H2020 DAEMON Project
Grant Agreement No. 101017109

Deliverable 2.3

Final DAEMON Network Intelligence framework and toolsets

Abstract

This document is the first deliverable of the third and last iteration of the DAEMON project, which builds on the results of the second iteration presented in D2.2, D3.2, D4.2, and D5.2 and covers the following key aspects to finalize the DAEMON Network Intelligence (NI) framework. Firstly, it provides a final update on the functional and non-functional requirements of the eight NI-assisted functionalities, assessing their risks and completion status. Secondly, it presents the final updates of the Network Intelligence Plane (NIP), which has evolved throughout the project lifetime into a unified framework incorporating operational hierarchy, orchestration, and the N-MAPE-K representation of NI components. Thirdly, it analyzes the specific needs that NI algorithms induce on the NIP, discusses the challenges NI algorithms pose in terms of management by the Network Intelligence Orchestrator (NIO), and provides functionalities and architectural designs to address such challenges. Additionally, the document includes a comprehensive literature review on integrating machine learning and NI in mobile network management, highlighting key trends and the unique contributions of the DAEMON project within that scope. All these findings inform the final updates to the project guidelines, emphasizing the importance of tailored NI design for 6G network management and the need to develop more interpretable models.

Document properties

Document number	D2.3
Document title	Final DAEMON Network Intelligence framework and toolsets
Document responsible	IMEC
Document editor	Miguel Camelo (IMEC) Paola Soto (IMEC)

Editorial team

Partner	Name	Surname	Sections
i2CAT	Esteban Ginés	Municio García-Aviles	Section 5, A, B
IMDEA	Antonio Marco Alan Sergi	Bazco-Nogueras Fiore Collet Alcalá	Sections 2, 6.1, 7, A, B Sections 2, 3.2, 4.1, 5.2, 7.2 Sections 7.1.2, 7.1.3 Section 7.2.4
IMEC	Miguel Paola	Camelo Soto	Sections 1, 2, 3.2, 4.1, 5, 6, 7.1, 7.2., A, B
NBL	Danny	De Vleeschauwer	Section 6.1, 7.2, A, B
NEC	Andres Josep Xavier	Garcia-Saavedra Salvat	Sections 2, 3.1, 3.2 4.2, 7.2
TID	Andra	Lutu	Sections A, B
UC3M	Marco Albert Vittorio Alberto	Gramaglia Banchs Prodomo García	Sections 2, 3.1, 3.2, 4.2, 6, 7
UMA	Lidia Joaquin Mercedes Monica	Fuentes Ballesteros Amor Pinto	Section 2, A, B
WINGS	Evangelos	Kosmatos	Section 5.3
ZSC	Ivan	Paez	Section 5.3

Target dissemination level	Public
Status of the document	Final
Version	1.0

Production properties

Reviewers	Evangelos Kosmatos – (WINGS) Brendan McAuliffe - (SRS) Nina Slamnik – (IMEC) Alexandros Kostopouos – (OTE) Marco Fiore – (IMDEA)

Document history

Revision	Date	Issued by	Description
0.1	01/05/2023	IMEC	First version with Table of Content
0.3	07/06/2023	All	First draft with all input
0.5	12/06/2023	IMEC	First complete draft edited with full content
0.7	19/06/2023	IMEC	Draft internally reviewed
0.9	26/06/2023	IMEC	Draft externally reviewed and first release candidate document
1.0	29/06/2023	IMEC	Final version

Disclaimer

This document has been produced in the context of the DAEMON Project. The research leading to these results has received funding from the European Union Horizon 2020 research and innovation programme under grant agreement no.101017109.

