Robust Learning and Reasoning for Complex Event Forecasting

Project Acronym:	EVENFLOW
Grant Agreement number:	101070430 (HORIZON-CL4-2021-HUMAN-01-01 – Research and Innovation Action)
Project Full Title:	Robust Learning and Reasoning for Complex Event Forecasting

DELIVERABLE

D3.2 – Interim Use Case Evaluation

Dissemination level:	PU - Public, fully open
Type of deliverable:	R - Document, report
Contractual date of delivery:	31 March 2024
Deliverable leader:	BSC
Status - version, date:	Final – v1.0, 2024-03-29
Keywords:	use cases, accomplishments, future plans

Funded by the European Union

This document is part of a project that is funded by the European Union under the Horizon Europe agreement No 101070430. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Commission. The document is the property of the EVENFLOW project and should not be distributed or reproduced without prior approval. Find us at www.evenflow-project.eu.

Executive Summary

This deliverable D3.2 provides an overview of the latest advancements and future directions for the EVEFLOW project, building on the foundational deliverable D3.1. It highlights progress across the project's three key use cases (Industry 4.0, Personalized Medicine, and Infrastructure Life Cycle Assessment), marking significant advancements in integrating AI with practical applications.

The Industry 4.0 use case deployed an autonomous mobile robot equipped with state-of-the-art navigation technologies, representing a significant step towards more efficient and proactive intralogistics operations. The Personalized Medicine use case has seen substantial advancements in analysing cancer progression, utilizing AI for synthetic data generation to reveal new insights into tumour dynamics and showcasing the potential for technological innovations in cancer research. The Infrastructure Life Cycle Assessment use case is advancing in its use of AI to boost equipment efficiency and predictive maintenance, aiming to significantly influence infrastructure management approaches.

The document also outlines upcoming strategic steps for each use case, focusing on rigorous testing, validation of research findings, and the expansion of current understandings through advanced data analysis techniques. Additionally, it outlines the project's approach to middleware infrastructure deployment, underscoring a comprehensive strategy for supporting these technological advancements.

Deliverable leader:	Davide Cirillo (BSC)
Contributors:	Guillermo Prol Castelo (BSC) Benjamin Blumhofer (DFKI) Karim Ladjeri (EKSO) Ioannis Christou (INTRA)
Reviewers:	Athanasios Poulakidas (INTRA), Karim Ladjeri (EKSO)
Approved by:	Athanasios Poulakidas, Dimitrios Liparas (INTRA)

Document History			
Version	**Date**	**Contributor(s)**	**Description**
0.1	2024-02-20	BSC	ToC preparation
0.2	2024-02-29	DFKI	Industry 4.0 UC draft
0.3	2024-03-11	INTRA	Middleware deployments
0.4	2024-03-26	BSC	Personalized UC, Introduction and Conclusions
0.5	2024-03-28	EKSO	Infrastructure Life Cycle UC and review
0.9	2024-03-29	INTRA	Review
1.0	2024-03-29	INTRA	QA and final version for submission

Table of Contents

Table of Figures

List of Tables

Acronyms and Abbreviations

Acronym/ Abbreviation	Title
AI	Artificial Intelligence
AMR	autonomous mobile robot
CAD	Computer-Aided Design
CEF	complex event forecast
CSV	Comma-Separated Values
DOL	dynamic Obstacle Layer
EoL	End of Life
KPI	Key Performance Indicator
ksps	kilo samples per second
LAB	Laboratory
LCA	Life Cycle Assessment
LiDAR	Light Detection and Ranging
MB	Megabyte
MEMS	Micro-Electro-Mechanical Systems
NAV2	Navigation 2
NCSR	National Center for Scientific Research
PCL	Predictive Costmap Layer
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
ROS2	Robot Operating System 2
RUL	Remaining Useful Life
SLAM	Simultaneous Localization and Mapping
TCGA	The Cancer Genome Atlas
UNIPA	University of Palermo
VAE	Variational Autoencoder
AI	Artificial Intelligence
AMR	autonomous mobile robot

1 Introduction

1.1 Project Information

EVENFLOW is developing hybrid learning techniques for complex event forecasting, which combine deep learning with logic-based learning and reasoning into neuro-symbolic forecasting models. The envisioned methods combine (i) neural representation learning techniques, capable of constructing event-based features from streams of perception-level data with (ii) powerful symbolic learning and reasoning tools, that utilize such features to synthesize high-level, interpretable patterns of critical situations to be forecast.

Crucial in the EVENFLOW approach is the online nature of the learning methods, which makes them applicable to evolving data flows and allows to utilize rich domain knowledge that is becoming available progressively. To deal with the brittleness of neural predictors and the high volume/velocity of temporal data flows, the EVENFLOW techniques rely on novel, formal verification techniques for machine learning, in addition to a suite of scalability algorithms for federated training and incremental model construction. The learnt forecasters will be interpretable and scalable, allowing for fully explainable insights, delivered in a timely fashion and enabling proactive decision making.

EVENFLOW is evaluated on three challenging use cases related to (1) oncological forecasting in precision medicine, (2) safe and efficient behaviour of autonomous transportation robots in smart factories and (3) reliable life cycle assessment of critical infrastructure monitoring.

Expected impact:

- New scientific horizons in integrating machine learning and machine reasoning, neural, statistical and symbolic AI
- Breakthroughs in verification, interpretability and scalability of neuro-symbolic learning systems
- Interpretable, verifiable and scalable ML-based proactive analytics and decision-making for humans-in-the-loop and autonomous systems alike
- Robust, resilient solutions in critical sectors of science and industry
- Accurate and timely forecasting in vertical sectors (healthcare, Industry 4.0, critical infrastructure monitoring)
- Novel FAIR datasets for scientific research
- Novel resources and approaches for verifiable, interpretable, scalable and knowledge-aware machine learning

Table 1: The EVENFLOW consortium.

Number[1]	Name	Country	Short name
1 (CO)	NETCOMPANY-INTRASOFT	Belgium	**INTRA**
1.1 (AE)	NETCOMPANY-INTRASOFT SA	Luxemburg	**INTRA-LU**

[1] CO: Coordinator. AE: Affiliated Entity. AP: Associated Partner.

Number[1]	Name	Country	Short name
2	NATIONAL CENTER FOR SCIENTIFIC RESEARCH "DEMOKRITOS"	Greece	**NCSR**
3	ATHINA-EREVNITIKO KENTRO KAINOTOMIAS STIS TECHNOLOGIES TIS PLIROFORIAS, TON EPIKOINONION KAI TIS GNOSIS	Greece	**ARC**
4	BARCELONA SUPERCOMPUTING CENTER-CENTRO NACIONAL DE SUPERCOMPUTACION	Spain	**BSC**
5	DEUTSCHES FORSCHUNGSZENTRUM FUR KUNSTLICHE INTELLIGENZ GMBH	Germany	**DFKI**
6	EKSO SRL	Italy	**EKSO**
7 (AP)	IMPERIAL COLLEGE OF SCIENCE TECHNOLOGY AND MEDICINE	United Kingdom	**ICL**

1.2 Document Scope

This document serves as a follow-up to the previous deliverable D3.1, extending the exploration and analysis of the progress of the three EVEFLOW use cases: Industry 4.0, Personalized Medicine, and Infrastructure Life Cycle Assessment. Building on the foundational work of describing data handling, requirements analysis, and scenario definition, this document aims to advance the project's objectives by providing a comprehensive update on the progress, challenges, and future directions of the EVEFLOW project. It ties the previously established groundwork of data management strategies, which encompass the collection, processing, and storage of data, with the understanding of use case requirements that have been carefully gathered, documented, and validated. Furthermore, it builds upon the initial scenario definitions that have been crafted to anticipate and mitigate potential challenges within the application domains of the project's use cases. This document is designed to ensure continuity and coherence in the project's development, facilitating a smooth transition from theoretical frameworks and preparatory analyses to practical application and strategic planning for the future. Finally, the document details the latest developments in deploying middleware infrastructure on the NCSR compute servers.

