

Executive Summary

Deliverable D4.4 of the HiDALGO2 project presents a comprehensive account of the significant advances achieved in High-Performance Data Analytics (HPDA) and Artificial Intelligence (AI) since the previous report, D4.3. It reflects the project's maturation from conceptual and prototypical developments into operational, scalable systems capable of supporting real-world environmental challenges. The document highlights not only methodological innovations but also the establishment of robust infrastructures and reproducible workflows that combine large-scale data processing with advanced AI models across multiple application domains.

The project's computational ecosystem has evolved into an integrated HPDA-AI platform built on Apache Spark, HDFS, and JupyterHub. This architecture now supports complex analytics workflows with enhanced provenance tracking, deterministic job orchestration, and secure user management, ensuring scalability, reproducibility, and interoperability among HiDALGO2 partners.

Within this technological framework, all five HiDALGO2 pilot domains demonstrate clear progress. The **Urban Air Project** has moved from an initial abstract concept to an operational, large-scale data analytics pipeline and an under-development AI workflow. Its HPDA component transforms computational fluid dynamics simulations of urban airflow and pollutant transport into actionable compliance assessments aligned with European air quality directives. The new data pipeline can process annual-scale simulations, performing analysis in three different levels of temporal granularity, in under five hours, identifying cases where pollutant concentration exceeds regulatory thresholds. In parallel, an AI-based surrogate emulator is being designed to replicate wind and pollutant dispersion patterns without the need for full-scale simulations – the objective being the development of a full-fledged AI application trained with simulation data, enabling predictive urban air quality management and faster scenario analysis.

The **Urban Buildings** pilot now operates a sophisticated data processing pipeline that aggregates detailed solar exposure simulations into city-scale hexagonal maps of parameterizable granularity, using distributed Spark-based computation to efficiently handle multi-gigabyte datasets. On top of this analytical foundation, the AI component has successfully employed graph neural networks to model and predict the impact of new constructions on solar accessibility and daylight availability for a fraction of the available datasets for this pilot. The vision of HiDALGO2 is to utilise the combination of scalable data processing and interpretable AI to provide urban planners with a powerful toolkit to assist their decision-making process for designing energy-efficient and sustainable built environments.

In the **Renewable Energy Sources** pilot, HPDA and AI methods have been fully integrated into a single operational forecasting framework. Ensemble-based analytics aggregate and weight meteorological simulations to produce reliable climatological insights, while neural networks trained on high-frequency data from a real photovoltaic

farm deliver accurate predictions of daily power generation. The workflow bridges weather modelling and renewable energy forecasting, enhancing resilience and planning under uncertain climatic conditions.

The **Wildfires** pilot successfully combines large-scale HPDA techniques for burn probability estimation with deep learning models that capture the spatiotemporal evolution of fire dynamics. Long short-term memory autoencoders and computer vision-based feature extraction methods have been introduced, enabling the characterization and retrieval of similar wildfire events for rapid response and scenario evaluation. These developments move beyond handcrafted analyses toward automated, data-driven prediction and pattern discovery.

A new pilot domain, **Material Transport in Water**, marks an important expansion of HiDALGO2's scope. Using the *ChannelFlow-Tools* pipeline, a collection of over ten thousand high-resolution lattice Boltzmann simulations has been generated, creating a 20-terabyte dataset suitable for training machine learning applications. On this foundation, a 3D U-Net surrogate model was trained to emulate flow dynamics with remarkable accuracy and computational efficiency. The result is a scalable, AI-driven approach to modelling environmental fluid processes that would otherwise require extensive high-performance simulation time.

Across all HiDALGO2 pilots, the project is targeting the convergence on a unified approach that couples high-performance data infrastructures with intelligent modelling, transforming raw simulation outputs into operational decision-support tools. The integration of distributed analytics, automated data management, and domain-specific AI demonstrates the maturity of the HiDALGO2 framework as a platform for environmental resilience and sustainability research.

In conclusion, D4.4 marks a decisive step forward in realizing HiDALGO2's vision of harnessing HPC and AI to address global challenges. It consolidates the technical foundations established in the earlier version of this report (D4.3) and transitions them into functional, scalable, and validated workflows. The work reported here not only delivers tangible improvements in computational and analytical capabilities but also establishes a replicable methodology for cross-domain applications. Looking ahead, future activities will focus on expanding dataset coverage, embedding physics-informed AI principles, enhancing workflow efficiency, and incorporating expert feedback to refine predictive accuracy and usability. Collectively, these advancements reinforce HiDALGO2's contribution to the European effort in developing high-impact, sustainable, and intelligent computational tools for understanding and managing complex global systems.

1 Introduction

1.1 Purpose of the document

This deliverable, D4.4, represents the second progress report on High-Performance Data Analytics (HPDA) and Artificial Intelligence (AI) developments within the HiDALGO2 project, covering activities and results from month 16 to month 34. It builds upon the initial findings and methodologies presented in Deliverable D4.3 [1] and precedes the upcoming Deliverable D4.5 at month 45. The purpose of this document is to detail advancements made towards the objectives of Work Package 4, focusing on scalable analytics frameworks, integrated AI pipelines, and domain-specific pilot applications that address pressing global challenges in urban environments, renewable energy, material transport, and environmental risk assessment.

1.2 Relation to other project work

The outputs reported herein directly advance Task 4.2 (HPDA workflows) and Task 4.3 (AI model development), leveraging and enhancing foundational capabilities established in Task 4.1 (data management and processing), Task 4.4 (visualization and semantic analysis), and Task 4.6 (uncertainty quantification). This deliverable also integrates innovations developed in Work Package 3, particularly T3.4, which advances exascale-capable HPC technologies that support efficient execution of the reported analytics pipelines. Synergistic interactions with WP5 pilots ensure that novel HPDA and AI tools are validated against real-world scenarios across energy, urban, environmental, and disaster resilience domains. Deliverable links to both preceding (D4.3) and future (D4.5) reports serve to document progressive refinement and scaling of methodologies.

1.3 Structure of the document

Chapters 2 through 4 provide a detailed narrative of the project's technical achievements. Chapter 2 focuses on the development of foundational HPDA frameworks, scalable data pipelines, and computational strategies enabling high-throughput analytics. Chapter 3 presents pilot-specific AI and HPDA advancements across the Urban Air Pollution (3.1), Urban Buildings (3.2), Renewable Energy (3.3), Wildfires (3.4), and Material Transport in Water (3.5) domains, highlighting both algorithmic innovations and application outcomes. Chapter 4 provides concluding remarks and outlines the roadmap for future technical developments leading to Deliverable D4.5. Comprehensive references and technical appendices accompany these chapters to ensure reproducibility, detailing data provenance, software environments, and experimental configurations.

1.4 Updates since D4.3

This section highlights the key progress and new work accomplished since the submission of Deliverable D4.3. It focuses on novel methods, data enhancements, improved results, and limitations addressed across the HiDALGO2 pilots and HPDA/AI workflows.

Pilot Domain	Key Updates Since D4.3
Urban Air Pollution	Data ingestion pipeline optimized; integrated static and dynamic datasets; shared HPDA/AI data infrastructure established.
Urban Buildings	Transition from concept to production Spark pipelines; enhanced graph neural network modelling; improved interpretability.
Renewable Energy	Operationalized HPDA ensemble aggregation; AI models retrained on detailed mesoscale datasets; climate impact analysis refined.
Wildfires	Shift to deep temporal LSTM representations; implemented expert evaluation protocol; added CV features; enhanced large-scale HPDA processing for burn probability mapping and sensitivity analysis.
Material Transport in Water	New pilot with automated HPC pipeline and ChannelFlow dataset; developed advanced 3D U-Net surrogates; first large-scale environmental fluid ML application in HiDALGO2.

2 Software stack and infrastructure updates

Chapter 2 details the foundational High-Performance Data Analytics (HPDA) frameworks and computational frameworks that support the advancements reported in this deliverable. It focuses on scalable data processing pipelines, uncertainty quantification techniques, and HPC workflow optimizations that enable the efficient handling of large and complex environmental datasets across diverse pilot domains. By establishing the core technical infrastructure and analytic building blocks, this chapter sets the stage for subsequent pilot-specific AI and HPDA applications discussed in Chapter 3 – unless explicitly stated otherwise.

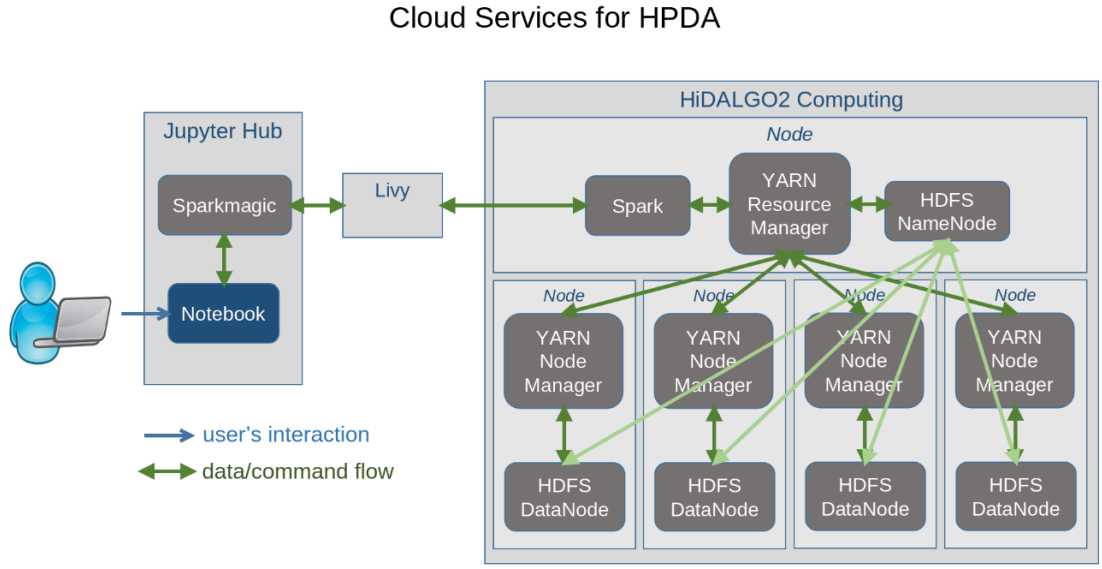

Figure 1: Cloud Services for HPDA.

The architecture described in this chapter is a core-enabling layer within the HiDALGO2 ecosystem, designed specifically to support HPDA workflows and, where applicable, AI-driven data manipulation and pre-processing tasks. This environment, built on cloud-native tools and distributed computing frameworks, is not part of the project's HPC infrastructure and does not include access to specialized accelerators such as GPUs. Instead, it provides flexible, scalable compute and storage resources for tasks that require interactive use, rapid iteration, and cross-partner collaboration – capabilities that fit HPDA requirements and, for AI, are primarily relevant in the context of feature engineering, dataset transformation, and pipeline automation prior to heavy model training.

By implementing standardized APIs, shared data models, and collaborative interfaces, the infrastructure facilitates seamless integration of these pre-processing and analytics steps across all domains of the HiDALGO2 project. Here, partners can contribute, refine, and deploy workflows supporting their respective pilot applications with minimal overhead, taking advantage of the cloud's ability to rapidly provision resources and accommodate diverse software stacks without the constraints typical of HPC environments. This approach enables continuous integration and distributed

development, with all data handling, large-scale ingestion, transformation, and initial analytics being performed efficiently and transparently for all project stakeholders.

The choice of cloud for these phases, instead of a traditional HPC setting, is deliberate. HPDA and data wrangling benefit from the cloud's inherent elasticity and interactive capabilities – making it possible to scale resources as needed, experiment with new processing chains, and quickly adapt to emerging requirements. By contrast, HPC centres are best suited for compute-intensive simulation workloads and large-scale, finalized AI model training that requires specialized hardware. For the HPDA and initial AI processing stages, the operational agility, software freedom, and shared access afforded by cloud infrastructure yield significant productivity and collaboration advantages, ensuring that the project's data ecosystems are both robust and adaptable.

It has already been established that the computation HPDA on massive volumes of data requires large, scalable, and distributed computing over the data. In the Big Data domain, data is managed by highly distributed storage systems, such as the Hadoop HDFS. HiDALGO2 has adopted this technology to offer a small cluster[1] with Hadoop HDFS where pilots can store and retrieve their data. Maintenance duties for this cluster are part of HiDALGO2 task T4.1. The analytics designed and implemented for the HiDALGO2 pilots process the HDFS data with a computing platform that has close and fast access to the data, as the storage and computing share the same infrastructure.

In HiDALGO2, a shared HPDA platform comprising JupyterHub [2] (gateway), SparkMagic [3] (notebook kernel), Livy [4] (HTTP bridge), Apache Spark [5] on YARN [6] (compute), and HDFS [7] (storage) was set up and deployed in the context of T4.1 and will be reported in the D4.2 deliverable. This stack enables reproducible HPDA and AI workflows for T4.2/T4.3 with notebook-first submission to Spark clusters and a common HDFS data lake. The architecture is presented in Figure 1.

All endpoints (HDFS, YARN, Livy, and UIs) are Kerberos-protected [8], and users operate with individual accounts; Spark/Livy jobs and HDFS access run under each user's identity for auditing and access control. Details on security setup are also covered in Deliverable 4.2.

Figure 1 shows the overall architecture of the HPDA platform:

- Notebooks (in the HiDALGO2 JupyterHub) leverage the SparkMagic library. This library is used to interact with remote Spark clusters from Jupyter notebooks through the Livy REST server. SparkMagic offers an API and some preconfigured kernels for Spark, Python, and R. In particular, by using the PySpark kernels, users can create a live Livy session and send Spark commands in Python notebooks to the Spark cluster.

[1] The size and capability of this cluster is constrained by the available resources, but enough to address the pilots' needs.

- The Livy server provides a gateway between Cloud-based services (such as the Jupyter Hub) and the Spark cluster. It enables multi-tenant submission of Spark jobs to Spark clusters for Web applications. Livy supports both Python and Scala for job submission.
- The Spark cluster runs on top of the YARN cluster and dispatches Spark jobs over the YARN computed nodes, mediated by the YARN resource manager. Spark is one of the de facto industrial standards for data analytics, with support for EDA and Machine Learning, both in batch and streaming.
- The YARN cluster is part of the Hadoop ecosystem and accompanies the installation of the HDFS, acting as a resource manager and job scheduler. YARN supports several workloads, such as MapReduce, Spark, and services like HBASE and others. By using YARN, Spark jobs are distributed across the available YARN nodes.
- The HDFS namenode (NN), which runs on the same node where the YARN resource manager runs, provides access to the data upon Spark requests.
- Both the YARN compute nodes and the HDFS data nodes (DN) are hosted on the same network of scalable nodes. In this way, the Spark jobs are running on the same nodes that host the data, getting fast access to it.

2.1 Data storage and access

HDFS is the main data lake for HPDA. The available HPDA cluster consists of 6 nodes, including a NN, a secondary NN, and 4 DNs – see Table 1. Additional data nodes can be added on demand. The DNs offer 16 Cores, 32 GB RAM, and 10 TB of storage for both data and computing per node. The NN (and secondary NN) has 8 cores, 16 GB RAM, and 540 GB of storage per node. Data is stored in the datanodes with replication r=1, resulting in the total storage space of 40 TB.

Table 1: HiDALGO2 Hardware resources for HDFS.

Layer	Service	#Nodes	#Cores	RAM	Storage
Storage/DN	HDFS	4	16	32 GB	10 TB
Storage/NN	HDFS	2	8	16 GB	540 GB

Access to HDFS is secured with Kerberos. Users require a Kerberos principal, which is used to log in with the `kinit` command, to get a token valid for 24 hours. Once the token is available on the user's computer, access to the HPDA services' Web portals, APIs, and CLIs for HDFS is granted.

2.2 Processing frameworks

Table 2 shows the hardware resources available for the HDPA cluster. There are 4 nodes with 16 Cores, 32 GB RAM, and 10 TB of storage for computing. Additional nodes can be added on demand. One additional node, with 8 cores, 16 GB RAM, and 540 GB of storage, hosts the YARN resource manager. The compute nodes are the same as the HDFS data nodes, so the YARN jobs are executed over the same nodes hosting the data.

Table 2: HiDALGO2 Hardware resources for HPDA.

Layer	Service	#Nodes	#Cores	RAM	Storage
Compute	YARN/Spark	4 (compute) 1 (manager)	14 2	28 GB 4 GB	10 TB 540 GB
Gateway	Jupyter Hub	1	32	94 GB	3 TB

Table 3 shows the endpoints for services included in the HPDA cluster.

Table 3: HiDALGO2 HPDA Services.

Service	Endpoint
HDFS	http://sophora-42.man.poznan.pl:9870
YARN	http://sophora-42.man.poznan.pl:8088
Livy	http://sophora-42.man.poznan.pl:8998

The versions of the HPDA software stack are:

- JupyterLab: 4.3.5,
- SparkMagic: 0.22.0,
- Livy: 0.8.0,
- Spark: 3.5.3,
- Hadoop/YARN: 3.3.6,
- Python: 3.13.7.

Users must install the SparkMagic library in the Jupyter Hub portal, accessing it with their account, and following the instructions officially provided by the development team of SparkMagic:

https://github.com/jupyter-incubator/SparkMagic/blob/master/README.md

2.3 Infrastructure Usage Details

To use the HDPA platform in a notebook, users need to select the PySpark kernel (see Figure 2) and issue Spark commands in the notebook cells. In the first executed cell, a Livy session is created with the requested resources. Users define default resources in the `~/.SparkMagic/config.json` file, but they can also be overridden in the notebook by adding a `%%configure -f {}` block, as shown in Figure 3. Once the session is created, the following cells are submitted for execution as Spark jobs (see Figure 4). Then, users can check the status of the Spark jobs executed in the YARN portal (see Figure 5).

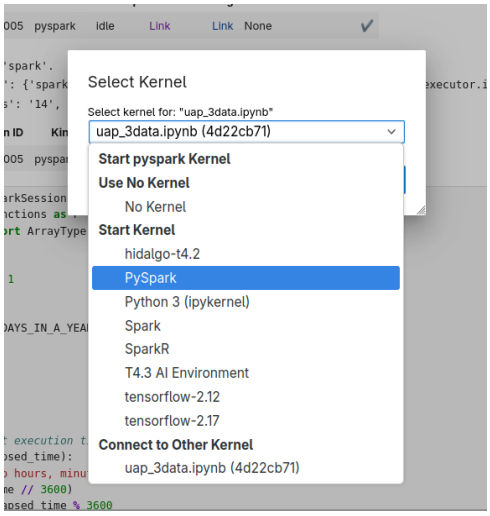

Figure 2: SparkMagic kernels in a Jupyter notebook.

```
# Spark session defaults for WP4
%%configure -f
{
  "conf": {
    "spark.executor.instances": "4",
    "spark.executor.cores": "8",
    "spark.executor.memory": "12g",
    "spark.driver.memory": "12g"
  }
}
```

Figure 3: Override Spark session default configuration.

Further technical details of this computing platform will be reported in D4.2.

3 HPDA and AI integration per pilot

The detailed advancements achieved in applying High-Performance Data Analytics (HPDA) and Artificial Intelligence (AI) across the HiDALGO2 project's diverse pilot domains are presented in this chapter. Building upon the foundational frameworks and data infrastructures outlined in earlier chapters, this section focuses on the development, deployment, and integration of scalable analytic workflows and AI models tailored to address complex environmental, urban, and energy system challenges.

The chapter is organized by pilot, showcasing tailored methodological innovations, data processing pipelines, and modelling results in the domains of Urban Air Pollution, Urban Buildings, Renewable Energy, Wildfires, and Material Transport in Water. Each pilot illustrates a unique combination of HPDA and AI capabilities, emphasizing reproducibility, scalability, and operational readiness on high-performance computing platforms.

Through these case studies, Chapter 3 highlights how the project's core objectives are advanced by bridging large-scale data analytics with state-of-the-art AI methodologies, in a quest towards offering actionable insights and tools to support climate resilience, sustainable urban planning, and environmental risk management.