Table of Contents

List of Figures

List of Tables

List of Acronyms

3GPP	3rd Generation Partnership Project
5G	Fifth Generation
5GC	5G Core
5GPPP	5G Infrastructure Public Private Partnership
6G	Sixth Generation
AARES	Automated Anomaly RESponse
AC	Admission Control
AF	Application Function
AI	Artificial Intelligence
AMC	Automatic Modulation Classification
AMR	Automatic Modulation Recognition
API	Application Programming Interface
ARQ	Automatic Repeat Query
B5G	Beyond 5G
BNN	Binarized Neural Networks
BOPS	Bits Operations
CAWRS	Compute-AWare Radio Scheduling
CD	Continuous Deployment
CFORE	Capacity FOREcasting
CI	Continuous Integration
CM	Component Manager
CNN	Convolutional Neural Network
CSOI	Creation Selection Optimization and Instantiation
CT	Control Theory
DL	Deep Learning
DNN	Deep Neural Network
DoA	Description of the Action
DRL	Deep Reinforcement Learning
DT	Digital Twin
EAWVNF	Energy-AWare VNF control & orchestration
ENI	Experiential Networked Intelligence
ETSI	European Telecommunications Standards Institute
FEC	Forward Error Correction
FL	Federated Learning
FR	Functional Requirement
GNN	Graph Neural Network
GP	Gaussian Processes
HARQ	Hybrid ARQ
IBSSI	In-Backhaul Support for Service Intelligence
IQ	In-Phase and Quadrature
KPI	Key Performance Indicators
MAC	Medium Access Control
MAE	Mean Absolute Error
MANO	Management and Orchestration
MAPE-K	Monitor-Analyze-Plan-Execute over a shared Knowledge
MDAF	Management Data Analytics Function

MDAS	Management Data Analytics Services
ML	Machine Learning
MLOps	Machine Learning Operations
MSE	Mean Squared Error
MSLE	Mean Squared Logarithmic Error
MTERM	Multi-Timescale Edge Resource Management
NF	Network Function
NFR	Non-Functional Requirement
NFV	Network Function Virtualization
NFV-O	Network Function Virtualization Orchestrator
NI	Network Intelligence
NIC	Network Interface Card
NICS	Normalized Inference Cost Score
NIF	Network Intelligence Function
NIF-C	NIF Component
NIFD	Network Intelligence Function Descriptor
NIO	Network Intelligence Orchestrator
NIP	Network Intelligence Plane
NIS	Network Intelligence Service
NISD	Network Intelligence Service Descriptor
NIS Wconf	NIS Workflow Configuration
N-MAPE-K	Network MAPE-K
NN	Neural Network
NS	Network Service
NWDAF	Network Data Analytics Function
OAM	Operations, Administration, and Maintenance
O-CU	Open Central Unit
O-DU	Open Distributed Unit
OPEX	Operational Expenditures
OSM	Open Source Mano
PISA	Protocol Independent Switch Architecture
PolicyIC	Policy Interpreter and Configuration
PPO	Proximal Policy Optimization
RA	Resource Allocation
RAN	Radio Access Network
RIC	Radio Intelligent Controller
RIS	Reconfigurable Intelligent Surfaces
RL	Reinforcement Learning
RNN	Recurrent Neural Network
RT	Real-Time
RU	Radio Unit
SDK	Software Development Kit
SDO	Standard-Defining Organization
SLA	Service-Level Agreement
SLMANO	Self-Learning MANO
SMO	Service and Management Orchestration
SNR	Signal-to-Noise Ratio

TC	Traffic Classification
TES	Thresholded Exponential Smoothing
TLS	Transport Layer Security
UE	User Equipment
VIM	Virtual Infrastructure Manager
VNF	Virtual Network Function
VNFM	Virtual Network Function Manager
WP	Work Package
XAI	Explainable AI

Executive summary

This is the third and last public deliverable of WP2 of the DAEMON project. It builds upon the material of the previous deliverable of WP2, i.e., D2.2 [1], and on activities and results achieved during the second iteration of the project in WP3 D3.2 [2], WP4 D4.2 [3], and WP5 D5.2 [4]. As a result, the document describes the following content.