1.3 Document Structure

This document is structured to provide a comprehensive understanding of the interim evaluation of the EVENFLOW use cases, elaborating on their developments and future directions. The content is organized into distinct chapters, each focusing on a pivotal aspect. Below is a detailed outline of the chapters included in this document:

Chapter 1 sets the stage for the document, offering readers a concise overview of the EVEFLOW project's background, its significance, and the core objectives it aims to achieve. This section is designed to provide a solid foundation for understanding the broader context of the discussions that follow.

Chapter 2 "Use Cases Objectives" delves into the specific aims of the project, particularly focusing on its innovative use cases: Industry 4.0, Personalized Medicine, and Infrastructure Life Cycle Assessment. This chapter aims to elucidate how each use case contributes to the overarching goals of the EVEFLOW project, highlighting their potential impacts and benefits in their respective fields.

Chapter 3 "Status Update on EVENFLOW Use Cases Development" provides a current snapshot of the project's progress, offering insights into the development stages of the use cases. This chapter critically examines the achievements and challenges encountered thus far, shedding light on the practical aspects of bringing the project's ambitious visions to life.

Chapter 4 "Roadmap for Future Steps" outlines the strategic direction for the continuation of the project. This chapter presents a forward-looking perspective, detailing the planned actions, milestones, and objectives for the next phases of the EVEFLOW project. It serves as a guide for readers to understand the future trajectory and the expected developments in the project.

The document concludes with the **Conclusions** section, which synthesizes the key findings, insights, and lessons learned throughout the course of the project. This section also reflects on the project's impact and its contribution to the fields of Industry 4.0, Personalized Medicine, and Infrastructure Life Cycle Assessment, offering a closing evaluation of the project's status and future prospects.

Appendix A "Middleware infrastructure deployment" provides a comprehensive update on the deployment of middleware infrastructure on the NCSR compute servers for data handling, detailing steps from Kubernetes cluster deployment to infrastructure access. This section serves as a technical guide, covering the installation of core components, services, and essential Kubernetes and Kafka commands.

2 Use Cases Objectives

2.1 Summary

This chapter is dedicated to outlining the objectives within the EVEFLOW project's use cases. The **Industry 4.0** use case aims at achieving autonomous transportation of sensitive cargoes within a factory setting efficiently and on time. It seeks to enhance robot navigation by incorporating forecasting capabilities into the cost maps, allowing for proactive path planning and collision avoidance through timely replanning. The **Personalized Medicine** use case is focused on utilizing latent representation learning models to analyse cancer progression data. The goal is to track probabilistic trajectories of data points in latent space to predict adverse outcomes and improve molecular characterization. For the **Infrastructure Life Cycle Assessment** use case, the objective is to leverage AI in the manufacturing sector to improve overall equipment efficiency, minimize waste, and facilitate predictive maintenance. This includes the application of smart technologies in water distribution and critical infrastructure for early detection of leaks and cost reduction in infrastructure management.

2.2 Industry 4.0

This use case aims to explore Neurosymbolic AI within the context of Industry 4.0, focusing specifically on the application of intralogistics. The primary objective is to develop predictive capabilities for anticipating delays and potential collisions within intralogistics operations.

The overarching goal is to proactively avoid situations that could lead to collisions with robots, thereby enhancing safety and efficiency in industrial environments. To achieve this, we employ a multi-tiered approach, considering various abstraction levels. The initial phase of the project concentrates on the navigation of individual robots.

During this phase, our focus is on predicting the movements of other robots operating on the shop floor to prevent scenarios where robots are in proximity and must engage in avoidance manoeuvres, which can significantly impact operational time. To facilitate this, we are developing a semantically predictive map and path-planning algorithms.

Our development efforts are built upon the ROS2 framework, leveraging its capabilities to facilitate seamless integration and efficient implementation of our predictive systems. In the latter part of the project, attention will shift to evaluating the system-level dynamics. This involves analysing how events involving multiple robots and production modules interact, leading to delays in the overall system.

By thoroughly examining these interactions, we aim to identify bottlenecks and inefficiencies, ultimately devising strategies to mitigate delays and optimize the performance of Industry 4.0 intralogistics systems.

2.3 Personalized Medicine

The Personalized Medicine use case of the EVEFLOW project is designed to address critical gaps in cancer research, specifically in understanding the molecular dynamics of cancer progression. This objective stems from the acknowledgment of the current limitations within the field, including the significant scarcity of comprehensive molecular data across the various stages of tumour progression. Such limitations are mainly due to the challenges in obtaining extensive samples that cover the entire trajectory of cancer development in humans. Given this context, the use case aims to leverage vast repositories of public anonymous human molecular data, such as those found in The Cancer Genome Atlas (TCGA), which includes RNA expression data profiling among others, despite its fragmented nature. The overarching goal is to leverage artificial intelligence (AI) techniques, particularly those involving synthetic data generation like Variational Autoencoders (VAE), to fill the gaps in our understanding of tumour progression, with a specific emphasis on the stages of breast invasive carcinoma.

Given the critical challenges currently faced in cancer research, the Personalized Medicine use case is tailored to address the following objectives:

- **Dynamic reconstruction from static data:** Develop methodologies capable of piecing together the dynamic progression of cancer from static, isolated data points, or "snapshots." This involves using advanced AI tools to infer the progression over time.
- **Handling complex data:** Address the challenge of working with high-dimensional molecular data that, in the case of breast cancer stages, also suffers from low separability. This objective focuses on employing and refining AI techniques that can effectively reduce dimensionality while preserving critical information, thereby improving the analysis and interpretation of cancer progression.
- **Forecasting clinically-relevant molecular changes:** Aim to accurately predict future changes in gene expression that have significant clinical relevance. By forecasting these changes, the use case intends to provide actionable insights that can lead to the development of prevention actions or targeted treatment strategies, ultimately improving patient outcomes. This involves not only identifying potential molecular targets but also understanding their evolution over the course of the disease.

By meeting these objectives, the Personalized Medicine use case aims to significantly advance the field of cancer research, offering new avenues for the diagnosis, treatment, and management of breast cancer, and also addressing a pervasive challenge in biomedicine: the challenge of understanding complex biological changes over time with limited data points. This endeavour not only has the potential to set a precedent for tackling similar issues across a broad range of biomedical inquiries, where dynamic events are often only captured as static moments in time.

2.4 Infrastructure Life Cycle Assessment

This use case is developing a digital twin of a water pipe section, which will identify and forecast LCA states and incidents ahead of time and with high accuracy. This will enable EKSO to make educated, data-driven decisions about maintenance, service, and repairs of the pipes, including the implementation of refurbishment operations when required sizing and locating them at the best. The use of the EVENFLOW platform will focus on locating and forecasting of non-trivial LCA events such as defects, malfunctions, and prediction of the End of Life (EoL) or the RUL of the pipe, as well as LCA states that lead to poor performance. For the implementation of the CEF, design characteristics of the pipe will be considered as domain knowledge. Using the digital twin and datasets collected from a pipe, we will also implement an engine for what-if simulations.

The EVENFLOW platform will be also used to formally verify the LCA states of the pipe based on the investigation of perturbations in the sensory input that may lead to different LCA outcomes. The interpretable nature of EVENFLOW forecasts will be used to identify sequences of events that could prolong the lifetime of the pipe and/or improve its environmental performance (i.e. reduce CO2 emissions). EVENFLOW will also enable the development of an on-line learning loop, which will provide continuous feedback and insights about the optimal maintenance of the pipe. Interpretability of forecasts will also increase transparency and boost the adoption of EVENFLOW by EKSO and its customers.