3.1 Urban Air Project (UAP)

3.1.1 Scope, Objectives & Inputs

The HiDALGO2 Urban Air Project (UAP) is dedicated to modelling air pollution concentrations and wind-related indicators within urban environments at very high spatial and temporal resolution. By combining detailed computational fluid dynamics (CFD) simulations with advanced data analytics and machine learning, UAP delivers actionable information that supports urban planners and policymakers in understanding how pollutants disperse and accumulate in densely built areas, and how wind conditions impact both air quality and pedestrian comfort.

To address the enormous computational demands of simulating city-scale airflow and pollutant transport, the project leverages high-performance computing (HPC) resources capable of resolving millions of grid cells. The resulting simulation outputs – initially produced in EnSight Case Gold [9] format and totaling approximately 1.2 TB – are converted into Apache ORC (Optimized Row Columnar) [10] format (around 1 TB) for efficient downstream processing. This transformation yields two main data artifacts: i) a single *geography* file per location/urban environment describing the three-dimensional urban mesh structure, and ii) timestep-based *snapshot* files containing velocity vectors, pressure fields, and nitrogen oxide (NO_x) concentrations for every grid cell.

Within this framework, UAP pursues two complementary, equally critical computational approaches. The first, High-Performance Data Analytics (HPDA), systematically aggregates simulation results to identify locations where nitrogen dioxide (NO_2) concentrations breach the EU Directive 2024/2881 [11] thresholds. After filtering the three-dimensional grid to retain only cells whose centres of mass lie below two meters of height from street level – corresponding to the human exposure urban environment – hourly averages of NO_2 concentration are computed and compared against the regulatory limit. Annual exceedance counts are then tallied, generating a notification in the case where the permitted number of violations is exceeded. This process is repeated for the detection of daily and yearly averages of pollutant concentrations. Implemented on the Apache Spark platform (introduced in 2.2) and using the storage service of Apache HDFS (see 2.1), this HPDA pipeline has been rigorously benchmarked and shown to scale linearly as dataset sizes grow, enabling timely compliance assessment even for year-long simulations.

In parallel, the project's Artificial Intelligence (AI) track aims to develop a surrogate emulator that can predict the temporal evolution of wind patterns and pollutant dispersion without executing full-order CFD simulations. By training on structured datasets that couple state variables with their corresponding time-stamps, boundary conditions, and initial conditions, the AI model learns how to reproduce the dynamic behaviour of urban airflow and pollutant transport under varying environmental scenarios. This emulator can offer a rapid, resource-efficient alternative for scenario testing, policy evaluation, and proactive air quality management, complementing the HPDA-based compliance assessments with predictive capabilities.

3.1.2 HPDA application[2]

3.1.2.1 Data Pre-processing

Before applying the HPDA workflow, a critical pre-processing step transformed the raw simulation outputs from their original format into a structure optimized for large-scale distributed analytics. The Urban Air Project simulation data was initially delivered in EnSight Case Gold binary format, with results organized into monthly case directories spanning a full calendar year. Within this structure, simulation outputs for the three-dimensional urban mesh grid were stored as binary files distributed across computational nodes, with temporal snapshots recorded at 600-second intervals. Each snapshot captured the complete state of the flow field, including three-component velocity vectors, pressure distributions, and nitrogen oxide (NOx) pollutant concentrations across all grid cells.

To enable efficient querying and distributed processing within the Apache Spark framework, this binary data underwent systematic conversion into Apache ORC

[2]Code repository: https://git.hidalgo2.eu/hidalgo2-group/hid-hpda-uap

format. The transformation produced two complementary file categories that together constitute the analysis-ready dataset. First, a static geography file encodes the three-dimensional mesh topology – defining the spatial coordinates, connectivity, and volumetric structure of the urban grid cells. Second, a collection of temporal snapshot files stores the time-varying flow field and pollutant concentration data, with each file containing velocity, pressure, and NOx values indexed by grid cell and timestep. A unique identifier links each geography file to its corresponding temporal snapshots, ensuring referential integrity.

This pre-processing pipeline not only provided the structured input foundation required for the HPDA compliance algorithms but also established a shared data infrastructure accessible to both the HPDA and AI development teams, eliminating redundant transformation efforts and ensuring consistent data provenance across both methodological tracks.

The High-Performance Data Analytics (HPDA) implementation for the Urban Air Project has been specifically designed to assess regulatory compliance with EU Directive 2024/2881 regarding nitrogen dioxide ($\mathrm{NO_2}$) pollution limits. The core algorithm evaluates air quality threshold exceedances through systematic hourly aggregation and analysis of high-resolution simulation data.

3.1.2.2 HPDA Workflow

To ensure direct relevance to human exposure assessment, the three-dimensional simulation grid undergoes spatial filtering to retain only computational cells with centres of mass positioned below two meters altitude. For these pedestrian-level cells, $\mathrm{NO_2}$ concentrations are aggregated at multiple temporal resolutions to ensure full compliance with EU Directive 2024/2881. Hourly averages are computed and compared against the $200\ \mu g/m^3$ limit, with each exceedance flagged. In addition, daily mean concentrations are evaluated against the $50\ \mu g/m^3$ threshold, and, finally, the annual average against the respective $20\ \mu g/m^3$ threshold. Threshold exceedances are systematically flagged when this limit is surpassed, with the algorithm maintaining comprehensive counts of total violations in all three levels of temporal granularity. In strict compliance with the directive requirements, when violation frequencies exceed the permissible limits (more than three violations in hourly granularity, eighteen in daily granularity and a single violation in annual granularity in the span of a calendar year), the fact is identified and the data-set is marked accordingly. A simple version of the algorithm covering the hourly granularity is presented in Figure 4.

Algorithm. The algorithm has been implemented on the Apache Spark distributed computing platform, utilizing the infrastructure configuration detailed in paragraph 2.2 of this deliverable. This technical architecture enables efficient processing of the substantial simulation datasets generated by UAP while ensuring computational scalability for both short-term tactical analyses and long-term strategic simulations.

Algorithm 1 Compliance Check with EU Directive (EU) 2024/2881 for NO_2 Pollution – Hourly Aggregation

Constants:
NUM_SNAPSHOTS_IN_AN_HOUR ← 1
HOURLY_THRESHOLD ← 200 $\mu g/m^3$ ▷ Max 3 exceedances/year
NUM_HOURS_IN_A_YEAR ← 365 × 24

Input:
geography ← 3D grid cells with coordinates (ORC)
simulation ← NO_2 concentrations per cell (ORC)

procedure FILTERLOWCELLS
    geography ← Add "centre_of_mass" column via UDF
    cells_low ← Filter cells with $z \leq 2$m
    sim_low ← Join(cells_low, simulation, on = "id")
end procedure

procedure HOURLYAGGREGATION
    hourly_df ← GroupBy(sim_low, $\left\lfloor \frac{\text{timestep}}{\text{NUM\_SNAPSHOTS\_IN\_AN\_HOUR}} \right\rfloor$ as "hour", "id")
        Aggregate: avg_nox ← AVG("nox")
    hourly_df ← AddColumn(hourly_df, "violation", IF avg_nox > HOURLY_THRESHOLD THEN 1 ELSE 0)
end procedure

procedure COUNTYEARLYVIOLATIONS
    hourly_violations ← GroupBy(hourly_df, "hour")
        Aggregate: violation ← MAX("violation")
    yearly_violations ← GroupBy(hourly_violations, $\left\lfloor \frac{\text{hour}}{\text{NUM\_HOURS\_IN\_A\_YEAR}} \right\rfloor$ as "year")
        Aggregate: total_violations ← SUM("violation")
    Output: yearly_violations ▷ Ensure ≤ 3 per year
end procedure

FILTERLOWCELLS()
HOURLYAGGREGATION()
COUNTYEARLYVIOLATIONS()

Figure 4: The algorithm implemented in the UAP HPDA task (addressing hourly aggregation)

Benchmark results. Comprehensive full-scale experiments have been executed to assess the scalability and computational efficiency of the HPDA workflow under increasing data volumes. The benchmarking methodology employed a systematic approach, progressively increasing the temporal extent of analysed datasets from one month of simulation data through three months and six months, culminating in analysis of a complete annual dataset. This incremental scaling approach enabled thorough evaluation of execution time scaling characteristics and identification of potential computational bottlenecks within the processing pipeline. It should be noted that at this point, we are more interested in achieving highly scalable performance rather than

studying the qualitative results of the analysis, which will be part of our future objectives.

Figure 5 presents the measured execution times across all benchmark scenarios. The testbed for this experiment was the Apache Spark and HDFS cluster presented in Chapter 2. More specifically, the workload was executed using 4 Spark Executors, each equipped with 8 virtual cores and 12 GB of memory. The input size of the whole data set is approximately 1 TB in ORC format. The results demonstrate close to linear scaling behaviour, with execution times progressing from approximately 25 minutes for one month of data analysis, extending to just over one hour for three-month datasets, nearly four hours for six-month analyses, and approaching five hours for complete annual processing. These benchmarking results confirm that the HPDA pipeline exhibits reliable scalability with increasing dataset dimensions, demonstrating robust capability to handle long-term simulations at full operational scale while maintaining execution times suitable for practical urban air quality monitoring and planning applications.

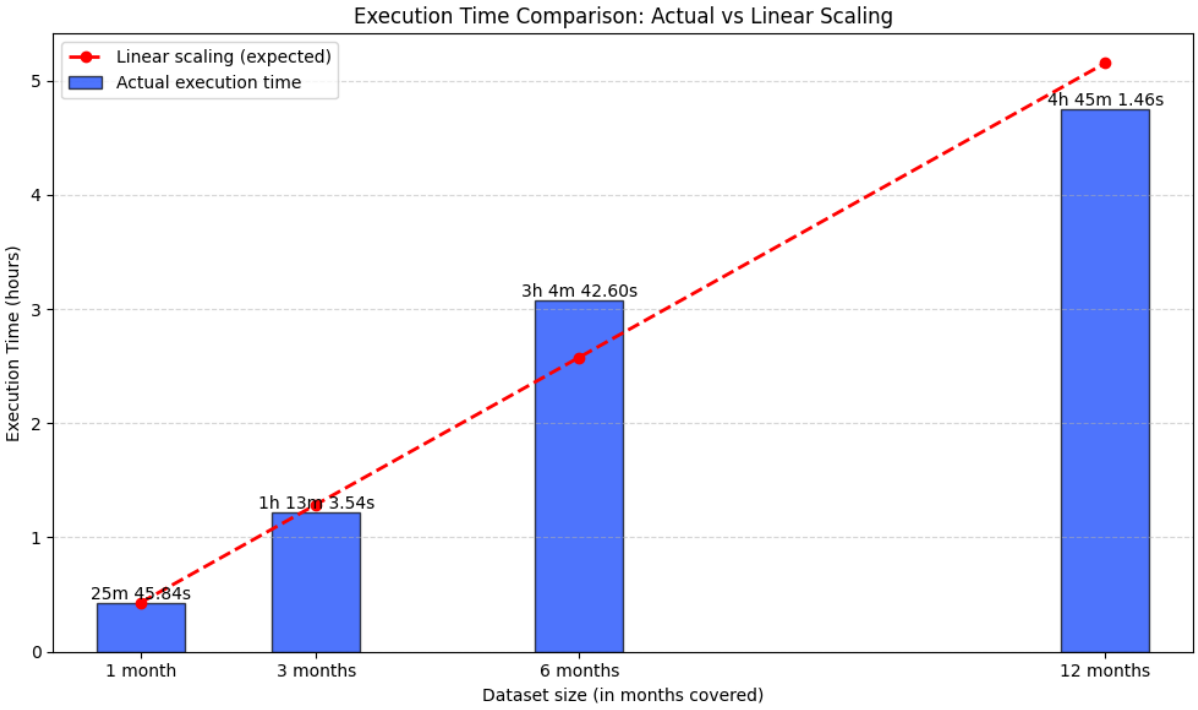

Figure 5: Execution time vs data-set temporal range (UAP HPDA).

3.1.2.3 Future Development Roadmap

Planned enhancements include exploration of additional analytical use cases, with particular emphasis on developing forecasting applications, where historical simulation data is utilized to train AI models for predictive air quality management. This approach will integrate advanced machine learning methodologies into the existing workflow, extending functionality beyond compliance monitoring toward proactive environmental management. Additional optimization efforts will focus on algorithmic refinement and

data processing pipeline enhancement to improve computational efficiency and reduce resource consumption.

3.1.3 AI application

The UAP full-order model (FOM) simulations are mathematically represented as sequences of snapshot series $S_{conf}=[X_1,time_1,parBC_1,parIC_1,...]$. The primary objective of the AI component is to construct a trained emulator based on S_{conf} that can predict the system state X at specified temporal points, given boundary conditions, $parBC$ and initial conditions, $parIC$. In other words, the emulator serves as a surrogate model capable of reproducing complex pollutant dispersion patterns and wind-related indicators with significantly reduced computational requirements.

A fundamental challenge lies in the substantial scale and complex format of simulation outputs generated by the full-order model. The raw FOM datasets are extremely large and cannot be directly processed by conventional AI methodologies without significant pre-processing. Consequently, the initial development phase required extensive application of HPDA techniques to transform, compress, and restructure simulation outputs into formats optimized for efficient querying and machine learning processing. Only following this critical pre-processing stage could the data be converted into structured inputs suitable for AI algorithm training and inference. This pre-processing stage has been merged with the pre-processing routine described earlier in 3.1.2.1.

The construction of the UAP AI emulator represents an active area of ongoing research and development within the project. Since the publication of Deliverable D4.3, where the use case was initially defined and a preliminary processing pipeline was outlined, development efforts have focused on addressing the practical computational and infrastructure challenges inherent in working with large-scale simulation datasets. This phase has necessitated intensive collaboration with the HPDA development team to ensure efficient data processing, storage optimization, and reliable access protocols, alongside the procurement and configuration of appropriate computational infrastructure resources.

With these foundational technical components now established, the project has reached a critical transition point where experimental validation using the proposed AI pipeline represents the immediate next milestone in UAP emulator development. The infrastructure and pre-processing capabilities are now in place to support the intensive training and validation phases required for surrogate model development, marking a significant advancement from the conceptual framework presented in D4.3 toward practical implementation and performance evaluation.

3.1.4 Summary

Since the initial framework outlined in Deliverable D4.3, the Urban Air Project has transitioned from conceptual design to robust operational workflows in both HPDA and AI tracks. On the HPDA side, compliance algorithms have been fully implemented

within an Apache Spark environment and rigorously benchmarked across datasets ranging from one month to a full year, demonstrating linear scaling and reliable execution times of under five hours for annual simulations. The regulatory compliance pipeline now assesses exceedances at hourly, daily, and annual resolutions in strict accordance with EU Directive 2024/2881, automatically identifying zones with violations beyond permitted limits.

In parallel, foundational infrastructure for the AI-based emulator has been established. High-volume CFD outputs have been ingested, transformed, and compressed into analytics-ready formats, and the end-to-end pre-processing pipeline is now in place. This groundwork enables the imminent transition to training and validating surrogate models that predict pollutant dispersion and wind indicators without full-order simulations.

Combined, these dual tracks can form a cohesive UAP workflow: the HPDA component provides robust, regulation-driven analysis of current simulation data, while the AI emulator aims to extends the project's reach into forecasting and rapid scenario exploration – once complete. This integrated approach can ensure that UAP not only meets present compliance requirements but also possibly empower decision-makers with predictive insights for effective urban environmental management.

3.2 Urban Building (UB) [3]

3.2.1 Scope, Objectives & Inputs

The Urban Buildings pilot develops advanced simulation tools to forecast energy consumption, thermal comfort, and indoor air quality at both building and neighbourhood scales, thereby supporting efforts to reduce energy waste, enhance urban sustainability, and improve the well-being of inhabitants. By capturing interactions between individual buildings and their urban context, this pilot reveals how factors such as solar exposure and shading influence both energy demand and occupant comfort, informing more resilient and human-centric city planning. A full technical description of the Urban Buildings workflows is available in Deliverable D5.3 [12].

Within this framework, the High-Performance Data Analytics team focuses on transforming detailed solar simulation outputs into scalable, actionable insights by aggregating three-dimensional building geometry into H3 [13] hexagonal grids. Starting from an EnSight Case Gold dataset of roughly 600 MB – which describes each building façade as triangular surface elements – the pipeline converts this data into approximately 10.9 GB of Apache ORC files optimized for distributed querying. Geolocation files encode the urban mesh, while time-series snapshots record solar

[3] Shared code repository for HPDA and AI: https://git.hidalgo2.eu/hidalgo2-group/hid-ai-hpda-ub

shading coefficients for each surface triangle at every timestep. Leveraging Apache Spark, the HPDA workflow computes aggregate solar exposure metrics across the hex grid, pinpointing optimal areas for photovoltaic deployment as well as zones of over- or under-exposure with direct implications for public health, comfort, and urban design.

Concurrently, the Artificial Intelligence team builds on the same simulation foundation in collaboration with the UNISTRA group to create predictive models of how new constructions alter the solar exposure of urban environments. By systematically removing individual buildings from the simulation domain and capturing the resulting evolution of shading coefficients across neighbouring structures, a comprehensive dataset is generated to reflect the influence of each building on its surroundings. This full-scale dataset is represented as a large NetworkX [14] graph, where nodes correspond to buildings and edges encode the magnitude of mutual solar interference. The AI pipeline trains graph-based neural networks to predict the presence and weight of these influence edges, enabling rapid evaluation of proposed developments and their impact on solar intake, daylight availability, and thermal comfort in adjacent spaces.

Combined in tandem, these HPDA and AI approaches aim to provide a unified toolkit: HPDA delivers robust, hex-grid–based mapping of current solar conditions at urban scale, while AI offers predictive insights into how future construction scenarios reshape building-to-building solar interactions. We envision that this integrated capability will be able to empower stakeholders to design energy-efficient, comfortable, and sustainable urban environments.

3.2.2 HPDA application

3.2.2.1 Data Pre-processing

Before executing the HPDA workflow, a comprehensive pre-processing step transformed the raw solar simulation outputs into a structure optimized for distributed analysis on Apache Spark. The Urban Buildings pilot received two complementary EnSight Case Gold binary datasets representing a study region in Strasbourg. The first dataset comprises temporal simulation results spanning one calendar month with hourly snapshot intervals, recording solar shading coefficients for each triangular surface element of the urban building façades. The second dataset provides static building metadata that associates each surface triangle with its corresponding building identifier (`building_id`) and building element identifier (`building_element_id`), enabling spatial aggregation and building-level analysis.

The pre-processing pipeline restructured these inputs into two distinct categories of Apache ORC files designed for efficient querying and analysis. The geography file encodes the static geometric description of the urban environment, assigning each triangular façade element a unique hash identifier (`hash_id`) alongside its three vertex coordinates in three-dimensional space and the associated building metadata inherited

from the source dataset. This geometry file serves as the spatial reference framework for all subsequent analyses. The simulation files capture the temporal evolution of solar exposure by linking each hourly timestep to the relevant surface triangles via their hash identifiers and recording the corresponding solar shading coefficients computed by the simulation solver.

This pre-processing transformation ensures dataset consistency through deterministic hash-based linkage between geometry and temporal data, delivers efficient columnar storage enabling rapid filtering and aggregation operations, and provides seamless integration into the subsequent HPDA workflow implemented on the Apache Spark platform. The resulting ORC file structure supports both the hex-grid aggregation algorithms and serves as a potential input source for future AI-based solar prediction models.

3.2.2.2 HPDA Workflow

The HPDA workflow for the Urban Buildings pilot translates detailed solar exposure data from three-dimensional building meshes into an intuitive hexagonal grid representation using H3 indexing. First, each triangular facet of the building surface is analysed to compute its geometric area and geographic centroid. These centroids are mapped to H3 hexagons, allowing each triangle's solar shading coefficient to be weighted by its area and aggregated into a single exposure value per hexagon. By computing monthly averages of these exposure values, the pipeline produces a scalable, city-wide portrait of solar dynamics that captures seasonal and spatial variations across the urban fabric (see Figure 6).

