Grant number: 883286
Project duration: Sep 2020 – Aug 2022
Project Coordinator: Joe Gorman, SINTEF

Horizon 2020: Secure societies
SU-INFRA02-2019
Security for smart and safe cities, including for public spaces
Project Type: Innovation Action

http://www.impetus-project.eu

IMPETUS Project Deliverable: D4.2

Data analytics and ingestion-time access control final report

Dissemination Status: Public

Authors: Marco Anisetti, Annalisa Appice, Claudio Agostino Ardagna, Alessandro Balestrucci, Chiara Braghin, Edoardo Cavallo, Michelangelo Ceci, Ernesto Damiani, Nicola Di Mauro, Stefano Ferilli, Goffredo Haus, Corrado Loglisci, Donato Malerba, Francesca Mazzia, Costantino Mele, Paolo Mignone, Antongiacomo Polimeno (CINI), Radu Popescu (SIV)

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

About IMPETUS

IMPETUS (Intelligent Management of Processes, Ethics and Technology for Urban Safety) is a Horizon 2020 Research and Innovation project that provides city authorities with new means to improve the security of public spaces in smart cities, and so help protect citizens. It delivers an advanced, technology-based solution that helps operational personnel, based on data gathered from multiple sources, to work closely with each other and with state-of-the art tools to detect threats and make well-informed decisions about how to deal with them.

IMPETUS provides a solution that brings together:

- *Technology*: leverage the power of Internet of Things, Artificial Intelligence and Big Data to provide powerful tools that help operational personnel manage physical and cyber security in smart cities.
- *Ethics*: Balance potentially conflicting needs to collect, transform and share large amounts of data with the imperative of ensuring protection of data privacy and respect for other ethical concerns - all in the context of ensuring benefits to society.
- *Processes*: Define the steps that operational personnel must take, and the assessments they need to make, for effective decision making and coordination - fully aligned with their individual context and the powerful support offered by the technology.

Technological results are complemented by a set of *practitioner's guides* providing guidelines, documentation and training materials in the areas of operations, ethical/legal issues and cybersecurity.

IMPETUS places great emphasis on taking full and proper account of ethical and legal issues. This is reflected in the way project work is carried out, the nature of the project's results and the restrictions imposed on their use, and the inclusion of external advisors on these issues in project management.

The cities of Oslo (Norway) and Padova (Italy) have been selected as the site of practical trials of the IMPETUS solution during the project lifetime, but the longer-term goal is to achieve adoption much more widely.

The work is carried out by a consortium of 17 partners from 11 different EU Member States and Associated Countries. It brings together 5 research institutions, 7 specialist industrial and SME companies, 3 NGOs and 2 local government authorities (the trial sites). The consortium is complemented by the Community of Safe and Secure Cities (COSSEC) – a group established by the project to provide feedback on the IMPETUS solution as it is being developed and tested.

The project started in September 2020 with a planned duration of 30 months.

For more information

Project web site: https://www.impetus-project.eu/
Project Coordinator: Joe Gorman, SINTEF: joe.gorman@sintef.no
Dissemination Manager: Snježana Knezić, TIEMS: snjezana.knezic@gmail.com

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

Executive Summary

Background and objectives: Big Data techniques and solutions are pushing towards a data-driven ecosystem where decisions are supported by more accurate analytics and (privacy-aware) data management. The complexity of a data-driven ecosystem is amplified by the heterogeneity and diversity of infrastructures/tools, on one side, and by the data themselves, on the other side. Data are in fact intrinsically heterogeneous, and collected at high rates and with different formats from different sources in dynamic and collaborative environments.

A major objective of the project and the focus of Work Package 4 is to provide a Big Data solution in the smart city domain that contributes to increase the security in public spaces. In particular, it supports the definition and execution of Big Data analytics for anomaly detection and event classification, driven by an access control system ensuring proper data management along the whole analytics pipeline.

The above objective introduces the need to redesign conventional platforms, tools, and methods used to collect, store, manage, and analyse data, while reviving the conflict between the need of protecting and the need of sharing data. So far, performance and scalability requirements have been mainly addressed, but no proper attention has been paid to the problem of balancing data sharing and data protection considering the peculiarities of Big Data ecosystems.

Furthermore, anomaly detection algorithms (e.g., Growing Hierarchical Self-Organizing Map – GH-SOM) require multiple iterations over the input dataset and are difficult to manage on larger datasets. Some of them are therefore designed to handle datasets composed of numerical attributes only. This aspect represents an important limitation of current systems, as most modern real-world datasets are characterized by mixed attributes, that is, numerical and categorical. In addition, most anomaly detection algorithms do not guarantee output that is interpretable by the user but merely provide feedback in the form of a Boolean response (i.e., normal or anomalous).

D4.2 presents the final version of the Big Data solution focusing on the integration between the data governance approach for data management and protection, and the analytics algorithms and pipelines for anomaly detection and event classification.

Approach: This document presents the extended version of the solution described in D4.1 [1], which is built on a customization of the traditional Apache-based Big Data platform.[1] The proposed Big Data platform is refined with new components and an extended access control system that enforces data protection requirements both on data collected from the smart cities partners of the IMPETUS project/Consortium (the ingestion-time access control described in D4.1 [1]) and on the data analysed throughout the analytics pipeline.

The data protection approach is driven by an extensive investigation of the key security requirements for Big Data governance. It is based on data annotations and secure data transformations performed before any data processing tasks, that is, it is platform-independent and can be implemented in any Big Data environments. The data governance approach supports ad hoc transformations that protect data during all steps of an analytics pipeline.[2]

The Big Data platform is finally complemented with a more accurate version of the algorithms for the construction of anomaly detection and event classification models, which guarantee large-scale efficient and distributed computation on time series data. Data governance and data analytics solutions are then integrated in a coherent and consistent solution and connected to the IMPETUS platform.

Results and conclusions: The Big Data solution implemented in this document is an improvement of the current state-of-art on data preparation and analysis, as well as privacy protection. The developed data governance methodology introduces a common and shared access control layer, whereas in existing ad hoc solutions each service is monitored for data governance compliance separately.

[1] We note that a detailed description of the components of our Apache-based Big Data platform can be found in D4.1 [1].
[2] We note that the solution in D4.1 only applied access control enforcement at ingestion time, before the analytics pipeline was even implemented.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

The proposed approach supports a data governance strategy that can be applied to all layers and services of a smart city architecture independently or to the whole smart city ecosystem by simply changing or adding new access control policies. This approach departs from traditional ad hoc solutions, where every single service should be updated according to the new data governance strategy.

Likewise, the data analytics methodology introduces the possibility to handle datasets composed of mixed attributes, making it applicable to address many real-world issues, including the financial and cyber-security domains. The proposed approach supports interpretability of the output by the end-user, by introducing a feature ranking, to better understand whether the identified anomaly represents a threat or not.

Finally, the resulting integrated approach permits the enforcement of access control policies at each step of the Big Data analytics pipeline, supporting emerging and future scenarios where a coalition of organizations provides and executes services orchestrated within the pipeline.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

Table of Contents

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

List of Figures

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

List of Tables

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

List of Abbreviations

Table 1: List of Abbreviations

Abbreviation	Explanation
AC	Access Control
API	Application Programming Interface
BDE	Big Data Engine
CRUD	Create, Read, Update, Delete
CSV	Comma-Separated Values
DAG	Directed Acyclic Graph
ELT	Extract, Load, Transform
ETL	Extract, Transform, Load
GDPR	General Data Protection Regulation
GH-SOM	Growing Hierarchical Self-Organizing Map
HDFS	Hadoop Distributed File System
HiveQL	Hive Query Language
HQL	Hibernate Query Language
JAR	Java Archive
JSON	JavaScript Object Notation
LSH	Locality Sensitive Hashing
ML	Machine Learning
NoSQL	Not only SQL
ODBC	Open Database Connectivity
OLAP	Online Analytical Processing
PII	Personal Identifiable Information
RDBMS	Relational Database Management System
REST	Representational state transfer
SOC	Security Operation Center
SQL	Structured Query Language
UI	User Interface
URL	Uniform Service Locators
XACML	eXtensible Access Control Markup Language
XML	Extensible Markup Language

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

List of Definitions

Table 2: List of Definitions

Term	Definition/explanation
Access Control	The process of permitting or restricting access to applications at a granular level, such as per-user, per-group, and per-resources (NIST SP 800-113).
Access Control System	A set of procedures and/or processes, normally automated, which allows access to a controlled area or to information to be controlled, in accordance with pre-established policies and rules (NIST SP 800-152).
Architecture	The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution (IEEE 1471).
Coalition	An alliance, especially a temporary one, of people, factions, parties, or nations.
Data Annotation	The process of adding relevant and further information to a single/set of data. Such information is called metadata and is assigned to a piece of data (generally a resource) in order to better frame it in terms of context, meaning, origin, usage and access mode.
Data Lifecycle	A data lifecycle refers to the entire period of time that data exist in a system. It encompasses all the stages that data go through, from first capture onward (in general, data creation, ingestion or collection, processing, analysis, publication or visualisation, archiving).
Data Lake	An ecosystem of data management tools (software components) to handle data peculiarities like the absence of a predefined structure. Every tool involved in the data lake should have some degree of distribution to fit the need of parallelism in computing pipelines.
Data sanitization	Data sanitization refers to a plethora of solutions and processes aimed to protect sensitive data in a document, message, database. It includes anonymization, generalization, suppression, masking, and so on.
Distributed system	A collection of independent computers that appear to the users of the system as a single computer.[3]
Obligation	An operation specified in a policy that should be performed in conjunction with the enforcement of an authorization decision (NIST SP 800-95 from OASIS XACML 2.0 Specification).
Pipeline	A sequence of tasks to achieve a specific goal. Each task can be parallelized on multiple executors improving the performance. Pipelines are suitable jobs for processing-oriented distributed systems.
Policy enforcement	The process of taking an access decision based on a policy.
Security policy (or access control policy)	A set of rules that governs all aspects of security-relevant system and system element behavior (NIST SP 800-160 Vol. 1).
Software ecosystem	A set of tools, normally loosely coupled, designed to work in synergy. They are integrated in a software architecture by means of specific configurations.

[3] Distributed Operating Systems, A. Tanenbaum, Prentice Hall, 1994

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

1 About this deliverable

1.1 Intended readership/users

The primary audience of this document is specialists, researchers, policy officers, safety and security operators, and practicians interested in Big Data management and Big Data analysis.

The audience can then be split into:

- **Primary Users.** All the Consortium Partners including: *i)* partners in charge of the IMPETUS platform and tools development that can be interested in the architectural approach at the basis of the data management and analytics solutions; *ii)* partners responsible for the specification of the privacy-preserving mechanisms that can be interested in the data management approach with particular reference to the access control mechanism and its data transformations; *iii)* partners involved in the definition of the ethics framework that can be interested in the data management approach.
- **Secondary Users.** Any possible adopters (with particular reference to COSSEC members) of the project results and, in particular, the data management approach and the data analytics algorithms for anomaly detection and event classification. More in details, it includes: *i)* external audience such as other R&D projects, system developers and engineers, data scientists, COSSEC members that can benefit by the technical aspects of data management, access control, and data analytics; *ii*) EU stakeholders (e.g., EU officers and policy makers) that can be interested in the proposed data management approach; *iii)* SMEs involved in the domain of data management, data protection, and data analytics that can be interested in the results described in this document.

The deliverable (or parts of it) may also be of interest to a wider group with an interest in data management and transformation to support collection, analysis and manipulation of data, and in data analysis based on anomaly detection and event classification.

1.2 Why would I want to read this deliverable?

D4.2 presents the final version of the data management and analytics solutions, which were first introduced in D4.1 [1]. D4.2 describes an integrated solution focusing on all aspects at the basis of a proper Big Data analytics process in the smart city domain, from data collection to data analysis, via data governance and protection. It also clarifies the overall benefits such a solution can bring to a scenario related to public space safety and security.

D4.2 develops through three main aspects of interest, which contribute to the integrated data management and analytics solutions for anomaly detection and event classification.

- A *data management layer* built on an innovative access control system that enforces data protection requirements along the overall analytics pipeline, from data collection from the partner cities to data analysis. In other words, this document shows how the ingestion-time access control described in D4.1 can be extended to monitor and enforce access to data during the whole big data pipeline execution.
- A *data analytics layer* with a more accurate version of the algorithms for the construction of anomaly detection and event classification models. This layer guarantees large-scale efficient and distributed computation on time series data, extending the anomaly detection solution in D4.1 and presenting the methodology behind the event classification task.
- The integration of data management and data analysis approaches in a coherent framework, presenting some examples of their application in the real context of the cities, partners of the IMPETUS project.

In summary, D4.2 clarifies how the ability to analyse large volumes of data can increase the safety and security of smart cities, boosting the adoption of smart processes even in critical domains such as healthcare and transportation, while enforcing data protection at run time.

1.3 Structure

The document is organized in four main chapters, as follows.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

Chapter 2 describes the final version of the access control solution for Big Data analytics pipelines. It firstly summarises the building blocks presented in D4.1 (Section 2.1). It then presents the updated version of the attribute-based access control methodology in D4.1 applied at both ingestion and analytics time (Section 2.2). The access control methodology is extended to cover the entire Big Data pipeline, meaning that data ingestion is generalized to an activity where data are collected, stored, and provided to a service node in the analytics pipeline.

Chapter 3 presents the updated version of the data analytics algorithms for anomaly detection (Section 3.2) and event classification (Section 3.3). The algorithms are extended with an approach to feature ranking that can be used in future version of the access control described in Chapter 2 to select between alternative applicable policies and transformations, thus limiting data loss and still preserving privacy.

Chapter 4 presents an end-to-end integration of the solutions proposed in Chapters 2 and 3, applied in real use cases at the partner cities. It first describes our ingestion-time access control, discussing data collection, data ingestion, and access control policy enforcement (Section 4.1). It further presents the integration between the access control system for big data pipelines in Chapter 2 and the designed data analytics in Chapter 3 (Section 4.2). It finally presents how the proposed solution is connected to the IMPETUS platform (Section 4.3).

Chapter 5 finally draws the conclusions and some additional developments that could be undertaken after the duration/conclusion of the project.

1.4 Other deliverables that may be of interest

D4.2 is related to the following deliverables:

- D1.2. “Requirements for public safety solutions”, providing requirements for data analytics and ingestion-time access control.
- D3.1 “Tool Development initial report” and D3.4 “Tool Development final report”, providing a description of the functionalities of the algorithms of the Physical Threat Intelligence (PTI) tool used as a starting point for the definition of enhanced Big Data analytics in this deliverable.
- D4.1 “Data analytics and ingestion-time access control initial report”, presenting the approach extended in this deliverable.
- D4.3 “Interface Design and Big Data Visualization”, presenting the User Interface (UI) of the entire IMPETUS platform that also visualises the alarms and events retrieved in D4.2 through the PTI tool.
- D5.2 “Initial mechanisms to preserve privacy in the secure smart city” and D5.4 “Mechanisms to preserve privacy in the secure smart city”, describing privacy-enhancing technologies that can be adopted in the data transformation process at the basis of the ingestion-time access control solution proposed in this document.

D4.2 is also indirectly related to:

- D5.1 “Initial ethical framework” and D5.3 “Ethical framework”, presenting the IMPETUS ethical framework that includes a discussion on how to develop, deploy and operate a trustworthy (i.e., lawful, ethical and robust) AI-based solution.
- D6.1 “Initial Concepts of Operation” and D6.3 “Operational framework- concepts of operations”, developing strategies and operating guidelines, aimed to support organizations in adopting the IMPETUS platform and tools.
- D7.1 “Validation Plan”, D7.2 “Acceptance Pilot Report”, presenting the process and activities undertaken to validate the solutions provided by IMPETUS platform.

1.5 Synergy with other projects/initiatives

No synergy.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

2 Access control in Big Data analytics pipelines

The conflict between the need of protecting and sharing data is hampering the spread of Big Data applications. In this context, access control systems are increasingly playing a central role, though existing solutions are not general and scalable enough to address the software and technological complexity of Big Data ecosystems. D4.1 proposed an innovative solution where traditional data management components are enriched with an ingestion-based access control system that enforces access to data in a distributed, collaborative, and multi-party Big Data environment.

This chapter introduces an updated version tailored to support Big Data ingestion at the partner cities, and its integration with an advanced access control system working both at ingestion and analytics time.

It extends the approach in D4.1 [1] by defining an innovative access control system working at each step of Big Data pipelines, from ingestion to analytics, protecting and managing data along the whole pipeline execution. In other words, access control policies are evaluated and enforced before data are provided as input to each node of the pipeline, implementing ingestion, preparation, processing, analytics, or reporting task. Moreover, ad hoc transformations are defined in addition to the ones natively provided by the Big Data Engine (BDE), increasing the flexibility of the data governance approach.

2.1 Ingestion-time Access Control – Summary from D4.1

This section summarises the fundamentals of the ingestion-based access control system, as introduced in D4.1 [1].

The implemented Big Data solution is composed of two sequential procedures (Figure 1): *i)* ingestion procedure, involving data sources and ingestion pipelines, *ii)* analytics procedure, involving analytics and visualization pipelines. We recall from D4.1 that Figure 1 depicts a streaming pipeline integrating ingestion and analytics procedures into a single pipeline for prediction (dashed arrows) and a typical batch processing or model creation analytics (solid arrows).

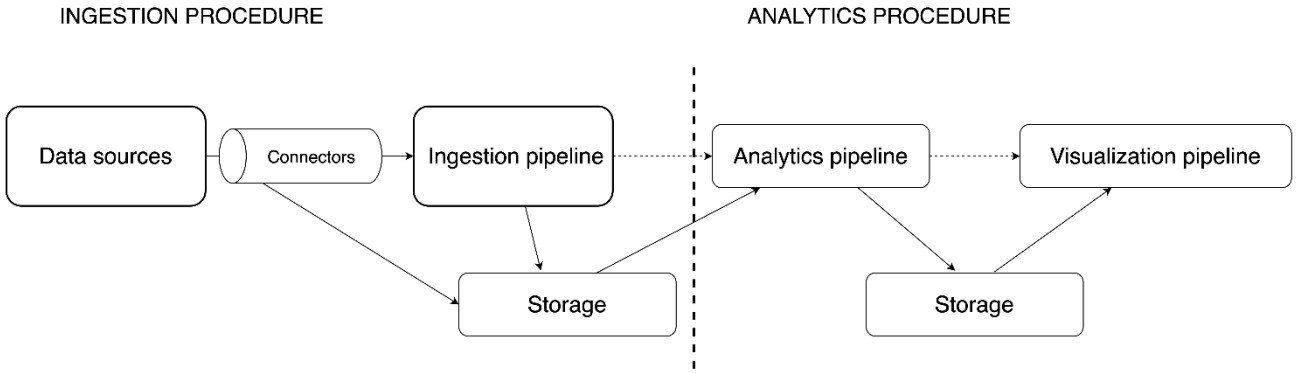

Figure 1: Architectural data flow.

The ingestion procedure is implemented as an advanced ETL (Extract/Transform/Load) where a data transformation is applied within the ingestion pipeline, if required by the security policy.

The aim of this advanced ETL method is to enforce access control at ingestion time before an access request on the data lake is received. This approach permits to have a fine-grained control on data access privileges in the framework of a Big Data computation, where data access can be granted prior to specific transformations (e.g., pseudo anonymisation) in strict connection with the access rights of the Big Data user. More specifically:

- *data sources* are the source of row data, connected to the ingestion pipeline via specific connectors (e.g., Kafka queue, API, specific ingestion tools);
- *ingestion pipeline* is a pipeline responsible to transform data flows (batch, micro-batch or stream) before storing them;

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

- *storage* is a distributed space where data are permanently stored. Data storage behaves differently depending on the data format. Depending on whether data are structured or not, there could be different storage solutions, different query languages, and different types of operations that can be performed on data;
- *analytics pipeline* is a pipeline responsible to retrieve values from the data after ingestion (e.g., build a Machine Learning - ML - model, or apply a model for prediction);
- *visualization pipeline* is a pipeline designed to prepare (e.g., aggregate) data to be graphically visualised. It usually includes some processing tasks to lower the dimension of the plot or to aggregate data.

2.1.1 Access Control Architecture and Policies

The ingestion procedure described in D4.1 has been implemented using a Policy-Aware ingestion pipeline, which is a pipeline extended with transformations triggered by access control policies [2]. These policies are based on *i)* the identity of the user asking for a specific ingestion and *ii)* the annotations on the ingested data.

Figure 2 shows the ingestion-time access control architecture.

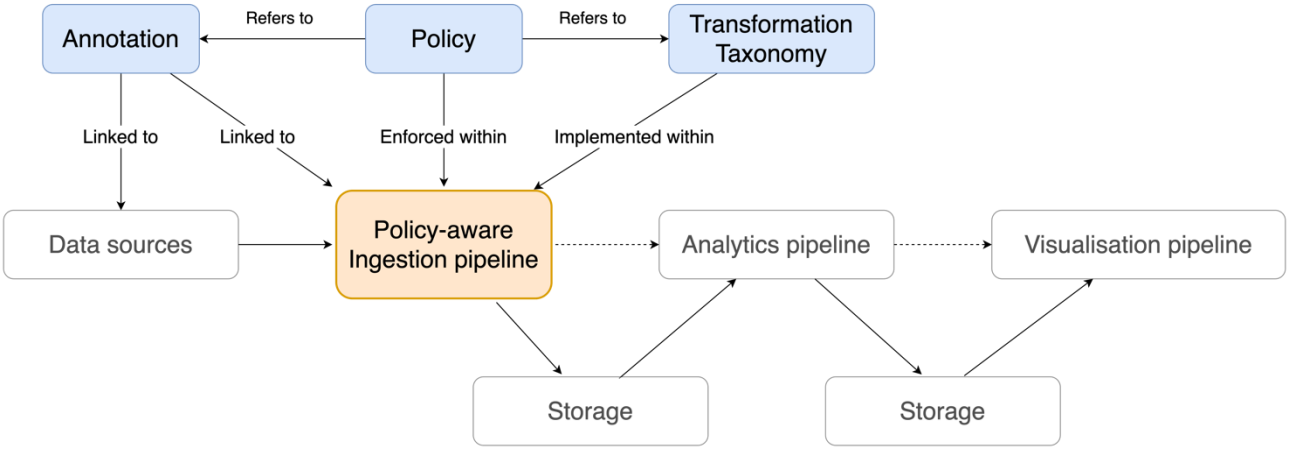

Figure 2: Policy aware ingestion pipeline structure

The methodology builds on and extends the policy structure of traditional attribute-based access control systems, involving data transformations instead of the traditional obligations.

```
<subject, action, object, context, data transformation>
```

where:

- the `subject` is the actor or the service who makes a request to access a certain object. It can be expressed either as a unique identifier or as a role/group the subject belongs to;
- the `action` declares the operation (e.g., read, write, update, delete, select) to be performed on the object for which permission is requested;
- the `object` defines the data, system component or service to be accessed;
- the `context` is a set of attributes (independent of a particular subject, object or action) that are relevant for the authorization decision;
- the `data transformation` is a type of obligation that sanitizes data resources to guarantee a minimum level of access to data.

Compared to traditional access control systems, a positive reply is always returned as a result of a policy enforcement. A deny access to data happens in extreme cases when data transformations delete all data resources.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

2.1.2 Data Ingestion

The data ingestion approach used in D4.1 builds the access control system described in Section 2.1.1. To integrate the transformations triggered by the access control policies, the ingestion pipeline in Figure 3 is structured as ETL extended with security and privacy-aware transformations ranging from pruning and reshaping to encrypting/decrypting or anonymizing the full resource or part of it.