First, it provides the final update on the functional and non-functional requirements of the eight NI-assisted functionalities (Reconfigurable Intelligent surfaces control - RISC, Multi-timescale Edge resource management – MTERM, In-backhaul support for service management – IBSSI, Compute-aware radio scheduling – CAWRS, Energy-aware VNF control and orchestration – EAWVNF, Self-learning MANO – SLMANO, Capacity forecasting – CFORE, and Automated anomaly response – AARES) tackled by DAEMON at the end of the WP2. Although no new updates were added to the functionalities, we assess the risks to achieve the requirements and its current completion status. For the requirements that were not finalized at the time of this deliverable, we also specify what is required to successfully finalize it and in which deliverable (e.g., WP3 D3.3, WP4 D4.3, or WP5 D5.3) the results will be provided.

Second, it presents the final updates of the Network Intelligence Plane (NIP), a collection of modules and interfaces responsible for managing NI within the network. In this deliverable, the NIP has evolved, and it is presented as a unified framework that brings together (i) the operational hierarchy of NI components and their orchestration and (ii) the N-MAPE-K representation of the NI components. By doing so, we make another step forward toward the vision of a complete NIP initially presented in D2.2 [1].

Third, in addition to the unified DAEMON framework, we also identify and present in detail the specific needs that NI algorithms pose on the NIP. Moreover, we analyze their specificity in terms of challenges towards the procedures for NI management at the Network Intelligence Orchestrator (NIO) level. We also devise and describe the functionalities that the NIO shall provide to support such requirements and how they fit the whole architecture together. The architectural design is complemented by presenting and discussing the interfaces required to allow communication between NIP components and with external entities such as the RAN controller and the 5G Core systems. These interfaces are also enablers for designing the set of procedures that address the needs and challenges introduced in this document.

Fourth, this document provides the final, comprehensive overview of the literature review carried out by the project, focused on the integration of machine learning and NI in mobile network management. The survey highlights key trends in current research and showcases the distinctive contributions made by the DAEMON project. The insights that originated from this analysis also support our final updates to the project guidelines, including new ones, for practical NI design. As in D2.2 [1], these guidelines focus on two main directions: i) NI design tailored to the needs of B5G network management, orchestration, and control, and ii) NI design that considers the use of more traditional, more straightforward, or interpretable models to avoid overburdening the system with data-heavy models and promotes the utilization of models that are easier to understand and interpret.

We closed this document with additional closing remarks and two appendices containing complementary information related to the functional requirements and the literature review.

1 Introduction

Let us recall the twofold target of Work Package 2 (WP2) of the DAEMON project:

- Design an overall architecture for the harmonized integration of Network Intelligence (NI) in Beyond 5G (B5G) systems.
- Develop a knowledge base and a rigorous methodology for the development of Artificial Intelligence (AI) and Machine Learning (ML) tools that effectively support Network Intelligence functionalities.

By achieving the twofold target above, WP2 is contributing to pursuing the following objectives within the DAEMON project.

- ***Objective 1.1***. To enable and drive the coordination and cross-compatibility across NI deployed in different network domains operating at different timescales.
- ***Objective 1.2***. To enable NI deep into the network infrastructure.
- ***Objective 3.1***. To adjust AI techniques to the specific necessities of the network environment and operations and to develop novel AI hybrid approaches.
- ***Objective 3.2***. To introduce appropriate and tailored cost functions for the networking context that can be used for training AI techniques.
- ***Objective 3.3***. To develop novel AI techniques that can dynamically adapt to available network resources by trading off accuracy with, e.g., inference latency or computational complexity.

During the third public deliverable of the project's WP2, which was built on the material of D2.2 [1], we are presenting the following contributions.