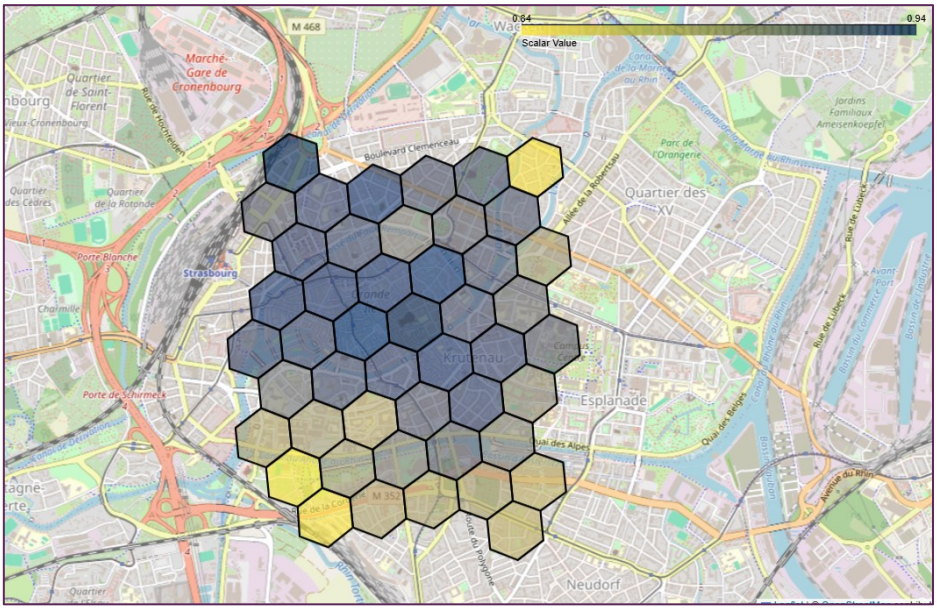

Figure 6: Visualisation of the HPDA workflow output for UB.

Beyond solar exposure, the algorithm (Figure 7) also calculates building density within each hexagon by summing the surface areas of all constituent triangles. By correlating density and exposure metrics, the workflow reveals patterns such as how tightly built

areas may experience reduced solar accessibility. This correlation coefficient serves as a quantitative indicator of the interplay between urban form and solar availability.

Built on the Apache Spark platform described in Chapter 2, the implementation leverages Spark's distributed processing capabilities to ensure a robust and flexible environment, even when handling moderate-sized urban datasets. Benchmarking experiments adjusted executor counts and memory allocations across the cluster but showed negligible performance variation, reflecting the sufficiency of the available resources for the UB dataset scale.

Algorithm 2 Aggregate Solar Exposure to H3 Hexagons

Inputs:
- 3D mesh of building triangles (vertices + timestep solar coefficients [0,1])
- H3 hexagon resolution (default: 9)

Outputs:
- Monthly average solar exposure per hexagon
- Correlation between building density and solar exposure

Procedure:

1. **Preprocess Mesh:**
   a. Calculate 3D area of each triangular cell
   b. Convert vertex coordinates to geographic (lat/lon)
   c. Compute triangle centroids (barycenters)
2. **Hexagon Mapping:**
   Assign each triangle to an H3 hexagon based on its centroid
3. **Weighted Exposure Calculation:**
   a. Join mesh data with solar coefficients
   b. Compute area-weighted solar exposure per triangle
   c. Aggregate to hexagon-level monthly averages:

$$\text{Hexagon Exposure} = \frac{\sum(\text{Solar Coefficient} \times \text{Normalized Area})}{\sum \text{Normalized Areas}}$$

4. **Analysis:**
   a. For each building, identify its dominant hexagon (largest area contribution)
   a. Calculate solar exposure per hexagon (monthly & hourly averages)
   b. Calculate building density per hexagon
   c. Compute correlation: Building Density vs. Solar Exposure

Return:
- Hexagon exposure map
- Density-exposure correlation coefficient

Figure 7: The base algorithm for the aggregation of Solar Exposure values for UB HPDA.

The current version of the HPDA workload has been tested against the modestly sized dataset already described in 3.2.2.1 – which it can execute within 1-2 minutes. Expanding to city-scale datasets is part of the future roadmap.

3.2.2.3 Future Development Roadmap

Future work will focus on expanding the range of use cases supported by this workflow. Efforts will also be directed toward integration of the algorithm into broader workflows within the pilot and across the HIDALGO2 platform, as well as performing benchmarking against significantly larger datasets. In parallel, optimization of data processing and aggregation steps will be pursued to improve computational efficiency further and reduce resource requirements for city-scale applications.

3.2.3 AI application

The AI use case addresses the problem of understanding how new constructions affect solar exposure within urban environments by formulating it as a supervised learning task. The city is represented as a graph: nodes correspond to buildings, and edges capture the shading relationship, i.e., whether one building affects the solar intake of another. The goal of the model is to predict which edges are activated when a new building is introduced, effectively quantifying how solar exposure evolves in response to urban development.

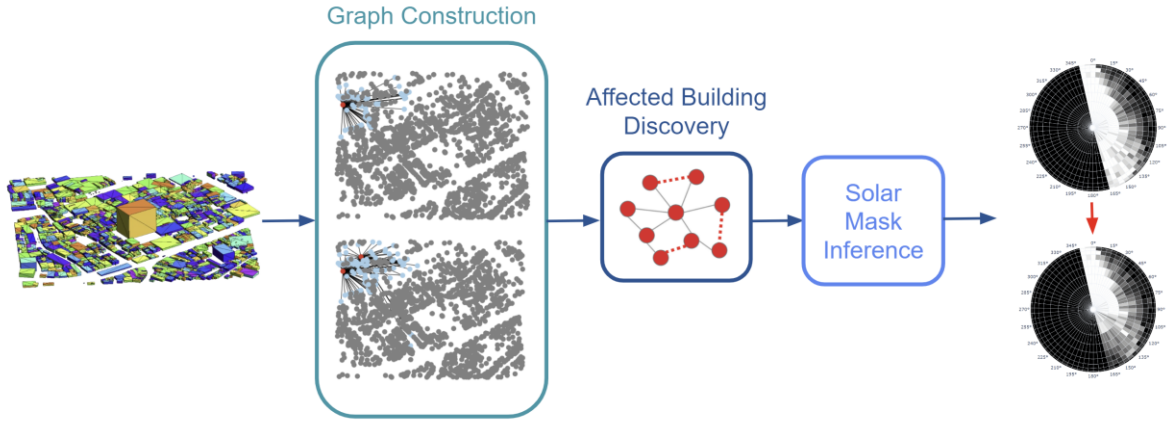

Figure 8: Pipeline of the AI framework for affected building prediction. Solar exposure relations are captured using a graph. A neural network is trained using the graph for Affected Building Discovery to facilitate Solar Mask Inference only on newly constructed buildings.

The pipeline developed for this exact problem is presented in Figure 8. The custom training dataset described in 3.2.1 enables the construction of an "*Affected Buildings Network*", where edges indicate that removing one building significantly alters the solar exposure of another. Once the graph is constructed, it becomes the basis for training a Graph Neural Network (GNN) to perform link prediction, as displayed in Figure 9. This pipeline and details on training were also provided in deliverable D4.3. Briefly, while training, a portion of edges is deliberately removed, while the remaining structure

is fed to the model for training. The GNN, specifically a two-layer Graph Convolutional Network (GCN) [15], learns node embeddings through message passing, and a decoder then predicts the likelihood of missing edges. Different decoding strategies have been tested, including a simple dot product classifier and a more expressive multi-layer perceptron (MLP). Additional experimental variations include treating the graph as directed or undirected, exploring different node features such as building height or location, and fine-tuning the solar mask threshold to balance accuracy with realism.

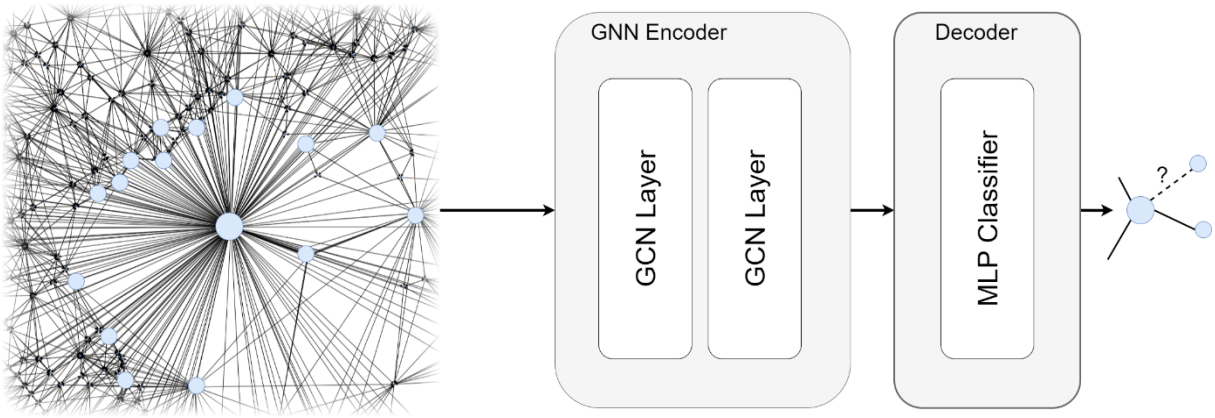

Figure 9: Architecture of the Graph Neural Network trained for Link Prediction (Affected Build-ing Discovery). During inference the Decoder is used to determine whether an edge should exist between two specific nodes (buildings).

Specifically, in the latest version of the dataset, the information provided differs from earlier releases. While previous datasets explicitly flagged affected buildings, the most recent version delivers a complete solar landscape. This shift makes it necessary to introduce a threshold to determine when a change in solar exposure is significant enough to justify the creation of an edge. Solar exposure is aggregated across all available surface-level coefficients of a building, covering the entire time span. Using the simplest aggregation method (averaging) results in a relatively narrow distribution of affectedness scores. As a result, defining an appropriate threshold for these values has become a central aspect of the ongoing research. Figure 10 further highlights that this threshold must be set with care. Because affectedness scores tend to be very close to one another, a strict threshold is required. Earlier experiments demonstrated that while dense graphs can yield high Area Under the Curve (AUC) values, these results can be misleading. In contrast, pruned and carefully curated graphs provide more reliable and interpretable outcomes.

In the previous deliverable, initial experiments highlighted several important trends. Building location proved to be the most informative node feature, and undirected graph structures consistently yielded stronger performance. While cases where the threshold is violated produced slightly lower AUC scores, the predicted affected buildings aligned more closely with spatial proximity, suggesting that the approach captures meaningful real-world relationships. Training loss analysis presented in D4.3 further confirmed that the models successfully fit the data, converging toward near-zero loss. Overall, these

findings supported the hypothesis that incorporating building location and adopting undirected graphs improves performance in transductive link prediction tasks on affected building networks.

In this deliverable, we build on that foundation by testing our established setup on the enhanced dataset provided by the pilot. The focus is on adapting our implementation to the new data format, requiring minor adjustments in code and function design, while preserving the core methodology. So far, integration has been successful, particularly in adapting the algorithm to a custom dataset where approximately one seventh of the buildings are removed and used to train the GNN. However, reporting results from this subset would not be sufficiently informative, as the reduced dataset leads to a significant loss of information. The next step is therefore to carry out GNN training on

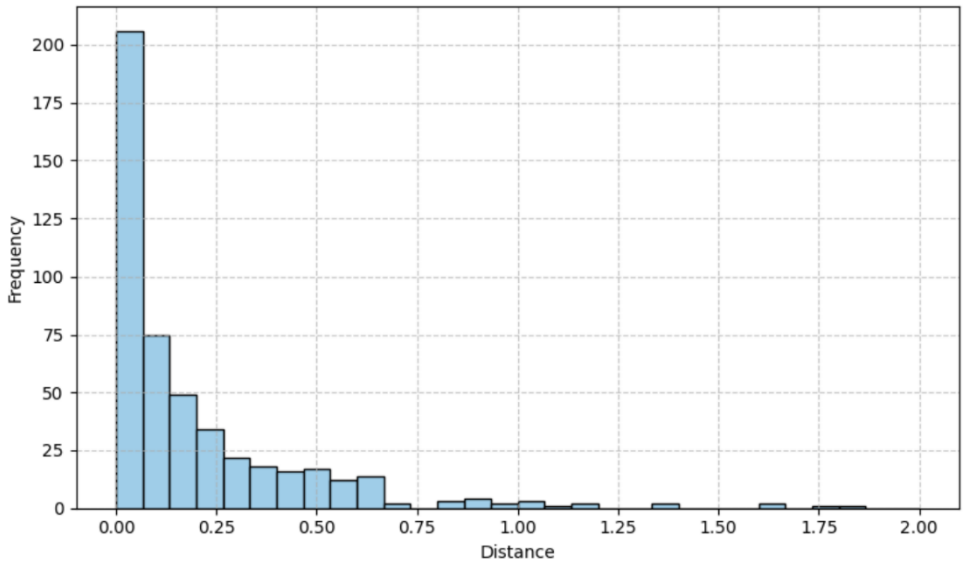

Figure 10: Histogram of Euclidean distances of averaged building solar masks.

the full custom dataset, in which each of the ~650 buildings is removed individually in turn.

Once optimized and tuned, the resulting GNN can be evaluated directly using standard binary classification metrics, without the need for human intervention, ensuring a straightforward and reproducible assessment process. At the same time, presenting results to engineers and urban planners – the ultimate users of the tool – remains essential. Through explanation techniques such as those outlined in Deliverable D4.3 [16], and thanks to the inherent transparency of graph-based representations, experts will be able to verify that the predicted solar shading relations are both valid and practically useful, thereby reinforcing trust in the system while enhancing their decision-making workflows.

Looking ahead, our work could extend beyond building-only interactions to incorporate additional environmental factors such as vegetation and microclimatic variables. These elements, while not yet included, are expected to play an important role in capturing the full complexity of solar exposure in evolving urban landscapes. The use of the graph structure. Furthermore, since node embeddings capture similarities between

buildings, the approach can be extended to inductive scenarios, enabling predictions for new buildings not present in the original dataset.

3.2.4 Summary

Since Deliverable D4.3, the Urban Buildings pilot has matured its HPDA and AI workflows from prototype implementations to integrated, production-ready pipelines.

On the HPDA side, the solar-exposure aggregation algorithm has been fully implemented on Apache Spark and applied to the full Strasbourg dataset. Whereas D4.3 described the hex-grid methodology in concept, the current deliverable demonstrates end-to-end execution – including centroid mapping, area-weighted coefficient aggregation, and monthly averaging – and production of city-wide hex maps of solar exposure and building density. Initial benchmarks in D4.3 were limited to small test cases; these have now been replaced by full-scale experiments under varying resource configurations, confirming that Spark's distributed framework handles 10.9 GB ORC datasets, which can be executed in the compute cluster described in Chapter 2 in 1-2 minutes.

Concerning AI, earlier work in D4.3 outlined a Graph Neural Network (GNN) approach using synthetic "affectedness" networks derived from partial building removals. Today, this pipeline has been adapted to the enhanced full-scale dataset of ~650 buildings. Code modifications accommodate the shift from explicitly flagged edges to continuous solar-mask distributions, necessitating calibrated threshold selection to construct reliable graphs. Integration tests on one-seventh subsets succeeded, and the groundwork is laid to train the two-layer Graph Convolutional Network encoder and decoder on the complete dataset. Preliminary findings from D4.3 – demonstrating superior performance for undirected graphs and geographic features – have been reaffirmed, and the current focus is on scaling up training, tuning thresholds for interpretability, and preparing quantitative evaluations via standard binary-classification metrics.

Together, these developments mark a significant step beyond the conceptual and small-scale experiments of D4.3, delivering robust, scalable analyses and predictive models that directly support urban planning and policy decision-making.

3.3 Renewable Energy Sources (RES)

3.3.1 Scope, Objectives & Inputs

The Renewable Energy Sources (RES) pilot aims to empower owners, operators, and developers of renewable infrastructure with accurate, timely forecasts of energy production. Achieving this requires two intertwined components: high-resolution weather forecasts and robust data-driven models that correlate meteorological conditions with power output. Local-scale weather predictions – delivered at horizontal

resolutions down to a few meters – bring inherent uncertainties from model physics, boundary conditions, and input data quality. These uncertainties are managed through High-Performance Data Analytics (HPDA), while Artificial Intelligence (AI) methods uncover the complex relationships between weather variables and photovoltaic (PV) energy generation.

3.3.2 HPDA application

Weather forecasts inherently carry uncertainties stemming from boundary conditions, input data quality, and the numerical models themselves. Mitigating these uncertainties relies on two complementary strategies: statistical uncertainty quantification – performed with the mUQSA toolkit [17] and detailed in Deliverables D4.9 [18] and D4.10 – and ensemble analysis, which is the focus of the HPDA workflow.

The forecast model can be configured with different preconditioner setups, allowing to more accurate forecast of certain weather parameters. The initial data to start a forecast may come from observational data, different global/mesoscale weather prediction models, terrain and building data from different providers, resulting in different outputs of the RES. To mitigate the differences in results, HPDA processing analyses weather parameters across all ensemble members throughout the time series, computing weighted averages that serve as inputs to the AI module. This analysis also generates monthly climatological averages supporting climate change impact studies on renewable energy infrastructure.

In the RES pilot, HPDA is applied to the RES.PV application, which forecasts energy production from photovoltaic systems. PSNC operates a 1 MWp PV farm (Figure 11) for which daily forecasts are generated. Each ensemble simulation produces weather prediction outputs stored in NetCDF format [19] – approximately 100 MB for one hour of prediction over a 250×250 grid – that includes wind speed and direction, humidity, pressure, solar irradiance components, and other relevant parameters required by the

Figure 11: 1MWp photovoltaic farm owned by PSNC

RES.energy AI module. Four 12-hour forecasts are issued each day, yielding 120 ensemble simulations (~1.2 GB each) to be processed for monthly averages.

Currently, the HPDA pipeline is implemented as a Python script using NumPy arrays, with plans in place to port the solution to the HPDA software stack presented in Chapter 2. In its current version, the pipeline ingests each NetCDF file, applies parameter-specific weights reflecting historical ensemble performance, and computes weighted

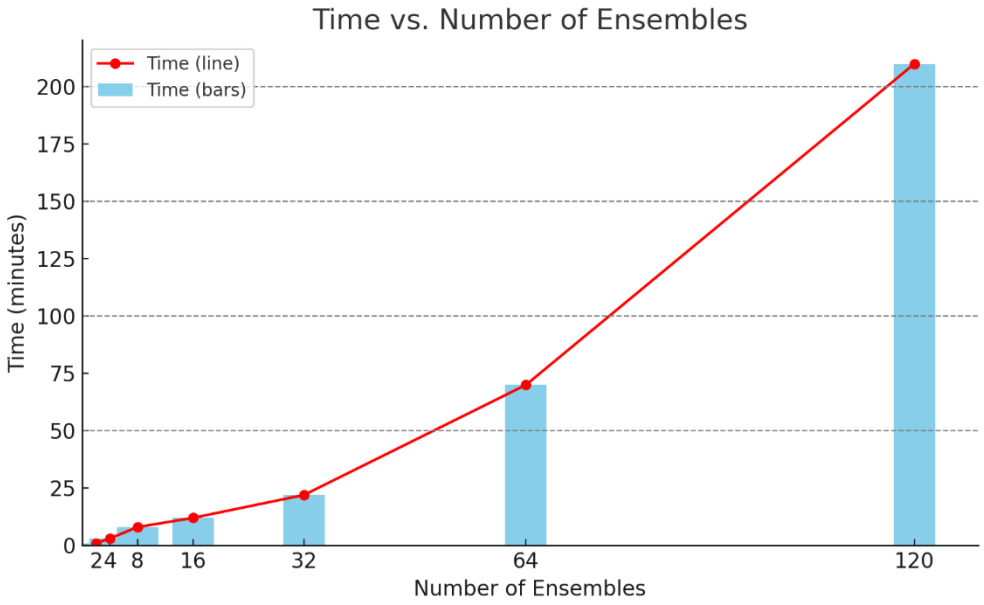

Figure 12: Processing time for HPDA ensemble analysis as a function of the number of ensemble members.

averages over all members and time steps. This flexible approach ensures that ensemble members demonstrating superior accuracy for particular variables contribute more heavily to the aggregate forecast. Although processing times for tens of ensembles remain acceptable, scaling to hundreds will strain this ad-hoc method (see Figure 12).