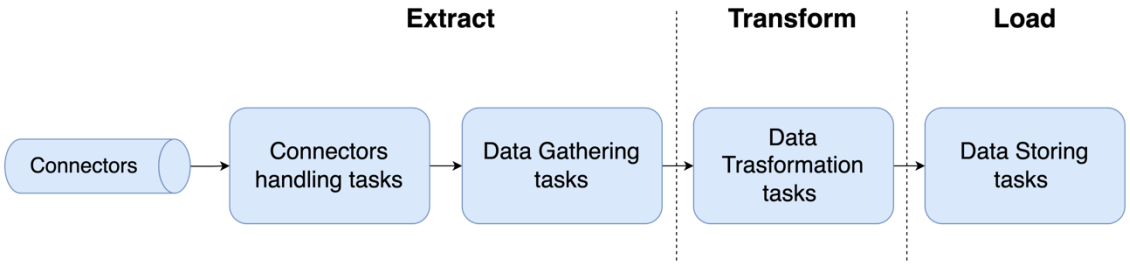

Figure 3: Our ETL Ingestion pipeline

The data extraction procedure includes:

i) *connectors* establishing the connection with the data sources;
ii) *connectors handling tasks* handling the different connection peculiarities;
iii) *data gathering tasks* collecting data;
iv) *data transformation tasks* transforming data formats;
v) *data storing/forwarding tasks* either saving or forwarding data to the analytics procedures.

2.1.3 Data Annotation and Transformation

The data annotations and transformations in D4.1 are at the core of our approach. The ETL ingestion pipeline depicted in Figure 4 is extended with data annotations and transformations that are used to sanitize the ingested data according to the access control policies.

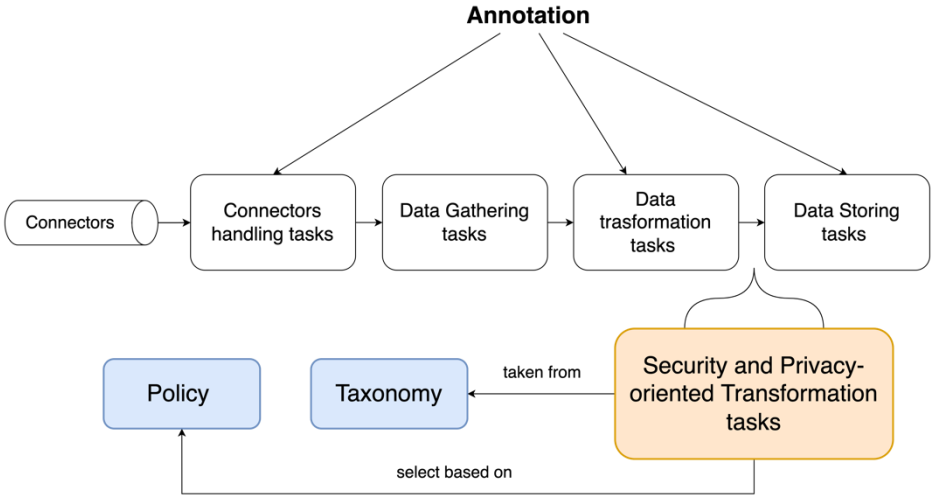

Figure 4: Annotations transformations and policies applied to our ETL ingestion pipeline.

Data annotations are applied to:

i) connection handling tasks, to annotate source of data;
ii) data transformation tasks, to annotate specific data fields;
iii) data storing/forwarding tasks, to provide post-transformation annotations.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

This labelling process consists of adding relevant information (called metadata) to a single/set of data.[4] It has two main advantages: *i)* it permits to identify, organise, and extract value from the (possibly unstructured) ingested raw data, *ii)* it enables resources categorization in different ways (e.g., by purpose, owner, environment, or other criteria), encompassing conventional database schema information.

In a distributed, collaborative, and multi-party Big Data environment, it is fundamental to agree on a consistent set of tag keys, called vocabulary. In D4.1 we identified a vocabulary for privacy classification of the data. More specifically, based on the General Data Protection Regulation (GDPR),[5] and on the advances in technology that have improved data collection and analytics, Personal Data (Personally Identifiable Information - PII) fall into three categories:

i) *explicit identifiers*, that is, any information that directly identifies individuals with certainty, such as national ID, assurance number, phone number, passport number;
ii) quasi-identifiers, a set of data attributes that could jointly or uniquely identify a subject when combined with publicly available data, such as ZIP code, date of birth, and address;
iii) sensitive information, data related to a person that do not permit direct identification. If linked to an individual, they reveal sensitive aspects of the individual private life.

The vocabulary we used for *Privacy classification* consists of a tag key *PrivacyClassification*, with three possible values (`PII-id`, `PII-quasi-id`, `PII-Sensitive`), representing the three above categories.

Annotations together with policies identify specific transformations to be executed in the data transformation task in Figure 4.

Security and privacy-aware transformations are special types of transformation tasks that are focused on compliance to regulations and standards, rather than simple format conversion. They need to have information about the semantics of the ingested data; this is normally provided via ad hoc design of pipelines (unfeasible in the dynamic context we are considering) or via data annotation. Data transformations are categorised depending on the security property they aim to guarantee:

- *data transformations for confidentiality (or integrity).* The conventional security mechanisms used to protect data are encryption or hashing. With the advent of multi-tenant distributed clouds and collaborative machine learning environments, different approaches have been proposed, such as Attribute-Based Encryption (ABE) [3], Identity-Based Encryption (IBE) [4] or Homomorphic Encryption [5];
- *data transformations for anonymity.* Data anonymization is the general term used to describe the process of hiding identifiable information that may lead to personal identification, so users described by such data remain anonymous. Transformations that can be used includes but are not limited to suppression or generalization of data, masking and encryption, distortion, and swapping.

A taxonomy of data transformations collects all functions used to sanitize the ingested data according to the security property they aim to guarantee. More details are available in D4.1 [1].

2.1.4 The Engine

We implemented the architecture using the Apache Big Data ecosystem of services as depicted in Figure 5. It is composed of three main types of components as follows:

- *Data and Resource Management* components focus on data collection and transformation before their storage for analytics. They are at the basis of data ingestion;
- *Access Control and Audit* components focus on data sanitization and access management;
- *Processing* components focus on the analytics of data collected at ingestion time. More details on these components are available in D4.1.

[4] In the Big Data scenario, “data annotation” and “(meta)tagging” are often used as interchangeable terms. A tag is a label consisting of a key and an optional value that is assigned to a resource.
[5] https://gdpr-info.eu/

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

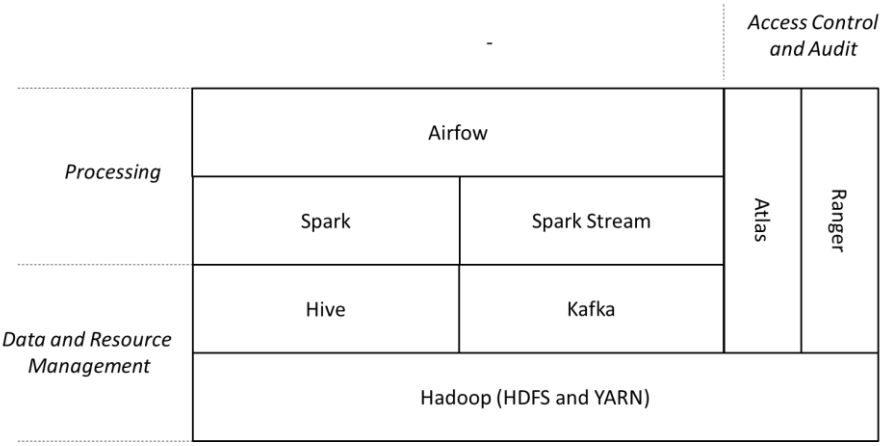

Figure 5: Big Data engine architecture

Figure 6 describes the data ingestion process from a technical standpoint by referring to those technologies and frameworks used at each specific step, from sourcing to storage. We recall that all technologies and frameworks are described in detail in D4.1 [1]. It shows the mapping between our methodology and our engine in the following areas.

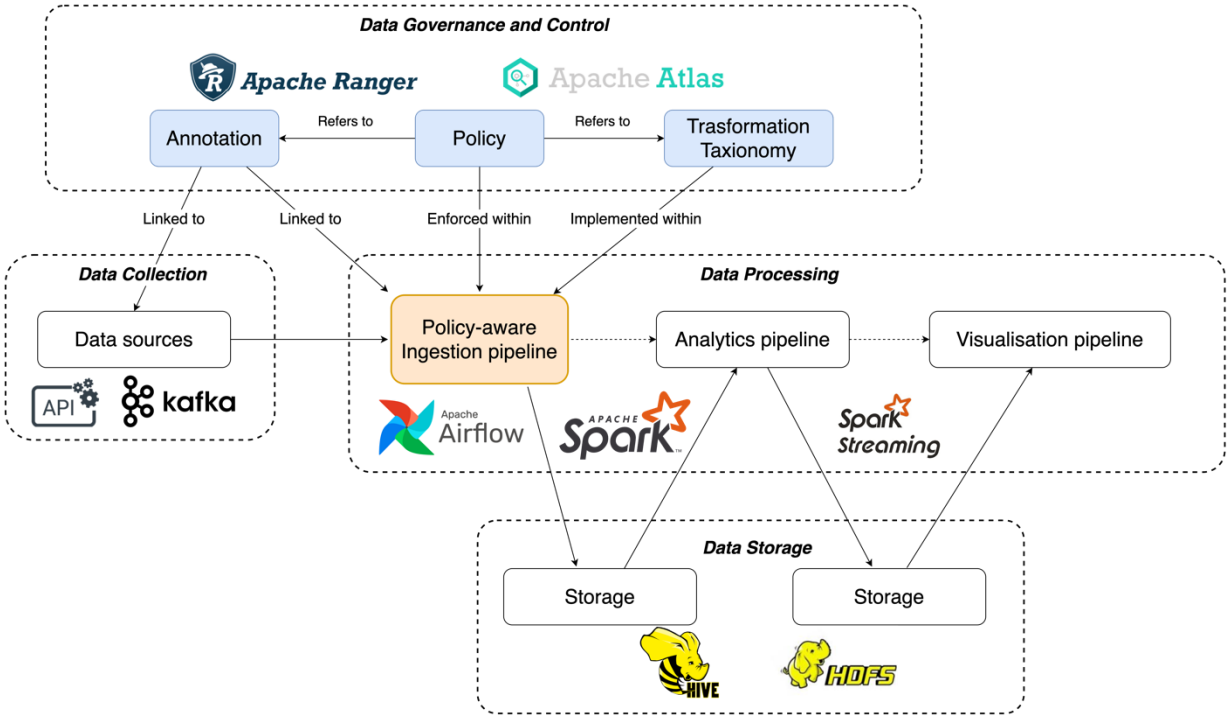

Figure 6: Technologies and frameworks in data ingestion and access control.

Data collection: It supports batch, micro-batch and streaming ingestion of data produced by sources (e.g., sensors) or other providers (e.g., cities). In the case of data ingested in batch form, the data collection process is executed through the source API.

Through these connectors, it is possible to technically input chunks of new data into the Big Data platform. Data may be collected either in an active (i.e., by directly picking up data -- event-based) or passive way (i.e., by a regular data receiving process -- time-based).

In the case of data ingested in streams, a Kafka-based queue mechanism is used. Each data source sends data towards a specific queue, that is, appropriately labelled to distinguish them from each other in terms of meaning, content, and the like.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

Data processing: It includes all the pipelines from ingestion to visualization. In our engine the pipelines are written and executed by Spark or Spark stream depending on the nature of the analysis to be carried out.

Spark can trace the execution of different jobs detailing the execution status. Pipelines are then managed together using Airflow. It can schedule a batch model processing pipeline aimed to update the model when enough data are available on the data lake. It can also schedule execution of the continuous prediction pipeline and synchronise it with visualization pipeline.

In our scenario, Airflow is also used to orchestrate single tasks in the pipeline when needed, for instance, in case of very complex analytics pipelines.

Data governance and control: It contains technologies to support policy-aware ingestion pipelines. More specifically, we adopted Apache Ranger for the definition, assignment, and enforcement of access control policies and Apache Atlas for annotations instrumenting the policies.

Data storage: Once the data are handled via the "policy-aware ingestion pipeline" module, they can be stored in the storage components of the data lake (e.g., Hive or HDFS) and made available for the analytics and visualization pipelines.

In the data storage area, technologies are primarily focused on ensuring data availability, fault tolerance, and reliable distribution of data on the nodes of the cluster. They are indirectly involved in the processing procedure because nodes are normally involved in the processing of the local data; when a reduction is needed, before being processed, the data should however be redistributed or organized.

This tight relation between storage and processing is partially overcome by in-memory processing of Spark, but it still exists when workflows of different pipelines are considered.

2.2 Analytics-time access control for Big Data pipelines

The access control solution in D4.1, summarized in Section 2.1, is extended in this deliverable to define a new access control system that can be performed at each step of the data analytics pipelines rather than only at ingestion time.

Access control is enforced both at ingestion time and before data are given as input to services in the data analytics pipeline. In the following of this section, we introduce:

i) the syntax of the new access control language for Big Data pipelines (Section 2.2.1);
ii) how the access control system is integrated in the Big Data pipelines (Section 2.2.2);
iii) a mechanism to define ad hoc transformations within access control policies, to increase the data governance flexibility and some metrics to analyse the level of manipulation done on the dataset according to the policies (Section 2.2.3).

2.2.1 Access Control Language

The big data scenario adds lots of complexity to data governance, especially from a *data protection* perspective.

For instance, Big Data is highly dependent on cloud-edge computing, which extensively uses multitenancy. Multitenancy permits sharing one instance of infrastructures, platforms, or applications by multiple tenants to optimize costs. This leads to common situations where a service provider offers subscription-based analytics capabilities in the cloud, or a single data lake is accessed by multiple customers.

Thus, having a big data pipeline where data and services belong to different organizations is quite common, and this is a serious risk of potential privacy and security violations.

A Big Data access control solution should then satisfy the following additional requirements that extend the requirements in D4.1:

- [**R1**]: Access control enforcement must protect data during their entire lifecycle. Access control policies must be enforced at each phase of the analytics process, guaranteeing that data are properly protected and shared only to authorized users and for authorized operations.
- [**R2**]: Access control policies must support the specification of dynamic access conditions based on the current coalitions among organizations and their agreement about how to share data and services.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

- [**R3**]: In collaborative Big Data scenarios, where sensitive data are shared among large coalitions of different organizations, access control enforcement should consider both the action to be performed on sensitive data and the coalition's sharing agreement.
- [**R4**]: In collaborative Big Data scenarios, traditional accept-deny enforcement is not always possible/wanted especially in latency-sensitive scenarios. Denied access to data should be modelled as a data preparation job, transforming data according to privileges modelled in access control policies.

The access control language proposed in this document is a refinement of the one in D4.1 and an extension of an attribute-based access control model that offers flexible fine-grained authorization capabilities and exploits XACML [6] notion of obligation to introduce pre-emptive data transformations.

In a collaborative Big Data scenario, pre-emptive data transformations are more suitable than denying access to data. Differently from the approach presented in D4.1, where access control was only performed at ingestion time, the new approach monitors every data processing operation along the whole pipeline (satisfying requirement **R1**).

As in any attribute-based access control model, the key elements of our model are defined as follows:

- A *subject* is a user or the service provider of a job that issues access requests to perform operations on objects. Subjects may have one or more attributes of the form (*aname*, *avalue*) pair, with *aname* the name of the attribute and *avalue* its value. In our case, there is always at least one attribute, specifying the organization the subject belongs to. We use the notation *u.aname* to refer to the value of attribute named *aname*. For short, we use the notation *u@X* to specify user *u* belonging to organization *X*, that is, to express the value of *u.organization*.
- An *object* is any data whose access is governed by the policy. It can be a file (e.g., a video, text file, image, etc.), a database, a table, a column, a raw, or a cell of a table. Like subjects, objects are assigned one or more attributes and have at least one, specifying the organization the object belongs to, expressed with the same notation *o@X* for object *o* belonging to organization *X*.
- An *action* is any job that can be performed within a Big Data platform, ranging from classical atomic operations on a database (e.g., CRUD operations varying depending on the data model) to coarser operations such as an Apache Spark Direct Acyclic Graph (DAG), Hadoop MapReduce, an analytics function call, or an analytics pipeline offered by a different service provider.
- The *environment* is defined by a set of dynamic attributes related to the specific context, such as time of the day, location, IP address, risk level, weather condition, holiday/workday, emergency (satisfying requirement **R2**).

The access control policy is then defined as a *set of policy rules* expressing access conditions based on the key elements and their attributes.

A *policy rule* is a function defined as:

$$\text{policy\_rule}_{obj}(subj, action, env, datatrans) \equiv \texttt{if}\ (cond\_expr == \texttt{true})\ \texttt{then}\ datatrans$$

where *cond_expr* is a propositional Boolean expression built on the composition of mathematical operators (>,<, =, +, -, *, /), logical operators ($\neg, \wedge, \vee$), set operators ($\in, \subset, \subseteq, \supset, \supseteq, \cap, \cup$), and logical quantifiers ($\forall, \exists$) on the resource *obj* and on arguments *subj*, *action*, *env* and their attributes. *datatrans* are the transformation functions, which support pruning, reshaping, encryption/decryption, or anonymization of the resource or part of it as described in Section 2.1.3.

In an attribute-based access control model, policies are limited only by the computational language and the richness of the available attributes. In our case, we exploit the *environment* field of our model to express the dynamic information on actual coalitions (satisfying requirement **R2** and **R3**).

In particular:

- A ***coalition*** *C* is a set of organizations $o \in O$, with *O* the set of the organizations associated with the specific scenario. The *environment* field of our model then contains both *O* and the current coalitions that can dynamically vary.
- A ***coalition agreement*** CoA_C is a set of policy rules associated with a coalition *C* that are added to the security policies of each organization, that is, it is a default set of permissions rules to be applied to all

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

coalition members. Obviously, the rules of a coalition agreement depend on the specific nature of a coalition.

Using the access control model described above, the IMPETUS consortium can be seen as a coalition, with a specific coalition agreement specifying data-specific rules for the different data ingested by the pilot cities.

For example, in the main streets entering Padua city-centre, several CCTVs have been installed to monitor the traffic in real time. These sensors read the car plates and a backend software service checks:

i) if the car meets the current emissions standards (as in many other cities, high polluting vehicles cannot enter the city centre);
ii) if the car insurance is valid/updated;
iii) if the car emissions review has been completed according to the regulations in force.

In case of violation, the software sends an alert to the Local Police's SOC at the municipality of Padua (denoted as `LocalPolDept`) indicating the street number, the plate number, and the car model.

Within the municipality, one operator of the SOC (denoted as *opt@PD*) can access the alert and all data ingested by the sensors, including car plates, and then contact a patrol on the field to stop the car and emit a fine. The rule allowing such an access can be expressed by the following rule:

$$\text{policy\_rule}_{\text{TMSdata}}(opt@PD, READ, env, -) \equiv \texttt{if}\ (opt.Dept == \texttt{LocPolDept})\ \texttt{then}\ give\_access$$

As another example, the Physical Threat Intelligence (PTI) tool at CINI (denoted as *PTI@CINI*), one of the tools of the IMPETUS platform, can access traffic data to find anomalies in the traffic volume. We note that, following GDPR requirements, the traffic data must be anonymised before any access is granted, that is, the car plate numbers must be hashed.

The rule allowing access to the traffic data by PTI tool after anonymization (*PlateAnonym*) can be expressed by the following rule:

$$\text{policy\_rule}_{\text{TMSdata}}(PTI@CINI, READ, env, PlateAnonym) \equiv \texttt{Do}\ \ PlateAnonym\ \texttt{then}\ give\_access$$

2.2.2 Pipeline Execution and Policy Enforcement

Figure 7 shows how the access control model presented in the previous section is integrated in the Big Data processing system in Figure 1. The new solution, differently from the one presented in D4.1, performs policy enforcement during the entire data lifecycle, not only at ingestion time.

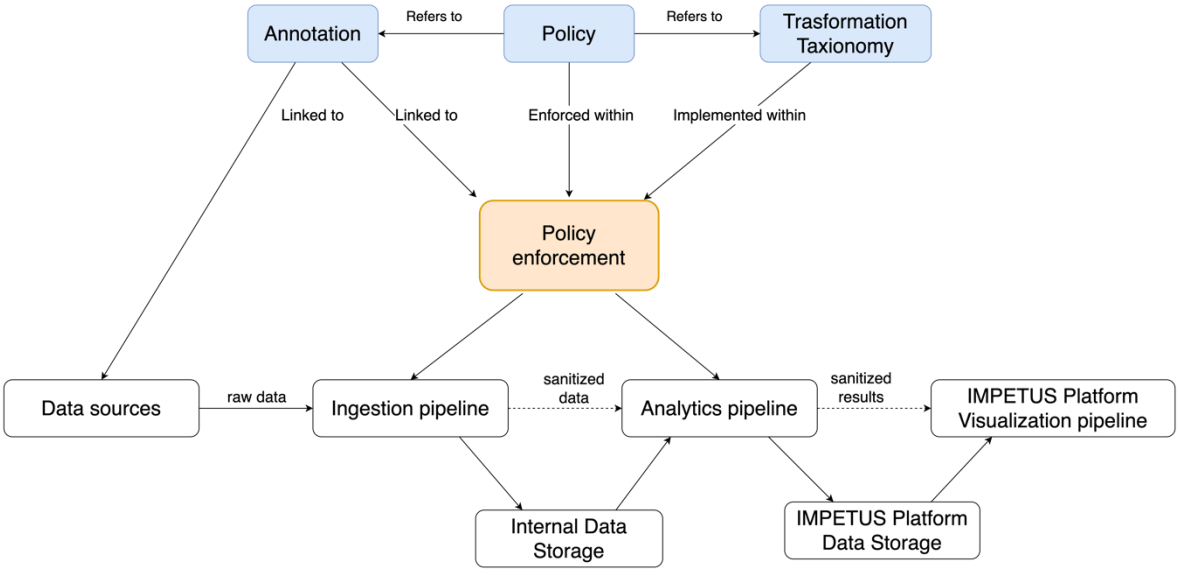

Figure 7: Policy Enforcement within the pipeline execution

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

Data protection is guaranteed by means of access control enforcement and data transformations prior to feed data to any processing tasks, from data ingestion to data analytics. As depicted in Figure 7, the execution of both ingestion and analytics pipelines is mediated by policy enforcement.

2.2.3 Ad Hoc Data Transformations and Data Quality Metrics

As already discussed in D4.1, Apache Ranger supports fine-grained authorization using resource-based and tag-based policies. In Apache Ranger, the policy specifies:

i) the resources a policy is applied to (e.g., HDFS files and directories, Hive databases, tables, and columns, HBase tables, column-families, and columns);
ii) the access conditions for specific users and groups.

In addition, Apache Ranger provides dynamic resource-based column masking capabilities to protect sensitive data. Data-masking policies enable to dynamically mask sensitive data as users access the data. It is possible to set filters for specific users, groups, and conditions.

The masking default types provided by Ranger are:

- **redact**, masking all alphabetic characters with "x" and all numeric characters with "n";
- **partial mask: show last 4**, showing only the last four characters of the cell value;
- **partial mask: show first 4**, show only the first four characters of the cell value;
- **hash**, replacing all characters with a hash of entire cell value;
- **nullify**, replace all characters with a NULL value;
- **unmasked (retain original value)**, no masking is applied;
- **date: show only year**, showing only the year portion of a date string and default the month and day to 01/01.

Data transformations are an essential part of the access control system. They can be applied both to the whole resource or part of it (i.e., databases, tables, columns and cells), increasing data governance flexibility. They provide additional functions, from pruning and reshaping to encrypting/decrypting, depending on the specific policies and data annotations.

D4.2 extends the solution in D4.1 with a mechanism fully integrated with Apache Ranger that supports the specification of ad hoc data transformations. Ad hoc data transformations implement custom data manipulations and support the definition of access control policies that address specific needs emerging in the considered scenario.