The primary source of data for the use case will be from MEMS/vibration sensors on the pipe, complemented with relevant domain knowledge. EKSO has already collected datasets from refurbished pipes, but additional data will be collected, and labelled. Furthermore, additional datasets will be considered in order to increase the accuracy of the analysis and provide opportunities for better forecasts.

3 Status Update on Use Cases Development

3.1 Summary

This chapter provides updates on the progress of the EVENFLOW use cases. In the **Industry 4.0** use case, an autonomous mobile robot (AMR) developed on the Festo Robotino platform incorporates advanced sensing, navigation, and predictive planning technologies, showcasing enhanced capabilities for dynamic environment navigation and operational optimization in intralogistics. The **Personalized Medicine** use case leverages AI for synthetic data generation and neuro-symbolic technologies to explore breast invasive carcinoma progression, revealing novel insights into cancer stage transitions through RNA-Seq data analysis and pathway enrichment, highlighting AI's potential to uncover new dynamics in tumour progression. The **Infrastructure Life Cycle Assessment** use case is developing the datasets analysis gathered from a pipe section, with the aim of identifying states and incidents ahead of time and with high accuracy.

3.2 Industry 4.0

The autonomous mobile robot (AMR) is based on the Festo Robotino platform, which was optimized specifically for our use case according to the state of the art. This platform was developed for educational purposes in 2014. It is characterized by an omnidirectional drive that allows it to move in all directions and rotate on the spot.

This platform was extended by us to transport production objects and products within an intralogistics system. In order to provide a state-of-the-art industrial robot for the use case, it had to be equipped with additional hardware and updated with the latest software. A LiDAR (Light Detection and Ranging) sensor from the RoboSense- Helios series is used for more precise environment detection to get precise 3D scans of the surroundings. By emitting and receiving laser beams, the LiDAR measures distances to objects and outputs a point cloud, which is further processed in the robot system.

To further increase the accuracy of detection and localization, a ZED2 stereo camera from Stereolabs was mounted in the direction of travel. The depth analysis and the additional Built-in sensors like a Inertial Measurement Unit (IMU), barometer & magnetometer enables a sensor fusion of both sensor systems. This enables us to reliably detect dynamic environments. In addition, image recognition can be realized by the ZED2 for specific tasks.

ROS2 (Robot Operating System 2) is used as the robot system for the AMR. ROS2 is a modular and real-time capable open-source framework that offers comprehensive support and integration for a wide range of hardware from different manufacturers. Its strength lies in its modular architecture, which enables flexible and efficient development, and a community-oriented approach that greatly simplifies and accelerates research and development in the field of robotics. It also offers enhanced security features, cross-platform compatibility and a rich ecosystem of tools, libraries and packages that work together to provide developers with robust, scalable, and secure applications. In the context of autonomous mobile robots, the navigation system plays an essential role. In ROS2, NAV2 is used as part of the navigation stack for navigation challenges. It provides functional modules for autonomous path planning, obstacle avoidance, localization and mapping that enable robots to move autonomously in environments.

One aspect of NAV2 is the costmap approach, which maps the environment in a grid-based format and assigns a "cost value" to each cell (or pixel). This value represents the passability or safety associated with moving through that cell. A distinction is made between a global and local map, which have different influences on navigation. The global cost map uses a previously created static map of the environment, which contains fixed obstacles such as walls, furniture or other static objects. This map is often created using a separate mapping process such as SLAM (Simultaneous Localization and Mapping). The global cost map is used to plan a path from one point to another over longer distances within the known environment. It considers the entire robot footprint and applies inflation zones around obstacles to ensure a safe distance that takes into account the size of the robot. This cost map does not change in real time but can be updated periodically to reflect changes in the static environment. In contrast, the local cost map is focused on the robot's immediate environment and uses real time data from on-board sensors such as laser scanners, cameras or sonars to generate a map that is constantly updated.

This partially dynamic map contains moving and emerging obstacles in time segments that are not included in the global map. The local cost map is responsible for short-term navigation and simple obstacle avoidance. It enables the robot to react to changes in its immediate environment and adapt accordingly. Each cost map comprises several layers that represent different information and environmental features:

- The **Static Layer** reflects the static elements of the environment based on a previously created map.
- The **Obstacle Layer** is updated more frequently by sensor information and captures temporary obstacles that appear over a time period.
- The **Inflation Layer** extends the cost values around known obstacles to create a safety buffer and prevent collisions.

This layer structure provides a basic framework for navigation to be able to navigate autonomously in simple environments. In a highly dynamic environment and busy environments, such as city centers and Industry 4.0 environments with human interaction, the basic layers reach their limits. Especially the handling of dynamic objects, people and the differentiation between the different objects, which require individual, special safety distances, is a current problem. In addition, there is a lack of predictive approaches that can predict the trajectories of unknown objects, people or robots. These predictive approaches should enable predictive planning to increase safety and avoid downtimes caused by blockages. In the industrial environment, blockages and in the worst case, deadlocks are particularly critical, as they hinder intralogistics and just-in-time production. Such downtimes can cost the company a large sum each year.

Our approach is designed to solve these problems by adding additional costmap layers as plugins. Figure 1 shows the navigation pipeline with state of the art (marked in blue) and beyond state of the art (marked in orange). In addition, the normal navigation pipeline is represented from the blue arrow and with our additional approach (orange arrows) is shown.

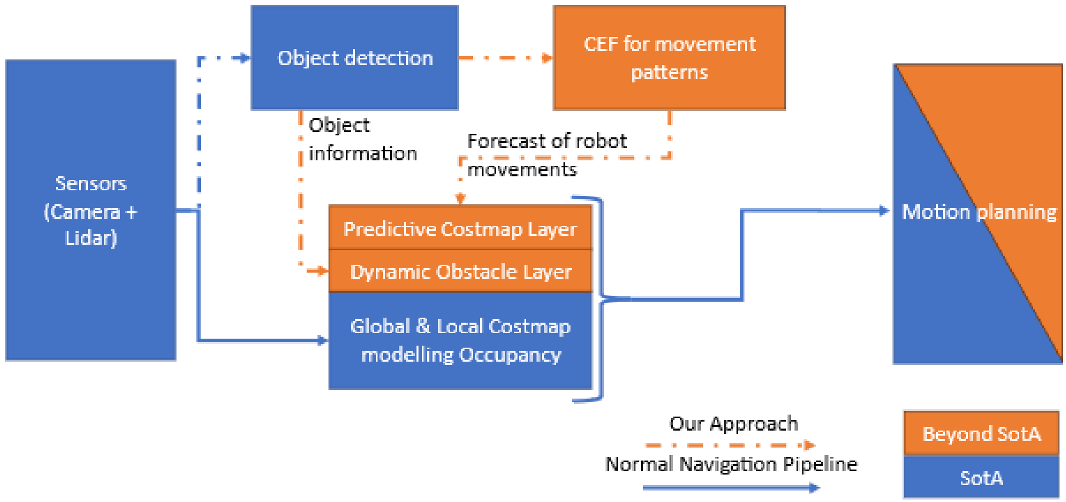

Figure 1. Navigation pipeline (blue: state of the art; orange: beyond state of the art).

The first new costmap layer serves as the basis for the integration of the EVENFLOW complex event forecast (CEF). This "dynamic Obstacle Layer" (DOL) captures dynamic objects and assigns their specific captured properties through their semantic information. This allows detected dynamic moving objects to be described and mapped in the costmap for robot navigation. With the stereo camera aligned in the direction of travel, semantic recognition of the objects can be carried out in the specific local field of view. This semantic information, such as object class, position, orientation, velocity vector, action state, and object dimensions, is integrated into our first costmap layer. This knowledge allows us to assign a defined safety distance to each detected object. Furthermore, the velocities are approximated with a Kalman filter and additionally blocked in the costmap as the direction of movement of the object. This approach allows for short-term predictive driving planning, which thus enables the robot to react adaptively and flexibly to unexpected changes in the environment.