- We provide the final update on the functional and non-functional requirements of the eight NI-assisted functionalities tackled by DAEMON. Although no new updates were added to the functionalities, we assess the risks to achieve the requirements and its current completion status.
- We evolve the Network Intelligence Plane (NIP) towards the Network Intelligence Stratum, an architecture that emerges from a collection of modules and interfaces responsible for managing NI within the network, and it is now a unified framework that brings together (i) the operational hierarchy of NI components and their orchestration and (ii) the N-MAPE-K representation of the NI components. As a result, the original NIP architecture is transformed from a purely separate plane to a more orthogonal approach where Network Intelligence Functions (NIFs) and Network Intelligence Services (NISs) can effectively be integrated into the traditional planes (data, control, and management).
- We identify and detail the specific needs that NI algorithms pose on the NIP towards the procedures for NI management at the Network Intelligence Orchestrator (NIO) level. We also devise and describe the functionalities that the NIO shall provide to support such requirements and how they fit the whole architecture together.
- We complete the architectural design with the interfaces required to allow internal and external communication of the NIP and its components, together with the set of procedures that address the needs and challenges introduced in the previous item.
- We summarize the outcomes of the comprehensive literature review carried out by the project and focus on the integration of machine learning and NI in mobile network management. The survey highlights key trends in current research and showcases the distinctive contributions made by the DAEMON project. The insights that originated from this analysis also support our final updates to the project guidelines on the limits of AI and hybrid approaches for NI and customized and adaptable AI for NI.
- We provide the final updates to the project guidelines, including new ones, on the pragmatic design of NI covering the following three aspects: (i) by deriving general guidelines for the design of dedicated loss functions that are perfectly aligned with the actual performance metrics, (ii) designing a methodology for self-learning AI models that dynamically and automatically balance costs and efficiency, and (iii) developing elastic NI models capable of adapting their complexity to the context, trading off (computational) complexity for accuracy, responsiveness or energy efficiency as needed.

The first four contributions, which are realized as a full architectural design, complete Objectives 1.1 and 1.2. On the other hand, the last two contributions, a comprehensive literature review and a set of guidelines, complete Objectives 3.1, 3.2, and 3.3 of this project.

1.1 Connecting the second and third iterations of the DAEMON project

To provide a clear understanding of the progress made in the DAEMON project, we situate D2.3 within the project's overall work plan schedule. Figure 1 presents the original Gantt diagram, illustrating the three iterative phases emphasized in the diagram and the scope of D2.3. Each iteration consists of specific phases: (i) the design of the NI framework and NI models, which is conducted in WP2, (ii) the implementation of NI-assisted functionalities based on the design, carried out in WP3 and WP4, and (iii) the evaluation of NI-assisted functionalities in dependable settings, executed in WP5. The iterative nature of the work plan allows each iteration to inform and build upon the preceding one. This flexible structure enables the identification and resolution of emerging issues in the developed solutions, ensuring a comprehensive approach to problem-solving throughout the project.