Hive provides several manipulation functions that the user can extend to define his own User Defined Functions (UDFs). The functions must be packaged as a JAR (Java Archive) file and registered in Hive (this can be done by uploading the JAR file to the host hive machine or HDFS, and issuing a specific query). Once the UDF has been registered in the system, it can be used in the query phase.

The standard UDF takes a single value as input and provides a single value as outputs, that is, the UDF is called for each row of the query result. All functions, both those provided by the system and UDFs, can be used as custom masking option in Ranger. Figure 8 shows an example of UDF Java class that takes a string as input and returns a lower case version of it as output; the class needs to be packed in a JAR file to be registered in Hive (Figure 9) and then used in Ranger (Figure 10).

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

```java
package com.microsoft.examples;

import org.apache.hadoop.hive.ql.exec.Description;
import org.apache.hadoop.hive.ql.exec.UDF;
import org.apache.hadoop.io.*;

@Description(
    name="LowerUDF",
    value="returns a lower case version of the input string.",
    extended="select LowerUDF(UPPERCASENAME) from table limit 10;"
)

public class LowerUDF extends UDF {

    public String evaluate(String input) {
        if(input == null)
            return null;

        return input.toLowerCase();
    }

}
```

Figure 8: Example of a Java UDF

```
CREATE FUNCTION custom_lower
AS 'com.company.hiveudf.custom_lower'
USING JAR
'hdfs:///warehouse/tablespace/managed/LowerUDF-1.0-SNAPSHOT.jar';
```

Figure 9: Example of registration of a UDF

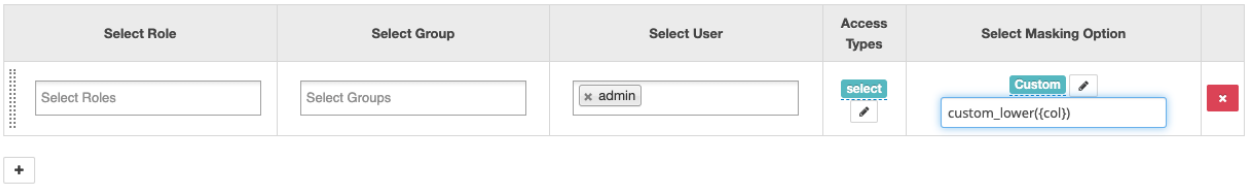

Figure 10: Example of policy using the UDF, {col} is a placeholder, it represents the column value targeted by the policy.

Ad hoc transformations can be further extended to analyse and quantify the level of manipulation done on the dataset according to the policies. The impact on the analytics accuracy can be then modeled by keeping track of the data quality degradation (if any) occurred due to the enforced policies.

Data quality degradation can be calculated according to three main metrics defined as follows:

i) *rate of manipulation*, as the rate between the manipulated data and the total amount of data. We note that this rate can take the specific manipulation into account (e.g., data deletion vs data generalization);
ii) *weighted rate of manipulation*, as the rate between the manipulated data and the total amount of data, weighted on the importance of the specific datapoint or data feature. For instance, the rate of manipulation can be weighted according to a feature ranking executed on the specific analytics;

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

iii) statistical manipulation, as the impact a data manipulation has on the statistical distribution of the features in the dataset.

The selected metrics have to quantify the utility of anonymized datasets, by measuring the loss of information caused by sanitization. If multiple policies satisfy a privacy criterion, utility metrics can be an appropriate instrument to select the most useful policy to be enforced.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

3 Data Analytics

This chapter presents an overview of the data analytics pipeline to be performed on data coming from the pilot cities, that is, anomaly detection and event classification. The data analytics pipeline introduced in this deliverable extends and refines the content in D4.1. It also introduces the methodology behind the algorithm used to perform event classification.

3.1 Background

In this section, some background definitions are provided, to fully understand the important notions underlying the methods.

The neighbourhood $N(p)$ of an object p is defined as the set of objects whose distance from p, according to a given measure, is within a threshold eps. Formally:

$$N(p) = \{q\ s.t.\ dist(p, q) < eps\}$$

An object p is called *core object* with respect to eps and $\min P\ ts$ if it has at least $\min P\ ts$ objects in its neighbourhood $N(p)$. Therefore, p is a *core object* if $|N(p)| \geq \min P\ ts$.

An object p is *directly density-reachable* from an object q if $p \in N(q)$ and q is a *core object*.

An object p_n is *density-reachable* from an object p_1 if there exists a chain of objects $p_1, p_2, \ldots, p_n$, such that for each pair of objects (p_i, p_{i+1}), p_{i+1} is *directly density-reachable* from p_i, w.r.t. eps and $\min P\ ts$.

An object p is *density-connected* to an object q if there exists an object o, such that both p and q are *density-reachable* from o w.r.t. eps and $\min P\ ts$.

A *cluster* is a non-empty subset of objects, where each pair of objects (p, q) is *density-connected*.

Non-core objects belonging to at least one cluster are called *border objects*, whereas objects not belonging to any cluster are named *noise objects*.

3.2 Anomaly detection

Anomaly detection is a machine learning task that refers to the problem of identifying data that do not conform to patterns observed in historical data. These patterns represent the expected behaviour in normal conditions. Therefore, anomaly detection is usually performed through a data-driven algorithm to construct a model that will be able to detect a specific measurement/object/instance/observation as anomalous with respect to the historical data already seen.

Anomaly detection is a very general task and is implemented in several real-world applications, such as the detection of anomalies in car traffic, air pollution, pedestrian trajectories, and so on [7].

For ML-based anomaly detection algorithms, it is possible to visualise what happens geometrically to the considered observations under analysis. Figure 11 illustrates from a geometrical point of view how the anomalies could be represented in a 2-dimensional data set. Specifically, there are two normal regions, N_1 and N_2, since most observations lie in these regions. The points that are sufficiently far away from the regions, e.g., points o_1 and o_2, and points in region o_3, are considered as anomalies.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

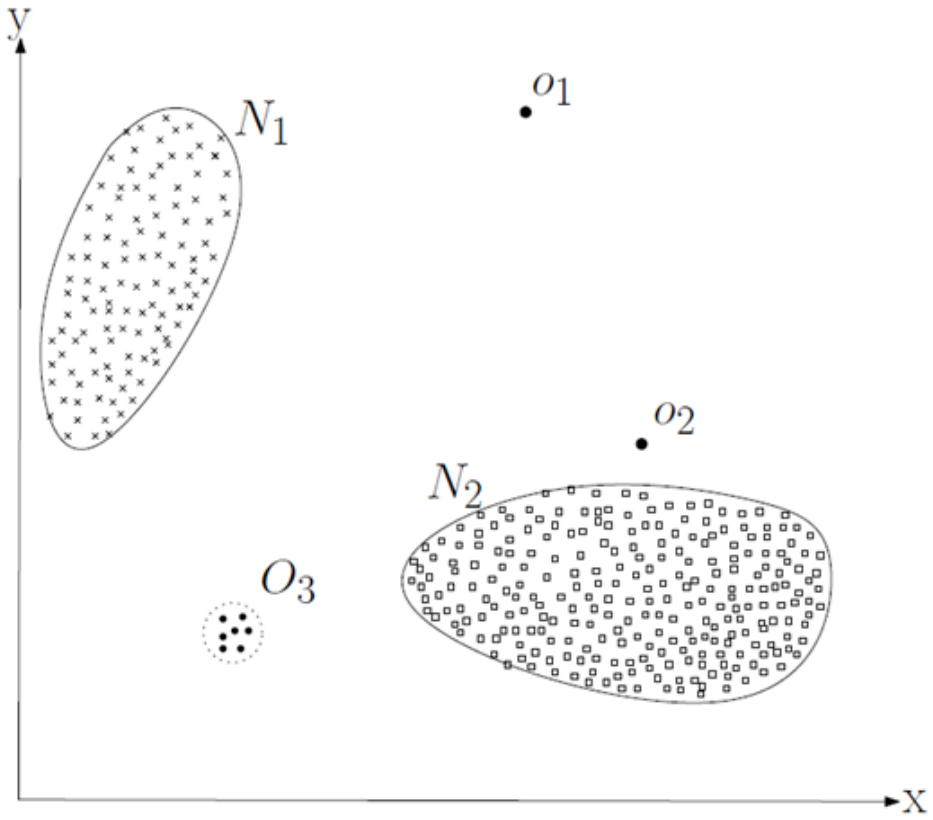

Figure 11: Anomalies in a 2-dimensional data set.

3.2.1 Spark-GHSOM

This section provides an overview of the Spark-GHSOM algorithm.

Spark-GHSOM is a machine learning algorithm supposed to automatically identify anomalies from geo-referenced sensors data.

Anomaly detection is usually performed by training a model (henceforth anomaly detector) that is capable of catching anomalies from data. For the purposes of the IMPETUS project, three possible phases to train an accurate anomaly detector were identified: the *i)* initial phase, *ii)* update phase, and *iii)* identification phase.

In the initial phase, a weak predictive model is trained, that represents a “coin-flip” function with low predictive capability. This model needs to be trained to improve the performance. To this aim, a batch-learning approach is considered, that is, the model is trained through a data batch (a full matrix) to overcome the weakness of the starting function. The algorithm in this phase is in the "initial state".

After the first stage, the anomaly detector performs better than the previous starting function and it could be ready to take the sensor data as input to identify possible anomalies. However, since the distributions could vary also in normal cases (e.g., considering air pollution, it is normal that during the weekend CO2 in the air could be lower than during working days), an additional update phase has been introduced, that aims to enable further training of the anomaly detector with the possibility to reuse the previously learned models for future sensor data analyses. This phase allows to store a pre-trained predictive model for anomaly detection as a file that could be subsequently plugged into the detector. Furthermore, to avoid training the anomaly detector from scratch, at this stage a previous pre-trained version of the model can be further trained with a possible reduction of the training time. The algorithm in this phase is in the "update state".

In the identification phase, the anomaly detector is ready to process sensor data to detect possible anomalies because it already analysed historical data. The knowledge of historical data is recorded and encoded within the model’s neuron vectors. At this phase, the anomaly detector could be placed in production to work actively with the real scenario data, since it shows better performance due to greater stability. A general schema of the anomaly detection is shown in Figure 12.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

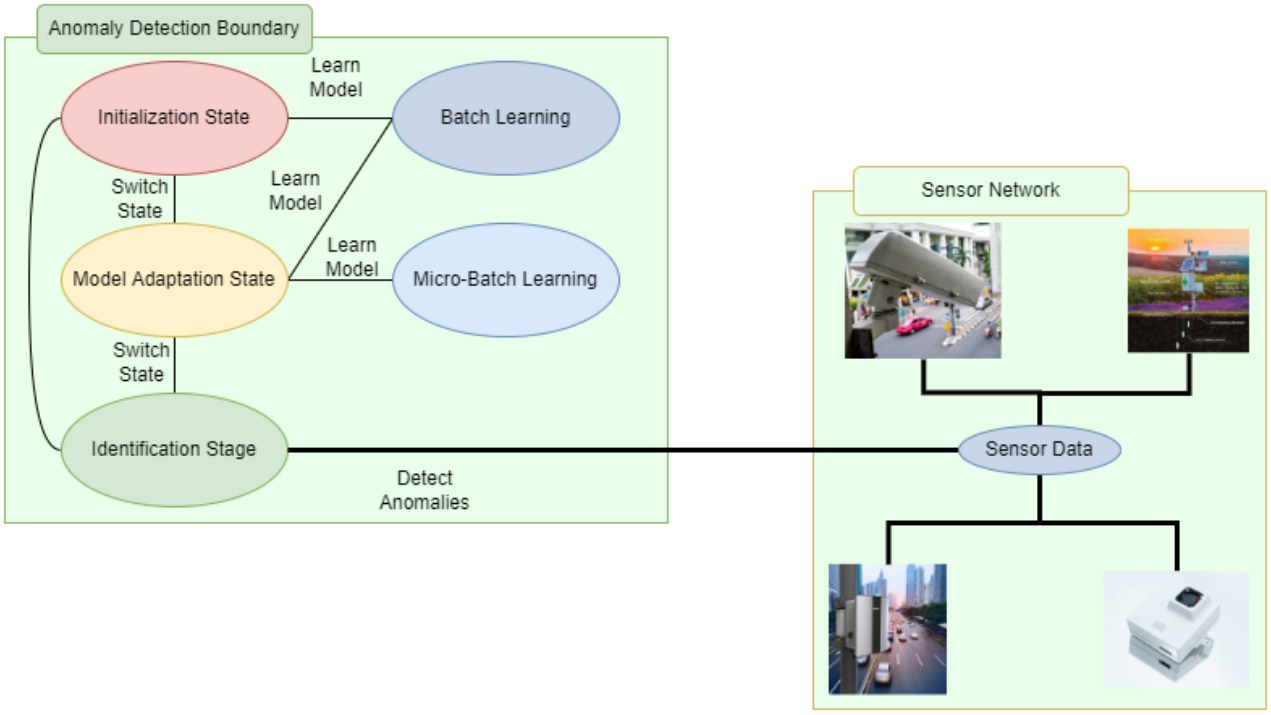

Figure 12: Anomaly detection for the identification of possible anomalies from sensor data.

Spark-GHSOM can handle the available sensor data containing spatial (e.g., GPS coordinates) and temporal information (e.g., timestamp) and a set of descriptive variables that are acquired by the specific sensor for the monitoring of the city. For instance, independently from the type of the sensor (e.g., traffic or air pollution), the anomaly detector acts with the same approach. Indeed, the anomaly detector works with values usually indicating the level of different things: pedestrians' concentration, traffic level, temperature, humidity, level of PM2.5, level of CO2, and so forth, that are automatically captured and transmitted by the sensor network. Therefore, in the real situation, the anomaly detector will analyse the data coming from different sensors and it will be able to judge the data as a normal or anomalous.

The anomaly detector can process different kinds of output depending on the level of detail. The simplest approach provides feedback for the current novel data with respect to the previously seen data in the form of a Boolean response. This kind of output represents the most common among the anomaly detectors and it could support raising an alert if the response is equal to "anomaly" (see Figure 13 as an example).

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

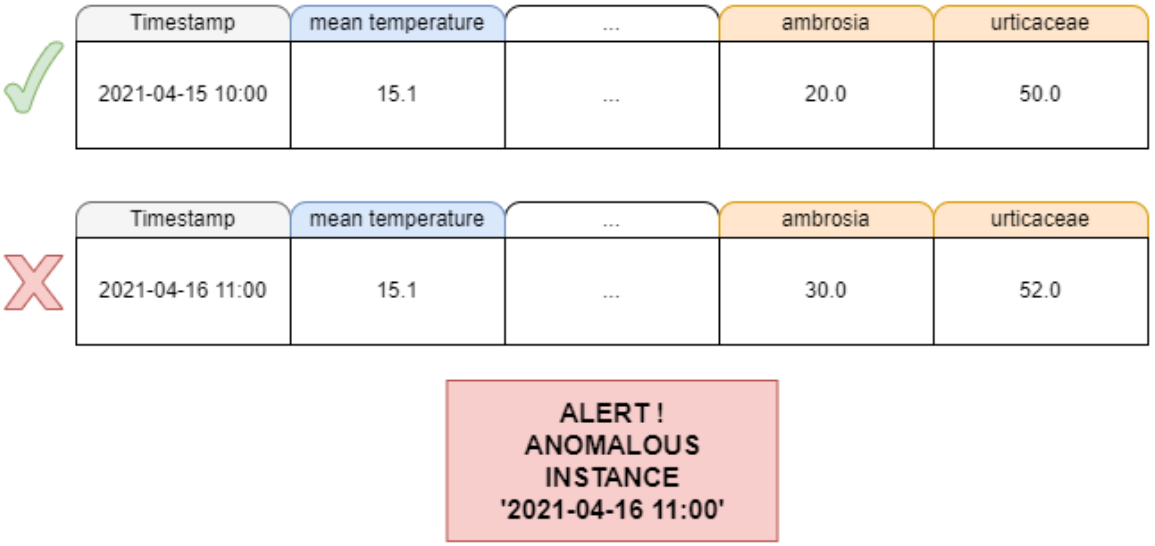

Figure 13: When an anomaly is detected, the system raises an alert indicating the timestamp and GPS coordinates.

The described approach is simple to handle and transmits the prediction as a binary response (e.g., anomaly/normal, 0/1, true/false). However, it makes difficult for the end user interpreting the raised alert/anomaly. Therefore, a more informative approach could be considered by combining the previous one with a ranking of the descriptive variables with their importance, indicating the contribution to catch the anomaly (see Figure 14). This characteristic of Spark-GHSOM concurs to improve the interpretability of the output provided by the tool, so that the user can understand the reason of the decision (anomalous/not anomalous) suggested by the system [8].

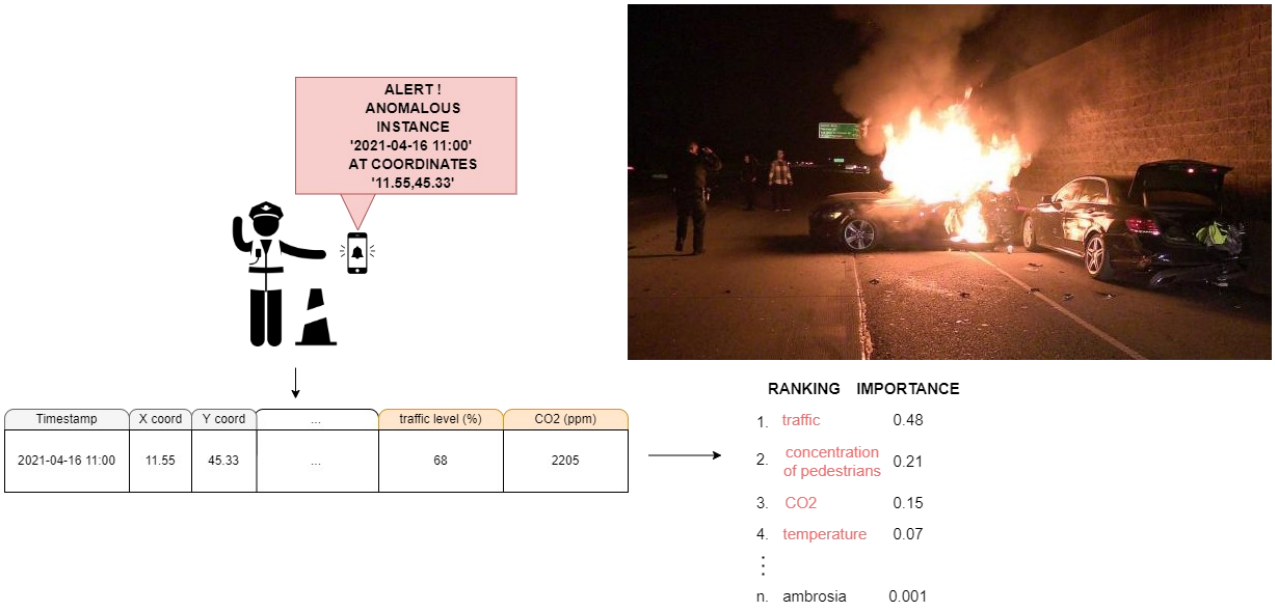

Figure 14: When an anomaly is detected, the system raises an alert and provides a feature ranking according to the features importance in detecting the anomaly.

3.3 Event classification

This section provides an overview of the event classification task where we consider the data as a stream. Therefore, we extended the algorithm DENCAST [9] to work properly after parameter tuning.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

Starting from a predefined set of threats (e.g., fire, car accident, attack with guns), similarly to the anomaly detection task, the event classification task aims to classify the current unlabelled sensor data under analysis as a particular threat or as a normal case. Therefore, in addition to the anomaly detector, the event classifier would be able to also indicate the type of threat under analysis.

For this task, similarly to anomaly detection, a training phase and an event identification phase are foreseen. The main difference with the anomaly detection task is that usually the classification of an instance is guided by the learning of a predictive model in a supervised setting. This means that the data set used for the training of the model must be annotated by describing the possible threats for the real scenario. However, this could be demanding to obtain, since it is necessary to hire data annotators to make the data fall into specific categories.

To overcome this problem, unsupervised algorithms could be also considered for the classification task as for the anomaly detection. These algorithms are usually less accurate than the supervised ones since they exploit less informative data avoiding considering predefined classes.

3.3.1 DENCAST

This section provides a more detailed description of the DENCAST tool, together with the extensions designed.

The method presented in D4.1 [1], called DENCAST, is able to perform the event classification task. It represents a novel distributed algorithm implemented in Apache Spark, that performs density-based clustering and can exploit the identified clusters to solve both regression and classification tasks. However, as already mentioned in this section, the classification of an instance is typically guided by the learning of a predictive model in a supervised manner, using labelled data, which should also contain the type of threat being observed. In a streaming context, this could be demanding to obtain.

For this reason, in this deliverable, an extension of the DENCAST model is proposed, that is capable of: *i)* conducting unsupervised learning of the model, (i.e., without manually annotated data), by identifying density-based clusters; *ii)* working in a streaming context and *iii)* making the prediction step through the use of feature ranking, a method designed to improve the interpretability of the output provided by the model.

The following describes the key features of the DENCAST model:

- it relies on the neighbourhood graph, a low-dimensional representation of the observed objects. In this way, the model needs only object identifiers and their neighbourhood relationships to work properly, therefore using a simpler representation of the objects, requiring limited space resources. The neighbourhood graph is efficiently built from high-dimensional data through the use of the LSH method;

- the tool is implemented using the Apache Spark framework, for this reason it is fully distributed. As a consequence, DENCAST can analyse large scale datasets without incurring computational bottlenecks;

- the identified density-based clusters can be exploited to classify possible threats. If the real description of what a cluster is representing is not present (i.e., when the ground truth is missing), then a cluster is described through the ranking of the components of its centroid vector that most contributed to build the cluster. This aspect of the tool allows to improve the interpretability of the output provided by the model.

Given the dataset D consisting of n unlabelled objects, represented by d descriptive attributes, a distributed variant of LSH [10] is applied to identify an approximate neighbourhood graph. The obtained graph is composed of a node for each unlabelled object in D and an undirected edge for each pair of nodes $< u, v >$, that appear similar enough according to the low-dimensional representation of the objects, obtained after the application of LSH, and to a threshold called $\min Sim$.

The method uses the neighbourhood graph, which can be considered an approximate low-dimensional representation of the input objects and their distances, instead of using objects represented in their original feature space. This design choice allows to significantly reduce the space and the time needed for the identification of the neighbours of each node and identification of core objects steps.

The proposed method for density-based clustering then maps each unlabelled node to a cluster, by propagating cluster identifiers from core objects through their neighbours. This approach is iterative and thus requires a

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

stopping criterion based on a threshold, called $labelChangeRate$, with the aim to avoid unnecessary iterations, which would lead to slight changes during the cluster assignment step.

Similar to existing density-based clustering algorithms, the method is able to discard objects that are considered noise or outliers, i.e., objects deemed too distant in the feature space from the identified clusters.

Subsequently, the method re-associates each node in the obtained clusters with the original features of the corresponding objects. The obtained clusters are finally used to identify potential threats from the data received by the city sensors, in the form of streaming.

The neighbourhood graph constitutes a low-dimensional representation of the unlabelled objects under analysis.

Formally, the neighbourhood graph of an object p is defined as the set of objects whose distance from p, according to a given measure, is within the threshold eps. As already described in Section 3.1, the neighbourhood graph of an object p is defined by the following equation:

$$N(p) = \{q\ s.t.\ dist(p\ ,\ q)\ <\ eps\}$$

In this way, the model needs only object identifiers and their neighbourhood relationships to work properly, therefore using a simpler representation of the objects, requiring limited space resources. The neighbourhood graph is efficiently built from high-dimensional data through the use of the LSH method.

As already described in D4.1 [1], in density-based clustering “dense” clusters of points are identified, making possible to learn arbitrarily shaped clusters and thus identify outliers in the data collection.

A popular density-based clustering algorithm, called DBSCAN [10], identifies a cluster starting from an arbitrary *core object* and retrieving all the objects that result *density-reachable* from it, with respect to eps and $\min Pts$ parameters.