We are currently developing a new layer for the integration of the EVENFLOW CEF, which will be used in the global and local costmap. In contrast to DOL, the Predictive Costmap Layer (PCL) can enable predictive travel planning over long distances. The PCL will expand the information of the DOL with the result of the CEF. A possible trajectory for the object can be calculated using the CEF information on the destination and the speed vectors of the detected object. The velocity information also makes it possible to determine the future positions of the object along the possible trajectory. This knowledge and the comparison of the calculated trajectories can be used, to determine bottlenecks and overlaps in the trajectories. These critical points are assigned higher costs in the PCL. By predicting potential movement paths of dynamic obstacles, the predictive cost map enables forward-looking planning and adaptation of navigation strategies. This further reduces the risk of safety-critical situations and contributes to smoother and more efficient navigation by incorporating not only the current but also potential future states of the environment into the planning.

In this Use Case significant progress has been made in developing a robust simulation environment to generate essential data for analysis and prediction. This simulation environment serves as a crucial component in our quest to enhance forecasting accuracy and efficiency within real-world factory settings.

Recognizing the necessity to simulate realistic scenarios reflective of actual factory operations, considerable effort has been devoted to ensuring the fidelity of the simulation environment. Leveraging CAD models and the layout of the testbed, we've strived to replicate the intricacies of the physical environment within the digital realm. However, this endeavour has not been without its challenges.

One of the primary complexities arises from the modular nature of modern production facilities, where objects are not statically positioned but dynamically rearranged based on operational needs. To address this, objects have been categorized into distinct classes, each governed by specific rules dictating their placement. For instance, certain production modules may only be situated at designated docking stations, necessitating careful consideration during simulation modelling.

Furthermore, to accurately emulate the behaviour of robots within the simulated environment, we've established a seamless interface between the simulation environment and ROS2. This integration ensures that the same navigation algorithms and control mechanisms are employed in both simulation and reality, fostering consistency and reliability in our analyses.

The initial phase of simulation involved the deployment of a single robot within the factory environment over a simulated period of 12 months. While this scenario provided valuable insights, it was deemed relatively straightforward to predict, prompting a re-evaluation of our approach. Through collaborative discussions with project partners, it was unanimously agreed upon to expand the scope of simulation by incorporating an additional robot.

However, the integration of a second robot posed substantial technical challenges, particularly within the Isaac Sim simulation environment. Adjustments were required to accommodate the randomization of target points for robot navigation, dynamic object placements, and seamless ROS2 integration. Compounding these challenges was the limited documentation available, necessitating extensive trial and error to overcome implementation hurdles.

Despite these obstacles which delayed the generation of appropriate data for neurosymbolic AI processing, significant progress has been achieved, culminating in the successful generation of datasets capturing various facets of robot behaviour and environmental dynamics. Each dataset encompasses a wealth of information, including visited targets, images captured by robot-mounted cameras, spatial coordinates, velocities, orientations, system states, prediction labels, and collision statuses.

Moreover, to facilitate future analyses and scenario replays, the generated simulation scenarios have been meticulously saved in .usd file format, ensuring reproducibility and continuity in our research endeavours. Additionally, detailed maps depicting the layout of the testbed for each scenario have been meticulously curated, providing invaluable context for subsequent analyses.

Looking ahead, the simulation environment will continue to serve as a cornerstone in our research efforts, enabling us to explore and refine forecasting models under diverse and evolving conditions. By iteratively enhancing the fidelity and sophistication of our simulations, we aspire to push the boundaries of predictive accuracy and operational optimization within intralogistics contexts.

In summary, while the journey to develop and refine our simulation environment has been fraught with challenges, the outcomes thus far underscore our unwavering commitment to advancing the frontiers of Complex Event Forecasting within intralogistics. Through collaborative innovation and persistent dedication, we remain steadfast in our pursuit of transformative insights and solutions for real-world challenges in industrial automation and logistics management.

3.3 Personalized Medicine

The Personalized Medicine use case aims to unveil new insights into the evolution of breast invasive carcinoma with the use of AI-based synthetic data generation and neuro-symbolic technologies. Below, we detail the latest updates and significant findings from our ongoing research in this critical area.

Systematic Literature Review on AI Applications in Cancer Progression

BSC performed a meticulous literature review aimed at uncovering the landscape of AI applications for the temporal reconstruction of cancer stages, adhering to the PRISMA guidelines (www.prisma-statement.org). We delved into major academic databases such as Google Scholar, Scopus, and Web of Science, covering publications from 2017 to 2022. Our findings revealed a concentration on AI-based studies that focused solely on cancer subtyping or the molecular characterization of tumour progression. Notably, we encountered an absence of AI-driven research dedicated to mapping cancer progression through time. This gap in the literature highlights the challenges that characterize this effort (see Chapter 2) but also the innovative potential of our work.

Processing and Analysis of Breast Invasive Carcinoma Data

BSC processed bulk RNA-Seq data of breast invasive carcinoma sourced from The Cancer Genome Atlas (TCGA). We compiled a dataset encompassing gene expression values for nearly 9,000 genes across almost 900 women. These individuals were categorized based on the stage of their tumour (I, II, III) and the tumour's anatomical location (lobular and ductal, with the latter being more prevalent). To construct temporal sequences from this cross-sectional data, we devised pseudo-trajectories using patient similarity metrics. This methodology enables us to propose a potential temporal sequence of cancer stages among women with similar expression profiles, despite the heterogeneity of the available data

(Figure 2). This approach is not only necessitated by the static nature of our data but is also biologically plausible, considering the underlying principles of tumour plasticity (i.e. the tumour's ability to optimize its survival strategy within its host environment effectively).

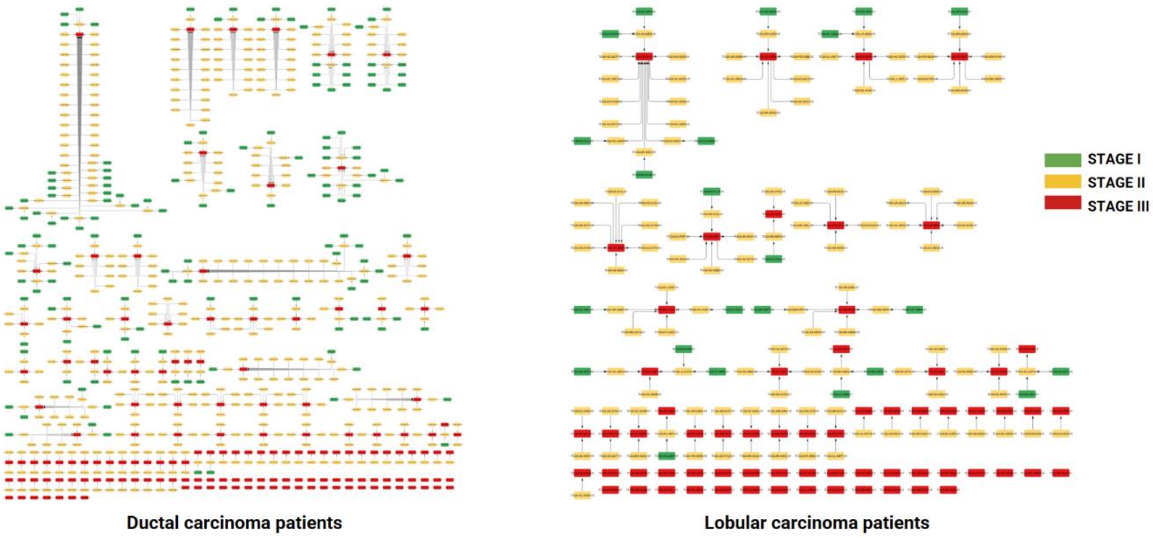

Figure 2. Pseudo-trajectories of breast invasive carcinoma stages.