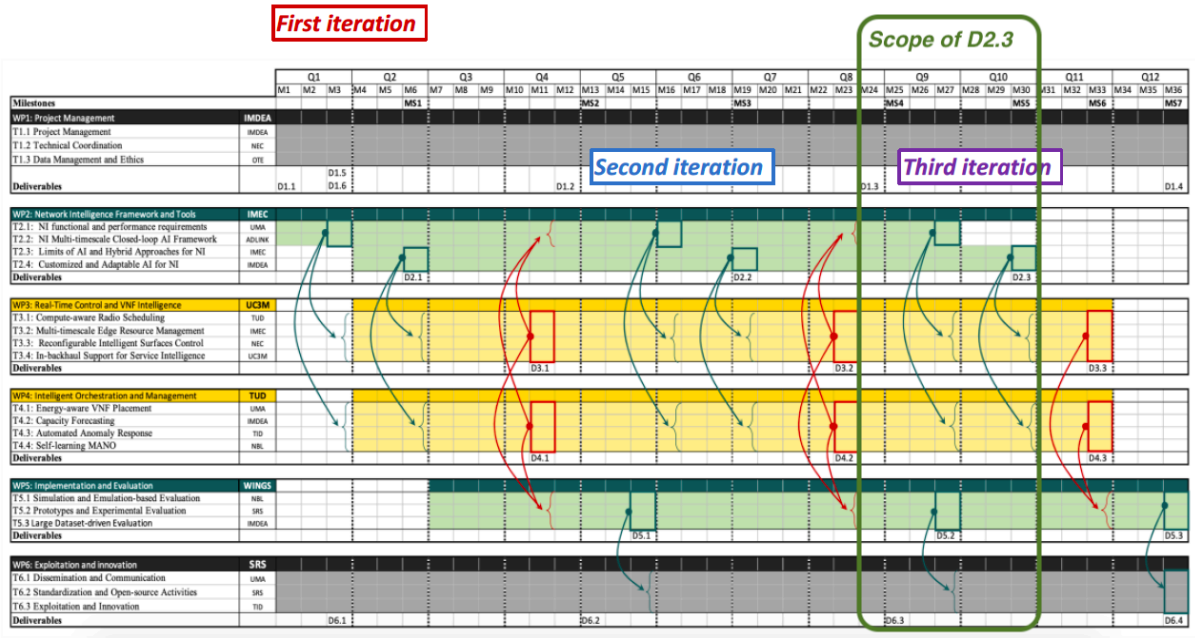


Figure 1. Gantt diagram of the DAEMON project, with the three iterations of the work plan highlighted and the scope of D2.3.

Notice that although D2.2 [1] provided a complete technical foundation towards WP2's main objective and DAEMON's related sub-objective at the end of iteration 2, several research activities were still open or raised during the third iteration of this WP:

- The definition and description of the functional and non-functional requirements were expected to undergo minor changes as they reached a high level of maturity during the second iteration. However, it was crucial to analyze their actual development status in order to assess the progress of their completion and manage the associated risks. This analysis was based on the outcomes of WP3, WP4, and WP5 during the second iteration, reported in deliverables D3.2 [2], D4.2 [3], and D5.2 [4], respectively. By doing so, WP2 could provide feedback to WP3, WP4, and WP5 if further actions were needed to ensure the fulfillment of all the requirements.
- The initial design of the NIP and its requirements provided a high-level view of what the NIP can achieve towards an AI-native architecture for 6G. However, the challenges that raise when orchestrating NIFs/NISs could not be identified until the end of the second iteration when WP3 and WP4 updated the design of their algorithms with the outcomes of D2.2 [1]. As a result, further progress was achieved in defining which components should be part of the Network Intelligence Orchestrator (NIO), determining the necessary interfaces for communication within the NIP, and establishing procedures to address these challenges.
- Although the first set of practical guidelines for the design of NI algorithms that are tailored to mobile network environments and the ones that identify the limits of applying AI/ML in networking were presented in D2.2 [1] based on the outcomes of D3.1[5], D4.1[6], and D5.1[7], further improvements on the guidelines, including new ones, and clear identification of limitations and challenges of them were derived at the end of the third iteration based on the outcomes from D3.2 [2], D4.2 [3], and D5.2 [4]. Moreover, the final iteration of the literature review was focused not only on recent developments in NI functionalities but also on identifying how the guidelines proposed in DAEMON provided a new state of the art in this domain.

1.2 Role of Deliverable 2.3 across iterations

The present deliverable acts as the connecting document between the second and third iterations of the project. Namely, ***D2.3 uses the conclusions and results of the second iteration to pave the way for the NI design, development and testing activities of the final iteration of the DAEMON project***. The content of this document, therefore, describes the work carried out in WP2 during the third iteration of the project. As anticipated in the previous subsection, such work addressed the outgoing research activities after the second iteration and, more specifically, encompassed the following key aspects:

- ***Final updates on functional and non-functional requirements***. The document provides the ultimate update on the definition of the functional and non-functional requirements for the eight NI-assisted functionalities addressed by the DAEMON project. While no new updates were added to such requirements with respect to the descriptions in D2.2 [1], the final risks associated with meeting these requirements are re-assessed in line with their current completion status. For requirements that remain unresolved at this stage, the specific actions that are still needed for their successful finalization are outlined, indicating the future deliverable where relevant results will be presented.
- ***Final updates of the Network Intelligence Plane (NIP).*** The NIP, responsible for managing NI within the network, has undergone significant developments beyond those reported in D2.2 [1]. This document presents the final version of the NIP model, which now serves as a unified framework encompassing the operational hierarchy and orchestration of NI components, along with the N-MAPE-K representation of these components. This progress aligns with the vision set in D2.2 [1].
- ***Identification of specific needs and challenges when orchestrating NI***. In addition to the unified DAEMON framework, the document thoroughly identifies and presents the specific needs that NI algorithms impose on the NIP. It analyzes the challenges these needs present regarding the NI management procedures at the level of the Network Intelligence Orchestrator (NIO). The functionalities required from the NIO to address these needs are described, highlighting their integration within the overall architecture. Moreover, the document discusses the interfaces necessary to facilitate communication between NIP components and external entities like the Radio Access Network (RAN) controller and the 5G Core systems. These interfaces enable designing procedures that tackle the needs and challenges introduced.
- ***Comprehensive literature review and research gaps that the DAEMON project is tackling, along with associated novel guidelines, to achieve a pragmatic design of NI.*** The document offers a final and comprehensive overview of the literature review conducted on integrating machine learning and NI in mobile network management. It highlights key trends in current research, showcasing the distinctive contributions made by the DAEMON project. The findings from this analysis further support the final updates to the project guidelines, including new guidelines, for practical NI design. These guidelines, as previously outlined in D2.2 [1], focus on two main directions: i) NI design tailored to the needs of B5G network management, orchestration, and control, and ii) NI design that emphasizes the utilization of more traditional, simpler, or interpretable models to avoid overburdening the system with data-heavy models.

This document serves as the foundation for the subsequent stages of the third iteration of the DAEMON project. Specifically, it will guide the updated design and implementation of NI-assisted functionalities by i) ensuring the fulfillment of all their requirements; ii) verifying that the proposed solutions meet the project's Key Performance Indicators (KPIs) in terms of performance, aligning with the functional and non-functional requirements and NI design guidelines outlined in this deliverable (including any requirements not yet achieved); and, iii) delivering a final version of the NI functionalities that fully aligns with the detailed architecture, including interfaces, and NIP procedures presented in this document.

1.3 Relationship to the other deliverables of DAEMON

Based on the discussion in Section 1.2, the relationship of D2.3 with the other project deliverables of the second and third are described below.

- ***D2.2.*** This document builds upon the requirements, novel NIP, and guidelines for the pragmatic design of NI defined in D2.2 [1]. In addition, it extends the vision for a NI orchestration framework by defining more detailed functional blocks, interfaces, and procedures to orchestrate intelligence, realizing a final version of the DAEMON's proposal for an NI plane.
- ***D3.2*** and ***D4.2.*** This document considers the outcomes of these deliverables of the project, which presented journaled and improved NI algorithms that adhere to the updated requirements presented in D2.2 [1]. Moreover, after providing a suitable representation of the NI algorithm, which are compatible and can be managed by the initial NI plane design, and the updates on how these NI algorithms operate across network functionalities after D2.2 [1], these deliverables also exposed the set of challenges and needs that raised when the NIP will orchestrate them.

- ***D5.2***. A comprehensive performance assessment was conducted within D5.2 [4] for the updated NI-assisted functionalities developed in WP3 and WP4 during the second iteration of the project. Such an assessment established a clear connection to the project's KPIs and specifically linked them to the requirements specified in this document.
- ***D3.3*** and ***D4.3.*** According to the overall framework defined by the DAEMON project, the solutions related to Real-Time Control and VNF Intelligence, and Intelligent Orchestration and Management are developed by WP3 and WP4, respectively, always considering the requirements set in the context of Task 2.1 in WP2 (NI functional and performance requirements). As discussed below, most of the requirements set for these families are already meeting (and sometimes even exceeding) the requirements. However, for some of them, the necessary details needed to understand why such requirements are eventually met will be described in the last iteration of WP3 and WP4 and hence provided in D3.3 and D4.3. In these cases, we clearly indicate for each requirement if that is the case.
- ***D5.3.*** A final comprehensive performance assessment will be conducted for the updated NI-assisted functionalities developed in WP3 and WP4 in this deliverable. D5.3 will include the performance evaluations required to achieve the requirements that have not been validated at the time that D2.3 and depend on them. For example, D5.3 will provide a set of performance evaluations of the initial proofs-of-concept on coordinating a pair of NI algorithms and the NIP components involved in them (see Section 5.3.1), which are required to finalize the set of requirements associated with the NIP (see Section 2).