The novelty of the proposed approach relies on the formulation of a density-based clustering method that performs the exploration of the neighbourhood graph in a fully distributed way. This operation is performed by identifying all the reachable nodes of all the *core objects* simultaneously, by propagating the cluster assignment of all the *core objects* to their neighbours, until the cluster assignment appears stable enough. Figure 15 provides a graphical representation of the proposed clustering approach.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

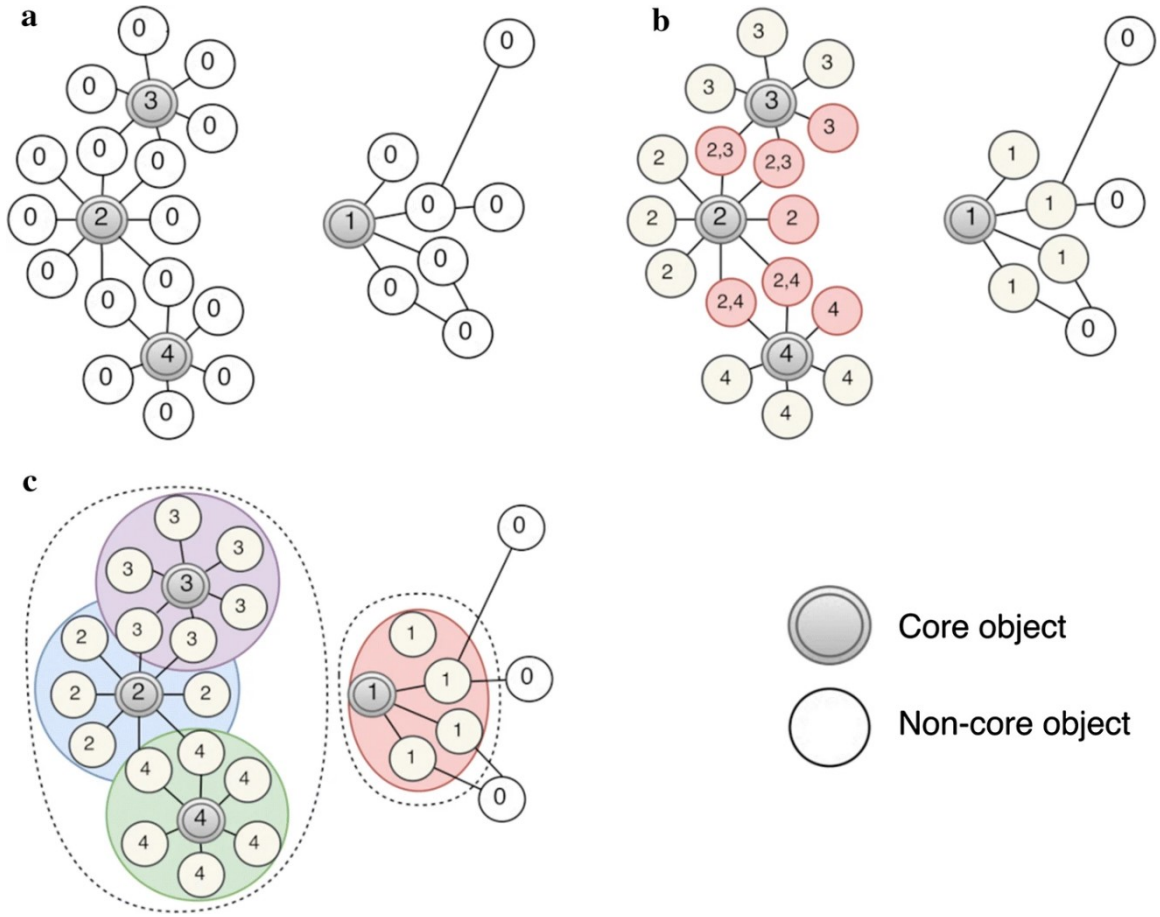

Figure 15: Graphic representation of the clustering approach. *i)* Initialization of cluster identifiers *ii)* Core objects propagate their cluster ID to their neighbours. Nodes highlighted in red receive multiple messages. *iii)* Multiple cluster IDs received by the same node are aggregated. Dotted lines represent the final clustering result, after some iterations [9].

Once the cluster creation step has been completed, for each obtained cluster the centroid vector is computed, i.e., the vector placed in a central position with respect to the set of vectors in the cluster.

Let C_k be the set of n vectors in the k -th cluster, with each vector v_i being composed of d components. Formally:

$$C_k = \{v_1 , \ \dots , \ v_i \ , \ \dots , \ v_n\}$$

$$v_i = [v_{i,1} , \ \dots , \ v_{i,d}]$$

The centroid c_k of the k -th cluster is the vector:

$$c_k \ = \ [c_{k\,,1} , \ \dots , \ c_{k\,,j} , \ \dots , \ c_{k\,,d}]$$

such that each component $c_{k\,,j}$ is computed as following:

$$c_{k\,,j} = \frac{1}{n}\sum_{i=1}^{n} v_{i\,,j}$$

The method only uses the centroid as the vector representing the cluster. The choice to only consider the centroid as the descriptor of the cluster is motivated by the fact that, in order to handle data provided in stream form, it is necessary to simplify the representation of the learned model.

During this phase, each test instance provided as input to the tool is associated with one of the clusters obtained above, i.e., the cluster whose centroid vector is closest to the test instance provided as input. Formally:

Let t be the test instance provided as input to the tool.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

$$t = [t_1, \ldots, t_d]$$

The test instance is associated to the centroid vector c_{sim}, such that:

$$dist(t, c_{sim}) = \arg\min([dist(t, c_1), \ldots, dist(t, c_n)])$$

where n is the number of clusters identified and $dist$ defines a vector distance measure (e.g., the Euclidean distance).

In an unsupervised setting, the prediction of the test instance is provided by the feature ranking vector, in the following form:

$$fr = [f_1, \ldots, f_i, \ldots, f_d]$$

where each component f_i is computed as the difference between the i-th component of vectors t and c_{sim}. Formally:

$$f_i = |t_i - c_{sim,i}|$$

finally, the vector fr is reordered such that each component $f_i \leq f_{i+1}$.

At this point, the closest centroid to the test instance, i.e., the vector c_{sim}, is updated. This operation is necessary to make the centroid more similar to the input vector. The updated centroid vector c_u occurs in the following form:

$$c_u = [c_{u,1}, \ldots, c_{u,i}, \ldots, c_{u,d}]$$

where each component $c_{u,i}$ is defined as a linear combination between the components $c_{sim,i}$ and t_i. Formally:

$$c_{u,i} = c_{sim,i} \cdot (1 - w) + t_i \cdot w$$

where w defines the weight of the linear combination, equal to $|C_{sim}|^{-1}$, the set of the identified centroids.

The experiments were conducted to evaluate the performance of DENCAST on different data: *PVItaly* data on energy production, *PVNREL* simulated photovoltaic data, *LightSource* solar energy production, *WindNREL* measurements of wind power plants, *bike sharing* data, data from *51 kW DC rooftop photovoltaic installations*. The effectiveness of Dencast predictive power was documented within the original manuscript. Here we report a brief summary.

The results in Table 3 show the average Root Mean Squared Error (RMSE) obtained by DENCAST and by all the considered competitor systems. The Multi-Target (MT) results are not available for the competitors Linear Regression (LR) and Isotonic regression (ISO). Additionally, ISO was not able to provide the results within 20 days of execution for some configurations of the PVItaly and PVNREL datasets. The results obtained by ISO were also poor in terms of RMSE and almost in line with the results obtained by the baseline (AVG). From the results obtained with different sizes of the training set we can observe that, in general, there is no significant variation. An important exception resides in the results obtained by AutoRegressive Integrated Moving Average (ARIMA), which leads to more errors when the size of the training set increases. This behaviour is reasonable since ARIMA only observes the time series described by the target variables.

Therefore, its predictions are negatively affected by objects that are too distant, in terms of timestamp, from the unlabeled objects in the test set. On the contrary, in some configurations, other methods took advantage of larger training sets. An example can be seen in the Bike sharing dataset (single-target setting), for which DENCAST obtains the best results only with the largest training set (90 days).

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

Table 3: Forecasting results in terms of average RMSE

Single-target (ST)			Multi-target (MT)			Avg MT	
30 days	**60 days**	**90 days**	**30 days**	**60 days**	**90 days**	**Improv (%)**	
PVItaly							
DENCAST	*0.1416*	*0.1384*	*0.1416*	*0.1002*	*0.103*	*0.095*	*29.2*
K-means	0.1471	0.1474	0.1469	0.1803	0.1744	0.1726	− 19.5
ARIMA	0.1508	0.1704	0.1925	0.1508	0.1704	0.1925	0
AVG	0.2032	0.2058	0.2065	0.249	0.249	0.249	− 21.4
LR	0.1516	0.1521	0.1532	N/A	N/A	N/A	N/A
ISO	0.2026	0.2	–	N/A	N/A	N/A	N/A
LSTM	0.2296	0.2296	0.2296	0.228	0.228	0.228	0.7
PVNREL							
DENCAST	*0.1519*	*0.1535*	*0.1529*	*0.114*	*0.1166*	*0.1193*	*23.7*
K-means	0.2366	0.2385	0.2397	0.2337	0.2198	0.215	4.8
ARIMA	0.2736	0.2843	0.3001	0.2736	0.2843	0.3001	0
AVG	0.2635	0.264	0.2647	0.3324	0.3324	0.3324	− 25.9
LR	0.2265	0.2274	0.2284	N/A	N/A	N/A	N/A
ISO	0.2621	–	–	N/A	N/A	N/A	N/A
LSTM	0.3367	0.3367	0.3367	0.2867	0.2867	0.2867	14.85
LightSource							
DENCAST	0.168	0.1618	0.168	*0.1222*	*0.1225*	*0.1263*	25.45
K-means	*0.1338*	*0.138*	*0.1332*	0.1658	0.1668	0.1672	– 23.43
ARIMA	0.1596	0.1729	0.192	0.1596	0.1729	0.192	0
AVG	0.1891	0.191	0.194	0.1267	0.1299	0.1355	*31.71*
LR	0.1651	0.1667	0.1687	N/A	N/A	N/A	N/A
ISO	0.1989	0.1977	0.1968	N/A	N/A	N/A	N/A
LSTM	0.2027	0.2027	0.2027	0.2123	0.2123	0.2123	− 4.73
WindNREL							
DENCAST	*0.2992*	*0.2813*	*0.2853*	0.3169	0.322	0.3343	− 12.5
K-means	0.3763	0.3844	0.3929	*0.2037*	*0.1761*	*0.1682*	*53.1*
ARIMA	0.3131	0.346	0.3757	0.3131	0.346	0.3757	0
AVG	0.3263	0.3452	0.348	0.4939	0.4939	0.4939	− 45.5
LR	0.3072	0.3146	0.3226	N/A	N/A	N/A	N/A
ISO	0.3494	0.3517	0.3627	N/A	N/A	N/A	N/A
LSTM	0.4858	0.4858	0.4858	0.3913	0.3913	0.3913	19.45
Bike sharing							
DENCAST	0.1117	0.1114	*0.1089*	*0.0848*	*0.0894*	*0.0926*	*19.6*
K-means	*0.1096*	*0.1096*	0.1136	0.2335	0.2301	0.2244	− 87.4
ARIMA	0.1646	0.1739	0.224	0.1646	0.1739	0.224	0%
AVG	0.1625	0.1638	0.1656	0.0926	0.0957	0.0992	41.6
LR	0.126	0.1278	0.1308	N/A	N/A	N/A	N/A
ISO	0.1759	0.1905	0.1996	N/A	N/A	N/A	N/A
LSTM	0.2424	0.2424	0.2424	0.2144	0.2155	0.2155	11.52
Photovoltaic installations							
DENCAST	*0.1263*	*0.128*	*0.126*	0.1205	0.132	0.1196	2.2
K-means	0.1408	0.145	0.1485	0.2128	0.2047	0.2059	− 43.7
ARIMA	0.1886	0.21	0.2263	0.1886	0.21	0.2263	0.00%
AVG	0.1985	0.2017	0.2051	0.2246	0.2246	0.2246	− 11.3
LR	0.144	0.1485	0.1538	N/A	N/A	N/A	N/A
ISO	0.4521	0.4388	0.4218	N/A	N/A	N/A	N/A
LSTM	0.1972	0.198	0.1975	*0.1157*	*0.1157*	*0.1157*	41.43

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

Focusing on the two different predictive settings (MT vs. ST), we observe that the MT setting provides advantages in most cases. In some datasets such a difference is significant. It is noteworthy that DENCAST is the system that benefits most from the MT setting (Table 3, last column). This result confirms that it is reasonable in many domains to combine the MT setting (which takes into account dependency between the values of the same time series), with a density-based predictive clustering solution.

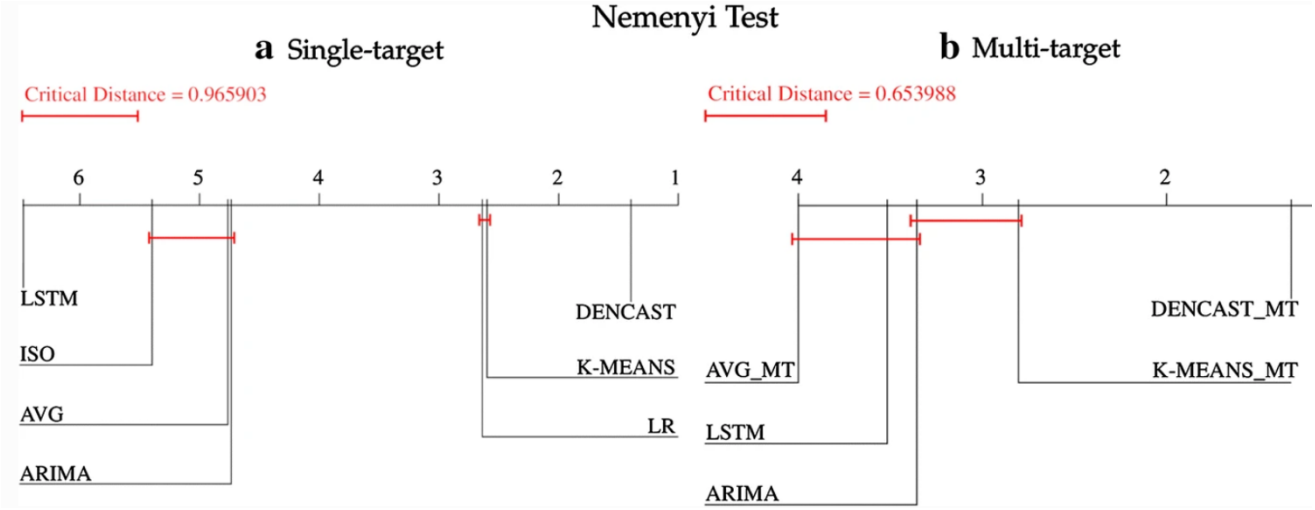

Figure 16: Nemenyi test for a single-target and b multi-target settings. If the distance between methods is less than the critical distance (at p-value = 0.05), there is no statistically significant difference between them.

Dencast resulted effective also for anomaly detection task for the IMPETUS data provided. In Figure 17, the methodological extension is illustrated.

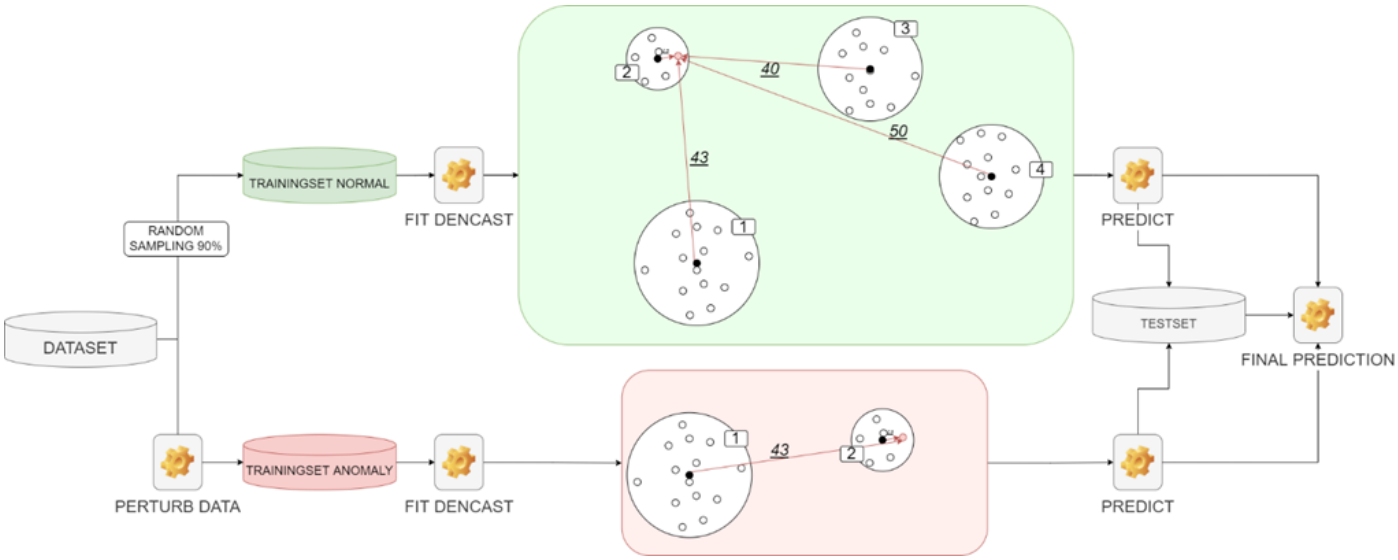

Figure 17: Dencast extension for anomaly detection on IMPETUS data. Two models were trained to discriminate among the normal and anomalous activities.

In this extension, Dencast learns a clustering model for each class (normal/anomaly). The models are then used as one-class classifiers in order to identify how much a new instance deviates from each class. In this way, it is possible to understand *i)* how much a new instance deviates from the "normal" instances and *ii)* if a new instance is closer to normal instances or closer to anomaly instances. In the former, only one model – that for the normal class – is used and the algorithm works in an unsupervised way. In the latter, both models are necessary and the algorithm works in a supervised learning setting.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

4 Integration

This chapter presents our Big Data system as the complete integration of the approaches presented in Chapter 2 (for data management) and Chapter 3 (for data analytics) at three different layers (Figure 18): *i)* data source ingestion; *ii)* data management and data analytics integration; *iii)* integration with the IMPETUS platform.

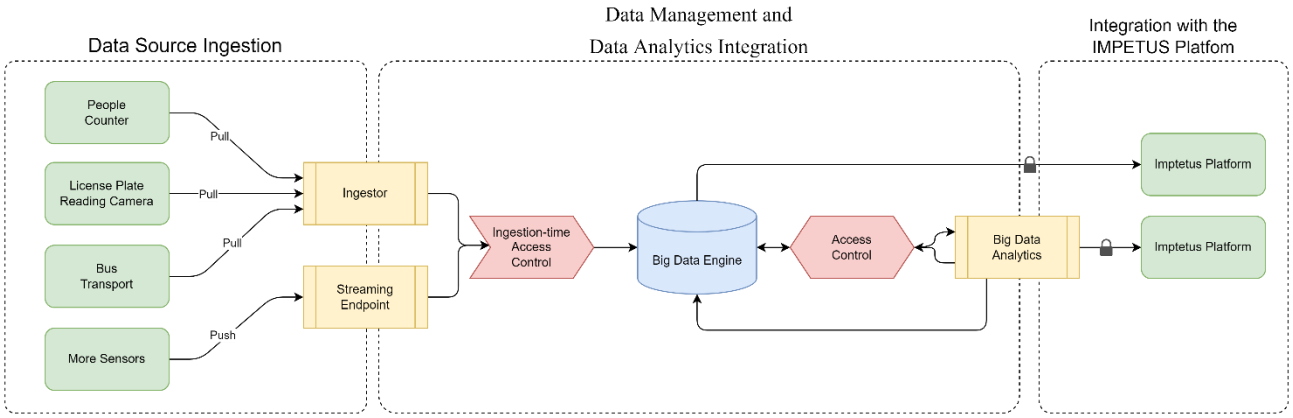

Figure 18: Integrated Big Data platform

Figure 19 depicts the flow implemented by the proposed Big Data platform with reference to our solution for data management and data analytics. We note that black arrows in Figure 19 refers to service executions, while dashed arrows refers to data transfer. Data producers (i.e., IMPETUS cities) connect their data sources to our Big Data platform using a data source ingestion service. This service sends raw data to the ingestion-time access control that enforces policies according to the targets of the ingestion process.
Sanitized data are produced as a transformation of raw data according to policy transformations and later stored in a data repository. IMPETUS services/users request the execution of an analytics pipeline. Before executing it, the access control system retrieves relevant sanitized data from the sanitized data repository and enforces policies according to the pair (service/user, pipeline). Analytics data are produced as a transformation of sanitized data according to policy transformation and later stored in a data repository of analytics data. Analytics pipeline is finally executed on analytics data.

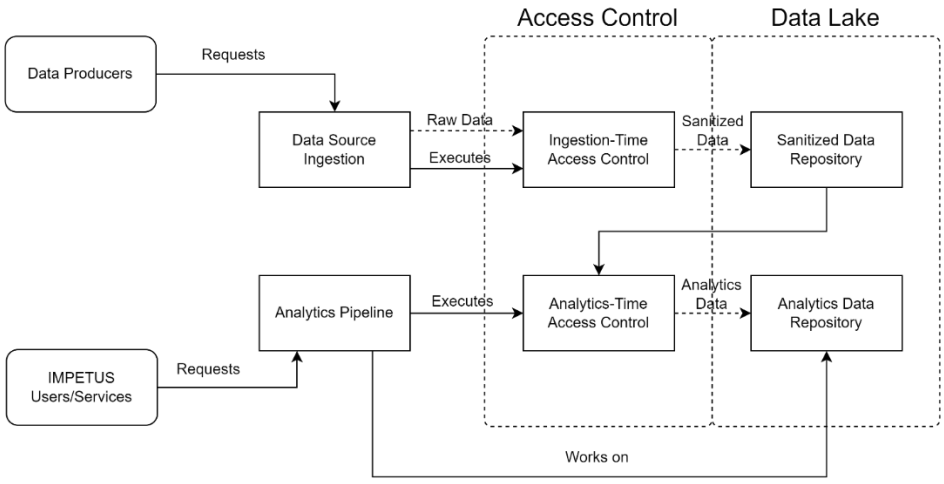

Figure 19: Integrated Big Data platform flow

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

The integration process in this chapter is exemplified using the scenarios developed in the IMPETUS cities for the acceptance pilots and live exercises.

4.1 Ingestion-Time Access Control

This section presents the integration of the Big Data platform, and its ingestion-time access control system, with batch and stream sources from the cities.

It describes the processes for collecting and ingesting data within the Big Data platform. Batch integration is done according to standard APIs; stream integration is done using Kafka queues as presented in Figure 20. Upon ingesting data collected at the partner cities, ingestion-time access control policies are enforced to generate sanitized data, providing a-priori filtering and transformation of data sources on the basis of the services/users that can access data when deposited. Sanitized data are then given as input to the analytics pipeline.