Pathway Enrichment Analysis

Given the high-dimensional nature of our dataset and the necessity for a neurosymbolic approach to manage a reduced number of variables, we transformed gene expression data into pathway enrichments. Our focus was on the lowest level of the Reactome hierarchy (https://reactome.org/), analysing 29 major pathways. This analysis shed light on the pathway patterns across the transitions from stage I to II to III (Figure 3). Interestingly, while the anatomical location showed minimal influence on these transitions, we identified some degree of separability between them overall, particularly between stages I to II and II to III. This level of distinction was not apparent when analysing the data solely based on gene expression values, suggesting the added value of our pathway-centric approach.

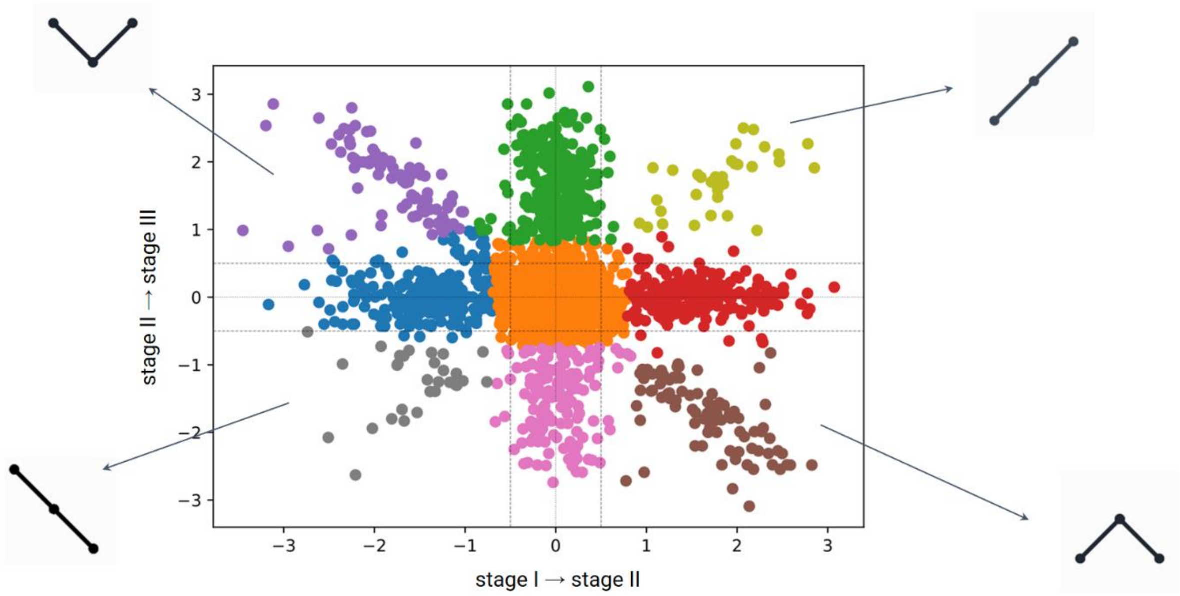

Figure 3. Pathway enrichment patterns across cancer stage transitions. Colours corresponds to KMeans clusters.

Variational Autoencoder and Synthetic Data Generation

To refine our analysis, BSC trained a Variational Autoencoder (VAE) using the expression data. The latent space generated by the VAE has been instrumental in synthesizing intermediate data points interpolated between patients within the pseudo-trajectories. Our ongoing evaluation focuses on how pathway enrichments evolve with the introduction of these synthetic intermediate points. This approach not only enriches our dataset but also enhances our understanding of the nuanced changes occurring between cancer stages, potentially offering new insights into the temporal dynamics of tumour progression.

3.4 Infrastructure Life Cycle Assessment

The Infrastructure Life Cycle Assessment use case is developing the datasets analysis gathered from a pipe section, with the aim of identifying states and incidents ahead of time and with high accuracy. In particular the LAB testing (at UNIPA), and the small scale pilot at EKSO premises (with preliminary AI processing) had led to the definition of the best set of technology (vibration sensors) already deployed in a full scale pilot on a real potable water pipe section (Figure 4), in operations and at present producing data continuously.

The main features of the Small Scale Pilot are the following:

1. Vibration time series from 1 sensor (event labelled) locally registered;
2. Frequency: 6,6 ksps;
3. Magnitude: 800MB on compressed CSV file;
4. Limited time frame measurements: 1 hour;
5. Different simulated leakage in distance and size.

The main features of the Full Scale Pilot are the following:

1. Vibration time series from 10 sensors (event labelled) remotely registered;
2. Frequency:1,6 ksps (each sensor-BUS main constraint);
3. Magnitude: 14MB/10min. (all sensors) in Binary format (Numpy zipped)
4. Continuous measurement: 24/7
5. Sigle leakage simulation.

Figure 4. Pipe section view for the Full Scale Pilot.

This is enabling EKSO to make educated, data-driven analysis regarding some major relevant phenomenons/defects, operational or structural that could affect the pipe efficiency:

- General Anomaly detection
- leak presence
- sizing leak
- locating leak
- Signal evolution in time
- Ageing phenomenons

These are the main KPI to base real time evaluation of the efficiency of the pipe status and operations to consequently make decisions about maintenance, service, and repairs of the pipes, including the implementation of refurbishment operations when required sizing and locating them at the best.

First results on the Full Scale Pilot, on the Leak detection and location, tell us that one sensor's recordings analysis alone is not enough: Pre-processing one sensor's recordings gives accuracy from 60% to 97%.

Progressing on a major complex scenario using closest sensor, the model (Figure 5) increases accuracy to 98.5%; if we use all 10 sensors, we increase accuracy to 99.8 %.

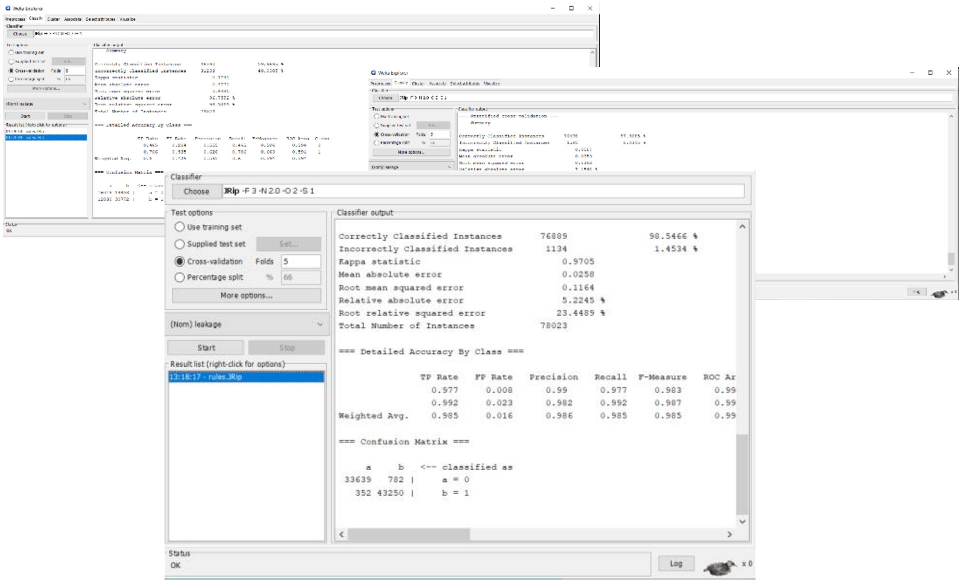

Figure 5. Model performances.

4 Roadmap for Use Cases' Next Steps

4.1 Summary

This chapter outline the next steps in the progress of the EVENFLOW use case. For **Industry 4.0**, upcoming steps include finalizing real robot navigation tests, enhancing predictive planning, and conducting comprehensive system evaluations in intralogistics to refine forecasting models. In **Personalized Medicine**, the focus will be on sharing our research through a detailed manuscript, validating our approaches to cancer progression, enhancing analyses of tumour dynamics and pathway enrichments, and generating synthetic data to better understand and predict cancer progression stages. The **Infrastructure Life Cycle Assessment** use case use case will progress in the spotted KPI analysis/processing with the aim of covering a wide set of information defining the most comprehensive scenario of the pipe status and operativity and take the best decisions for ordinary or extraordinary maintenance.