1.4 Structure of the document

The high-level structure of this deliverable is summarized as follows.

- Section 2 provides the final version of the requirements for the eight NI-assisted functionalities addressed by DAEMON at the end of WP2. Although no new updates were introduced in the description of the requirements, which reflects their maturity at this point of the project, the risks involved in meeting these requirements and their status of completion are assessed. For requirements that are not yet finalized, the document specifies what is needed for successful completion and indicates in which deliverable the results will be provided.
- Section 3 presents the final architectural design of the NIP as a unified framework. It incorporates the operational hierarchy of NI components, their orchestration, and the N-MAPE-K representation of these components. This progress aligns with the envisioned complete NIP architecture from D2.2 [1].
- Section 4 and Section 5 delve into the specific needs posed by NI algorithms on the NIP and the required procedures to address them. Specifically, Section 5 analyzes the challenges faced in managing NI at the Network Intelligence Orchestrator (NIO) level and outlines the functionalities the NIO should provide to address these needs. The architectural design presented in Section 3 is supported by discussing the necessary interfaces described in Section 4 for communication between NIP components and external entities like the RAN controller and the 5G Core systems. These interfaces facilitate the design of procedures that tackle the introduced needs and challenges.
- Section 6 offers a final and extensive overview of the literature review on integrating machine learning and NI in mobile network management. It highlights key trends in current research and emphasizes the distinctive contributions made by the DAEMON project. The findings from this analysis also support the guidelines presented in the following Section.
- Section 7 provides the final updates to the project guidelines for practical NI design. These guidelines, similar to D2.2 [1], focus on two main directions: i) NI design tailored to the needs of B5G network management, orchestration, and control; ii) NI design that prioritizes the use of more traditional, straightforward, or interpretable models to prevent system overload with data-intensive models and promote the use of models that are easier to understand and interpret.
- Section 8 summarizes and concludes the work carried out in WP2 during the third iteration.

In addition, this deliverable includes two appendixes, which are presented next.

- Appendix A details the full list of requirement trees for each NI-assisted functionality, as well as for the NI plane. These tree structures are too long to be included in the main body of the document, but they complement the content in Section 2.
- Appendix B reports the complete taxonomy table of the related works studied by the project as part of the literature survey. Again, the table is too large to be included in the main body of the document, but it completes the discussion in Sections 6 and 7 of the deliverable.

2 Updated Network Intelligence functional requirements

Requirements elicitation is the process of gathering and specifying the prerequisites for a software system. DAEMON initially identified the functional and non-functional requirements for the eight Network Intelligence (NI)-assisted functionalities presented in Section 3 of D2.1 [8]. The main objective of this task was to establish a set of requirements that the design and implementation of NI functionalities should comply with. The requirement elicitation involves not only the identification and collection of requirements, but also their analysis and refinement during the project lifecycle. Appendix A in D2.2 [1] includes the updated state of requirements after the first iteration.

In this third iteration (see Table 1), only two new requirements have been included (FR-IBSSI-002 and NFR-NIP-009). FR-MTERM-004, FR-MTERM-004.01, FR-MTERM-006 and FR-MTERM-007.00 updated their KPIs. Lastly, FR-AARES-000 reduces its risk after averaging the children's initial risks. Table 1 summarizes the progress and evolution of the functional requirements over time in WP2. It can be observed that the project is now in a mature stage from the requirement perspective, so only minor changes (or even no changes) are expected at the end of the project.