3.3.3 AI application

The **RES.energy** module establishes a data-driven correlation between forecasted weather parameters and the electrical output of photovoltaic (PV) installations. Central to this effort is PSNC's own 1 MWp PV farm, which provides high-fidelity generation records at 15-minute intervals over a full year, enabling robust model training. Feature importance analysis consistently identifies direct solar radiation as the dominant predictor, followed by visibility and ambient temperature, among other variables.

The AI model, whose architecture is illustrated in Figure 13, is implemented in PyTorch and trained on a single NVIDIA A100 GPU. Training on one year's worth of paired weather–generation data requires approximately five minutes, with runtime scaling linearly as additional data are incorporated. Although the current configuration is optimized for PSNC's farm, the methodology is fully adaptable to other PV site data –

requiring a fresh train cycle; retraining on site-specific data and minor architectural adjustments ensure portability across farms of varying capacities.

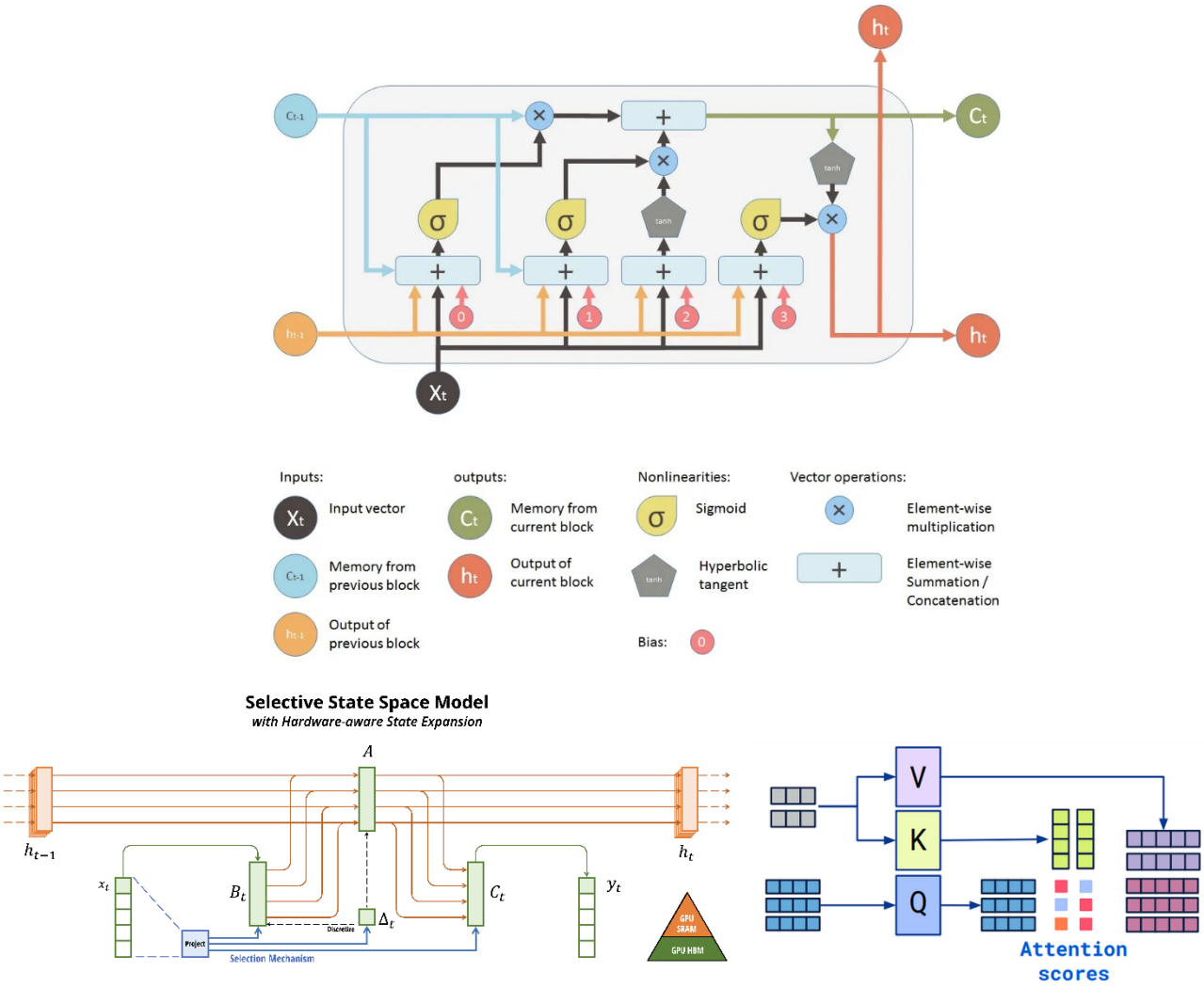

Figure 13: Proposed novel neural architectures for RES.energy, including the Mamba selective state-space model (left), which uses hierarchical state representations and selection mechanisms, and a transformer-based model with attention layers (right) for capturing long-range dependencies in weather–power data.

The neural network architecture (illustrated in Figure 13, simplified to three parameters for clarity) has been refined through systematic evaluation of multiple approaches:

- **Multi-Layer Perceptron (MLP)**: used to predict future output, treating each timestep independently, thus disregarding temporal dependencies.
- **LSTM Recurrent Neural Network (Historical)**: incorporating temporal dependencies by leveraging historical power generation values to predict future output.

- **LSTM Recurrent Neural Network (Historical & Future Weather)**: combining historical power generation data with forecasted meteorological parameters to predict the future power output.
- **State Space Model (SSM)**: attempting to capture deeper feature dependencies and temporal dynamics using more sophisticated architectures.

The third approach – **LSTM with both historical generation and future weather forecasting** – achieves superior performance, as expected from information-theoretic principles. The first approach sacrifices temporal information, significantly limiting predictive capability. The second approach captures temporal dynamics in generation patterns but lacks forward-looking weather information. Incorporating future meteorological forecasts substantially improves performance metrics (RMSE, R^2). The SSM architecture, while theoretically capable of capturing more complex dependencies, underperforms due to insufficient training data volume required for its higher parameter count, in practice.

From an implementation perspective, utilizing exogenous future variables (forecasted weather) while predicting extended time horizons requires roll-out procedures where predicted values at each timestep inform subsequent predictions. The training procedure processes yearly data partitioned into 96-timestep chunks (24 hours at 15-minute resolution), predicting energy production at each subsequent timestep. Training minimizes mean squared error loss using the ADAM optimizer [20] with ReduceLROnPlateau scheduling [21] over 60 epochs with an initial learning rate of 1×10^{-3}. The model architecture contains fewer than $50{,}000$ parameters, with inputs normalized using scikit-learn's PowerTransformer. Hyperparameters were systematically optimized using Hydra's parameter sweep API.

The initial RES.energy release utilizes openly available forecast data from OpenMeteo. Figure 14 demonstrates prediction accuracy for PSNC's PV farm: the blue line represents ground truth measurements, the red line shows model predictions, and the green line indicates direct solar radiation – a feature exhibiting strong correlation (0.9) with PV generation. Predictions closely track actual values while avoiding over-reliance on any single highly correlated feature, confirming that the model captures complex multi-parameter dependencies rather than simple univariate relationships.

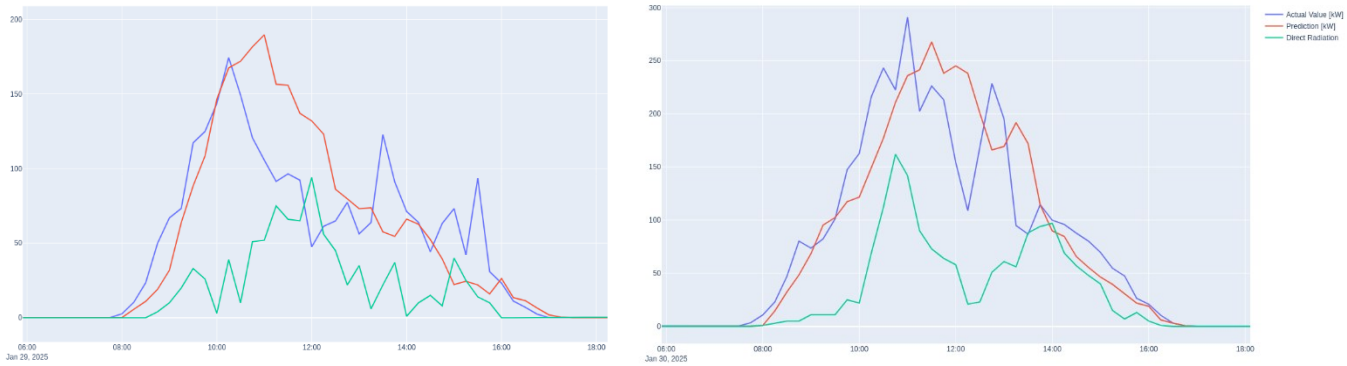

Figure 14: Accuracy of RES.energy 1st release – results for 29/01/2025 (left) and 30/01/2025(right).

3.3.4 Summary

Compared to the conceptual framework and initial prototypes presented in Deliverable D4.3, the RES pilot's HPDA and AI workflows have been significantly advanced and operationalized.

On the HPDA side, D4.3 introduced the idea of ensemble-based uncertainty mitigation and mentioned prototype processing tools. In this version, the HPDA pipeline has been fully implemented for the RES.PV application, ingesting 1.2 GB of daily ensemble NetCDF outputs from PSNC's 1 MWp farm and computing weighted monthly climatologies. The scripted NumPy-based workflow now applies parameter-specific weights, consolidates 120 daily ensemble members, and produces reliable aggregate statistics for both AI training and climate-impact analyses. This moves the methodology from concept to robust operational execution, with forthcoming migration to HDFS and Apache Spark for scalable production runs.

In the AI track, D4.3 outlined a machine-learning approach using open forecasts and preliminary model architectures. The current deliverable reports that RES.energy has been trained on one year of 15-minute PV generation data, completed in five minutes on an NVIDIA A100 GPU, and validated against OpenMeteo forecasts. Early validation demonstrates the model's ability to capture complex weather–power relationships beyond a single-variable dependence.

These developments mark a clear transition from D4.3's exploratory prototypes to realized, production-capable HPDA and AI solutions that deliver accurate, and moderately scalable energy production forecasts for renewable energy stakeholders.

3.4 Wildfires (WF)

3.4.1 Scope, Objectives & Inputs

The Wildfire pilot investigates how environmental and meteorological variables influence wildfire behaviour across spatial and temporal dimensions, addressing this challenge through complementary HPDA and AI methodologies. The **HPDA workflow** analyses variable importance in determining burn probability (BP) – the likelihood of fire occurrence – and examines their role in governing the progression and rate of fire spread through time. Meanwhile, the **AI application** pursues two parallel objectives: first, identifying the most relevant pre-computed simulations when new fire events unfold through learned similarity representations; and second, enriching the feature space through computer vision descriptors that capture visual aspects of fire perimeter evolution. Together, these approaches support wildfire risk assessment and

management by identifying which variables most strongly contribute to fire occurrence and propagation dynamics, while providing operational tools for scenario-based planning and response coordination.

The HPDA workflow processes high-resolution simulation rasters for the Rectoret study region, covering a 9 km² area discretised at 2-meter spatial resolution in the ETRS89/UTM31N coordinate system [22]. The dataset includes $10{,}584$ ASCII raster files totalling 133 GB of storage, each tagged with metadata describing ignition point coordinates and meteorological boundary conditions such as WS and WD. To contextualize fire spread within the broader environment, the analysis integrates terrain characteristics (aspect, slope, and elevation) and vegetation indicators, including biomass density indices ($\mathrm{bio3x3n}$ and $\mathrm{bio5x5n}$), and multi-scale continuity metrics ($\mathrm{cont6m}$, $\mathrm{cont10m}$, and $\mathrm{CONT\_NORM\_2\_20}$) that quantify fuel bed connectivity.

The HPDA workflow produces three primary output categories that support strategic planning across different operational contexts. First, spatially explicit burn probability maps quantify fire occurrence likelihood at each location within the study domain. Second, feature importance metrics derived from both classical machine learning models (e.g., random forests, gradient boosting) and deep learning architectures reveal which environmental variables exert the strongest influence on fire behaviour. Third, model interpretation techniques including saliency mapping, occlusion analysis, and perturbation-based sensitivity assessments provide spatial visualizations of where and how input features drive model predictions, enhancing the interpretability of black-box neural network architectures.

The AI application operates on a distinct but complementary dataset comprising $10{,}584$ wildfire simulations generated for the municipality of Barcelona. Each three-hour simulation, discretised at one-minute intervals, undergoes temporal feature extraction to produce sequences of handcrafted descriptors including fire perimeter area, bounding-box dimensions, orientation angle, eccentricity, moment of inertia, and centre of gravity. These temporal sequences serve as inputs to LSTM-based autoencoder architectures that learn compact latent representations capable of capturing essential fire evolution dynamics. In parallel, a computer vision feature enhancement pipeline processes snapshot imagery of fire perimeters to extract classical visual descriptors including Canny edge detection, Harris corner identification, SIFT (Scale-Invariant Feature Transform) keypoints and descriptors, SURF (Speeded-Up Robust Features), ORB (Oriented FAST and Rotated BRIEF), and HOG (Histogram of Oriented Gradients) features. These CV-derived features are integrated into the training dataset to enrich the representation and potentially capture shape and texture evolution aspects not fully expressed in handcrafted geometric descriptors. The challenge lies in developing similarity measures that operate in an unsupervised setting – since no labelled pairs of similar fires exist – while remaining robust to temporal shifts and enabling meaningful comparisons across fires of varying duration and complexity.

The AI application delivers temporal similarity matching through learned latent space representations, enabling retrieval of historically similar fire events when new scenarios emerge. The autoencoder architecture compresses each fire's temporal evolution into fixed-length embeddings that preserve essential spread dynamics while enabling efficient distance-based comparisons. To ensure practical utility, an expert-driven evaluation protocol has been established in collaboration with domain specialists, who assess whether retrieved simulations genuinely represent acceptable matches for source fires. This ground-truth validation framework guides model selection and refinement toward operationally relevant similarity measures. Future integration of the CV-enhanced features is expected to improve retrieval robustness by incorporating visual characteristics complementary to geometric feature sets.

3.4.2 HPDA application

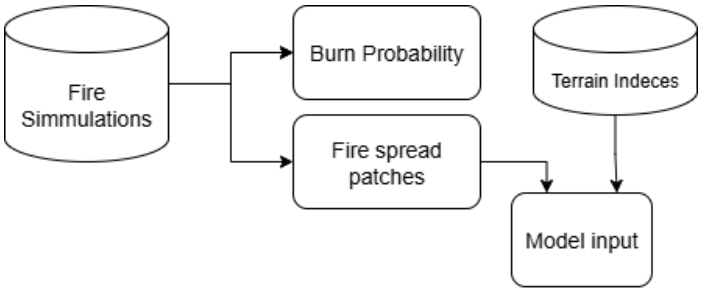

Figure 15: Workflow showing how fire simulations produce BP and spread patches, which are then combined with terrain and vegetation indices to create model inputs.

The HPDA workflow (Figure 15) applied in the Wildfire pilot aims to quantify how environmental and meteorological variables affect wildfire behaviour, focusing specifically on BP and fire spread dynamics. The analysis combines the ability to capture non-linear spatial interactions with interpretable, feature-level insights.

All computations are performed on a JupyterHub environment running on Linux, with the following specifications:

- Hardware: 2 nodes $\times$ 16 cores (32 cores total), 94 GB RAM (~81 GB available).
- Software: Python 3.12.8 stack including PyTorch (2.3.0), NumPy (1.26.4), SciPy (1.13.0), GeoPandas (0.14.4).
- Data size: ~133 GB total size of simulation rasters.

3.4.2.1 Burn Probability Analysis

The main focus in this section is the Burn Probability (BP). It is calculated for each point by dividing the number of times a fire occurred there by the number of simulations and multiplying it by 100. Figure 16 shows the spatial distribution of BP normalized between 0 and 100.

$$BP = 100 \cdot \frac{NF}{NS}$$

Where:

BP - Burn Probability, expressed in %

NF - Number of times fire passes through a point in the territory

NS - Total number of ensemble simulations

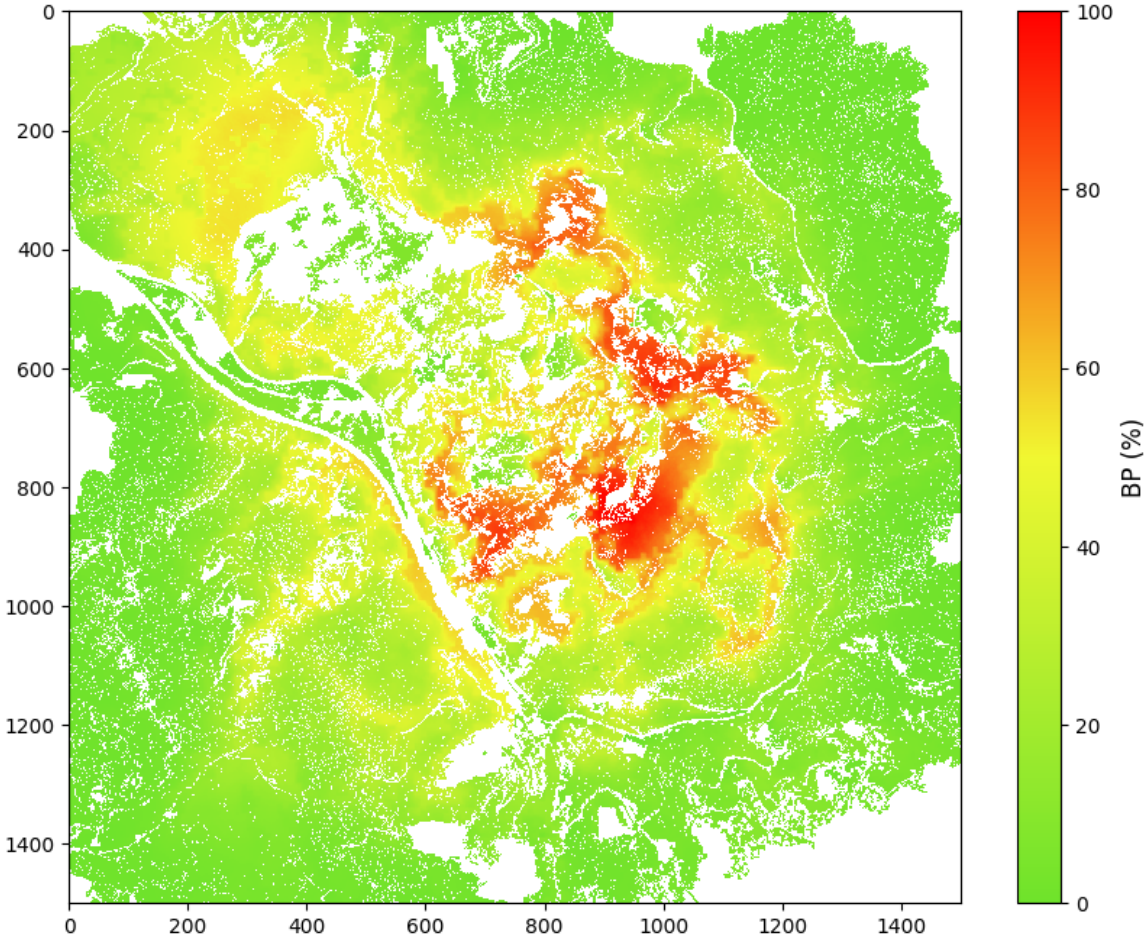

Figure 16: Plot showing spatial distribution of normalized BP, where green means low probability and red high probability.

Spatial hotspot analysis identifies areas of consistently high BP, concentrated primarily in regions of higher elevation. This suggests a localized interaction between topography and fuel continuity that enhances fire probability in specific zones.