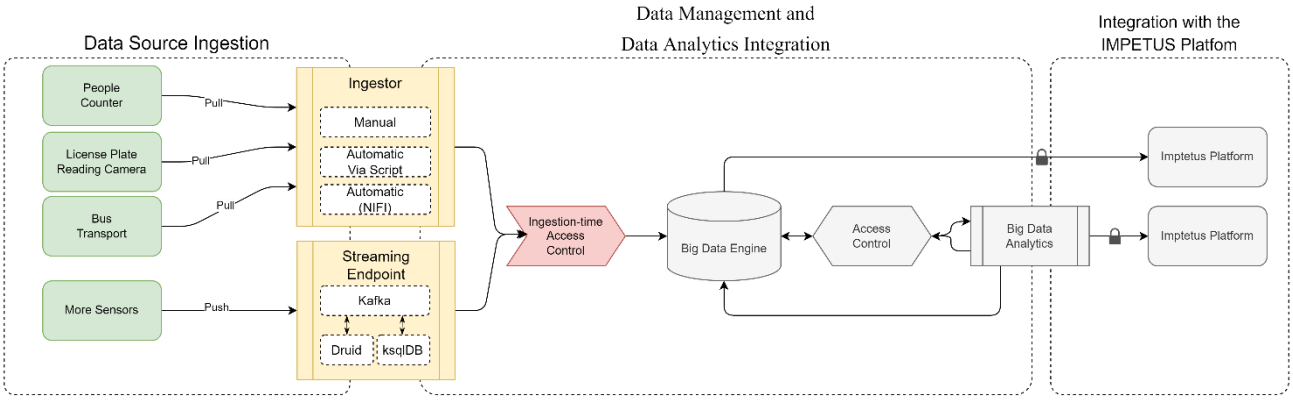

Figure 20: Data Source Ingestion – Mapping to the Engine

4.1.1 Data Sources

The Big Data platform provides a collection and ingestion process that is independent from the specific data sources and can be applied to any datasets. In this section, we consider four data sources collected from the partner cities, Padua and Oslo, as follows.

Traffic Control (Dataset 1 - Padua): It contains information retrieved by the car plate readers at the gates of plate-number monitoring system. The sensors recognize vehicles passing through the gates and record their license plate and timestamp along with other information. The data are downloaded from a centralized API. The API returns the results in JSON format. Only the fields deemed relevant for the analytics is extracted and stored. The received data contains the following information:

- **Date:** the date of the transit (dd-MM-yyyy)
- **Time:** the time of the transit (HH:mm:ss:SSS)
- **Cameraname:** a string indicating the name of the camera that has detected the presence of a car
- **Licence plate number:** licence plate number in clear text
- **Direction**: information about the direction of the vehicle relatively to the camera. The attribute can assume the value “LEAVING” or “APPROACHING”
- **GPS (Latitude and Longitude):** gps coordinates of the sensor

People Counter (Dataset 2 - Padua): It contains information about the people counters installed in Piazza Dei Signori. It is offered through a REST API, which returns data in an XML format. The received data are aggregated by a 5-minute time window and indicate the people who entered and left a specific entrance during a given time window. The received data contains the following information:

- **Date**: in ISO 8601 format
- **In**: Count in the direction of entry
- **Out**: Count in the exit direction
- **Completed**: Indicates whether the detections have completed

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

- **Valid**: Indicates whether the data is validated or not

Transports (Dataset 3 - Oslo): It contains information from a traffic monitoring service of public transportations. The received data contains the following information:

- **dateTime**: Timestamp for when the position/status was recorded/updated
- **LinkDistance**: Distance in meters between the previous stop (or current, if located at stop) and the next stop
- **Percentage**: How much of the total distance (percentage) that has been traversed at the time of the message
- **LineRef**: Reference to the line in question
- **DirectionRef**: Reference to the direction in question
- **PublishedLineName**: Name describing the line in question
- **OriginRef**: Reference to the Origin in question
- **OriginName**: Name describing the origin of the departure
- **DestinationRef**: Reference to the destination in question
- **DestinationName**: Name describing the destination of the departure
- **OriginAimedDepartureTime**: Origin aimed departure time
- **DestinationAimedArrivalTime**: Destination aimed arrival time
- **VehicleRef**: Reference to the vehicle in question
- **Delay**: Delay-time, defined as "PT0S" (0 seconds) when there are no delays
- **HeadwayService**: Field defining whether the service is a headway service
- **InCongestion**: Field defining whether the traffic is in congestion or other circumstances which may lead to further delays
- **InPanic**: Boolean field
- **GPS (Latitude and Longitude):** Position of the public vehicle, according to the timestamp recorded
- **StopPointRef**: Reference to the stop point in question
- **VisitNumber**: Number of the stop point in question
- **StopPointName**: Name of the stop point in question
- **VehicleAtStop**: Field defining whether the vehicle is at the stop
- **DestinationDisplay**: Name of the next destination

Air Quality (Dataset 4 - Oslo): It contains information about air quality coming from different locations in Norway. It is offered through a REST API which returns data in JSON format. The received data contains the following information:

- **ID**
- **Zone, Municipality, Area**: Three values containing information about the city where the station is installed (E.g., Store-Oslo, Oslo, Oslo).
- **Station**: Station name (E.g., Manglerud)
- **EOI**: Entity of Interest
- **Type:** Station Type
- **Component**: The component whose concentration is reported in the value (E.g., PM10)
- **From Time**: in ISO 8601 format
- **To Time**: in ISO 8601 format
- **Value**: The value of the concentration of the "component" component expressed in "unit" unit
- **Unit**: Unit component whose concentration is reported in “value” (E.g., ug/m^3)
- **Latitude**
- **Longitude**
- **Timestep**: Time increments between one measurement and another expressed in seconds (similar to to Time minus fromTime)
- **Index**
- **Color**
- **IsValid**
- **IsVisible**

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

Table 4 provides a summary of the four data sources, their type, and format. Examples in this section are based on these data sources.

Table 4: Data Sources.

City	Source	Source Type	Source Format
Padua	Traffic Control	Batch	CSV, JSON
	People Counters	Batch	XML
Oslo	Transports	Batch/Stream	XML
	Air Quality	Batch	JSON

4.1.2 Batch Ingestion

It is implemented through a batch agent (ingestor) that connects external sources of data with the Big Data engine. Data sources are collected from heterogeneous APIs and then ingested within the Big Data engine through ad hoc connectors. Connectors also prepare (e.g., enrich, merge, transform) data, before their storage, for further analysis.

Batch ingestion is executed according to the following key steps:

i) data are deposited in the distributed filesystem (HDFS) and they are made available to applications;
ii) data in HDFS are imported in specialized tools (e.g., Hive, Presto, Trino) according to a pre-defined data scheme. To this aim, relevant attributes in the datasets are selected, in the proposed scheme;
iii) upon defining the schema, data are imported. In our case, Hive uses HDFS to hold the data and a MYSQL database (Metastore) to hold the schema along with other information such as types, number of records and other metadata. The Metastore can also be used by other tools (e.g., Presto), this facilitates the centralization of the data and favours the diversification of access methods;
iv) once the data and the schema are associated with each other, it is possible to query the data warehouse through queries written in SQL/SQL Like (HQL) language.

Batch ingestion can then be implemented according to three different approaches: manual approach, automatic approach using scripts, or automatic approach using existing Big Data tools (e.g., Apache Nifi) as discussed in the following.

Manual approach: The import of data within HDFS is handled manually. Personnel of the municipality of Padua exports some batches of data in CSV format. The exported data are adapted and imported into the Big Data Platform. The import operation on the filesystem is performed using the integrated HDFS web interface.

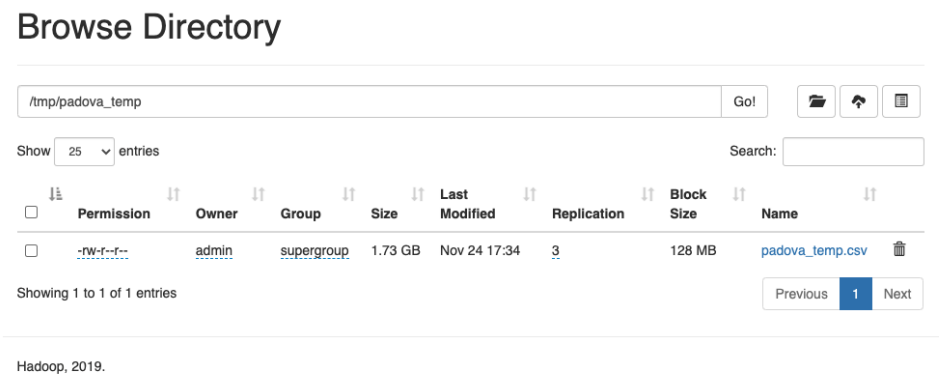

Figure 21: View of datasets in HDFS

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

The files are then loaded into Hive using the script in Figure 22. The script, written in Hive-QL, creates a Hive table and feeds it with data from a CSV file located on HDFS. Operation `CREATE` defines the schema, how Hive should determine row and columns delimitation, where the data are located, and some other properties (i.e., skip the first row because is the CSV header).

```
CREATE external table if not exists oslo.padova_temp
(hash string,`date` timestamp,cameraname string,dir string, lat decimal(8,6),long decimal(9,6))
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS textfile location'hdfs://namenode:9000/tmp/padova_temp/'
tblproperties ("skip.header.line.count"="1");
```

Figure 22: Hive Ingestion Script

Automatic via scripts: The import of data within HDFS is handled automatically using an ad hoc script. In the case of Padua, 9 sensors for counting people are installed in the square Piazza dei Signori. Through the APIs provided by the sensors, it is possible to download all the data provided by each sensor. The script first downloads the data, resulting in a set of XML files stored in HDFS. It then loads all the files from the filesystem, extracts the relevant information, and outputs a single CSV file, which is loaded into the platform (i.e., Hive).

```
# Import the required modules

from datetime import datetime, timedelta
it_holidays = holidays.country_holidays('IT', subdiv='PD')

URL = <API_URL>


def is_holiday(date):
    date = datetime.fromisoformat(date).date()
    return date in it_holidays


# Download the file from `url` and save it locally under `file_name`:
def download(path,begin,end):
    ...
    ...


download("files",begin,end)
files = glob.glob("*.xml")
files.sort(key=os.path.realpath)


#Extract data from file and store them in a temporary array
for filename in files:
    with open(filename, "r", encoding="utf-8") as file:
        xml = file.read()
        xml_to_dict= xmltodict.parse(xml)
        for x in xml_to_dict.get("ArrayOfTrafficHourly").get("TrafficHourly") or []:
            results.append(
                {
                    "date": x.get("Date"),
                    "in": x.get("In"),
                    "out": x.get("Out"),
                    "sensor_id": filename.split("_")[1].split(".")[0],
                    "holiday": int(is_holiday(x.get("Date")))
                }
            )

#Store the array as CSV
df = pd.DataFrame(results)
df.to_csv('output.csv', index=False)
```

Figure 23: Ingestion Script

Automatic approach using BDE tools: The import of data within HDFS is handled automatically using NiFi. NiFi is a tool that supports powerful and scalable directed graphs of data routing, transformation, and system

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

mediation logic. Its structure permits to connect each element of the flow through queues. Through this connection, the output of a step becomes the input of one or more subsequent steps (even parallel ones). Queues allow to create buffers between one step and the others. In this way, the fastest steps can complete their work asynchronously and deposit the results to be consumed by the slower ones. The slower steps, on the other hand, can process the inputs provided by the faster ones suitably with their own execution times.

Figure 24 shows the NiFi flow adopted to fetch the data from Oslo air quality data and ingest them in the Big Data engine. The flow downloads the data (in JSON format) and chunks them to reduce the impact on the memory by leveraging on the queue system integrated in NiFi. The chunks are then converted into CSV format. Each CSV file contains a number of elements at most equal to the maximum number of elements that a queue can hold.

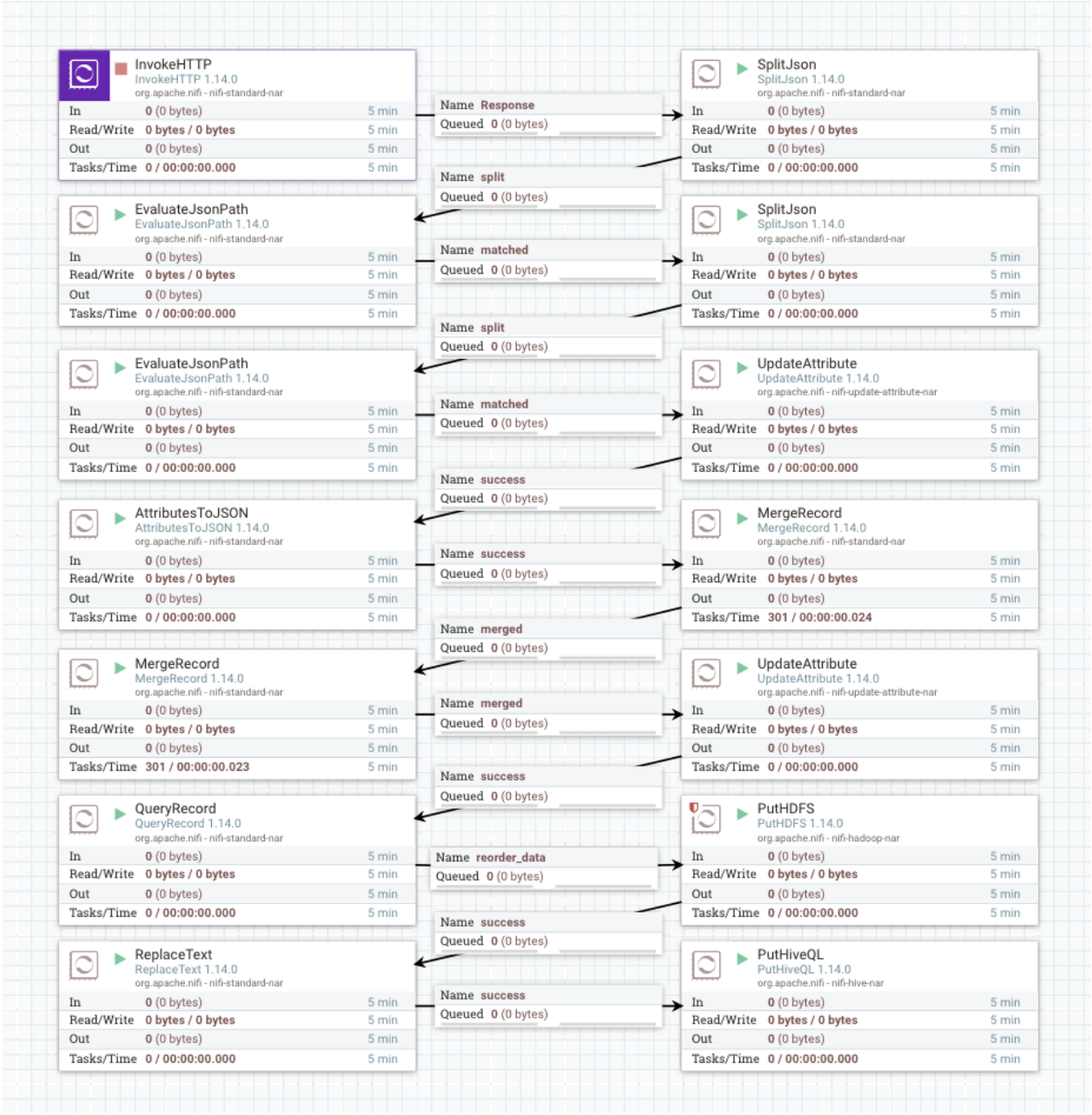

Figure 24: Ingestion flow using NiFi.

Each file is then loaded into Hive through the following query template:

```
LOAD DATA INPATH '/path/to/the/file.csv' INTO TABLE #{db}.#{table};
```

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

4.1.3 Stream Ingestion

It is implemented through a streaming endpoint in Kafka capable of accepting connections and receiving stream of data. Producers submit their data to a Kafka topic, which flows into the BDE.

Kafka performs a dual function within the system. It acts as an intercommunication channel between the components and as a buffer. The data can in fact be deposited inside Kafka to be consumed (even multiple times) later. However, Kafka and its streaming data management functionalities are more difficult to be integrated with our ingestion-time access control based on Apache Ranger for two main reasons:

i) Apache Ranger only supports a subset of its capabilities when integrated with Kafka. We can only enable or disable accesses to certain topics by certain users or groups of users, thus strongly limiting the working of our access control solution;
ii) online data transformation makes the management of data streams very complex. This is due to the lack of information about the shape (types, schemas) of the data flowing in the various topics. Adding the responsibility of schema extraction and management to Kafka should be avoided as Kafka bases its efficiency on its simplicity.

We therefore complemented Kafka with Apache Druid, an Apache tool offering functionalities to apply real-time transformations on a stream of data.

Apache Druid is a data store designed to quickly ingest massive quantities of data. Druid permits the ingestion of Kafka streams using a specific connector called “Kafka indexing service”. A guided step-by-step procedure permits to select the Kafka instance and the Topic from which to draw data. Druid then suggests the types of data for the received data and permits to apply transformations on the data themselves. After completing this quick configuration, the data start to flow into Druid and can be later consumed using simple queries.

Druid exploits Kafka's non-duplicating functionality to create a copy of the data on which it performs the operations requested by the users and open the way to an access control that also works on data streams.

Druid architecture is described in Figure 25 and is composed of the following components:

i) Deep storage - Shared file storage layer;
ii) Metadata Storage - Shared metadata layer;
iii) Data Servers - Process ingestion and queryable data;
iv) Query Servers - Execute query and route request to other components;
v) Master Servers - Coordinate ingestion workloads and data availability.

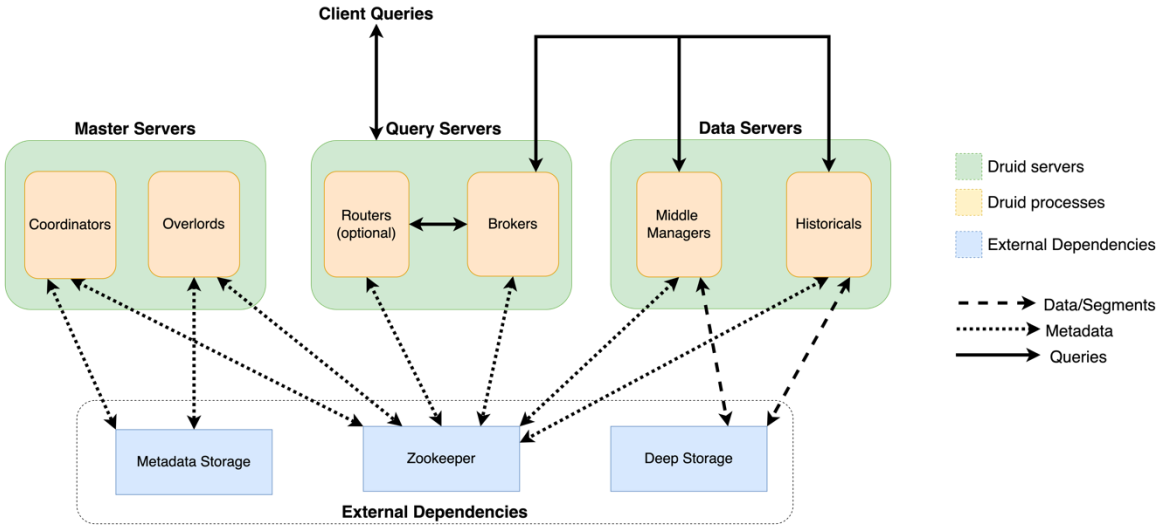

Figure 25: Druid Architecture

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

4.1.4 Ingestion-Time Access Control

Upon ingesting data sources, the access control system in Chapter 2 enforces policies to dynamically augment the ingestion pipeline with security and privacy-oriented transformation tasks and supports sanitization and transformation of data prior to the storage in the data lake.

As already described in D4.1 [1], transformations are instrumented by policies and data annotations, and triggered by policy enforcement. Policy selection is done according to the service (and corresponding users) managing and accessing data, while policy enforcement is done independently from the specific pipeline to be executed.

We present here a summary of ingestion-time access control for batch data (already presented in D4.1) and ingestion-time access control for stream data, both integrated within the Big Data platform in this document.

4.1.4.1 Ingestion-Time Access Control (Batch Data) – Excerpt from D4.1

Ingestion-based access control enforcement on batch data is managed by Apache Ranger and Apache Atlas and implements the approach in Section 2.1.2. We adopted Apache Ranger for the definition, assignment, and enforcement of access control policies and Apache Atlas for annotations instrumenting the policies.

Our solution regulates CRUD (Create, Read. Update, Delete) operations on tables, databases, tagged entities, files in the HDFS filesystem. It also supports filtering or modification of records based on specific permissions of the user making the request.

The proposed pipeline is composed of the following steps. First, raw data are stored in plaintext in a temporary table. Then, the appropriate policies are enforced on the data possibly organized in views. Finally, data are ingested in the BDE. In this way services or users without permissions have no access to the data in clear text.[6]

Figure 26 shows the Apache Ranger user interface to define an access control policy named "*grant select user1 on traffic*" on dataset 1. In particular, from the "Policy Details" panel, we can see that the policy targets all columns of table *traffic* in the database called *traffic*. In the lower panel, section "Allow conditions" specifies the permissions granted to the users as follows.

- *SmogAnalyst* can perform only select operations;
- *LocPolDept* can perform create, select, and update operations.

[6] We note that in case raw data cannot be stored in the system due to legal and privacy requirements, a compliant version of the data organized in tables is created and the stored. This can be accomplished by defining a policy on the table and then producing a materialized view that applies the policy, effectively creating a copy of the data that is compliant to legal and privacy requirements.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

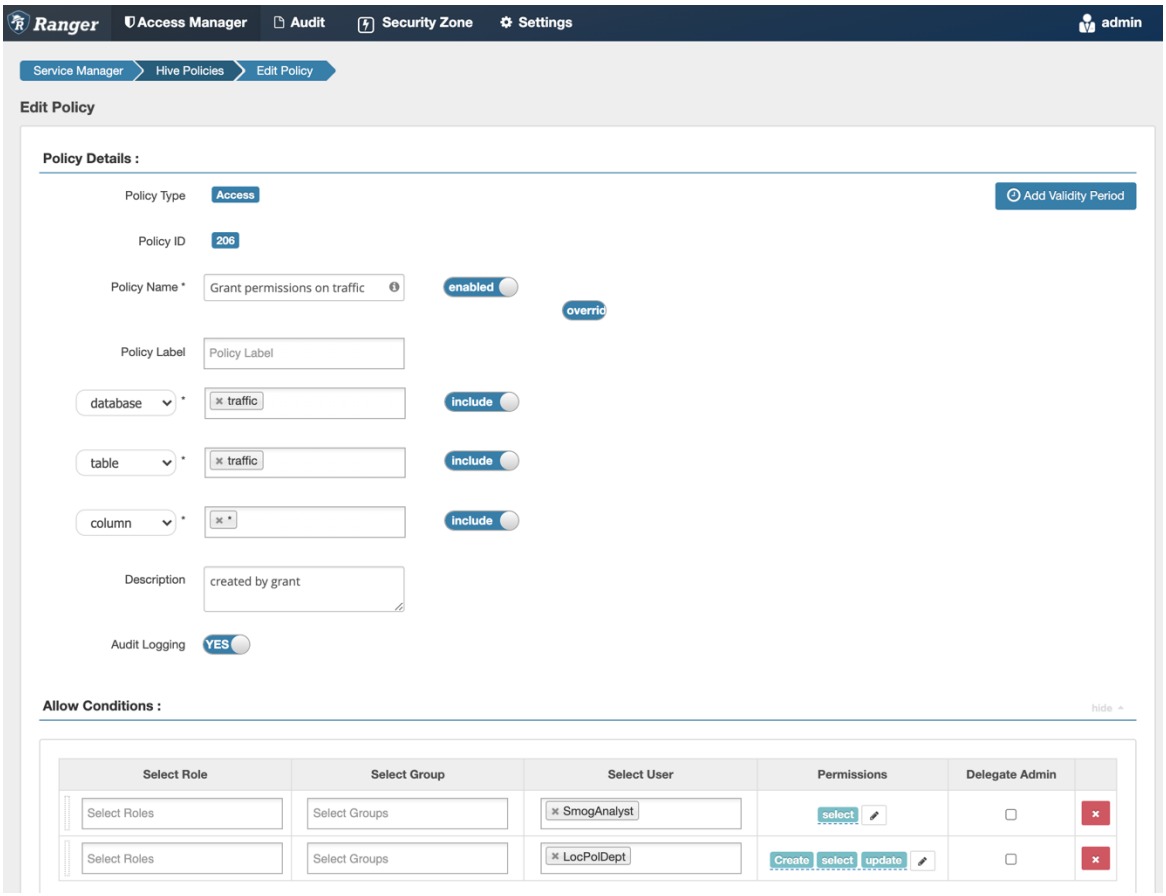

Figure 26: Definition of an Access policy.