4.2 Industry 4.0

In the next phase of our research project on Complex Event Forecasting in intralogistics, several key steps will be undertaken to advance our understanding and implementation of the system. First, we will finalize the evaluation of navigation on a real robot, which includes integrating the forecasting component. This integration will enable us to assess how effectively the navigation system incorporates forecasted complex events into its decision-making processes. Furthermore, we will conduct scenario-based evaluations to simulate various real-world scenarios and gauge the system's performance under different conditions.

Additionally, we will proceed with the evaluation at the system level of an intralogistics system. This evaluation will focus on capturing and analysing simple events such as skill initiation, skill state transitions, and monitoring information. By considering an entire intralogistics process, from goods retrieval from a warehouse through multiple handling systems to the storage of the finished product, we aim to comprehensively assess the system's ability to forecast and respond to complex events throughout the entire logistics workflow. Through these evaluations, we anticipate gaining valuable insights into the effectiveness and robustness of our forecasting models and their integration within real-world intralogistics environments.

4.3 Personalized Medicine

As we advance in the Personalized Medicine use case, our next steps consist in sharing our findings with the broader scientific community. Firstly, BSC is in the process of drafting a manuscript that delineates our achievements, in collaboration with NCSR. This document will highlight the innovative methodologies and significant discoveries made thus far, aiming to contribute valuable knowledge to the field of cancer research. Secondly, to reinforce the foundation of our work, we plan to further corroborate our assumptions underlying the construction of the pseudo-trajectories. This involves a detailed examination of tumour plasticity evidence in existing literature, alongside an in-depth analysis of the gene expression levels available to us. Thirdly, we aim to solidify our findings regarding the separability of stage transitions based on pathway enrichments. By establishing clear distinctions between the transitions, we can provide more precise insights into the biological underpinnings of cancer progression. Fourthly, the generation of synthetic instances interpolated in the VAE latent space between stages within each pseudo-trajectory represents a crucial step forward. Analysing changes in pathway enrichment through these synthetic instances will allow us to simulate the progression dynamics more accurately and explore potential intervention points. Lastly, we plan to leverage the neuro-symbolic machinery developed by NCSR for our use case, aiming to forecast the progression of cancer with greater precision.

4.4 Infrastructure Life Cycle Assessment

Regarding the short-term target related to the leak detection and sizing, data classification issues under consideration are the following:

- During training: Create a model for each sensor that answers the relevant question "Is there a leakage X meters to my right/left?"
- During inference: Run each produced model on each sensor's test data-set Compute leakage location as the average location of the locations given by the 100 models.
- During visualization: When detecting a leakage, if the leakage is not consistently (all the sensors together) detected for the next 1 minute or so, consider it FALSE ALARM.
- Prepare more data to train the models using leakages observed from different locations.

The main road map status of the KPI to be considered an analysed is the following:

1. Anomaly detection (done)
2. Leak presence (done)
3. Sizing leak: (under evaluation)
4. Locating leak: (under evaluation, through sensors interaction analysis)
5. Anomaly evolution on long run: (On going analysis in Combination with the expected Pipe life declared by the producer)
6. Ageing phenomena: On hold waiting #

5 Conclusions

This document, as a continuation of deliverable D3.1, offers an in-depth examination of the progress and forthcoming steps in the EVEFLOW project's three use cases: Industry 4.0, Personalized Medicine, and Infrastructure Life Cycle Assessment. By advancing from the initial stages of data handling and scenario definition, this update emphasizes the significant strides made in applying theoretical frameworks to practical challenges.

In **Industry 4.0**, we have seen the successful deployment of an autonomous mobile robot, showcasing cutting-edge navigation and forecasting capabilities, signifying a leap towards efficient and proactive intralogistics operations. The **Personalized Medicine** use case has made remarkable progress in the analysis of cancer progression, employing AI to generate synthetic data and uncover new insights into tumour dynamics, demonstrating the potential of technology in advancing cancer research. The **Infrastructure Life Cycle Assessment** continues to leverage AI for enhancing equipment efficiency and predictive maintenance, aiming to significantly impact infrastructure management.

Furthermore, this document outlines a series of strategic steps for each use case. Industry 4.0 is set to undergo rigorous testing and evaluations to refine its navigation and forecasting systems further. Personalized Medicine will focus on validating research findings and expanding the understanding of cancer progression through advanced data analysis techniques. The **Infrastructure Life Cycle Assessment** aim is to develop the widest set of information production with the most comprehensive evaluation scenario of the pipe status and operativity to take the best decisions for ordinary or extraordinary maintenance.

Finally, the provided details on middleware infrastructure deployment indicate the project's comprehensive approach to ensuring robust support for these advancements. In conclusion, this update highlights the EVEFLOW project's dedication to pushing the boundaries of AI and its applications in industry, medicine, and infrastructure management.

Appendix A Middleware Infrastructure Deployment

To properly handle data (both streams and static), the agreed upon middleware infrastructure on the NCSR compute servers has been mostly deployed, with the only missing aspect being the installation and configuration of the KeyCloak identity management (and oauth) Open-Source solution that is still pending.

In the next sections, we document in detail the steps needed to implement this deployment.

A.1 Kubernetes Cluster Deployment

A.1.1 VMs creation

The current cluster on NCSR uses 4 CPU cores, has 32GB RAM total, and 256GB secondary memory storage. It runs CentOS Linux release 7.9.2009. The network interfaces are as follows:

- evenflownode1: 10.10.10.1/24 and 143.233.226.125/25
- evenflownode2: 10.10.10.2/24
- evenflownode3: 10.10.10.3/24
- evenflownode4 :10.10.10.4/24

The admin can connect on ssh port: 40222 with public key authentication (on all VMs).

A.1.2 Prerequisites on all nodes

Add on `/etc/hosts` file:

10.10.10.1 evenflownode1
10.10.10.2 evenflownode2
10.10.10.3 evenflownode3
10.10.10.4 evenflownode4

Disable swap via the following commands:

```
free -h
blkid
lsblk
swapoff -a
comment swap line on /etc/fstab
mount -a
```

Ensure firewalld is disabled via the following commands:

```
systemctl status firewalld
systemctl disable firewalld
```

Disable SeLinux:

```
setenforce 0
sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/' /etc/selinux/config
```

Check for unique MAC on all nodes:

```
ip a
```

Check for unique product_uuid on all nodes:

```
cat /sys/class/dmi/id/product_uuid
```

Setup sysctl and modules for containerd and Kubernetes:

```
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-iptables = 1
net.bridge.bridge-nf-call-ip6tables = 1
net.ipv4.ip_forward = 1
vm.swappiness = 0
EOF

cat <<EOF | tee /etc/modules-load.d/k8s.conf
overlay
br_netfilter
EOF

modprobe overlay
modprobe br_netfilter
lsmod | grep br_netfilter
lsmod | grep overlay
```

A.1.3 Container Runtime

Install containerd:

Download and untar binaries:

```
curl -L -o containerd-1.6.19-linux-amd64.tar.gz
https://github.com/containerd/containerd/releases/download/v1.6.19/containe
rd-1.6.19-linux-amd64.tar.gz

tar Cxzvf /usr/local containerd-1.6.19-linux-amd64.tar.gz
```