It is further confirmed by the correlation analyses that were performed using both Pearson correlation (Figure 17) and Global Moran's I (Figure 18) to evaluate linear and spatial autocorrelation relationships between BP and terrain or vegetation variables.

Results indicate that vegetation continuity indices ($\mathrm{cont10m}$, $\mathrm{CONT\_NORM\_2\_20}$) and $\mathrm{bio5x5n}$ are strongly associated with BP (Figure 18), whereas slope and aspect provide moderate refinement. Forest fuel type shows minimal direct influence on BP.

The analysis was also conducted using XGBoost [20] to quantify how individual environmental features influence BP at the cell level. Figure 19 shows the XGBoost-derived feature importance heatmap, highlighting the relative contributions of each input variable to BP predictions. These results indicate that vegetation continuity indices are the most influential, with $\mathrm{CONT\_NORM\_2\_20}$ (0.26) and $\mathrm{cont10m}$ (0.14) being dominant contributors. Aspect (0.21) and slope (0.15) provide secondary refinement, $\mathrm{bio5x5n}$ has a moderate contribution (0.12), while forest fuel and $\mathrm{bio3x3n}$ have minimal influence. This confirms that vegetation structure, particularly continuity, is the primary driver of BP.

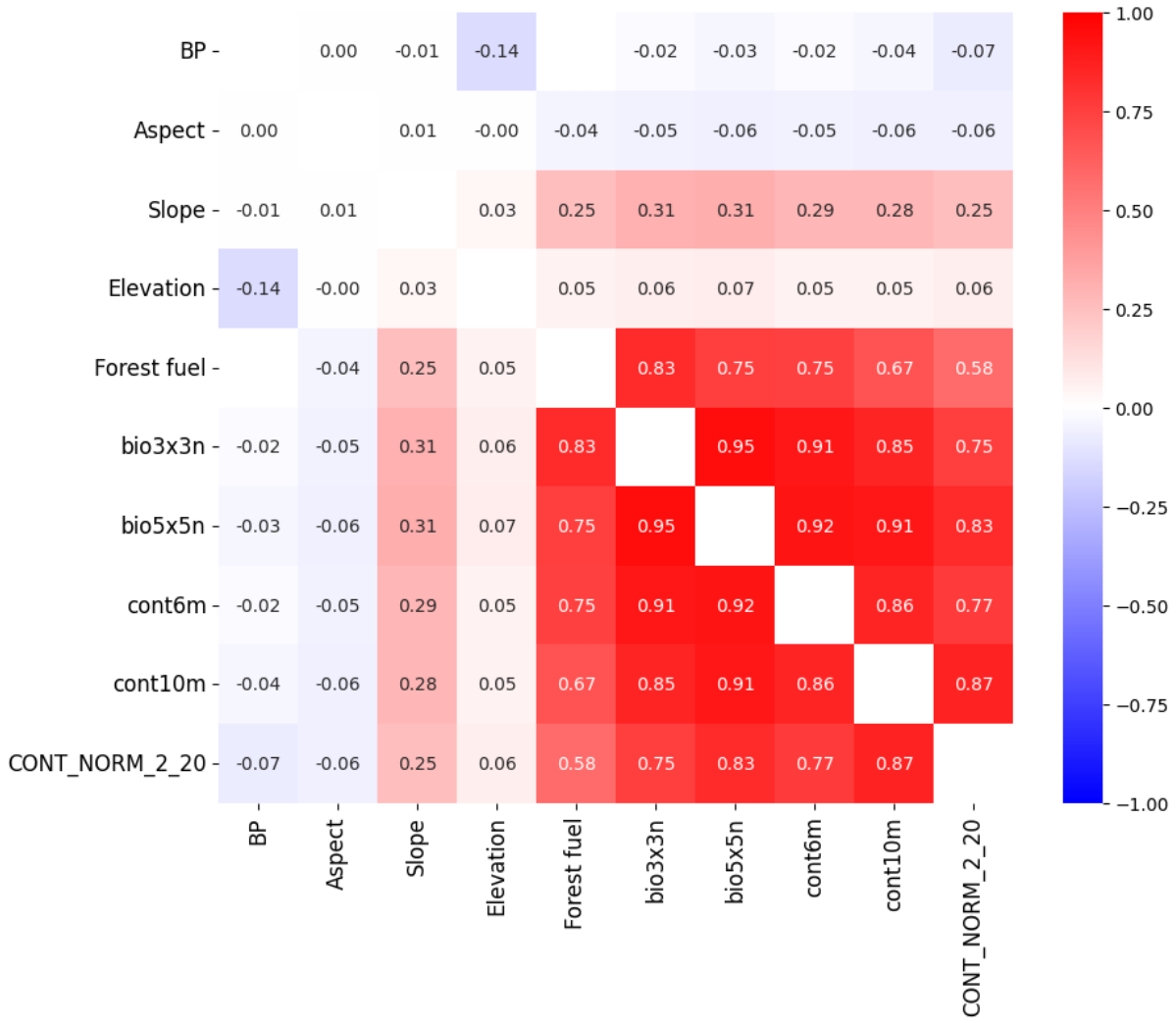

Figure 17: Feature correlation matrix showing a heatmap visualization of pairwise correlations between the features in the dataset. The values range from -1 (blue - strong negative correlation) to +1 (red - strong positive correlation), with 0 indicating no correlation.

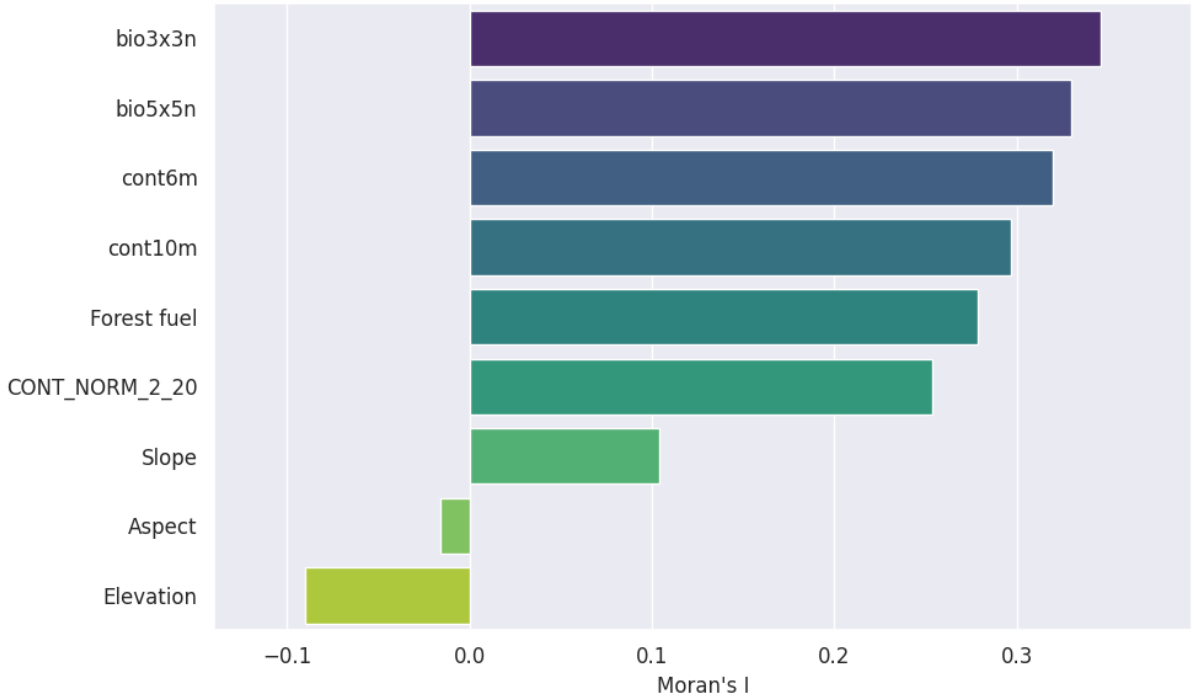

Figure 18: Spatial autocorrelation of selected features and BP, measured using Moran's I. Positive values indicate clustering of similar values, while negative values indicate dispersion. Features are sorted from highest to lowest Moran's I.

The analysis was also conducted using XGBoost to quantify how individual environmental features influence BP at the cell level. Figure 19 shows the XGBoost-derived feature importance heatmap, highlighting the relative contributions of each input variable to BP predictions. These results indicate that vegetation continuity indices are the most influential, with CONT_NORM_2_20 (0.26) and cont10m (0.14)

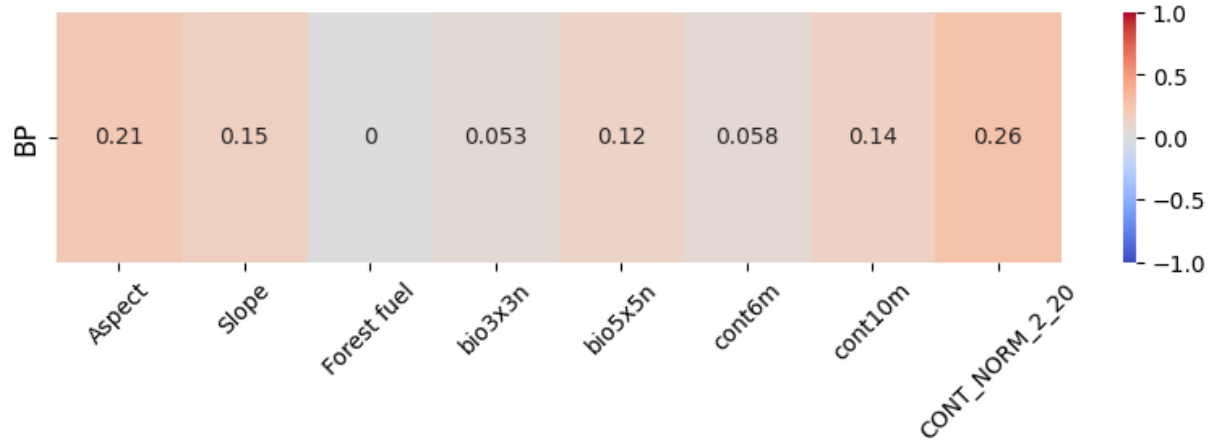

Figure 19: Feature importance scores obtained from the XGBoost model. The heatmap shows the relative contribution of each feature to the model's prediction of BP, with warmer colours indicating higher positive importance and cooler colours indicating negative or lower importance.

being dominant contributors. Aspect (0.21) and slope (0.15) provide secondary refinement, bio5x5n has a moderate contribution (0.12), while forest fuel and bio3x3n

have minimal influence. This confirms that vegetation structure, particularly continuity, is the primary driver of BP.

3.4.2.2 Deep Learning Analysis

To capture non-linear interactions between environmental variables and fire spread, a patch-based CNN was implemented. The model processes 128×128 raster patches incorporating terrain (aspect, slope), vegetation indices ($\mathrm{bio5x5n}$, $\mathrm{cont10m}$, $\mathrm{CONT\_NORM\_2\_20}$), wind conditions (WS, WD), and the fire stage at each time step.

For the feature importance assessment three complementary methods were applied:

- Perturbation Analysis: Evaluates changes in model predictions when parts of features are changed. This method identifies features critical to local prediction stability (Figure 21, Figure 22).
- Occlusion Analysis: Assesses global model reliance by masking input features and measuring prediction changes (Figure 20).
- Saliency Mapping: Computes gradient-based sensitivity of predictions with respect to each input feature (Figure 20).

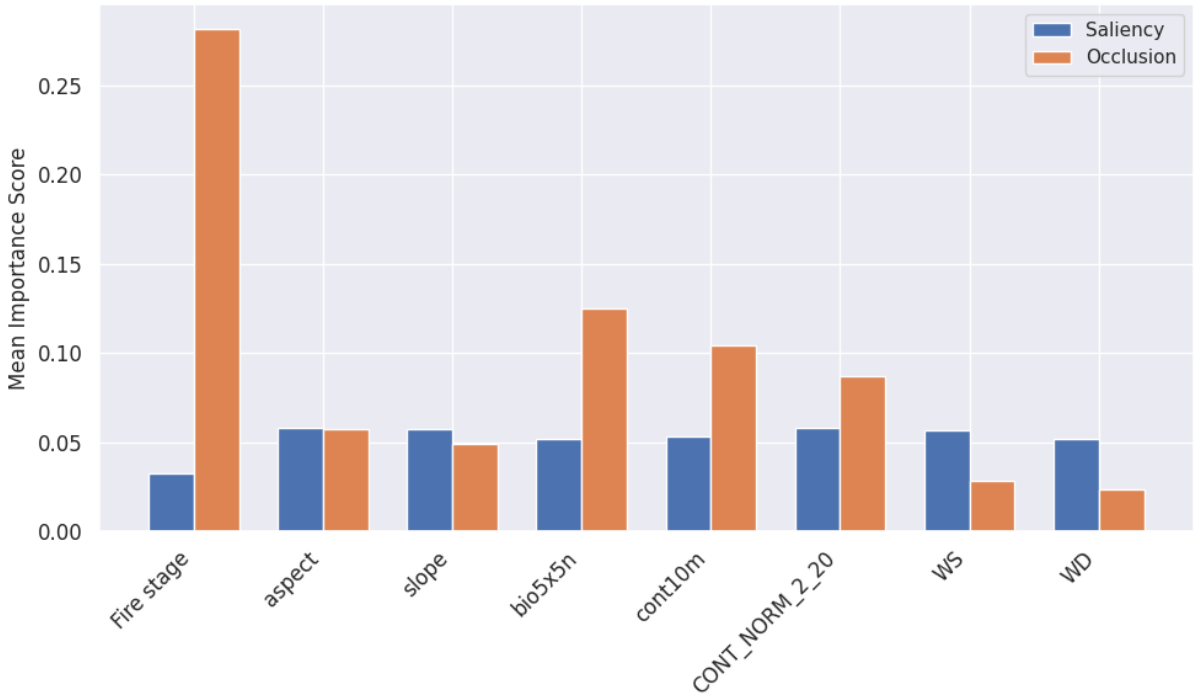

Figure 20: Comparison of feature importance across layers using two attribution methods: saliency and occlusion. Bars represent the mean importance score for each feature, with saliency scores shown as blue and occlusion scores as orange. Feature names are indicated on the x-axis.

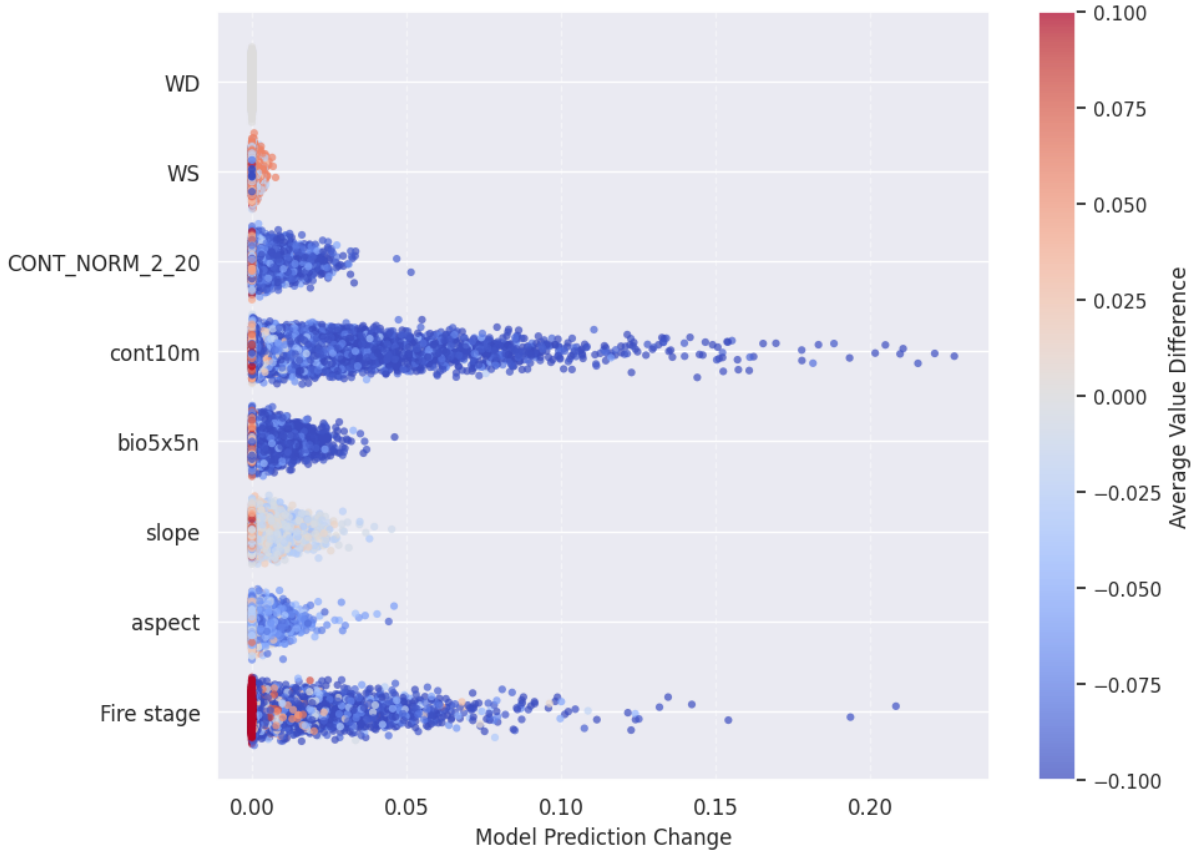

Figure 21: Feature-wise distribution of global prediction changes. Each point represents one analysed change in prediction accuracy associated with a feature, with colour indicating the magnitude of the feature value difference. The large number of points reflects multiple perturbations per feature, illustrating how variations in each feature influence the model's overall predictions.

Perturbation analysis identifies cont10m, bio5x5n, and fire stage as most locally influential, while aspect, slope, and wind have minimal effect. Occlusion shows fire stage dominates global predictions, with moderate contributions from bio5x5n and vegetation continuity indices. Saliency distributes importance more evenly, emphasizing terrain and continuity indices. Overall, vegetation continuity drives local sensitivity, fire stage governs global dynamics, bio5x5n provides moderate influence.

3.4.2.3 Analysis of the results

The analysis shows that forest fire forecasts in the Rectoret region are primarily based on vegetation continuity and fire stage metrics, with secondary refinement derived from bio5x5n, exposure, and slope. Classical ML highlights local sensitivities (cont10m and CONT_NORM_2_20), while deep learning reflects broader spatial and temporal dynamics, confirming the value of combining both approaches.

The minimal impact of wind and fuel type suggests that higher-resolution wind data and more detailed vegetation descriptors may be needed to improve forecasts. From

an application perspective, identifying vegetation continuity as the dominant driver provides clear guidance for risk assessment, fuel management, and targeted mitigation strategies.

Overall, this pilot confirms the crucial role of vegetation structure in forest fire risk modelling and demonstrates how HPDA and AI workflows can deliver interpretable, scalable insights for fire management across regions.