Apache Ranger can access to different security and privacy-oriented transformation tasks and trigger their execution.

Figure 27 shows where it is possible to define the Apache Ranger transformation actions. Note that custom transformations can be defined and added to this list (as described in Section 2.2.3).

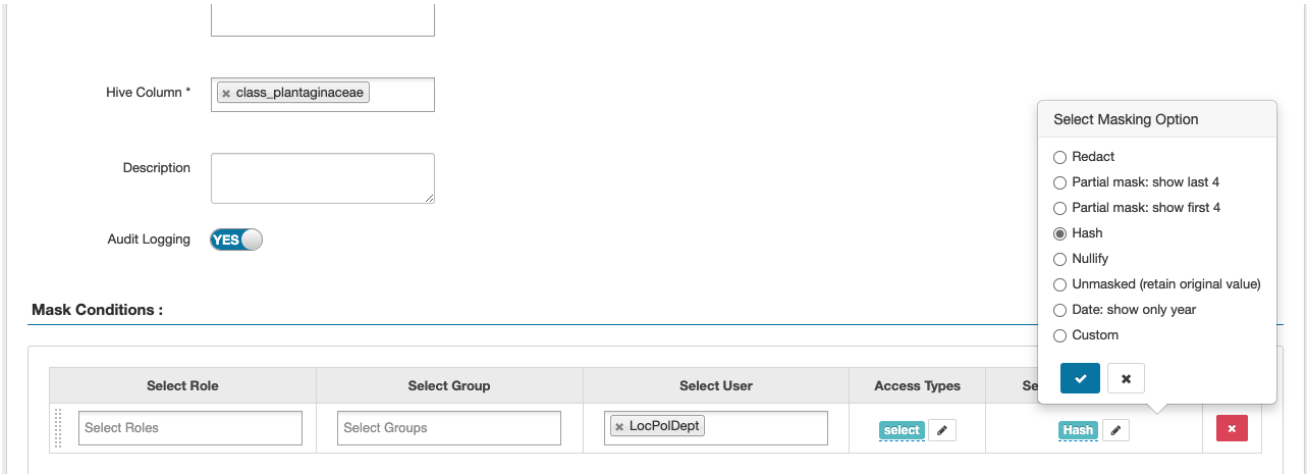

Figure 27: Apache Ranger transformation actions.

Apache Atlas is used to implement the *annotation* process by means of tags. The tags are used to define tag-based policies in Apache Ranger; Apache Atlas notifies the relevant components about tags when needed.

Figure 28shows a tag-based policy in Ranger using the defined tags. Atlas will also provide data lineage to improve the auditability of data flows.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

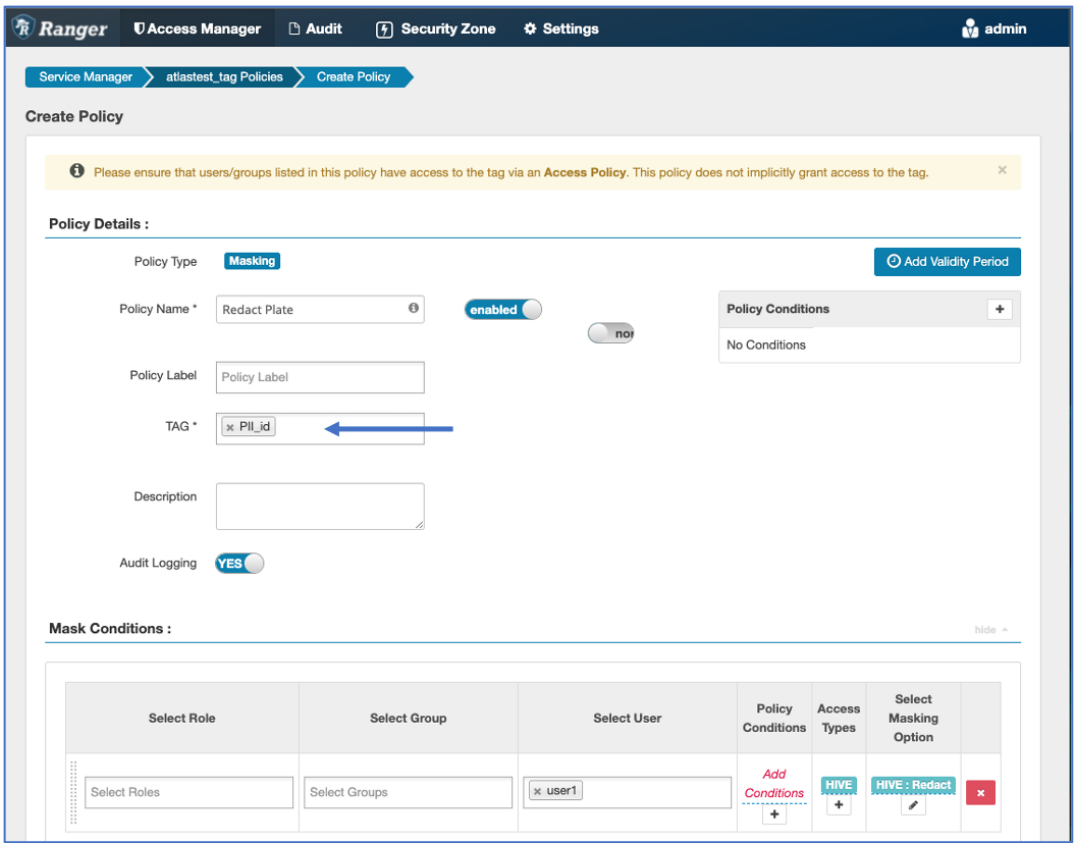

Figure 28: Tag-based masking policy

4.1.4.2 Ingestion-Time Access Control (Stream Data)

Ingestion-based access control enforcement on stream data is managed by Apache Druid, and works as follows:

- it first parses data and identifies column timestamp;
- it then defines some transformations and filtering based on data type;
- it further defines a schema for the stream data in a semi-automatic way. Druid analyses the incoming data flow and infers a plausible pattern, leaving to the user the responsibility of refining it (Figure 29);
- it finally executes a tuning process. At this point data are saved in the Druid datastore with the applied transformations. The original copy of the data is retained inside Kafka.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

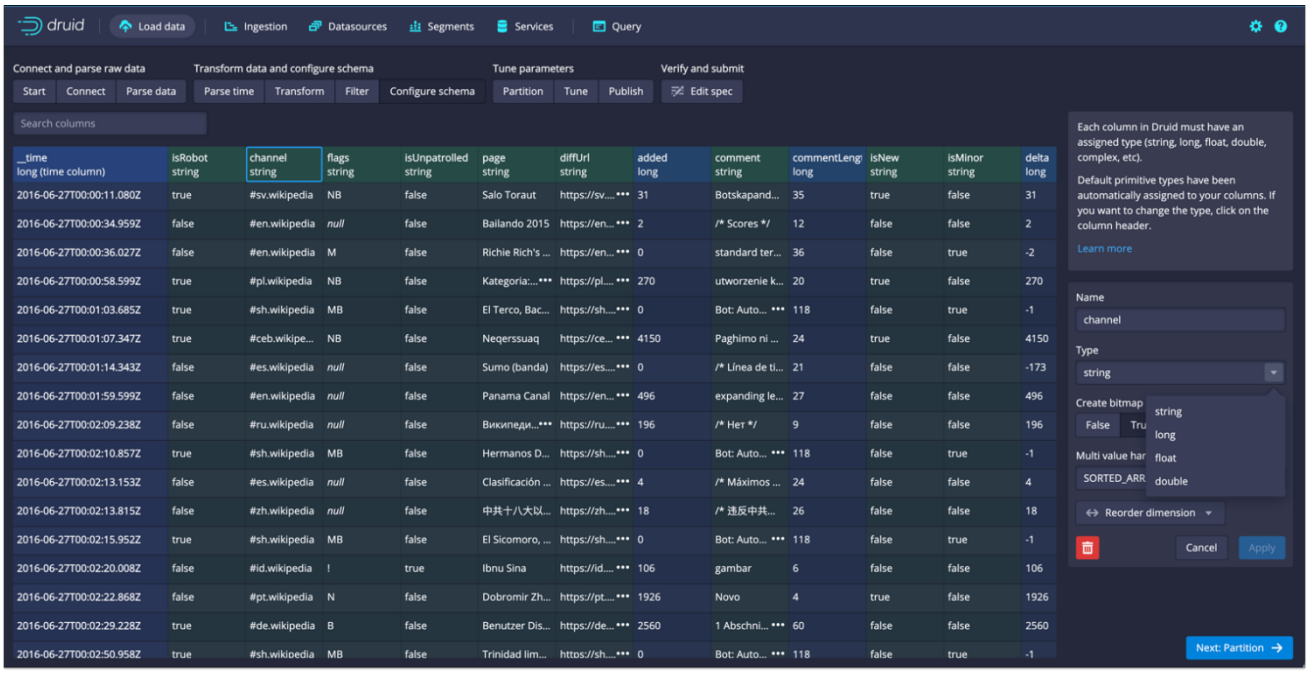

Figure 29: Druid Web UI – Parsing phase. In this phase Druid suggests data type; the user can specify the most suitable type

Correctly deserializing the messages permits to apply transformation queries to the data in a similar way to what has already been done in Section 4.1.4.1 for batch data. The ingested data can be retrieved with simple SQL queries, as shown in Figure 30. We note that Druid permits to monitor the result of the query live.

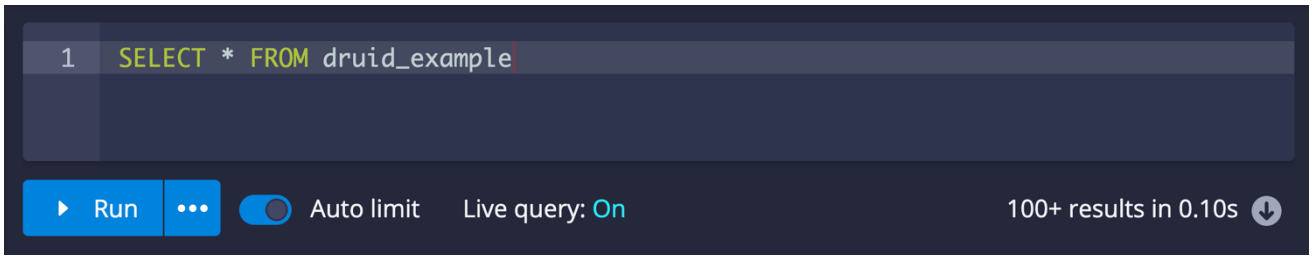

Figure 30: An example of Druid query

4.2 Data Management and Data Analytics Integration

The integration of data management in Chapter 2 and data analytics in Chapter 3 permits to mediate data analytics by access control enforcement (Figure 31). The latter works similarly to the ingestion-time access control in Section 4.1.4, while further transforming data based on the current user and the pipeline itself.

The extended access control system introduced in Chapter 2 then provides a full-fledge and complete integration within any Big Data analytics pipelines explained in Chapter 3 and protects data along all steps of an analytics pipeline, by enforcing policies before its execution.[7]

[7] We note that the solution in D4.1 only applied access control enforcement at ingestion time, before the analytics pipeline was even implemented.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

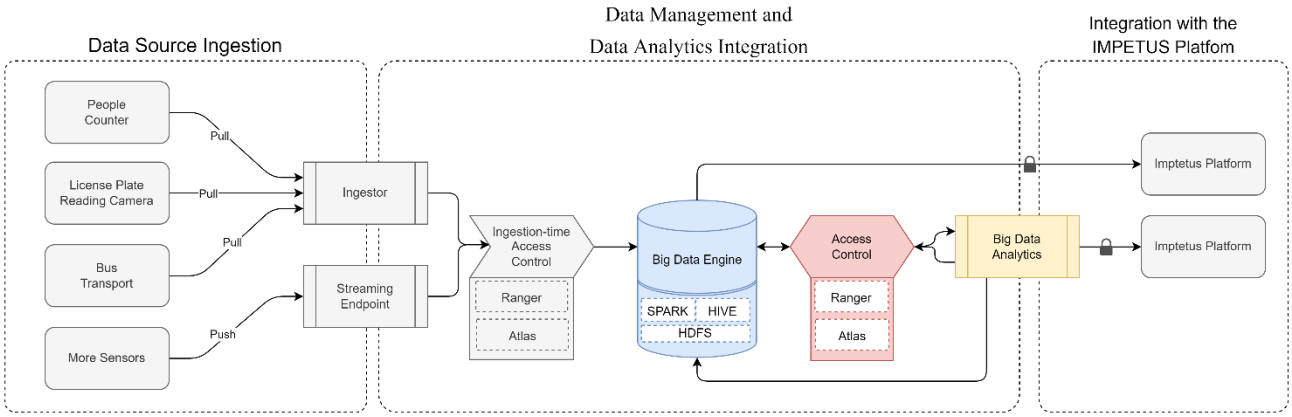

Figure 31: Data management and data analytics integration – Mapping to the Engine

More in detail, once data are transformed according to policies enforced at ingestion time (Section 4.1.4) and stored within the data lake, the analytics pipeline can be executed.

Additional data sanitization and transformations are applied depending on the analytics pipeline to be executed and the user executing it.

The access control system in Chapter 2 mediates each analytics pipeline execution, enforcing access control policies and in turn transforming the input data prior to their analysis within the data analytics pipeline.

This permits a fine-grained management of data protection, which reduces the amount of transformations and manipulations needed when an ingestion-time policy enforcement is applied alone. In the latter case, in fact, policies must take into account the most critical user or pipeline service accessing data and transform data according to its privileges, unnecessarily losing data quality.

When data sanitization is applied at analytics time, data transformation can be tuned on the basis of the specific user or service accessing data at a specific step of the Big Data analytics.

Two main policy enforcement steps are done at this stage to:

i) regulate access to HDFS and files stored in it (e.g., the machine learning model);
ii) regulate access to data, by sanitizing them prior to analytics execution.

Focusing on the analytics in Chapter 3, once the Spark-GHSOM executable obtains the sanitized data, it begins its training phase and produces an output model. The model is then saved to HDFS alongside with the Spark-GHSOM JAR itself. However, the HDFS filesystem acts as a repository for data originating from different sources and intended for different purposes, so it is not uncommon for multiple users to find themselves browsing it. To prevent some data (e.g., the model) from being accessed indiscriminately, access control comes into play. The reading and writing of the model and in general the permissions in the folder are granted only to the user or service under which the analysis is performed.

When the analytics phase is executed, an additional access control policy enforcement is executed, possibly resulting in an additional data transformation. In the case of data from Oslo (dataset 3), no further policy enforcement is needed.

In the case of data from Padua and specifically those concerning traffic control (dataset 1), a masking policy on the plate number of the involved vehicles is applied.[8]

A hash policy is enforced on the plate numbers, since annotated as sensitive. We note that the application of the hash policy does not particularly affect the quality of the data from the analytics side, since it is still possible to discriminate the different plates even if masked. Figure 32 shows the hashed plate numbers resulting from policy enforcement.

[8] We note that, for privacy reasons, the plate numbers were hashed before being distributed by the city of Padua. The additional hashing executed on the hashed values has no impact on the overall process, while it permits to test the technical soundness of our approach.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

	plate_id
1	1N6AD0C
2	KNDMG4C
3	5LMJJ3J
4	WAUDF78
5	1FAHP3E
6	1G6DE8E
7	4T1BF1F
8	1N4AL2A
9	KL4CJBS
10	5UXZV4C
11	2C4RVAA
12	WBA3T1C
13	WAUDFBF
14	4T1BF1F
15	3D7TT2C
16	WBAXA5C
17	JM1NC2E
18	3D73M4C

	plate_id
1	02e7bf8ebad1b0b1eba4cc7a8444b781b3a66bab74443d3a88fcc4cc249d5bb6
2	90f2663cac0c424f21d63d4186b90f11319bf8641dacbe58414b13c3e1402f57
3	a2293d78c9ec4ef016b991aff2cffa5cf212ad86d44eb518dc1951af5924d004
4	3bbc027aed97896a7ba675502ebe55e5b6398825624c717e1dd5a1b6e8729256
5	90f2663cac0c424f21d63d4186b90f11319bf8641dacbe58414b13c3e1402f57
6	63f0fbfa4f4929d1e81e7393369be880a12ad3ff366709c70f3afe605818aaba
7	4a377350ad793196070e0110322cbddfb83d009fbe960346d265bd2165722995
8	59a6cf062c65ccac39d10ec6c52c5d3d8040bd36c02077caa0b4a377c6ee9ade
9	d4a389b09e90c1f7cd327d456501c8d35bbbae0795702cad454cc80096b6da90
10	95249cde73f94bbde2837fb5da7870cdcf4eb9ccdf4a7efd67280592112f0bce
11	f5087f1a18c9967f04a84d7ea174f0ce7be6ce2bd10df1682c9861580c80afd8
12	b843a18e57991ab34cca7be7586e37949d0a42592899083cf2167ef210e0a0a4
13	b843a18e57991ab34cca7be7586e37949d0a42592899083cf2167ef210e0a0a4
14	38ba3591268cd140f6331c7c29863fd896f5fd0f536f0236b59536a57ff9387c
15	3daee7f1bd7466b38ede5344e3a226755f84f2612d0817a27f886b1c02c97338
16	5948e7343278750e8791689fe136a4c70e0938bb3dbad7253f4d5413c53e5b40
17	e8eaf9b830130304e81a9277956796fc023ad63b0c7d296f50e2f2fbdf32deaa
18	717e2549de8f1e792abffb10f41b35c3e95e9fed251f3844fd73e137fa1b3e1a

Figure 32: Plate IDs – Before and after applying hash policy (please note the data in the example are not real)

Upon analytics-time access control enforcement, the analytics pipeline is executed. Below, we present the results retrieved by executing the solution in this document with Spark-GHSOM (tool, for brevity) for anomaly detection on dataset 1, dataset 3, and dataset 4 from the two city pilots.

Integration results for the city of Oslo (Dataset 4)

During the integration procedure for the city of Oslo, the solution in this document has been tested using data coming from air quality monitoring sensors to identify pollutant concentrations deemed abnormal. Below is reported the list of stations used during the test session, together with the list of pollutants that each station can monitor:

- Hjortnes. NO, NO2, NOx, PM10 and PM2.5 pollutants.
- Loallmenningen. NO, NO2, NOx, PM1, PM10 and PM2.5 pollutants.
- Spikersuppa. PM10 and PM2.5 pollutants.

The location of the stations within the city of Oslo is shown in Figure 33. It is possible to observe that the considered stations are close to the Oslo Town Hall, where the Acceptance Pilot was held.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

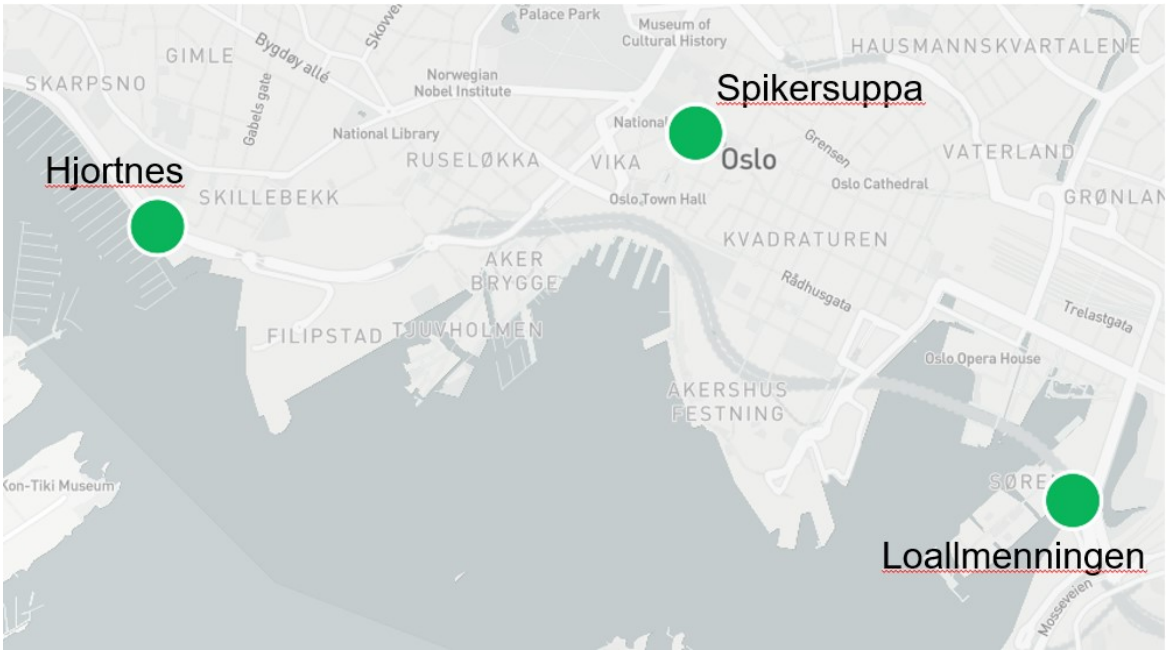

Figure 33: Location of the considered stations in the city of Oslo.

The period considered for training goes from January 2021 to September 2021, with an acquisition per hour, totalling 18.286 acquisitions from the chosen stations. The period considered for testing is October 2021, with an acquisition per hour, totalling 720 acquisitions from the chosen stations.

Figure 34 shows the concentrations per hour of NO, NOx, and NO2 pollutants during the identified test period, i.e., October 2021, from Hjortnes station. The choice fell on these pollutants because they are present within the top-3 of the feature ranking, for those time instants considered anomalous by the tool, indicated with black arrows within the graph.

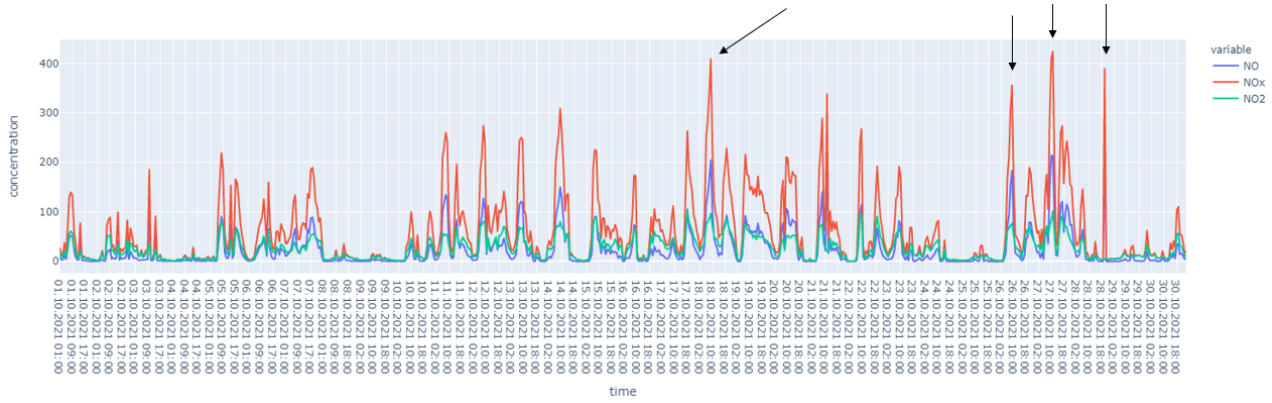

Figure 34: Concentrations per hour of NO, NOx and NO2 pollutants during October 2021, from Hjortnes station.

As shown in the graph, higher NO and NOx concentrations were recorded at the time points identified by the tool.