Configure service:

```
mkdir -p /usr/local/lib/systemd/system/
```

The contents of the file `/usr/local/lib/systemd/system/containerd.service` should be as follows:

```
# Copyright The containerd Authors.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

[Unit]
Description=containerd container runtime
Documentation=https://containerd.io
After=network.target local-fs.target
```

```
[Service]
ExecStartPre=-/sbin/modprobe overlay
ExecStart=/usr/local/bin/containerd

Type=notify
Delegate=yes
KillMode=process
Restart=always
RestartSec=5
# Having non-zero Limit*s causes performance problems due to accounting
overhead
# in the kernel. We recommend using cgroups to do container-local accounting.
LimitNPROC=infinity
LimitCORE=infinity
LimitNOFILE=infinity
# Comment TasksMax if your systemd version does not supports it.
# Only systemd 226 and above support this version.
TasksMax=infinity
OOMScoreAdjust=-999

[Install]
WantedBy=multi-user.target

systemctl daemon-reload
systemctl enable --now containerd
```

Configuration for Cgroup Driver:

containerd uses a configuration file located in `/etc/containerd/config.toml` for specifying daemon level options. The default configuration can be generated via:

```
mkdir -p /etc/containerd
containerd config default > /etc/containerd/config.toml.
```

To use the systemd cgroup driver in `/etc/containerd/config.toml` with runc:

The file `/etc/containerd/config.toml` should be like this:

```
[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc]
  ...
  [plugins."io.containerd.grpc.v1.cri".containerd.runtimes.runc.options]
    SystemdCgroup = true
```

Install runc:

```
curl -L -o runc.amd64
https://github.com/opencontainers/runc/releases/download/v1.1.4/runc.amd64
install -m 755 runc.amd64 /usr/local/sbin/runc
```

Install CNI plugins:

```
curl -L -o cni-plugins-linux-amd64-v1.1.1.tgz
https://github.com/containernetworking/plugins/releases/download/v1.1.1/cni
-plugins-linux-amd64-v1.1.1.tgz
mkdir -p /opt/cni/bin
tar Cxzvf /opt/cni/bin cni-plugins-linux-amd64-v1.1.1.tgz
```

Reference: https://github.com/containerd/containerd/blob/main/docs/getting-started.md

A.1.4 Kubernetes Cluster

Configure repository:

```
cat <<EOF | tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-
\$basearch
enabled=1
gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
```

Install Kubernetes Packages:

```
yum install kubectl-1.25.6 kubelet-1.25.6 kubeadm-1.25.6
```

Initialize control-plane:

```
kubeadm init --control-plane-endpoint "10.10.10.1:6443" --upload-certs --
kubernetes-version "v1.25.6"

systemctl enable kubelet.service
mkdir -p $HOME/.kube
cp  -i /etc/kubernetes/admin.conf $HOME/.kube/config
```

Install Calico:

```
curl
https://raw.githubusercontent.com/projectcalico/calico/v3.25.0/manifests/ca
lico.yaml -O
kubectl apply -f calico.yaml
```

Calico images needed to be loaded externally with “ctr -n=k8s.io images import calico.tar”

Verify image list:

```
ctr --namespace k8s.io i ls
```

Join workers:

```
kubeadm join 10.10.10.1:6443 --token xrkasv.tnljusxs7silezsy --discovery-
token-ca-cert-hash
sha256:bcab1e759eeb389c3ee9a1cd4c40e77e8f214c33c7ac6679990f767c2e734735

systemctl enable kubelet.service
```

Verify cluster nodes status from master node:

```
kubectl get pods -A
kubectl describe node evenflownode1
kubectl get all -A
```

Reference: https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/

A.2 Infrastructure Core Components Installation

A.2.1 Helm Package manager Installation

```
curl -fsSL -o get_helm.sh
      https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3
chmod 700 get_helm.sh
./get_helm.sh
```

A.2.2 Kubernetes Ingress Controller - Nginx

Get and Modify Helm Chart:

```
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
helm repo update
helm pull ingress-nginx/ingress-nginx
tar -xvf ingress-nginx-4.8.3.tgz
```

Make sure the file `ingress-nginx/values.yaml` looks as follows:

```
# -- Is this the default ingressClass for the cluster
    default: true

# -- Node labels for controller pod assignment
nodeSelector:
  node: ingress

# -- Annotations to be added to controller pods
  ##
  podAnnotations: {}
  replicaCount: 1
```

Configure hostNetwork to allow NGINX Ingress controller bind ports 80 and 443 directly to Kubernetes nodes' network interfaces. Also, configure dnsPolicy to ClusterFirstWithHostNet in order to use internal DNS resolver (kube-dns).

Reference:

https://kubernetes.github.io/ingress-nginx/deploy/baremetal/#bare-metal-considerations.

```
hostNetwork: true
dnsPolicy: ClusterFirstWithHostNet
```

Change the service type from Load Balancer to ClusterIP:

```
service:
...
  type: ClusterIP
```

Add toleration for scheduling pods on master nodes:

```
#under deployment section
tolerations:
  - effect: NoSchedule
    key: node-role.kubernetes.io/control-plane
```

```
    operator: Exists
```

Nginx Installation

Label master node:

```
kubectl label nodes evenflownode1 node=ingress
```

Create namespace:

```
kubectl create namespace ingress-nginx
```

Install the chart:

```
helm install ingress-nginx ./ingress-nginx --namespace ingress-nginx
```

Reference:

https://kubernetes.github.io/ingress-nginx
https://github.com/kubernetes/ingress-nginx

A.2.3 Longhorn Storage - Helm Installation

Nodes Preparation

Create `/data` directory on all worker nodes

Get and Modify Helm Chart

```
helm repo add longhorn https://charts.longhorn.io
helm pull longhorn/longhorn --version v1.4.1
tar -xvf longhorn-1.4.1.tgz

vi longhorn_old/longhorn/values.yaml
```

Configure the following:

```
defaultSettings:
...
  defaultDataPath: /data/longhorn

persistence:
...
  reclaimPolicy: Retain
```

Installation

Install the chart:

```
helm install longhorn ./longhorn_old/longhorn/ --namespace longhorn-system
--create-namespace
```

Create secret:

```
USER=username_ommited; PASSWORD=password_ommited; echo "${USER}:$(openssl
passwd -stdin -apr1 <<< ${PASSWORD})" >> auth
```

```
kubectl -n longhorn-system create secret generic basic-auth --from-file=auth
```

Create Longhorn Ingress Rule:

```
vi longhorn-ingress.yml

#place the following
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: longhorn-ingress
  namespace: longhorn-system
  annotations:
    # type of authentication
    nginx.ingress.kubernetes.io/auth-type: basic
    # prevent the controller from redirecting (308) to HTTPS
    nginx.ingress.kubernetes.io/ssl-redirect: 'false'
    # name of the secret that contains the user/password definitions
    nginx.ingress.kubernetes.io/auth-secret: basic-auth
    # message to display with an appropriate context why the authentication
is required
    nginx.ingress.kubernetes.io/auth-realm: 'Authentication Required '
    # custom max body size for file uploading like backing image uploading
    nginx.ingress.kubernetes.io/proxy-body-size: 10000m
    nginx.ingress.kubernetes.io/rewrite-target: /$2
spec:
  rules:
  - host: evenflownode1
    http:
      paths:
      - pathType: Prefix
        path: /longhorn(/|$)(.*)
        backend:
          service:
            name: longhorn-frontend
            port:
              number: 80

kubectl apply -f longhorn-ingress.yml -n longhorn-system
```

A.2.4 Login to Longhorn UI

Login via: http://143.233.226.125/longhorn/. Credentials are omitted from this public deliverable.