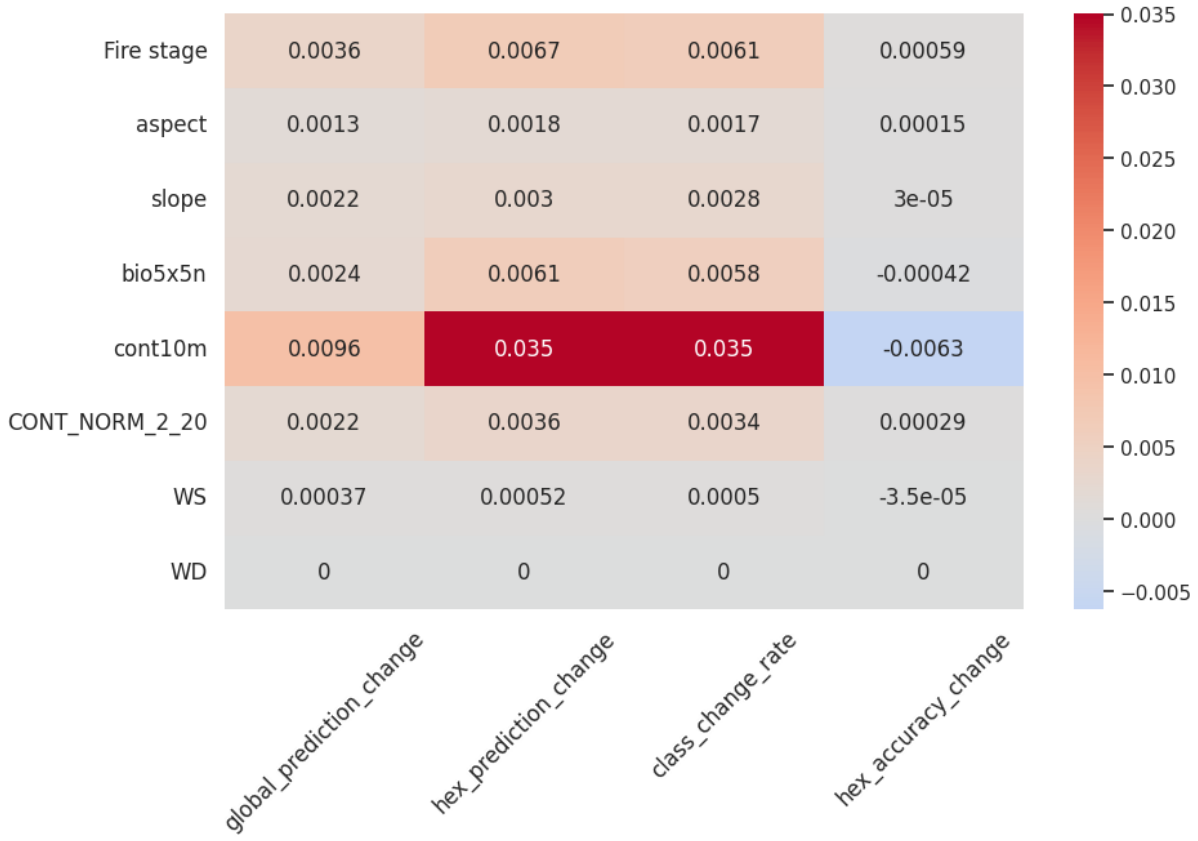

Figure 22: Mean impact of each feature across global and hex-level prediction accuracy changes, predicted class change rate, and hex-level accuracy change. Higher values indicate features that have a stronger effect on model predictions, classification outcomes, or local prediction accuracy.

3.4.2.4 Future Development Roadmap

The analysis will be extended to additional regions, including La Pedriza de Manzanares and Moralzarzal, to assess the generality of the results. Including additional environmental variables, such as soil moisture and smoke dispersion factors, could further improve prediction accuracy. Future research will also explore integrating higher-resolution meteorological data, particularly local wind fields, to better understand their role in fire dynamics. Furthermore, efforts will focus on improving model generalization through transfer learning methods and assessing operational implementation in near-real-time forest fire monitoring systems.

3.4.3 AI application[4]

3.4.3.1 LSTM Autoencoder for Wildfire Retrieval

Problem Formulation and Background. In the context of the Wildfires pilot, the AI use case is concerned with the problem of identifying the most relevant pre-computed simulations when a new fire event unfolds. To this end, we rely on a dataset comprising $10{,}584$ wildfire simulations generated for the Rectoret area in Barcelona. In the previous deliverable (D4.3), we implemented a feature-extraction and exploratory analysis pipeline: at each timestep, the fire perimeter was delineated and a set of handcrafted descriptors was computed, including *area*, *perimeter length*, *bounding-box length* and width, orientation angle, eccentricity, moment of inertia, and the *centre of gravity*. These descriptors were subsequently analysed (e.g., using PCA) to assess their discriminative capacity and their potential role as the foundation for more advanced models. In particular, it was emphasised that this representation could later support the development of trainable similarity measures, moving beyond purely handcrafted features. Nevertheless, several challenges remain. The absence of labelled pairs of similar fires requires the system to operate in an unsupervised setting. Furthermore, the similarity mechanism must be capable of capturing the fundamental drivers of fire evolution, recognising patterns even under temporal shifts, and enabling comparisons across fires of varying duration.

Autoencoder Architecture Design. To address the aforementioned challenges and to establish a trainable similarity mechanism capable of producing informative representations for retrieval, we designed an autoencoder-based architecture. The adopted solution is illustrated in Figure 23.

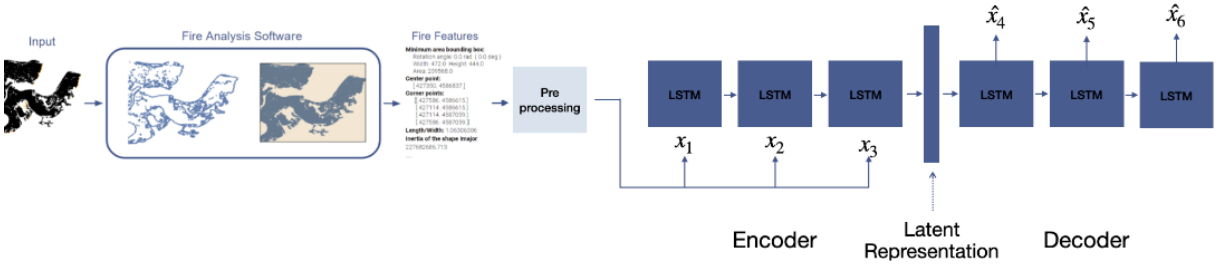

Figure 23: Architecture of the wildfire similarity model. A recurrent autoencoder is trained to encode the temporal evolution of wildfires into fixed-size latent vectors, which can subsequently be used for retrieval.

The architecture follows the classical autoencoder paradigm, adapted to sequential wildfire data. An **encoder**, implemented as a stack of LSTM layers, receives as input a sequence of features extracted at each timestep of a fire simulation. These features include descriptors such as area, perimeter, bounding-box width and height, eccentricity, orientation angle, moment of inertia, and the centre of gravity. The encoder compresses this temporal sequence into a fixed-length **latent vector**. This

[4] Code repository: https://git.hidalgo2.eu/hidalgo2-group/hid-ai-wf

latent vector is designed to capture the essential dynamics of the fire, regardless of the number of timesteps provided.

The **decoder**, also based on LSTM layers, takes this latent vector and attempts to reconstruct the original sequence of features. In doing so, the network is forced to store in the latent vector all the information necessary to describe the fire's evolution. The decoder is therefore not used at inference time but serves a critical role during training: it ensures that the encoder learns a meaningful and compact representation of each fire sequence. Once training is completed, the decoder is discarded, and the encoder alone is used to map any given fire into latent space, where similarity comparisons can be efficiently carried out.

Reconstruction-Based Training Paradigm. The reconstruction-based training paradigm compels the encoder to learn representations that are both compact and informative. By requiring the decoder to regenerate the entire temporal sequence of a wildfire from a single latent vector, the model enforces a strong information bottleneck. In practice, this means that the encoder must compress all relevant aspects of fire behaviour – such as growth dynamics, shape evolution, and directional trends – into a fixed-length representation. If important information is omitted, the decoder will fail to reproduce the sequence accurately, and the reconstruction error will remain high.

Through this process, the latent vector acquires the role of a distilled summary of the fire's evolution. It is not merely a reduced version of the original descriptors but a learned representation that reflects the underlying dynamics driving the spread. These representations are particularly valuable for retrieval: by embedding different wildfires into the same latent space, the model enables meaningful comparisons across events of varying length and complexity. Fires that evolve in similar ways are placed close together in this space, even if their trajectories differ in duration or temporal alignment, while dissimilar fires are naturally separated. In this way, the reconstruction objective ensures that the learned embeddings capture the essential structure needed for the similarity task.

Experimental Configuration. According to the experimental setup, each wildfire simulation spans a total duration of three hours, discretised at one-minute intervals, resulting in 180 timesteps. For every simulation, the first 150 timesteps were used as input for training the autoencoder, while the remaining 30 timesteps were held out for evaluation of the reconstruction quality. This consistent partitioning ensured that the model was always tested on a non-observed segment of each fire evolution. Beyond this baseline configuration, we also explored variations of the training process and, in particular, systematically examined different latent space dimensionalities in order to study the trade-off between compression and expressiveness of the learned representations.

Latent Dimensionality Analysis. Figure 24 shows the effect of latent space dimensionality on training loss. The smallest latent vectors, such as two dimensions, are unable to encode the complexity of wildfire dynamics, and as a result their

reconstruction error remains consistently high throughout training. At the opposite extreme, very large latent vectors (e.g., 256 dimensions) initially reduce the loss rapidly but quickly trigger early stopping, indicating overfitting and limited generalisation capacity. Between these extremes, intermediate latent sizes provide a more favourable balance: they converge smoothly, achieve lower reconstruction loss, and maintain stability across epochs. This behaviour suggests that moderate latent dimensionalities are better suited to capturing the essential dynamics of wildfire evolution without sacrificing generalisation, making them the most promising candidates for the similarity retrieval task.

Expert-Driven Evaluation Protocol. Reconstruction loss provides only an indirect indication of model performance and does not reflect the actual objective of the wildfire similarity task. The downstream application requires an evaluation of whether retrieved simulations are genuinely representative of a given source fire. Since no labelled pairs of similar fires exist, evaluation must instead rely on expert judgement.

In collaboration with the pilots, an evaluation protocol was established based on structured annotation. For each source simulation, a set of candidate outputs produced by the trained models is presented. Domain experts are then asked to indicate which of these candidates can be considered acceptable representations of the source. To facilitate this process, a dedicated annotation template was designed: the first column contains the ID of the source simulation, while the second column is filled with the indices of the acceptable candidates; if none are suitable, the entry remains empty.

This protocol enables the construction of a ground-truth signal tailored to the retrieval task, even in the absence of predefined labels. It also ensures that evaluation reflects

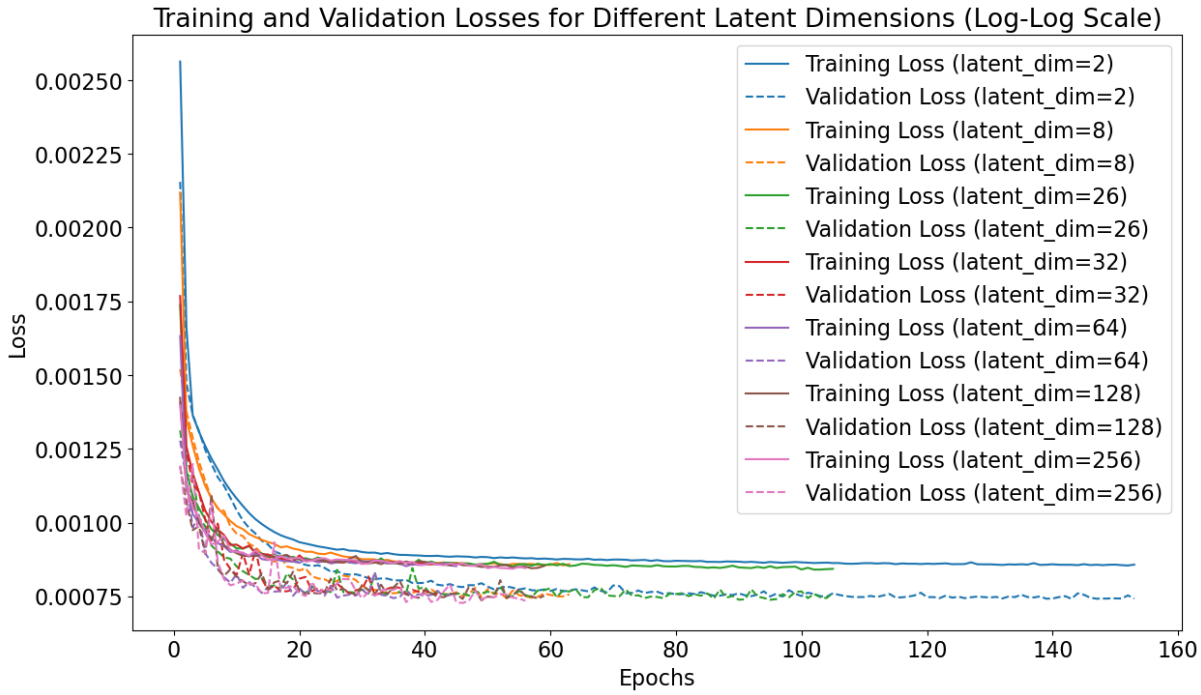

Figure 24: Training behaviour of the recurrent autoencoder architecture across different latent space dimensionalities.

both numerical fidelity and practical interpretability, since the assessment is guided by expert knowledge of wildfire behaviour. By systematically applying this framework across the dataset, the relative merits of different models can be compared, and the representation strategies that best support operational retrieval can be identified.

Future Development Roadmap. The overarching aim of the next phase is to determine the most effective similarity algorithm and make it available to the pilots through a simple interface. The primary driver for model selection will be the expert evaluation: the annotations collected from the pilots will serve as the reference against which competing models are compared. Using these annotations, we will quantify retrieval quality (e.g., agreement with expert choices, rank-based indicators) and identify the baseline configuration that consistently yields acceptable matches. This process also provides structured error signals (typical failure modes, sensitivity to sequence length, robustness to temporal shifts) that will guide subsequent design choices.

In parallel, a first version of an API will be provided to facilitate access to the retrieval service. The purpose of this interface is straightforward: allow pilots to submit queries and obtain candidate simulations without handling the training code. This enables rapid testing of the selected baseline and streamlines the feedback loop without prescribing any operational scenario.

Once the baseline has been established from the expert evaluation, the project will proceed with targeted experimentation informed by the results. The validated subset and the observed error patterns will be used to refine the modelling choices (latent dimensionality, regularisation, temporal pooling/attention, alternative sequence encoders, contrastive objectives for time-shift tolerance). The goal is to improve representation quality where the evaluation indicates clear headroom, rather than exploring unconstrained model space.

A further strand will focus on enriching the representation by embedding additional feature families into the model. In particular, computer-vision descriptors derived from perimeter imagery and related visual cues (see Section 3.4.3.1) will be incorporated to complement the existing handcrafted features. The expectation is that these inputs will capture aspects of shape and texture evolution that are not fully expressed in the current descriptors, thereby improving retrieval robustness.

Taken together, these steps – expert-driven selection, an accessible API for use and feedback, focused refinement based on evaluation outcomes, and integration of computer-vision features – define a clear path toward a working prototype that can be iteratively improved.

3.4.3.2 Computer Vision Feature Enhancement.

In a parallel development thread, Deep Learning models have been designed and trained to learn latent representations of the time snapshots of WF simulations, from a dataset of features that includes computed properties of the snapshot, such as the

minimum bounding box, its eccentricity, the centre of gravity, or the moment of inertia. To improve the performance of these models, the training dataset can be enriched with additional features resulting from applying a feature engineering process that computes image's feature descriptors common in Computer Vision (CV).

The inclusion of Computer Vision features enhances the Deep Learning models' ability to learn meaningful latent representations of WF simulation snapshots by providing rich, complementary structural and shape information beyond raw pixel values or standard computed properties. Techniques such as Canny edge detection [23] and Harris corner detection [24] extract boundaries and keypoints that help the model discern fire shape and spread patterns, while descriptors like SIFT [25] , SURF [26], ORB [27], and HOG [28] capture distinctive local texture, gradient, and object-level features present in the fire evolution imagery. Incorporating these engineered features into the training dataset enables the Deep Learning autoencoders to more effectively capture relevant spatial characteristics and variations in wildfire dynamics. This enriched representation supports improved performance in tasks such as similarity retrieval and prediction, as it allows the model to better discriminate between subtle differences in fire behaviour and morphology that may be crucial for accurate forecasting and scenario evaluation.

For each WF simulation's snapshot (see Figure 25a), we have computed the following CV features (see the other plots in Figure 25):

Canny Edge [23]: this multi-stage algorithm is used to detect the edge of the WF snapshot (see Figure 25b),

Harris Corner [24]: this algorithm detects image corners, that is, "*points whose local neighbourhood stands in two dominant and different edge directions*" (see Figure 25c),

SIFT [25] is an algorithm used to "*detect, describe, and match local features in images*" (see Figure 25d),

SURF [26] is a patented CV local feature detector and descriptor, an evolution of SIFT with much greater performance,

ORB [27] is another CV local feature detector, aiming to offer a fast and efficient alternative to SIFT (see Figure 25e)

HOG [28]is another feature descriptor for object detection.

These CV additional features have been computed for simulations of the Rectoret WF event provided by the WFs pilot. The added features are the following: the SIFT keypoints and descriptors, the SURF keypoints and descriptors, the ORB keypoints and descriptors, and the HOG features. The computation of Canny edges and Harris corners is also implemented, although not included in the enhanced WFs dataset, but they could be included if required to train the autoencoder model that shows the best performance for learning the latent representation of WFs simulations.

This feature enhancement implementation for WFs based on CV features is available as open source[5].

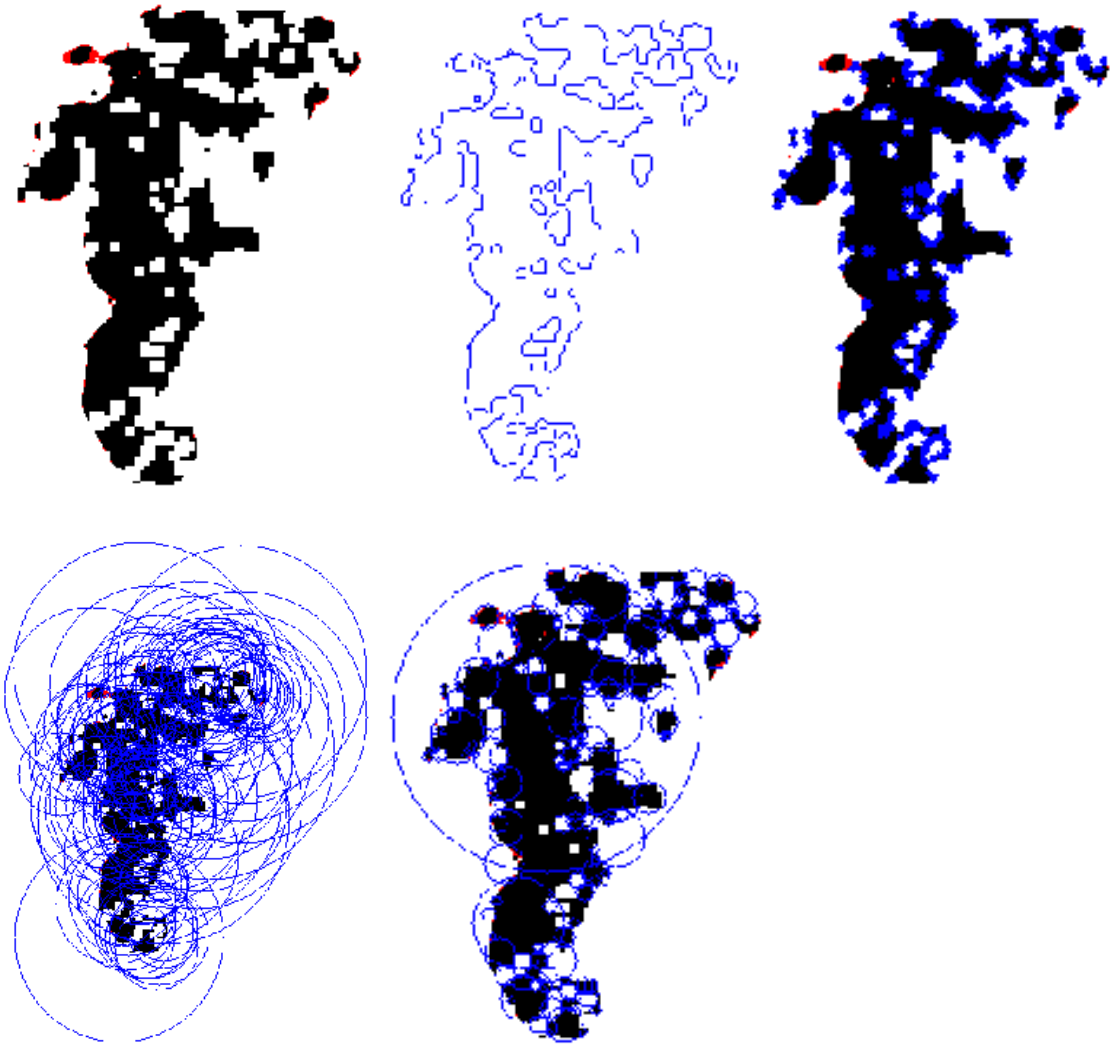

Figure 25: From top left to right and bottom, a) snapshot of a fire evolution, b) snapshot contour computed with the Canny Edge algorithm, c) corners detected with the Harris algorithm, d) features detected in the snapshot with the SIFT algorithm, e) features detected by the ORB algorithm.