Figure 35 shows the concentrations per hour of PM10 and PM2.5, the pollutants considered less relevant for the detection of situations deemed abnormal by the tool, which occurred during the test period.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

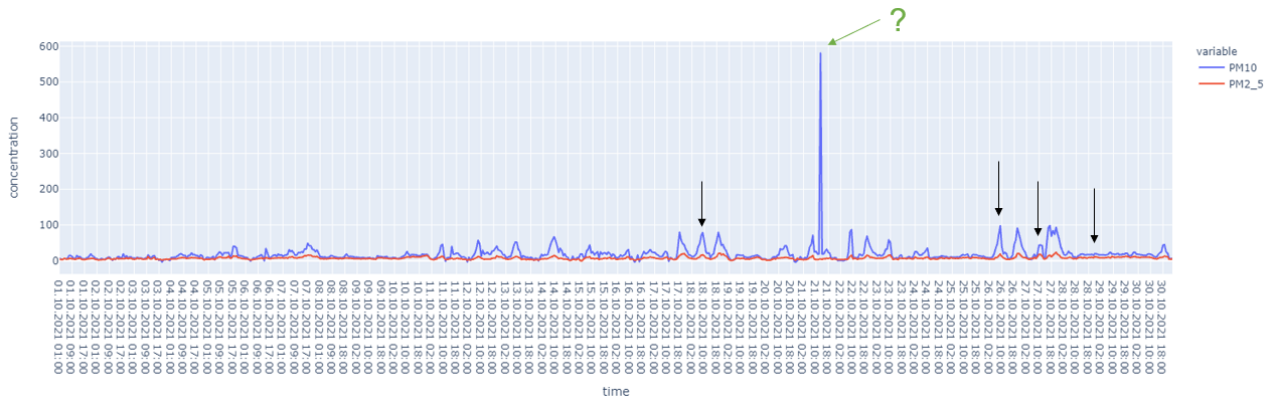

Figure 35: Concentrations per hour of PM10 and PM2.5 pollutants during October 2021, from Hjortnes station.

As in Figure 34, the black arrows indicate the time instants when the tool found abnormal situations. It is interesting to note that the tool did not find an abnormal situation during October 21 at 10 a.m., indicated with a green arrow in Figure 35, when very high concentrations of PM1 were recorded, even though at this time point the pollutant PM1 is correctly present in the first position of the feature ranking.

The motivation is because several pollutants are being observed by the tool and the sudden increase of concentrations of one of them is sometimes not sufficient to classify the time instant as a potential abnormal situation.

Figure 36 shows the concentration per hour of PM1 pollutant during the test period, from Loallmenningen station. For this station, PM1 is the most decisive pollutant for the detection of abnormal situations that occurred during October 2021.

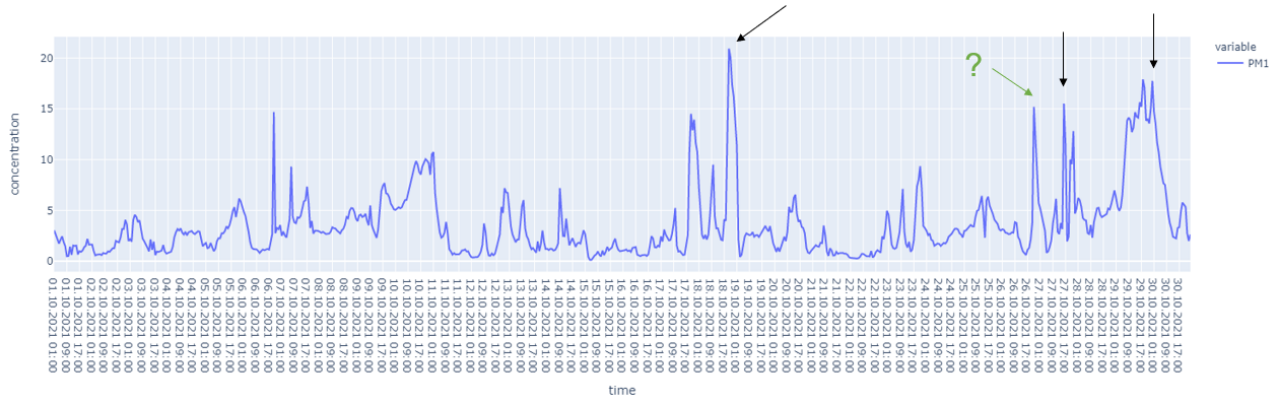

Figure 36: Concentrations per hour of PM1 pollutant during October 2021, from Loallmenningen station.

As in the previous graphs, black arrows indicate the time instants in which the tool detected abnormal concentrations of the pollutants considered. As expected, the tool was able to correctly detect high concentrations of the PM1 pollutant.

However, on October 26 at 9 p.m., indicated by the green arrow, the concentrations of PM1 were very similar to those of October 27 at 4 p.m., but only in the latter case, an anomalous situation was found by the tool. A more detailed graph is shown in Figure 37.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

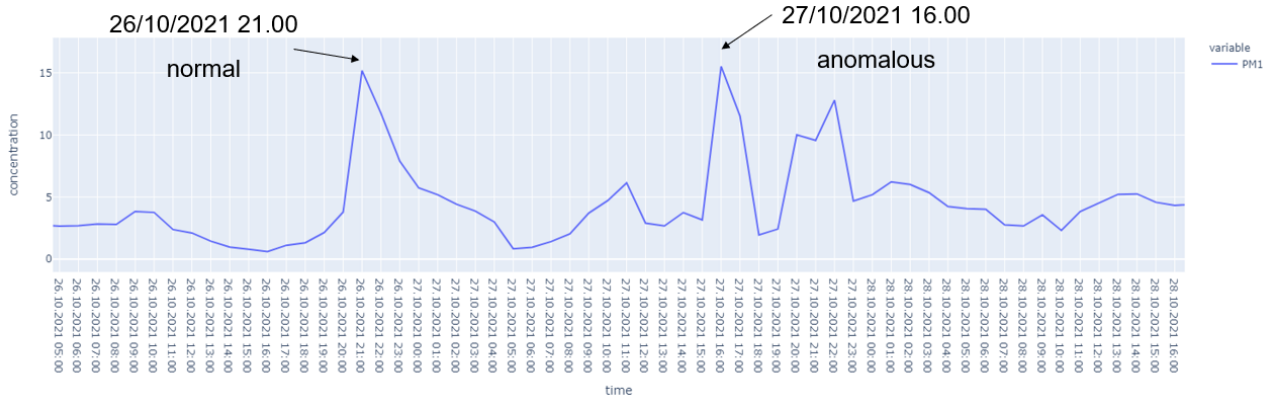

Figure 37: Concentrations per hour of PM1 pollutant during October 2021, from Loallmenningen station.

The reason is due to a sudden increase in concentrations of the remaining pollutants, which occurred on October 27 at 4 p.m. This situation, as shown in Figure 38, permitted the tool to identify an anomalous situation during this time.

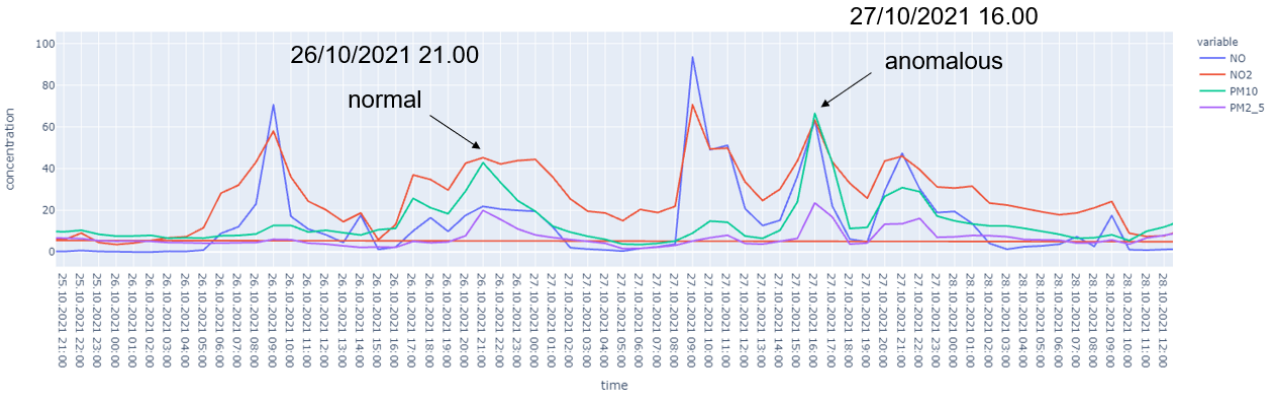

Figure 38: Concentrations per hour of NO, NO2, PM10 and PM2.5 pollutants during October 2021, from Loallmenningen station.

Figure 39 shows the concentrations per hour of PM10 and PM2.5 pollutants during the test period, from Spikersuppa station. The pollutants shown in the graph are the only ones the station can monitor.

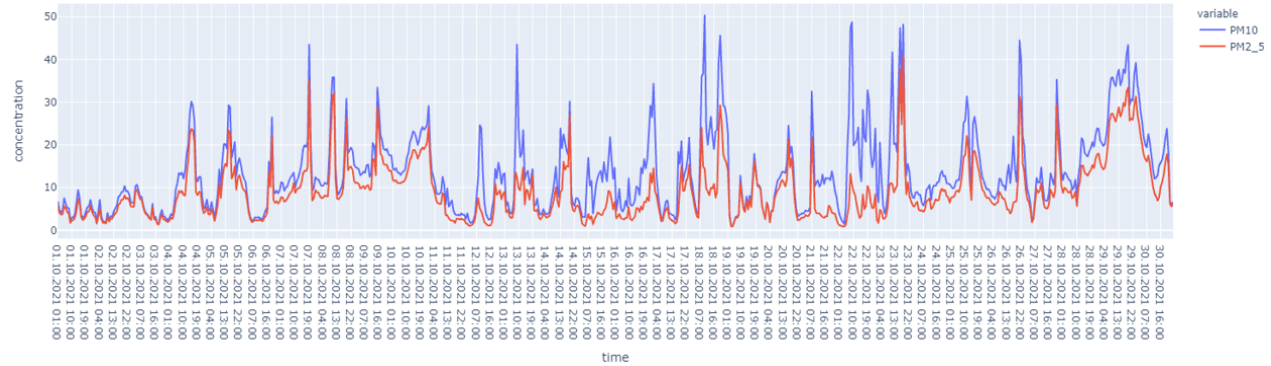

Figure 39: Concentrations per hour of PM10 and PM2.5 pollutants during October 2021, from Spikersuppa station.

As expected, the tool did not identify any situations deemed abnormal for this station, as the concentrations of October are quite regular.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

Integration results for the city of Padua (Dataset 1)

During the integration procedure for the city of Padua, the solution in this document has been tested using data coming from plate-number monitoring system (dataset 1) to identify vehicle transits deemed abnormal by the tool. The system can monitor the transit of vehicles at a total of 94 points in the city of Padua.

Vehicle transits have been aggregated in 5-minute time windows, for each sensor involved, to detect more relevant situations. The aggregation just described allowed the addition of the following derived descriptive variables to dataset 1:

- numVehicles: number of vehicles passing through the 15-minute time window.
- numUNKNOWN: number of vehicles passing through the 15-minute time window, with direction "UNKNOWN".
- numLEAVING: number of vehicles passing through the 15-minute time window, with direction "LEAVING".
- numAPPROACHING: number of vehicles passing through the 15-minute time window, with direction "APPROACHING".

The sum of numUNKNOWN, numLEAVING and numAPPROACHING for each time window considered is equal to numVehicles.

The period considered for training the tool is respectively November 23, 24, 25, 26, 27, 29, and 30, 2021 considering only the weekdays of the chosen week. The period considered for testing the tool is December 1, 2021.

Figure 40 shows daily vehicle transits with direction "UNKNOWN", "LEAVING" and "APPROACHING" in the street "Ponte 4 Martiri" vs "Padova centro", during the identified test period, i.e., December 1, 2021. The black arrows indicate the time instants when the tool found abnormal situations.

As shown in the graph, anomalies were found on this day between 5 p.m. and 9 p.m. In this time slot, it is possible to observe a high number of vehicles with directions "APPROACHING" and "LEAVING".

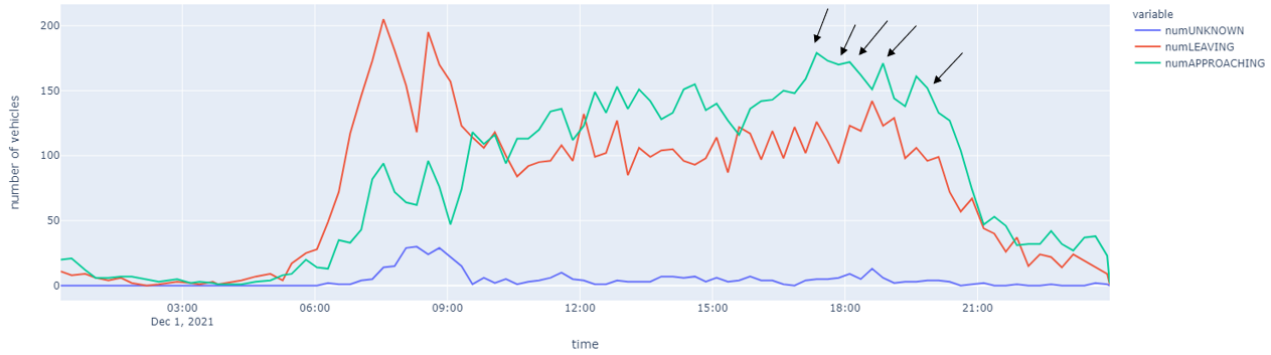

Figure 40: Daily vehicle transits with direction "UNKNOWN", "LEAVING" and "APPROACHING" during December 1, 2021, in the street "Ponte 4 Martiri vs Padova centro".

Figure 41shows daily vehicle transits with direction "UNKNOWN", "LEAVING" and "APPROACHING" in the street "Sito 1 – via Plebiscito", during December 1, 2021. Black arrows indicate the time instants when the tool found abnormal situations.

As shown in the graph, a high number of vehicles with direction "UNKNOWN" passed between 8 and 9 a.m.. The tool was able to identify this situation as abnormal.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

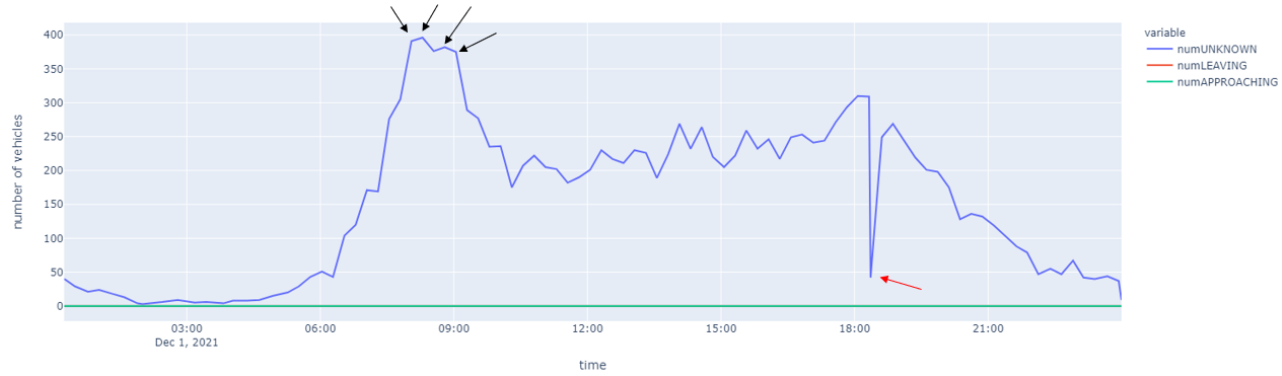

Figure 41: Daily vehicle transits with direction "UNKNOWN", "LEAVING" and "APPROACHING" during December 1, 2021, in the street "Sito 1 – via Plebiscito"**.**

To test the robustness of the tool, it was decided to simulate several anomalous situations to understand whether, in its current state, the tool was able to detect them.

The red arrow in Figure 41 indicates one of the simulated time instants. However, the tool was not able to classify the time instant as anomalous.

A different simulated time instant is shown in Figure 42, indicated with the red arrow. In this case, the tool has correctly classified the time instant as an anomaly.

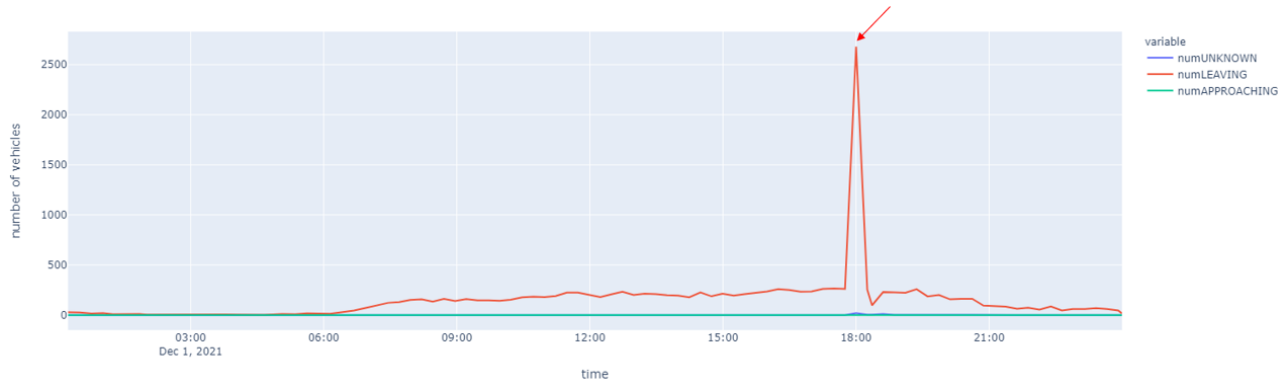

Figure 42: Simulated daily vehicle transits with direction "UNKNOWN", "LEAVING" and "APPROACHING" during December 1, 2021, in the street "Rotonda Grassi – Maroncelli corsia sinistra vs Grassi/Friburgo".

Integration results for quantitative analyses of public transport traffic (Dataset 3)

Further experiments were conducted for anomaly detection tasks with the aim to evaluate quantitatively the goodness of the learned predictive models. Specifically, for the public transport traffic analysis, we evaluated the ability of the proposed algorithm to identify anomalies automatically for the public transport traffic of Oslo. A week of data regarding Oslo's public transport GPS-tracked busses were considered to learn the predictive model in order to define the *normal behavior* of the public transport traffic. The instances are geotagged with latitude and longitude and each instance is timestamped according to the standard ISO 8601 with a resolution in seconds. The pipeline for the analyses of Oslo's public transport is illustrated in Figure 43.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

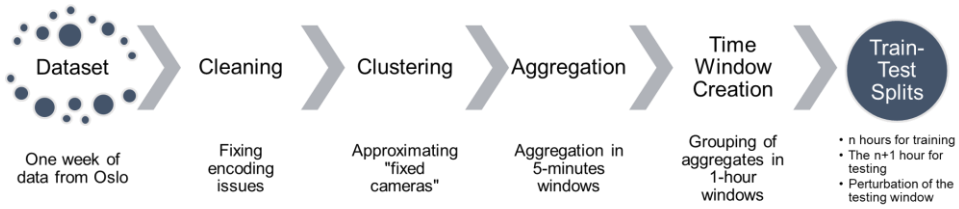

Figure 43: Data processing pipeline for Oslo's public transport analysis.

Therefore, starting from the week of data from Oslo traffic transport, we performed *data cleaning* in order to fix some encoding issues. For example, the characters "Ø" and "Å" have been encoded as *accented A* followed by non-printable characters. We replace the hexadecimal codes of the wrong characters with the right ones. We also removed a single instance that contained (almost) only null values.

The *clustering* then aims to identify a pair of coordinates indicating specific areas with *stationary cameras*. Such detected locations could be seen as strategic points on the map for monitoring the traffic.

Once such points were detected in the city, we performed the computation of *aggregated* data within 5 minutes according to how many buses passed through such specific points. This step was crucial since the data provided refer to movable points in the map making the aggregation operations unfeasible. Clustering on the geolocation was performed by exploiting K-Means algorithm [10]. The variables of the considered data were extended by considering the cluster identifier (cluster ID) and the cluster's centroid latitude and longitude to the data. Since K-Means algorithm needs the number of clusters to identify, we performed the well-known silhouette cluster analysis [11] with the aim to identify the number of areas for monitoring the traffic. According to silhouette analysis (as illustrated in Figure 44), we considered 100 different regions for monitoring the traffic.

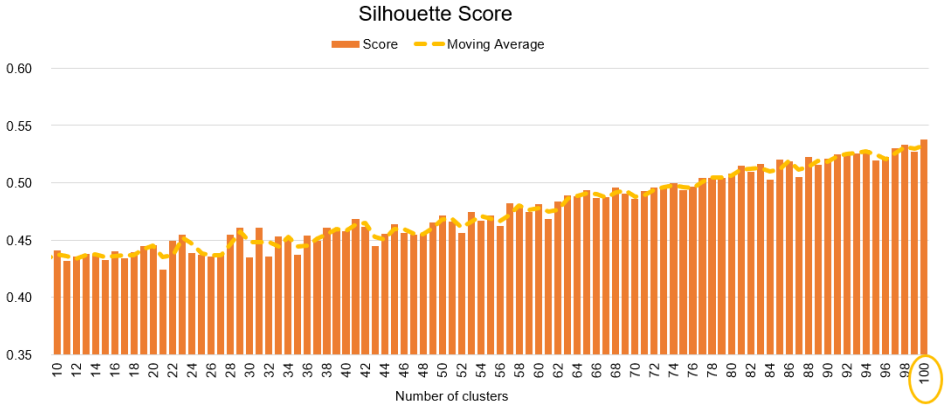

Figure 44: Silhouette cluster analysis

After the clustering of the GPS coordinates, the data are aggregated in 5 minutes *windows*. The instances are grouped by two levels: first the time, then the cluster id previously identified. Various new features are computed as part of the aggregation (e.g., the average of the buses "delay") for each identified clustered monitoring area. The categorical variables are encoded as one-hot vectors to allow aggregations. Therefore, the vectorial sum is considered as operation for the one-hot vectors for the final aggregated dataset. The aggregates are then split into various landmark windows: each window is of exactly 1 hour and it is saved in a separate file for easier management. Specifically, multiple train and test sets were created as illustrated in Figure 45.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

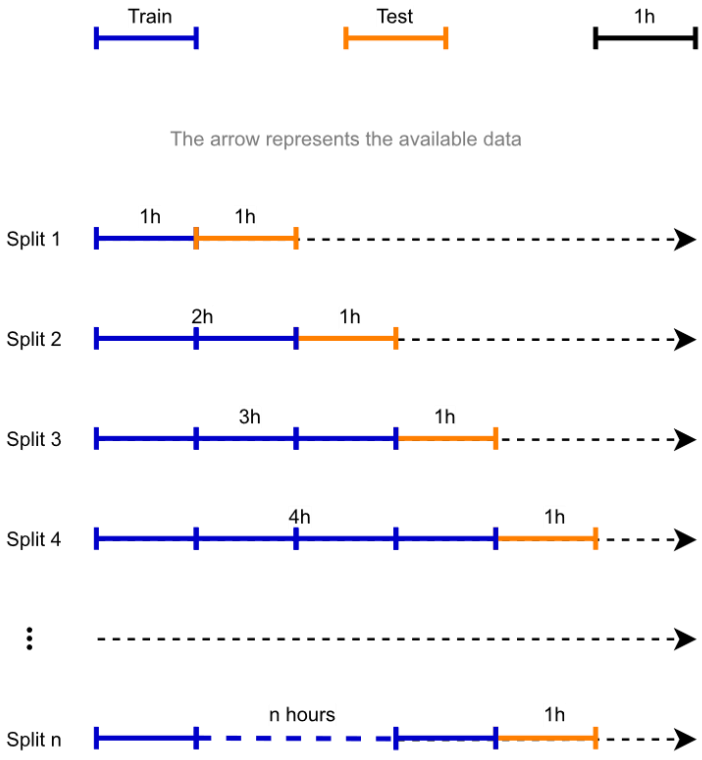

Figure 45: Train and test splits

The n-th split uses n hours for training, and the (n+1)-th hour is used for testing. The 10% of the available test windows are perturbed randomly selecting a set of columns for each instance and randomly assigning a new value for each selected feature. The remaining 90% of the available test windows are used as is (as normal windows). The aim of this setting is to build several anomaly detectors in order to evaluate different times how is the accuracy of the predictions in detecting anomalies and normal cases.