In case you want to configure any extra disks, create Longhorn directory on every worker node.

```
mkdir /data2/longhorn
```

From Longhorn UI
Node -> Select Node -> Operation: Edit node and disks -> Add Disk -> Path: /data2/longhorn -> Scheduling (Enable) -> Save

Reference:
https://longhorn.io/docs/1.3.1/deploy/install/install-with-helm/
https://longhorn.io/docs/1.3.1/deploy/accessing-the-ui/longhorn-ingress/

A.3 Services Installation

A.3.1 InfluxDB Installation

```
helm repo add influxdata https://helm.influxdata.com/
helm pull influxdata/influxdb2
tar -xvzf influxdb2-2.1.2.tgz
```

Install Influxdb with persistent storage

```
helm install influxdb --set persistence.enabled=true influxdata/influxdb2 -
-namespace influxdb-ns --create-namespace
```

To retrieve the password for the 'admin' user:

```
echo $(kubectl get secret influxdb-influxdb2-auth -o
"jsonpath={.data['admin-password']}" --namespace influxdb-ns | base64 –
decode)
```

Reference: https://github.com/influxdata/helm-charts/tree/master/charts/influxdb2

A.3.2 Login to Influxdb UI

Login via: http://143.233.226.125/. Credentials are omitted from this public deliverable.

A.3.3 Telegraf Installation

```
helm repo add influxdata https://helm.influxdata.com/
helm pull influxdata/influxdb2
tar -xvzf influxdb2-2.1.2.tgz
```

Install Telegraf with persistent storage

```
helm install telegraf --set persistence.enabled=true influxdata/telegraf --
namespace influxdb-ns
```

Telegraf installation comments:

"To open a shell session in the container running Telegraf run the following:

kubectl exec -i -t --namespace influxdb-ns $(kubectl get pods --namespace influxdb-ns -l app.kubernetes.io/name=telegraf -o jsonpath='{.items[0].metadata.name}') /bin/sh

To view the logs for a Telegraf pod, run the following:

kubectl logs -f --namespace influxdb-ns $(kubectl get pods --namespace influxdb-ns -l app.kubernetes.io/name=telegraf -o jsonpath='{ .items[0].metadata.name }')"

Reference: https://github.com/influxdata/helm-charts/tree/master/charts/influxdb2

A.3.4 Zookeeper cluster Installation

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

```
helm install zookeeper bitnami/zookeeper --set replicaCount=3 --set auth.enabled=false --set allowAnonymousLogin=true --namespace kafka-ns --create-namespace
```

Zookeeper installation comments:

ZooKeeper can be accessed via port 2181 on the following DNS name from within your cluster:

```
zookeeper.kafka-ns.svc.cluster.local
```

To connect to your ZooKeeper server run the following commands:

```
export POD_NAME=$(kubectl get pods --namespace kafka-ns -l "app.kubernetes.io/name=zookeeper,app.kubernetes.io/instance=zookeeper,app.kubernetes.io/component=zookeeper" -o jsonpath="{.items[0].metadata.name}")
kubectl exec -it $POD_NAME -- zkCli.sh
```

To connect to your ZooKeeper server from outside the cluster execute the following commands:

```
kubectl port-forward --namespace kafka-ns svc/zookeeper 2181:2181 & zkCli.sh 127.0.0.1:2181
```

Reference: https://github.com/bitnami/charts/tree/main/bitnami/zookeeper

A.3.5 Kafka cluster Installation

```
helm repo add bitnami https://charts.bitnami.com/bitnami

helm install kafka bitnami/kafka --set zookeeper.enabled=true --set kraft.enabled=false --set replicaCount=3 --set externalZookeeper.servers=zookeeper.kafka-ns.svc.cluster.local --namespace kafka-ns
```

Kafka installation comments:

"Kafka can be accessed by consumers via port 9092 on the following DNS name from within your cluster:

kafka.kafka-ns.svc.cluster.local

Each Kafka broker can be accessed by producers via port 9092 on the following DNS name(s) from within your cluster:

kafka-0.kafka-headless.kafka-ns.svc.cluster.local:9092
kafka-1.kafka-headless.kafka-ns.svc.cluster.local:9092
kafka-2.kafka-headless.kafka-ns.svc.cluster.local:9092

To create a pod that you can use as a Kafka client run the following commands:

kubectl run kafka-client --restart='Never' --image docker.io/bitnami/kafka:3.5.0-debian-11-r4 --namespace kafka-ns --command -- sleep infinity
kubectl exec --tty -i kafka-client --namespace kafka-ns -- bash

PRODUCER:
*kafka-console-producer.sh *

*--broker-list kafka-0.kafka-headless.kafka-ns.svc.cluster.local:9092,kafka-1.kafka-headless.kafka-ns.svc.cluster.local:9092,kafka-2.kafka-headless.kafka-ns.svc.cluster.local:9092 *
--topic test

CONSUMER:
*kafka-console-consumer.sh *
*--bootstrap-server kafka.kafka-ns.svc.cluster.local:9092 *
*--topic test *
--from-beginning"

Reference: https://github.com/bitnami/charts/tree/main/bitnami/kafka

A.4 Basic Kubernetes Commands

Get nodes of the Kubernetes cluster:
`kubectl get nodes`

Get more detailed info by adding "-o wide" to the end of the command:
`kubectl get nodes -o wide`

Output format in JSON:
`kubectl get pods -o=json`

Get info for all cluster pods:
`kubectl get pods -A`

Output format in YAML:
`kubectl get pods -o=yaml`

Get pods on Kubernetes core namespace:
`kubectl get pods -n kube-system (-o wide)`

Get pods related to Longhorn:
`kubectl get pods -n longhorn-system (-o wide)`

Get Nginx pod:
`kubectl get pods -n ingress-nginx (-o wide)`

Get pods related to zookeeper/kafka clusters:
`kubectl get pods -n kafka-ns (-o wide)`

Get pods related to influxdb:
`kubectl get pods -n influxdb-ns (-o wide)`

Get more info for a specific pod:
```
kubectl describe pod/<pod_name> -n <namespace_name>
kubectl describe pod/longhorn-admission-webhook-79779c487b-qcr4m -n longhorn-system
```

Get logs of a running pod:
```
lubectl logs pod/<pod_name> -n <namespace_name>
```

List pods, services, daemonsets, deployments, replicasets, statefulsets, jobs, and CronJobs in all namespaces, not custom resource types:
```
kubectl get all -A
kubectl get all -n <namespace_name>
```

Get an interactive shell on a single-container pod:
```
kubectl exec -it <pod_name> /bin/sh
```

Execute a command against a container in a pod:
```
kubectl exec <pod_name> -c <container_name> <command>
```

A.5 Kafka Commands

Specify pod to use:
```
export POD_NAME=$(kubectl get pods --namespace kafka-ns -l
"app.kubernetes.io/name=kafka,app.kubernetes.io/instance=kafka,app.kubernet
es.io/component=kafka" -o jsonpath="{.items[0].metadata.name}")
```

Topic creation:
```
kubectl --namespace kafka-ns exec -it $POD_NAME -- kafka-topics.sh --create
--bootstrap-server kafka-0.kafka-headless.kafka-ns.svc.cluster.local:9092 -
-replication-factor 1 --partitions 1 --topic mytopic
```

Topics list:
```
kubectl --namespace kafka-ns exec -it $POD_NAME -- kafka-topics.sh --list -
-bootstrap-server kafka-0.kafka-headless.kafka-ns.svc.cluster.local:9092
Run a kafka consumer:
kubectl --namespace kafka-ns exec -it $POD_NAME -- kafka-console-consumer.sh
--bootstrap-server kafka-0.kafka-headless.kafka-ns.svc.cluster.local:9092 -
-topic mytopic --from-beginning
```

Run a kafka producer:
```
kubectl --namespace kafka-ns exec -it $POD_NAME -- kafka-console-producer.sh
--broker-list kafka-0.kafka-headless.kafka-ns.svc.cluster.local:9092,kafka-
1.kafka-headless.kafka-ns.svc.cluster.local:9092,kafka-2.kafka-
headless.kafka-ns.svc.cluster.local:9092 --topic mytopic
```

A.6 Infrastructure Access

There are two ways to access Demokritos' infrastructure:

1. Direct access to linux master VM
2. WebUIs' access using a web browser

Both ways require credentials that are omitted from this public deliverable.