3.4.4 Summary

On the HPDA side, forest fire forecasting in the Rectoret region has demonstrated that burn probability is primarily driven by vegetation continuity metrics and fuel connectivity indices, with machine learning and deep learning methods offering complementary perspectives on spatial sensitivity and dynamic fire propagation processes. While D4.3 established the baseline analytical framework, the current deliverable reports comprehensive feature importance analyses across multiple model architectures,

[5] https://git.hidalgo2.eu/hidalgo2-group/hid-ai-wf/-/tree/cv-features?ref_type=heads

including classical random forest and gradient boosting approaches alongside deep neural network interpretability techniques such as saliency mapping, occlusion analysis, and perturbation-based sensitivity assessments. These advances provide fire managers with both aggregate variable importance rankings and spatially explicit sensitivity maps that reveal where specific environmental factors exert their strongest influence on fire occurrence and spread dynamics.

In the AI track, the evolution from D4.3's handcrafted feature extraction pipeline to the current autoencoder-based similarity retrieval system represents a fundamental methodological shift. Whereas D4.3 focused on exploratory principal component analysis of geometric fire descriptors, the current work implements trainable LSTM encoder-decoder architectures that learn compact latent representations capturing essential temporal dynamics of fire evolution. Systematic experiments across latent dimensionalities (2 to 256 dimensions) revealed that intermediate representations provide optimal trade-offs between compression and expressiveness, avoiding both underfitting and overfitting failure modes. Critically, the establishment of an expert-driven evaluation protocol in collaboration with pilot partners provides ground-truth validation signals absent in D4.3, enabling data-driven model selection based on operational relevance rather than purely reconstruction metrics.

A parallel advancement introduced since D4.3 is the computer vision feature enhancement infrastructure, which systematically extracts classical visual descriptors – Canny edges, Harris corners, SIFT, SURF, ORB, and HOG features – from fire perimeter imagery. This feature engineering pipeline has been implemented and applied to Rectoret simulations, with the enhanced dataset now available for integration into future autoencoder training campaigns. This represents a strategic investment in richer input representations that may capture shape and texture evolution aspects not fully expressed in the original geometric feature set.

Regarding future work, HPDA efforts will expand the burn probability analysis to additional geographic regions beyond Rectoret, integrate higher-resolution environmental and meteorological data layers, and investigate real-time implementation pathways to enhance operational relevance for active fire management scenarios. The AI roadmap prioritizes deploying a first-version API to enable pilot teams to submit retrieval queries and provide structured feedback, followed by targeted model refinement guided by expert evaluation outcomes and integration of the CV-enhanced features into production similarity models. Together, these developments establish a robust foundation for operational wildfire decision support tools that combine spatial risk assessment with historical scenario retrieval capabilities.

3.5 Material Transport in Water (MTW)

3.5.1 Scope, Objectives & Inputs

Within the HiDALGO2 framework, the Material Transport in Water (MTW) pilot addresses the critical need for scalable, high-fidelity simulations of particle-laden flows and their efficient surrogate modelling. This newest addition to the HiDALGO2 pilot portfolio – introduced after Deliverable D4.3 – focuses on modelling pollutant transport in aquatic environments, with particular emphasis on microplastics and chemical contaminants in river systems. While classical computational fluid dynamics (CFD) solvers based on the Lattice Boltzmann Method (LBM) [29] remain the backbone of accurate flow computations, their application in large-scale ensemble studies or long-time horizon scenarios is often prohibitively expensive. To overcome this limitation, the MTW pilot extends beyond traditional solver development into the creation of **data-driven surrogate models** capable of delivering **fast, accurate flow field predictions** without the full computational burden of direct numerical simulation.

This dual approach – combining high-fidelity LBM simulations with machine learning surrogates – directly supports the MTW pilot's main objectives: i) enabling real-time "*what-if*" analyses for environmental impact assessment, ii) accelerating scenario exploration for pollution mitigation strategies, and iii) reducing both the environmental and financial footprint of HPC usage. The scope of this contribution within Work Package 4 is explicitly two-fold, encompassing both the **infrastructure for data generation** and the **systematic creation of training datasets**.

Pipeline Development. To standardize the path from conceptual geometry to machine-learning–ready datasets, we created *ChannelFlow-Tools* [30], a configuration-driven, end-to-end HPDA framework. This pipeline automates the entire workflow, from programmatic geometry generation to machine-learning–ready datasets, ensuring reproducibility and scalability across thousands of simulation cases. By encapsulating domain expertise into reusable configuration templates, *ChannelFlow-Tools* eliminates manual bottlenecks and enables rapid iteration on geometry families, flow regimes, and data sampling strategies.

Dataset Development. Built on this pipeline infrastructure, the team generated the **ChannelFlow dataset**, a corpus of approximately $10{,}000$ three-dimensional channel flow simulations featuring embedded obstacles of varying complexity. Each simulation aims to i) capture flow regimes from laminar to transitional turbulence and ii) include diverse geometries and spatial placements to maximize the representativeness of wake structures, shear layers, and vortex interactions characteristic of real-world material transport scenarios.

The geometric diversity that is required of this dataset is achieved through procedurally generated obstacles exported as STL [31] surface meshes and embedded within a standardized channel domain of $[0,\ 2048]\times[0,\ 512]\times[0,\ 512]$ lattice units (LU). Six primitive shape families are supported: **cuboid**, **cone**, **cylinder**, **sphere**, **torus**, and

wedge, instantiated with varied *dimensions*, *orientations*, and *positions* within a designated region of interest ($x \in [146, 1800]$). Single-obstacle and multi-obstacle configurations (up to $3–4$ objects per case) are generated systematically.

Parameterization ranges are tailored to each geometry family to capture realistic obstacle scales and orientations:

- **cuboid:** dimensions independently vary between $100\text{-}500$ LU along each axis.
- **cylinder:** radii vary between $50\text{-}200$ LU; heights extend up to 400 LU; tilt angles sampled uniformly across $0–360°$.
- **sphere:** radii range between $50\text{-}250$ LU; optional cut angles enable partial-sphere geometries.
- **wedge:** structures span lengths of $100–500$ LU with opening angles between $15°$ and $120°$;
- **torus:** major radii extend between $100\text{-}250$ LU; minor radii between $20–60$ LU.

These systematic variations yield geometrically diverse obstructions that generate representative wake structures, separated shear layers, and complex vortex interactions directly applicable to understanding pollutant transport pathways in aquatic environments.

3.5.2 HPDA application

3.5.2.1 Pipeline Development and Toolchain

A central contribution of the MTW pilot within HiDALGO2 is the development of *ChannelFlow-Tools*, a fully automated, end-to-end pipeline that transforms raw obstacle definitions into machine-learning–ready CFD datasets. This comprehensive framework integrates geometry synthesis, signed-distance field (SDF) generation, solver automation, resampling, and provenance tracking into a single configuration-driven system. Each stage is designed with reproducibility, scalability, and high throughput on HPC resources as core priorities – hallmarks of High-Performance Data Analytics (HPDA) where large-scale execution and transparency are as critical as raw computational performance.

Geometry Generation and Simulation Setup. The pipeline architecture begins with automated obstacle creation through CadQuery [32] and the OpenCASCADE [33] kernel. Objects that belong to the six primitive geometry families are procedurally instantiated with randomized parameters drawn from Hydra configurations with parameterizable dimensions and spatial orientations. Each candidate geometry undergoes rigorous validation against feasibility constraints: bounding-box inclusion within the channel region of interest, minimum volume thresholds, non-intersection with existing obstacles, and clearance margins of at least $2\Delta x$ to ensure numerical stability. Sampling policies support both uniform random distributions and Sobol [34] low-discrepancy sequences, with the latter enabling structured coverage of high-dimensional parameter spaces while retaining deterministic reproducibility across

runs. Accepted geometries are fused into watertight STL meshes and paired with YAML sidecars that encode complete simulation metadata, including Reynolds numbers, inlet velocity profiles, and periodic boundary conditions. Together, these artifacts form atomic, self-contained units of dataset creation.

Signed Distance Field (SDF) Generator. Following geometry generation, the pipeline voxelises each obstacle configuration into signed distance fields using OpenVDB's level-set representation, where $\varphi < 0$ indicates solid interior regions, $\varphi > 0$ marks the fluid domain, and $\varphi = 0$ defines the precise obstacle interface. This representation captures smooth geometry encodings while preserving numerical robustness for downstream machine learning applications. Critically, co-registration ensures that identical origin points and voxel spacing are maintained across all resolution levels, allowing perfect spatial alignment between geometry inputs and flow field targets. The pipeline supports three standard grid resolutions – $128\times32\times32$, $256\times64\times64$, and $512\times128\times128$ – carefully balancing geometric fidelity against storage requirements and computational throughput. Outputs include dense NumPy arrays as the primary machine-learning input format, with optional VTK and VTU volume exports enabling visual quality assurance. Batch generation is fully parallelized via SLURM job scheduling, with launcher scripts managing thousands of input-output pairs across compute nodes, achieving speedups of $35\text{-}40\times$ on 40-core FAU cluster configurations.

Automation Framework. The automation framework serves as the backbone of *ChannelFlow-Tools*, enabling ensemble-scale execution spanning thousands of cases without manual intervention. Its fundamental design principle ensures that each simulation case is materialized into a self-contained run capsule, embedding geometry files, metadata records, solver configurations, and job submission scripts. These capsules guarantee idempotency – rerunning any case yields identical outputs without duplication or data corruption. Deterministic orchestration ensures that SLURM jobs are submitted in reproducible order, with Sobol sequence indices and configuration hash values preserved across re-runs. Comprehensive provenance capture logs Hydra configuration snapshots, random number generator seeds, git commit identifiers, solver version information, and loaded environment modules for each run, enabling complete audit trails. Failure isolation mechanisms allow individual job failures to trigger targeted retries without impacting the broader submission sequence or altering subsequent random draws, ensuring consistent dataset coverage. The system scales efficiently to support thousands of simultaneous jobs distributed across HPC resources while maintaining submission-order determinism, transforming massive ensemble campaigns of $10{,}000$+ runs into manageable, transparent workflows.

Solver Orchestration. Simulations are executed using the waLBerla Lattice Boltzmann Method framework [35], configured with cumulant collision operators and Smagorinsky large-eddy simulation closures to maintain stability across high Reynolds number flows. Geometry embeddings are derived from SDF-based classifications that distinguish fluid and solid voxels with sub-lattice accuracy. Boundary conditions follow

standardized dataset policies: velocity inlet and pressure outlet conditions in streamwise directions, with link-wise no-slip wall treatments on transverse faces. Target Reynolds numbers are achieved through inlet velocity scaling, with local subgrid contributions added dynamically during simulation execution. To produce low-variance training targets suitable for supervised learning, the solver performs online time-averaging of flow fields, with automated stationarity checks monitoring flux balance convergence and velocity field stabilization before output writing.

Resampling and ML Grid Export. The final resampling stage converts raw simulation outputs into machine-learning–ready Cartesian grids at three standard resolutions. Using ParaView's PointVolumeInterpolator [36], three-dimensional velocity and density fields are resampled onto consistent voxel layouts that maintain perfect co-registration with their corresponding SDF geometry inputs. Multiple interpolation kernels and spatial footprints – including linear, Gaussian, and Shepard methods – are supported via configuration parameters, allowing transparent trade-offs between accuracy and computational cost. This critical transformation compresses solver output data, typically 1.8 GB per case in VTU format, into lightweight standardized NumPy arrays of approximately 40 MB per case, enabling efficient ingestion by neural network architectures while preserving multi-resolution fidelity across the three supported grid scales.

3.5.2.2 Dataset Development

Complementing the pipeline infrastructure, the second major contribution of the MTW pilot within HiDALGO2 is the systematic creation of the **ChannelFlow dataset**, a large-scale, machine-learning–ready corpus of obstructed channel flow simulations. Unlike ad hoc datasets assembled for individual studies, ChannelFlow is built entirely on top of the standardized *ChannelFlow-Tools* pipeline, ensuring that every case is generated, simulated, and post-processed under transparent, reproducible conditions. This foundational approach makes the dataset not only a valuable training resource for surrogate model development but also a reference benchmark for HPDA-driven CFD-ML integration across the broader computational fluid dynamics community.

Scale and Diversity of Cases. The pilot's dataset includes roughly $10{,}000$ simulations covering six obstacle families across laminar to turbulent Reynolds number regimes, capturing varied flow phenomena relevant to environmental scenarios. This parametric coverage provides a training corpus that is simultaneously industrially relevant – capturing diverse blockage scenarios encountered in environmental applications – and scientifically controlled through canonical channel flow boundary conditions and reproducible computational settings.

Dual Data Formats for Dual Purposes. Each simulation is strategically exported in two complementary formats that serve both traditional CFD analysis and modern machine learning workflows.

High-fidelity VTU outputs [37] preserve raw solver fields in VTK unstructured grid format, maintaining full waLBerla solver resolution including detailed lattice-level velocity, pressure, and turbulence statistics. At approximately 1.8 GB per case, this format supports in-depth fluid dynamics analysis, turbulence characterization, and cross-validation against alternative CFD solvers.

- For **machine learning** (ML) applications, solver outputs are resampled to structured Cartesian grids at three standard resolutions – $128\times32\times32$, $256\times64\times64$, and $512\times128\times128$ voxels. Each ML-ready sample includes time-averaged velocity and density fields alongside a co-registered signed distance field that encodes obstacle geometry in a compact, differentiable representation suitable for neural network architectures. These arrays, stored in **compressed NumPy format**, average approximately 40 MB per case – compact enough to enable large-scale training on standard GPU memory configurations yet sufficiently detailed to preserve wake structures, shear layers, and turbulence features critical for accurate surrogate modelling.

This dual-format strategy effectively bridges CFD and ML communities, allowing researchers to either analyse detailed VTU outputs for physical validation or directly train models on standardized arrays without redundant pre-processing overhead.

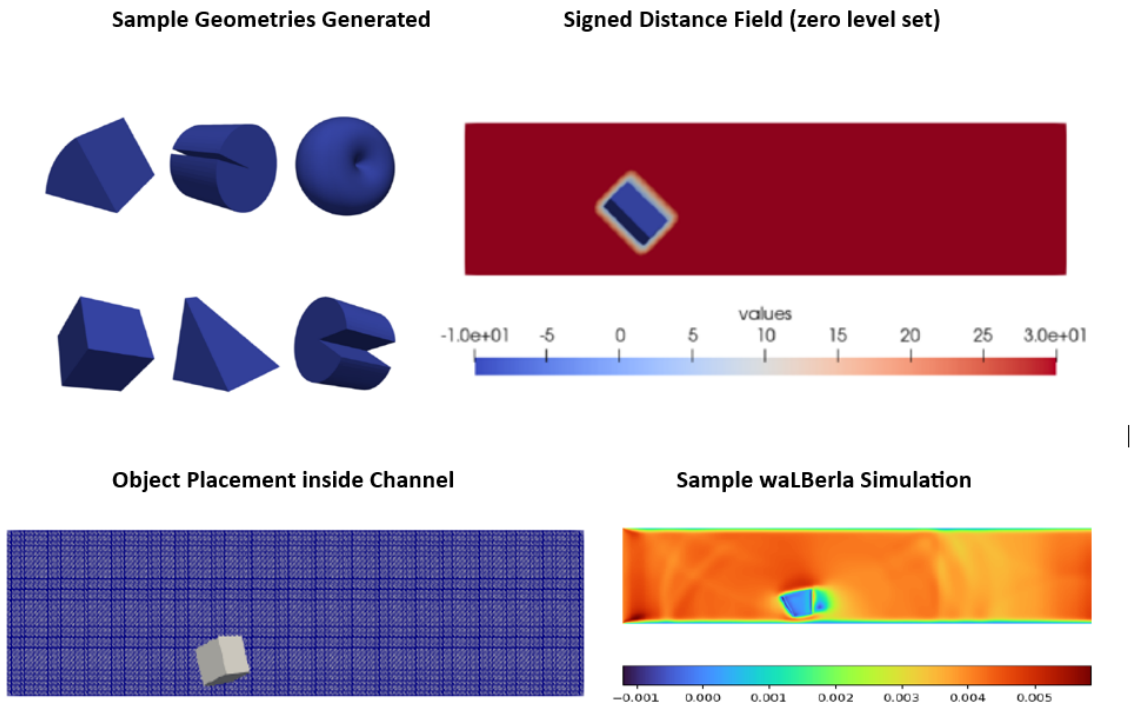

Figure 26: Sample of ChannelFlow toolbox's capabilities

Storage and HPC Throughput Considerations. The complete dataset totals approximately 20 TB of storage, hosted on the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) HPC cluster infrastructure. This scale underscores the fundamental HPDA challenge of managing, curating, and providing efficient access to

multi-terabyte datasets without creating bottlenecks in storage I/O or network bandwidth. To address these requirements, the pipeline implements throughput-oriented strategies across all processing stages:

- **batch processing** of SDF generation and field resampling distributes workloads across 40-core compute nodes, achieving wall-time speedups of 35-$40\times$ compared to serial execution;
- **compressed array storage** for ML applications reduces per-sample footprint from approximately 1.8 GB in VTU format to just 40 MB in NumPy format, enabling an order-of-magnitude reduction in storage requirements;
- and **case-level encapsulation** bundles geometry definitions, metadata records, run manifests, and simulation outputs into atomic units, enabling selective reruns and efficient dataset regeneration without reprocessing entire campaigns.

This architectural design ensures that the dataset infrastructure can scale seamlessly to larger ensemble campaigns while remaining navigable and fully reproducible, satisfying core HPDA requirements of data volume management, processing velocity, and computational verifiability.

Quality Assurance and Filtering. Dataset quality is ensured through automated verification checks deployed at multiple stages of the generation pipeline. Geometric validation enforces clearance tolerances, verifies watertight mesh topology, and confirms non-intersection constraints before simulation initialization. Flow stability is monitored through Reynolds number targeting under combined cumulant collision and Smagorinsky large-eddy simulation stability constraints [38], with runtime checks identifying divergent cases. Temporal consistency is guaranteed via online stationarity detection mechanisms, where velocity fields undergo time-averaging only after transient decay phases complete and mass flux balances converge to steady-state tolerances. Non-stationary or numerically unstable cases trigger automated flagging for extended integration periods or dataset exclusion. Resolution-aware fidelity preservation is achieved through multi-resolution export strategies that maintain vortex topology and turbulence structures across the three supported grid scales while carefully balancing storage costs and computational requirements.

3.5.2.3 Future Development Roadmap

Regarding future work, several strategic enhancements will extend the capabilities and impact of the *ChannelFlow-Tools* pipeline and dataset infrastructure. Dataset refinement and categorization efforts will introduce advanced analytical tools to systematically clean, filter, and taxonomically organize simulations according to multiple physical and geometric criteria, including Reynolds number regimes, obstacle geometry families, and flow classification (laminar, transitional, turbulent). This structured categorization will enable targeted model training strategies where neural

network architectures can be specialized for particular flow regimes or obstacle configurations, potentially improving surrogate model accuracy through domain-specific optimization.

Physics-based validation mechanisms represent a critical next step toward ensuring both dataset integrity and surrogate model reliability. Planned enhancements include automated verification checks based on fundamental conservation laws – mass, momentum, and energy balance constraints – alongside enforcement of divergence-free velocity field conditions and validation against established turbulence statistics such as energy spectra and Reynolds stress profiles. These physics-informed quality gates will operate at both dataset generation time, ensuring that only physically consistent simulations enter the training corpus, and at inference time, validating that surrogate model predictions satisfy fundamental fluid mechanics principles.