The same evaluation procedure was also used

Results

In Figure 46, an hour-by-hour confusion matrix is illustrated. Stacked green bars indicate the correct predictions, while the red ones the wrong predictions. The red text in the date indicates that the window is perturbed (anomaly). The top label contains the total number of instances in the test set.

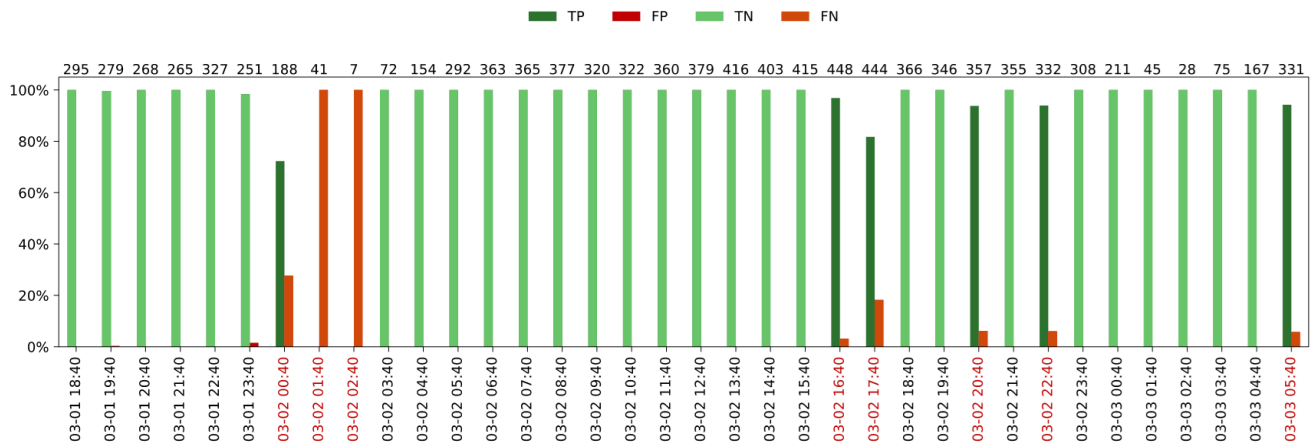

Figure 46: Hour-by-hour histogram

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

During normal hours, the anomaly detector results are effective since it avoids false positives. Most of the normal scenarios that occurred during different time slots (in the morning, afternoon, evening, and night) were recognized as normal situations: 99.7% accuracy. Only 5 false positives were retrieved at the beginning when the model was still unstable (anomalies detected during the night [01:40-03:40]). Moreover, the lack of data, due to the lack of public transport late in the night (or early in the morning, only 48 instances), further complicated the problem. During the day, after 22 hours of training, the anomaly detector appears to be more stable and capable to predict most of the anomalies occurred during the two-hours anomalous time slot [16:40-18:40] in the afternoon: improvement of 29% w.r.t. the previous unstable model. After 26 hours of training, the anomaly detector becomes further stable and capable to predict most of the anomalies occurred during the anomalous time slot [20:40-21:40] in the evening: improvement of 32.4% w.r.t. the previous unstable model. After 28 hours of training, the anomaly detector becomes further stable and capable to predict most of the anomalies occurred during the anomalous time slot [05:40-06:40] in the evening/early morning: improvement of 32.7% w.r.t. the previous unstable model. The global results are showed in Table 5.

Table 5: Results of Oslo public transport (Dataset 3) in terms of accuracy, precision, recall, and F1-score obtained by Spark-GHSOM

Accuracy	91.33%
Precision	99.77%
Recall	85.74%
F1-Score	**92.22%**

The F1-Score is highlighted because it represents a good metric to evaluate the behavior of the predictive models as harmonic mean between the precision and recall. This aspect is important to define a good model to detect both the anomalies and normal cases.

Experiments were conducted also for car traffic analysis for the city of Padua (Dataset 1). The results show very high anomaly identification rates for this dataset (see Table 6).

Table 6: Results of Traffic control of Padua (Dataset 1) in terms of accuracy, precision, recall, F1-score obtained by Spark-GHSOM

Accuracy	99.33%
Precision	97.56%
Recall	100.00%
F1-Score	**98.76%**

Two examples of feature ranking follow. The first ranking shows that when the number of approaching cars is too high, sparkGHSOM considered the variable (“no_approaching”) as the most important to detect the scenario as anomalous. Moreover, the second example shows that some variables could be may not affect the detection of anomalies (see the latitude “y” and the “month” of the year with the lowest importance).

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

```
"timestamp" : "2022-06-27T17:35:00+01:00",
   "coordinates" : [45.446143, 11.893692],
   "ranking" : [
    {
     "feature" : "no_approaching",
     "value" : 150.0,
     "importance" : 0.28
    },
    {
     "feature" : "no_unknown",
     "value" : 150.0,
     "importance" : 0.27
    },
    {
     "feature" : "no_leaving",
     "value" : 0.0,
     "importance" : 0.01
    },
...
   ],
   "anomaly" : "true"
```

```
"timestamp" : "2022-06-27T17:35:00+01:00",
   "coordinates" : [45.434977, 11.89712],
   "ranking" : [
    {
     "feature" : "no_approaching",
     "value" : 150.0,
     "importance" : 0.29
    },
    {
     "feature" : "no_unknown",
     "value" : 150.0,
     "importance" : 0.28
    },
    {
     "feature" : "no_leaving",
     "value" : 0.0,
     "importance" : 0.01
    },
...
    {
     "feature" : "y",
     "value" : 11.9,
     "importance" : 0.0
    },
    {
     "feature" : "month",
     "value" : 6.0,
     "importance" : 0.0
    }
   ],
   "anomaly" : "true"
```


In Table 7 we also report the results obtained by Spark-GHSOM for the people count dataset.

Table 7: Results of People count of Padua (Dataset 2) in terms of accuracy, precision, recall, F1-score obtained by Spark-GHSOM

Accuracy	88.88%
Precision	75.00%
Recall	100.00%
F1-Score	**85.71%**

Two examples of feature ranking for anomaly detection follow, indicating that an increasing number of pedestrians “count_people_in” is prominent in describing an anomalous scenario.


```
"timestamp" : "2022-06-27T17:30:00+01:00",
   "coordinates" : [45.408038845838625, 11.873190997963357],
   "ranking" : [
    {
     "feature" : "count_people_in",
     "value" : 100.0,
     "importance" : 0.29
    },
    {
     "feature" : "count_people_out",
     "value" : 0.0,
     "importance" : 0.0
    }
   ],
   "anomaly" : "true"
```

```
"timestamp" : "2022-06-27T17:30:00+01:00",
   "coordinates" : [45.40814832316576, 11.873665355634742],
   "ranking" : [
    {
     "feature" : "count_people_in",
     "value" : 100.0,
     "importance" : 0.28
    },
    {
     "feature" : "count_people_out",
     "value" : 0.0,
     "importance" : 0.0
    }
   ],
   "anomaly" : "true"
```


Finally, in Table 8 we also report the results obtained by Dencast for the public transport traffic analysis of Oslo (Dataset 3). These results are obtained by using the algorithm in the supervised learning setting, that is, by learning two cluster models (one for the normal class and one for the anomaly class) and labelling new instances as normal if the closest cluster for the normal class is closer than the closest cluster for the anomaly class; anomaly otherwise. As it is possible to see, Dencast achieves 94.74% F1-score by detecting correctly all the existing anomalies within the data (recall 100%). Moreover, F1-score is in line with the results obtained by

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

Spak-GHSOM, although Dencast is used in a supervised learning setting and Spark-GHSOM is used in an unsupervised learning setting.

Table 8: Results of Oslo public transport (Dataset 3) in terms of accuracy, precision, recall, and F1-score obtained by Dencast in a supervised learning setting.

Accuracy	99.33%
Precision	90.00%
Recall	100.00%
F1-Score	**94.74%**

Summary. During the integration sessions, the tool was able to identify potential situations deemed to be abnormal.

In the city of Oslo, higher than normal concentrations of pollutants were identified. Regarding the public transport traffic, quantitative analyses showed that the predictive models for anomaly detection achieved good results in terms of accuracy, precision, recall, and F1-score when sufficient data were provided for training stage.

In the city of Padua, on the other hand, abnormal vehicle transits and abnormal pedestrian flows were correctly identified.

We also report the results obtained for the task of event classification, where the system Dencast is used.

To further improve the results and increase the detection of anomalous situations, we considered to include time information among the descriptive variables, to identify potential variations in data distributions at certain periods of the day.

4.3 Integration with the IMPETUS Platform

This section presents how the integrated solution in Section 4.2 is connected to the IMPETUS platform, supporting the ingestion of the results retrieved from the execution of the analytics algorithm in Chapter 3, or the ingestion of data collected from the cities and sanitized according to the approach in Chapter 2.

The raw data (i.e., those data that did not enter the analytics pipeline yet) stored in the *Big Data Engine* and the results of *Big Data Analytics* are sent to the IMPETUS platform via an encrypted connection on two different Kafka queues (Figure 47).

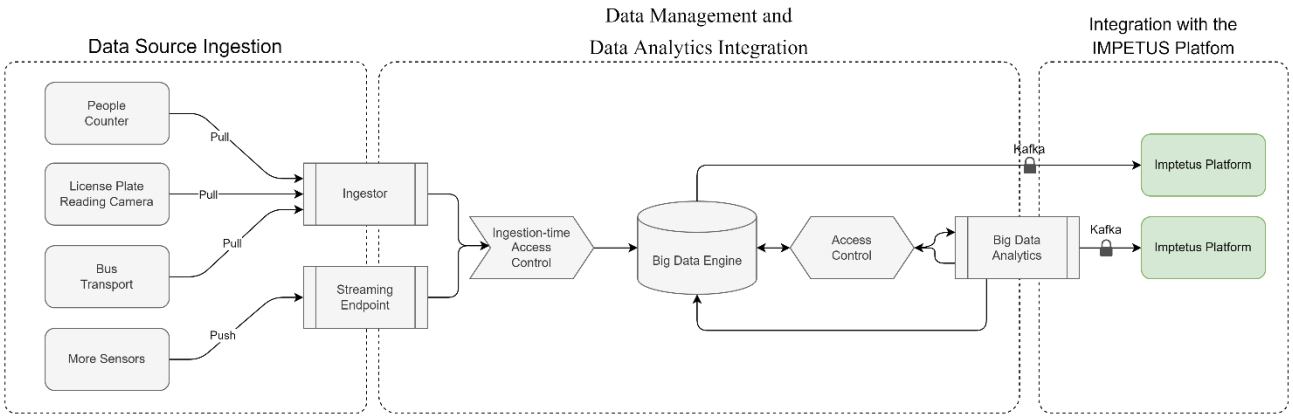

Figure 47: Integration with the IMPETUS Platform

In particular, the data that must be available to the user are sent to a message broker and then stored in the IMPETUS Platform. The role of the message broker is implemented with Apache Kafka. It ensures

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

asynchronous communication between the users and the platform, buffering of the data sent to the platform, and scalability.

The communication with Apache Kafka is encrypted using Transport Layer Security (TLS). The data flowing between the Big Data platform, which is hosted outside the cities networks, and the IMPETUS platform is secured and cannot be inspected by third parties.

The access to the Apache Kafka is not public. The connections are authenticated based on usernames and passwords. In order to send data to the platform, the Big Data platform has to use the credentials of the platform developers.

Once authenticated, the Big Data platform is authorized to write and read only from certain *topics*. Other tools integrated in the IMPETUS platform have no access to the data published by the solution described in this document.

The data resulting from the analytics can be classified as:

- Alerts – they contain the detected anomalies
- Streams – a relevant sample of the input data that can be used by the platform operators to better understand the alerts

The format of the data sent to the platform is JSON. The proposed format for the alerts is:

Field		Description
Header	Source	Tool_Name
	Timestamp	2020-08-30T18:00:00+01:00
	Criticality	The critical level of the notification. This could INFO, MEDIUM, HIGH
	Description	This field contains a description of the message. This is the text initially visible to the user of the platform.
Payload		This field contains the payload of the message. This contains information that can be accessed by the user of the platform. This information details the description text.

An example of output regarding a single test instance follows.

```
"timestamp" : "2022-06-28T10:50:00+01:00",
     "coordinates" : [45.427622, 11.879039],
     "ranking" : [
       {
         "feature" : "no_approaching",
         "value" : 150.0,
         "importance" : 0.3
       },
       {
         "feature" : "no_unknown",
         "value" : 150.0,
         "importance" : 0.28
       },
       {
```

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

```
  "feature" : "hour",
  "value" : 10.0,
  "importance" : 0.08
},
{
  "feature" : "day",
  "value" : 28.0,
  "importance" : 0.08
},
{
  "feature" : "minute",
  "value" : 50.0,
  "importance" : 0.08
},
{
  "feature" : "cameraname",
  "value" : "LT_2_010B_LT VIANELLO-BUONARROTI VS ROTONDA BUONARROTI",
  "importance" : 0.08
},
{
  "feature" : "datetimedayoftheweek",
  "value" : 3.0,
  "importance" : 0.07
},
{
  "feature" : "x",
  "value" : 45.43,
  "importance" : 0.02
},
{
  "feature" : "no_leaving",
  "value" : 0.0,
  "importance" : 0.01
},
{
  "feature" : "y",
  "value" : 11.88,
  "importance" : 0.0
},
{
  "feature" : "month",
  "value" : 6.0,
  "importance" : 0.0
```


The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

```
        }
      ],
      "anomaly" : "true"
    }
```

Such a test instance represents a detected anomaly (“anomaly”:”true”). The tool also indicates the reasons why it is considered an anomalous instance at that specific time (timestamp) in that specific place (coordinates). Indeed, the most important variables (features) under analysis are shown at the top of the provided ranking indicating the most important reason why the anomaly occurred. In this specific example, the first block

```
"feature" : "no_approaching",
"value" : 150.0,
"importance" : 0.3
```

indicates that the most prominent “alarming” variable is the number of approaching cars nearby the main plaza of Padua *because* the number of cars is 150. These data contribute with importance of 30% on the instance to detect this observation as anomalous.

We finally note that the results retrieved by the algorithms in this document provide the end users, and in particular the SOC operators, with spatio-temporal information on anomalies that might also be used as indicators of other and more critical events. For instance, an unexpected traffic jam or changes in the paths followed by public transportations could be due to an incident or a terroristic attack that might be promptly managed by SOC operators at the city.

The effectiveness of the triggered anomalies and events strongly depends on the ability to report them to the SOC operators in a clear and proper way. To this aim, a User Interface (UI) will be implemented and integrated in the IMPETUS platform dashboard to visualise anomalies and events captured by the implemented algorithms. The UI will be reported in deliverable D4.3 and will provide the end users with information about the specific anomalies and events, the geographical areas impacted by them, and the list of reasons (with their contribution) for raising a particular alarm.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

5 Conclusions

D4.2 presented an advanced Big Data solution for the smart city domain, supporting analytics for anomaly detection and event classification, and allowing fine-grained data management and protection.

It also described how this solution has been integrated into the IMPETUS platform and applied to the scenarios of the cities of Padua and Oslo.

During the integration, we refined the data governance approach proposed in D4.1 to achieve full data management. Moreover, we extended the analytics algorithms in terms of interpretability and quality, expanding the interconnection capabilities of our approach and selecting suitable services from the Apache ecosystem (e.g., NiFi, Druid) for security purposes.

Results of our integrated solution have been finally provided, demonstrating its application in the real environment of the cities part of the IMPETUS consortium.

The contribution of D4.2 lies in two main areas.

First, we implemented a new data governance solution based on access control. The data governance solution enriches traditional data management components with an access control system that enforces access to data in a distributed, multi-party Big Data environment. More specifically the data governance solution presented in this document extends the previous solution described in D4.1 by allowing full data governance at data analytics time.

Access control is enforced at each step of a data analytics pipeline, supporting a finer-grained data governance, meaning that the access control system works at both ingestion and analytics time.

To the best of our knowledge, this is the first attempt to bring fine-grained data control and multi-tenancy into a Big Data architecture grounded on the Apache ecosystem.

Second, we defined new anomaly detection and event classification algorithms. The considered state-of-the-art methods for anomaly detection and event classification have been updated to be compliant with and tailored on the IMPETUS platform:

- *i)* the anomaly detection task was updated to produce interpretable outputs which show not only the anomalies detected, but also the ranking of the variables that most contributed to detecting the anomalies. This feature helps for better explaining the real motivations of the physical threats detected;
- *ii)* the training stage now allows saving pre-trained, well-configured models through in-lab experiments in order to reuse them also for the real scenarios of Padua and Oslo. In this way, it is possible to perform the time-consuming step of parameters tuning in advance, with the aim of reducing the effort during live tests;
- *iii)* the output format is easily interpreted by other tools and shared via Apache Kafka in a human and machine-readable format.

In summary, the proposed Big-Data platform implements a solution that can be customised to ingest, protect and analyse any type of data, collecting anomalies and events suitable to the specific needs of the final user. The anomalies and events returned by the solution provide spatio-temporal information of the existence of a potential problem that cannot be simply detected neither by manual analysis, nor by other classical tools that only define a simple alert threshold.

Possible improvements built on the results of D4.2 can focus on:

- *i)* fostering the adoption of the Big Data solution in different scenarios, with heterogeneous technologies, data formats and standards;
- *ii)* extending our access control system to manage access to analytics parameters and configurations, providing an improved governance of analytics at low cost;
- *iii)* providing a dynamic approach to the autotuning of ML parameters to increase the overall algorithms usability.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

6 References

[1] M. Anisetti, A. Appice, C. A. Ardagna and et al., D4.1 - Data analytics and ingestion-time access control initial report, IMPETUS Project, 2021, 2021.

[2] M. Anisetti, C. A. Ardagna, C. Braghin, E. Damiani, A. Balestrucci and A. Polimeno, "Dynamic and Scalable Enforcement of Access Control Policies for Big Data," in *ACM MEDES*, Virtual, 2021.

[3] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. of ACM CCS*, Alexandria, USA, November 2006.

[4] X. Boyen and B. Waters, "Anonymous hierarchical identity-based encryption (without random oracles)," in *Proc. of CRYPTO*, Santa Barbara, USA, August 2006.

[5] F. Armknecht, C. Boyd, C. Carr, K. Gjøsteen, A. Jäschke, C. A. Reuter and M. Strand, "A guide to fully homomorphic encryption," *IACR Cryptology ePrint,* vol. 2015, no. 1192, 2015.

[6] R. C. Hill and H. Lockhartr, "eXtensible Access Control Markup Language (XACML) Version 3.0," in *OASIS Standard*, https://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html, 2013.

[7] A. Malondkar, R. Corizzo, I. Kiringa and M. Ceci, "Spark-GHSOM: Growing Hierarchical Self-Organizing Map for large scale mixed attribute datasets," *Information Sciences,* vol. 496, pp. 572-591, 2019.

[8] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter and L. Kagal, "Explaining Explanations: An Overview of Interpretability of Machine Learning," in *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, 2018.

[9] R. Corizzo, G. Pio, Ceci, M. and et al., "DENCAST: distributed density-based clustering for multi-target regression," *J Big Data,* vol. 6, no. 43, 2019.

[10] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *Proc. of 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

Members of the IMPETUS consortium

SINTEF	SINTEF, Strindvegen 4, Trondheim, Norway, https://www.sintef.no	Joe Gorman joe.gorman@sintef.no
Institut Mines-Télécom	Institut Mines Telecom, 19 place Marguerite Perey, 91120 Palaiseau, France, https://www.imt.fr	Joaquin Garcia-Alfaro joaquin.garcia_alfaro@telecom-sudparis.eu
UNÎMES	Université de Nimes, Rue du Docteur Georges Salan CS 13019 30021 Nîmes Cedex 1, France, https://www.unimes.fr	Axelle Cadiere axelle.cadiere@unimes.fr
cini	Consorzio Interuniversitario Nazionale per l'Informatica, Via Ariosto, 25, 00185 – Roma, Italy, https://www.consorzio-cini.it	Donato Malerba donato.malerba@uniba.it
Università degli Studi di Padova	University of Padova, Via 8 Febbraio, 2 - 35122 Padova, Italy, https://www.unipd.it	Giuseppe Maschio giuseppe.maschio@unipd.it
Entrepreneurship Development Centre for BIOTECHNOLOGY and MEDICINE	Biotehnoloogia ja Meditsiini Ettevõtluse Arendamise Sihtasutus, Tiigi 61b, 50410 Tartu, Estonia, https://biopark.ee	Sven Parkel sven@biopark.ee
SIMAVI Software Imagination & Vision	SIMAVI, Complex Victoria Park, Corp C4, Etaj 2, Șos. București – Ploiești, nr. 73 – 81, Sector 1, București, Romania, https://www.simavi.ro	Gabriel Nicola Gabriel.Nicola@simavi.ro Monica Florea Monica.Florea@simavi.ro
THALES	Thales Nederland BV, Zuidelijke Havenweg 40, 7554 RR Hengelo, Netherlands, https://www.thalesgroup.com/en/countries/europe/netherlands	Johan de Heer johan.deheer@nl.thalesgroup.com
CINEDIT INTELLIGENT VIDEO ANALYTICS	Cinedit VA GmbH, Poststrasse 21, 8634 Hombrechtikon, Switzerland, https://www.cinedit.com	Joachim Levy j@cinedit.com

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.

INSIKT INTELLIGENCE	Insikt Intelligence, Calle Huelva 106, 9-4, 08020 Barcelona, Spain, https://www.insiktintelligence.com	Dana Tantu dana@insiktintelligence.com
SIXGILL	Sixgill, Derech Menachem Begin 132 Azrieli Tower, Triangle Building, 42nd Floor, Tel Aviv, 6701101, Israel, https://www.cybersixgill.com	Benjamin Preminger benjamin@cybersixgill.com Ron Shamir ron@cybersixgill.com
XM CYBER	XM Cyber, Galgalei ha-Plada St 11, Herzliya, Israel https://www.xmcyber.com	Lior Barak lior.barak@xmcyber.com Menachem Shafran menachem.shafran@xmcyber.com
Comune di Padova	City of Padova, via del Municipio, 1 - 35122 Padova Italy, https://www.padovanet.it	Enrico Fiorentin fiorentine@comune.padova.it Stefano Baraldi Baraldis@comune.padova.it
Oslo	City of Oslo, Grensen 13, 0159 Oslo, Norway, https://www.oslo.kommune.no	Osman Ibrahim osman.ibrahim@ber.oslo.kommune.no
Institut za sigurnosne politike ISP	Institute for Security Policies, Kruge 9, 10000 Zagreb, Croatia, http://insigpol.hr	Krunoslav Katic krunoslav.katic@insigpol.hr
TIEMS	International Emergency Management Society, Rue Des Deux Eglises 39, 1000 Brussels, Belgium, https://www.tiems.info	K. Harald Drager khdrager@online.no
UniSMART Fondazione Università di Padova	Unismart – Fondazione Università degli Studi di Padova, Via VIII febbraio, 2 - 35122 Padova, Italy, https://www.unismart.it	Alberto Da Re alberto.dare@unismart.it

The research leading to these results has received funding from Horizon 2020, the European Union's Framework Programme for Research and Innovation (H2020) under grant agreement n° 883286.