Workflow optimization and scalability improvements will focus on further reducing time-to-solution for large ensemble campaigns through enhanced automation, robust checkpointing mechanisms enabling fault-tolerant execution, and refined parallel decomposition strategies optimized for next-generation HPC architectures. Particular emphasis will be placed on achieving fully reproducible dataset generation across heterogeneous computing environments, with deterministic execution guarantees preserved even when distributing workloads across different cluster configurations or cloud-based HPC resources.

Finally, a comprehensive statistical analysis of the dataset will characterize its coverage properties, parametric variability, and representativeness across the target application domain. This effort will produce standardized dataset documentation in the form of "dataset cards" – structured metadata records that document sampling strategies, coverage gaps, known biases, and recommended use cases – to support reproducibility and establish the ChannelFlow corpus as a community-wide benchmarking resource for CFD-ML surrogate modelling research.

3.5.3 AI application

A critical dimension of the MTW pilot's contribution to HiDALGO2 is the development of an advanced surrogate modelling framework capable of rapidly predicting three-dimensional channel flows with embedded obstacles across diverse Reynolds number regimes. The ultimate vision driving this effort is to construct a general-purpose predictive model trained on the comprehensive **ChannelFlow dataset**, which can subsequently be adapted and fine-tuned for MTW-specific environmental applications – ranging from fundamental obstruction flow physics to complex scenarios involving particle-laden transport and contaminant dispersion in aquatic systems.

Advanced U-Net Surrogate Model. Building upon the well-established U-Net architecture originally developed for image segmentation, the team has designed and implemented an advanced 3D U-Net [39] that integrates multiple architectural enhancements to maximize representational capacity for complex turbulent flow fields.

The enhanced architecture features substantially deeper and wider encoder-decoder pathways incorporating repeated convolutional blocks with expanded channel dimensions reaching up to $2{,}048–4{,}096$ feature channels in the deepest network layers. Residual connections are integrated within both encoder and decoder blocks to improve gradient flow during backpropagation and enhance training stability across the deep network hierarchy. Multi-scale convolutional kernels – spanning $3\times3\times3$, $5\times5\times5$, $7\times7\times7$, and $9\times9\times9$ spatial extents – enable simultaneous capture of fine-scale turbulence structures such as vortex filaments and large-scale flow features including wake regions and recirculation zones. Additionally, Convolutional Block Attention Modules (CBAM) provide both channel-wise and spatial attention mechanisms, enabling the network to dynamically focus computational resources on the most physically relevant flow regions during inference.

These architectural enhancements substantially increase the model's capacity to represent complex fluid dynamics phenomena. While a standard 3D U-Net baseline contains approximately 80 million trainable parameters, the advanced architecture encompasses roughly 511 million parameters – a six-fold increase that demands careful orchestration of high-performance computing resources for both training and inference operations.

Training Strategy and HPC Integration. To manage this computational scale, the training workflow leverages PyTorch Distributed Data Parallel (DDP) [40] for data-parallel scaling across multiple GPU devices, combined with DeepSpeed ZeRO-2 [41] optimization for memory-efficient model parallelism that partitions optimizer states and gradients across devices. Initial testing on multi-GPU configurations featuring four NVIDIA A100 accelerators demonstrates epoch throughput improvements of $4–5\times$ compared to single-GPU baseline implementations, confirming the scalability of the distributed training approach.

The training corpus consists of the complete **ChannelFlow dataset** spanning approximately $10{,}000$ unique simulations (20 TB), with each case resampled into multi-resolution structured arrays at three standard grid scales: $128\times32\times32$, $256\times64\times64$, and $512\times128\times128$ voxels. Network inputs comprise signed distance field (SDF) representations of obstacle geometries, encoding the three-dimensional boundary geometry in a compact, differentiable format suitable for convolutional operations. Target outputs are time-averaged velocity vector fields and density distributions extracted from converged LBM simulations. Pre-processing experiments systematically compared min-max normalization and standard z-score scaling strategies, with empirical results demonstrating that standard scaling yields more stable training convergence and better gradient conditioning across the deep network layers.

Initial Experiments and Current Results. To validate the advanced architecture's capacity to represent complex flow physics, the team first conducted controlled overfitting experiments on small data subsets. Training on just 16 samples, the

advanced U-Net achieved an L1 reconstruction loss of 0.09, compared to 0.32 for the standard baseline architecture – demonstrating substantially superior expressive capacity. Qualitative analysis of predicted velocity fields confirmed that the advanced model captures sharper wake structures and finer-scale vortical features compared to the smoother predictions of the baseline network. Expanding the training corpus to $1{,}000$ samples with proper train-validation splits, both architectures were systematically evaluated. While the advanced model consistently outperformed the baseline across multiple metrics, prediction accuracy remained below expectations given the substantial architectural enhancements, highlighting a critical insight: the model's capacity exceeds the diversity and volume of the current training subset.

Current results remain preliminary and reveal important directions for future development. The advanced architecture demonstrates clear capacity advantages and produces sharper feature recovery than standard U-Net implementations, yet struggles to achieve robust generalization when trained on limited subsets of the full $10{,}000$-case dataset. Predicted velocity fields in regions of high-velocity gradients – particularly in object wakes and shear layers – exhibit excessive smoothing compared to ground-truth LBM solutions, indicating insufficient exposure to the full diversity of flow phenomena present in the complete dataset. Training convergence patterns confirm that broader coverage of Reynolds number regimes and richer geometric obstacle distributions are necessary to achieve the stable accuracy required for production deployment in environmental transport modelling scenarios. These findings establish a clear roadmap: full-dataset training campaigns leveraging the complete ChannelFlow corpus and the HiDALGO2 HPC infrastructure represent the critical next phase for advancing surrogate model fidelity toward operational MTW applications.

3.5.4 Summary

The Material Transport in Water (MTW) pilot represents an entirely new contribution to HiDALGO2 that was not present in Deliverable D4.3. Introduced after the previous deliverable's publication, MTW addresses critical environmental challenges in modelling pollutant transport and particle-laden flows in aquatic systems, with particular emphasis on microplastics dispersion and chemical contaminant tracking in river networks.

Since its inception, the MTW pilot has delivered two major integrated contributions that exemplify the synergistic combination of HPDA and AI methodologies central to Work Package 4. On the HPDA side, the team developed *ChannelFlow-Tools*, a comprehensive end-to-end pipeline infrastructure that automates the complete workflow from procedural geometry generation through Lattice Boltzmann Method simulation execution to machine-learning–ready dataset export. This configuration-driven framework implements reproducible execution across thousands of simulation cases on HPC resources, achieving $35\text{-}40\times$ throughput improvements through intelligent parallelization and achieving full provenance tracking for scientific

reproducibility. Building upon this infrastructure, the team systematically generated the ChannelFlow dataset – a 20 TB corpus including ~10,000 unique channel flow simulations spanning Reynolds numbers from 100 to 15,000 involving obstacles that can be classified to six elementary geometric families. The dataset's dual-format architecture serves both traditional CFD validation workflows (via 1.8 GB VTU outputs) and modern ML training pipelines (via compressed 40 MB NumPy arrays), effectively bridging computational fluid dynamics and machine learning communities.

From an HPDA perspective, the ChannelFlow dataset represents not merely a static corpus of simulation results but rather a comprehensive data production framework operating at an industrial scale. Its principal contributions span the four defining dimensions of HPDA systems:

- *Volume* – encompassing approximately 10,000 distinct simulations totalling 20 TB across wide parametric ranges of geometry and flow conditions;
- *Velocity* – automated throughput-oriented generation on HPC resources with reproducible scaling characteristics documented across multiple compute configurations;
- *Variety* – systematic geometric and flow regime diversity spanning six obstacle families and Reynolds numbers from laminar through turbulent regimes; and
- *Veracity* – comprehensive quality control through stationarity gates, geometric feasibility filters, and complete provenance tracking ensuring dataset integrity and scientific reproducibility.

Complementing the HPDA infrastructure, the AI track developed an advanced 3D U-Net surrogate modelling architecture containing 511 million trainable parameters – six times larger than standard baselines – with integrated residual connections, multi-scale convolutional kernels, and attention mechanisms optimized for complex turbulent flow prediction. Initial validation experiments demonstrated the architecture's superior expressive capacity, achieving L1 reconstruction losses of 0.09 compared to 0.32 for baseline models on controlled overfitting tests. Training infrastructure leverages PyTorch Distributed Data Parallel and DeepSpeed ZeRO-2 optimization across multi-GPU configurations, demonstrating 4-5× epoch throughput improvements on four NVIDIA A100 accelerators. While preliminary results on 1,000-sample training subsets confirm the model's capacity advantages, prediction accuracy in high-gradient flow regions reveals that full-dataset training campaigns are necessary to achieve robust generalization.

As a whole, these HPDA and AI contributions establish MTW as a foundational pilot demonstrating how systematic dataset generation infrastructure can enable data-driven surrogate modelling at scale, positioning the framework for future adaptation to environmental transport applications across the HiDALGO2 platform.

4 Conclusions

Deliverable D4.4 marks substantial progress in the development and operationalization of High-Performance Data Analytics (HPDA) and Artificial Intelligence (AI) methods within the HiDALGO2 project. Building on the foundational work from Deliverable D4.3, this report demonstrates significant advances across all pilot domains, underscoring HiDALGO2's role in delivering scalable, reproducible, and impactful analytics for complex environmental and urban challenges.

The Urban Air Pollution pilot has refined its data integration and analytical workflows, establishing a unified data infrastructure that serves both HPDA and AI analyses. This shared foundation improves pollutant dispersion modelling and supports more detailed airflow assessments.

The Urban Buildings pilot has transitioned from conceptual algorithms to full-scale Apache Spark implementations paired with graph neural networks that provide accurate, interpretable solar exposure predictions. These tools facilitate urban planning decisions aimed at sustainability and energy efficiency.

In Renewable Energy, new HPDA pipelines and AI models integrate high-fidelity mesoscale weather ensembles and photovoltaic production datasets, improving forecast accuracy and climate impact assessment capabilities compared to previous iterations.

The Wildfire pilot advances understanding through deep temporal modelling using LSTM autoencoders combined with computer vision feature augmentation, supported by expert evaluation frameworks and prototype APIs. This integrated approach better captures fire dynamics, enabling improved scenario retrieval and risk assessment.

A new contribution, the Material Transport in Water pilot, introduces an end-to-end HPC pipeline generating a vast, multi-terabyte machine-learning dataset, together with an advanced 3D U-Net surrogate model that offers rapid simulations of complex channel flows, expanding HiDALGO2's environmental modelling portfolio.

Across all pilots, further enhancements in HPC workflow management, provenance capture, and distributed AI training ensure analytical robustness and scalability. The reported work paves the way for wider geographic coverage, increased data granularity, and incorporation of physics-informed validations.

Moving towards the future, the continued refinement of AI models, deeper domain expert engagement, and expansion of datasets will be pivotal in translating HiDALGO2 innovations into operational resilience tools. Deliverable D4.4 thus represents a key milestone on the path toward impactful, HPC-enabled environmental analytics that address urgent global sustainability challenges.

References

[1] G. Stamou *et al.*, *HiDALGO2 D4.3 Advances in HPDA and AI for Global Challenges*. 2024. doi: 10.13140/RG.2.2.10065.95848.

[2] I. Colonnelli *et al.*, "Distributed workflows with Jupyter," *Future Generation Computer Systems*, vol. 128, pp. 282–298, 2022.

[3] *jupyter-incubator/sparkmagic*. (Sept. 16, 2025). Python. jupyter-incubator. Accessed: Oct. 01, 2025. [Online]. Available: https://github.com/jupyter-incubator/sparkmagic

[4] *apache/incubator-livy*. (Sept. 29, 2025). Scala. The Apache Software Foundation. Accessed: Oct. 01, 2025. [Online]. Available: https://github.com/apache/incubator-livy

[5] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in *2nd USENIX workshop on hot topics in cloud computing (HotCloud 10)*, 2010. Accessed: Oct. 01, 2025. [Online]. Available: https://www.usenix.org/event/hotcloud10/tech/full_papers/Zaharia.pdf

[6] V. K. Vavilapalli *et al.*, "Apache Hadoop YARN: yet another resource negotiator," in *Proceedings of the 4th annual Symposium on Cloud Computing*, Santa Clara California: ACM, Oct. 2013, pp. 1–16. doi: 10.1145/2523616.2523633.

[7] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The hadoop distributed file system," in *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)*, Ieee, 2010, pp. 1–10. Accessed: Oct. 01, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5496972/

[8] C. Neuman, S. Hartman, K. Raeburn, and T. Yu, "The Kerberos Network Authentication Service (V5)," Internet Engineering Task Force, Request for Comments RFC 4120, July 2005. doi: 10.17487/RFC4120.

[9] Computational Engineering International, "EnSight User Manual," Computational Engineering International. Accessed: Oct. 01, 2025. [Online]. Available: https://dav.lbl.gov/archive/NERSC/Software/ensight/doc/Manuals/UserManual.pdf

[10] X. Zeng, Y. Hui, J. Shen, A. Pavlo, W. McKinney, and H. Zhang, "An Empirical Evaluation of Columnar Storage Formats," *Proc. VLDB Endow.*, vol. 17, no. 2, pp. 148–161, Oct. 2023, doi: 10.14778/3626292.3626298.

[11] *Directive (EU) 2024/2881 of the European Parliament and of the Council of 23 October 2024 on ambient air quality and cleaner air for Europe (recast)*. 2024. Accessed: Oct. 01, 2025. [Online]. Available: http://data.europa.eu/eli/dir/2024/2881/oj/eng

[12] L. Torres *et al.*, *HiDALGO2 D5.3 Research Advances for the Pilots*. 2023. doi: 10.13140/RG.2.2.19390.46400.

[13] J. Bousquin, "Discrete Global Grid Systems as scalable geospatial frameworks for characterizing coastal environments," *Environmental Modelling & Software*, vol. 146, p. 105210, 2021.

[14] A. Hagberg, P. J. Swart, and D. A. Schult, "Exploring network structure, dynamics, and function using NetworkX," Los Alamos National Laboratory (LANL),

Los Alamos, NM (United States), 2008. Accessed: Oct. 06, 2025. [Online]. Available: https://www.osti.gov/biblio/960616

[15] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” Feb. 22, 2017, *arXiv*: arXiv:1609.02907. doi: 10.48550/arXiv.1609.02907.

[16] A. Dimitriou, M. Lymperaiou, G. Filandrianos, K. Thomas, and G. Stamou, “Structure Your Data: Towards Semantic Graph Counterfactuals,” Mar. 11, 2024, *arXiv*: arXiv:2403.06514. Accessed: Apr. 16, 2024. [Online]. Available: http://arxiv.org/abs/2403.06514

[17] B. Bosak, P. Kopta, M. Kulczewski, and T. Piontek, “mUQSA – An Online Service for Uncertainty Quantification and Sensitivity Analysis,” in *Computational Science – ICCS 2025 Workshops*, vol. 15911, M. Paszynski, A. S. Barnard, and Y. J. Zhang, Eds., in Lecture Notes in Computer Science, vol. 15911. , Cham: Springer Nature Switzerland, 2025, pp. 57–70. doi: 10.1007/978-3-031-97570-7_6.

[18] F. Galeazzo *et al.*, *HiDALGO2 D4.9 Uncertainty Quantification*. 2025. doi: 10.13140/RG.2.2.20533.79845.

[19] Unidata, *Network Common Data Form (netCDF)*. (2025). UCAR/Unidata, Boulder, CO. Accessed: Oct. 01, 2025. [Online]. Available: https://doi.org/10.5065/D6H70CW6

[20] “XGBoost Documentation — xgboost 3.0.5 documentation.” Accessed: Oct. 03, 2025. [Online]. Available: https://xgboost.readthedocs.io/en/stable/

[21] PyTorch Developers, “ReduceLROnPlateau — PyTorch 2.8 documentation,” PyTorch documentation. Accessed: Oct. 06, 2025. [Online]. Available: https://docs.pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

[22] IOGP (International Association of Oil & Gas Producers), “ETRS89 / UTM zone 31N (EPSG:25831).” 2019. [Online]. Available: https://epsg.io/25831

[23] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 2009.

[24] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Alvey vision conference*, Manchester, UK, 1988, pp. 10–5244. Accessed: Oct. 01, 2025. [Online]. Available: https://bmva-archive.org.uk/bmvc/1988/avc-88-023.pdf

[25] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the seventh IEEE international conference on computer vision*, Ieee, 1999, pp. 1150–1157. Accessed: Oct. 01, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/790410/

[26] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded Up Robust Features,” in *Computer Vision – ECCV 2006*, vol. 3951, A. Leonardis, H. Bischof, and A. Pinz, Eds., in Lecture Notes in Computer Science, vol. 3951. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417. doi: 10.1007/11744023_32.

[27] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in *2011 International conference on computer vision*, Ieee, 2011, pp. 2564–2571. Accessed: Oct. 01, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6126544/

[28] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, Ieee, 2005, pp. 886–893. Accessed: Oct. 01, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/1467360/

[29] G. G. McNamara and G. Zanetti, “Use of the Boltzmann equation to simulate lattice-gas automata,” in *Lattice Gas Methods For Partial Differential Equations*, CRC Press, 2019, pp. 289–296. Accessed: Oct. 03, 2025. [Online]. Available: https://api.taylorfrancis.com/content/chapters/edit/download?identifierName=doi&identifierValue=10.1201/9780429032738-16&type=chapterpdf

[30] S. Kavane, K. Kulkarni, and H. Koestler, “ChannelFlow-Tools: A Standardized Dataset Creation Pipeline for 3D Obstructed Channel Flows,” Sept. 17, 2025, *arXiv*: arXiv:2509.15236. doi: 10.48550/arXiv.2509.15236.

[31] Library of Congress, “STL (STereoLithography) File Format Family,” Sustainability of Digital Formats: Planning for Library of Congress Collections. Accessed: Oct. 03, 2025. [Online]. Available: https://www.loc.gov/preservation/digital/formats/fdd/fdd000504.shtml

[32] AU *et al.*, *CadQuery/cadquery: CadQuery 2.4.0*. (Jan. 15, 2024). Zenodo. doi: 10.5281/zenodo.10513848.

[33] *Open-Cascade-SAS/OCCT*. (Oct. 02, 2025). C++. Open Cascade SAS. Accessed: Oct. 03, 2025. [Online]. Available: https://github.com/Open-Cascade-SAS/OCCT

[34] I. M. Sobol, “Distribution of points in a cube and approximate evaluation of integrals,” *USSR Computational mathematics and mathematical physics*, vol. 7, pp. 86–112, 1967.

[35] M. Bauer *et al.*, “waLBerla: A block-structured high-performance framework for multiphysics simulations,” *Computers & Mathematics with Applications*, vol. 81, pp. 478–501, 2021.

[36] J. Ahrens, B. Geveci, and C. Law, “Paraview: An end-user tool for large data visualization,” *The visualization handbook*, vol. 717, no. 8, 2005, Accessed: Oct. 03, 2025. [Online]. Available: https://www.academia.edu/download/76144491/ParaView_An_End-User_Tool_for_Large_Data20211211-11702-17fcn1r.pdf

[37] W. Schroeder, K. M. Martin, and W. E. Lorensen, *The visualization toolkit an object-oriented approach to 3D graphics*. Prentice-Hall, Inc., 1998. Accessed: Oct. 03, 2025. [Online]. Available: https://dl.acm.org/doi/abs/10.5555/272980

[38] J. Smagorinsky, “General circulation experiments with the primitive equations: I. The basic experiment,” *Monthly weather review*, vol. 91, no. 3, pp. 99–164, 1963.

[39] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, vol. 9351, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., in Lecture Notes in Computer Science, vol. 9351. , Cham: Springer International Publishing, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[40] S. Li *et al.*, “PyTorch Distributed: Experiences on Accelerating Data Parallel Training,” June 28, 2020, *arXiv*: arXiv:2006.15704. doi: 10.48550/arXiv.2006.15704.

[41] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, “DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Virtual Event CA USA: ACM, Aug. 2020, pp. 3505–3506. doi: 10.1145/3394486.3406703.