Table 1. *Evolution of the functional requirements during the three iterations in WP2.*

Iteration	New	Updated	Reorganized	Total
Iteration 1	81	0	0	81
Iteration 2	24	18	12	102
Iteration 3	2	5	0	104

An important aspect to clarify in this iteration is that in Section 2.2 of D2.2 [1], we indicated that the NIP covered all the KPIs of the project. The main reason for this was to indicate that since the NIP will assist any NI functionality in achieving their associated KPIs, then those KPIs would also be covered by the NIP. However, in this document, we removed the KPI field from the NIP requirements to dispel any confusion. Moreover, the NIP is a novel component of future AI-native 6G architectures; therefore, their associated KPIs are still to be defined. As part of the effort to set a set of reference performance metrics, we expect to deliver a set of baseline reference values in D5.3 to achieve the completeness of the NIP requirements, as explained in more detail in Table 3.

According to the good practices recommended by the standard O/IEC/IEEE 29148:2018[1] on which we based our requirement elicitation documents, requirements have an associated risk level. In D2.2 [1], we preserved the initial risk estimations for the majority of the requirements and updated the risk management of a few of them after the first iteration.

In this deliverable, we have included new data to report the progress in meeting the requirements and how we mitigated the initial risks to achieve them after the second iteration. Table 2 shows a complete requirement table showing three new fields in a new section named ***Current Status***.

The ***Percent Complete*** reports up to what extent a requirement has been completed and validated at DAEMON. In order to estimate this value, we have analyzed the previous deliverables, the content of the current one, and all the papers and activities that report the work made in DAEMON until now.

The ***Risk Management*** field assesses how well the initially identified risks were tackled, taking into account we tried to reduce as much as possible their impact on requirements compliance. This value has been categorized as *Successful* (all identified risks were avoided or mitigated), *Effective* (some risks could not be fully canceled, but this only had a minor impact on the completion of the requirement), or *Partial* (some uncontrolled risks have impacted the requirement completion partially).

Considering the level of requirements completeness is strongly related to how well risks were managed, the ***Percent Complete*** and ***Risk Management*** values are correlated. Finally, the new ***Rationale*** field provides a justification for the ***Percent Complete*** and ***Risk Management*** fields. Within this text, we provide evidence supporting the Percent Complete and Risk Management values by referencing and linking to previous deliverables, featured articles, or sections of this deliverable.

[1] https://standards.ieee.org/standard/29148-2018.html

Table 2. *New fields are included to gather information about the Current Status of the requirement.*

FR-EAWVNF-001.00	
Description	DAEMON's EAWVNF shall measure the energy footprint of VNFs in terms of CPU usage.
Version	001M2
Owner	UMA
Priority	High
Risk	2
Risk Description	Calculating the cost of executing any kind of code, on specific hardware accurately is a complex task, since there are several factors that we need to quantify in order to calculate the energy footprint. The theoretical values given by CPU providers usually do not coincide with the real ones.
Rationale	We need to identify what are the factors that should be considered in the formula that calculates the global energy footprint of the VNFs instantiated for each application, in terms of computation. We know that the processor type of the device where a VNFs is running influences the energy footprint, but there are also other parameters that make the software provoke the hardware to consume more energy, like the size of VNF input.

K1	X	K2		K3		K4		K5		K6		K7		K8		K9	

Parents	FR-EAWVNF-001
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	In the placement solution presented in D4.2 [3], the energy model used to estimate the energy consumption explicitly includes the energy cost of computation calculated from the CPU cycles and the CPU frequency along with other factors. In the placement and autoscaling solution presented in D4.2 [3], the energy consumption model calculates the energy footprint of VNFs in terms of CPU usage according to the node in which VNFs are going to be deployed.

The new information provided in each requirement helps us understand the status of the functionalities assisted by NI. The information in Figure 2 and extended in Table 3 shows the completion percentage for the different functions. We can see that reported progress is above 70%, and some, such as NIP, CAWRS, or IBSSI, are almost or already completed. It is expected that there will be unfinished functionalities at this stage because some of them depend to a large extent on experimentation, the results of which will be obtained at the end of the project and will be presented in deliverables D3.3, D4.3, and D5.3, as listed and described in Table 3.

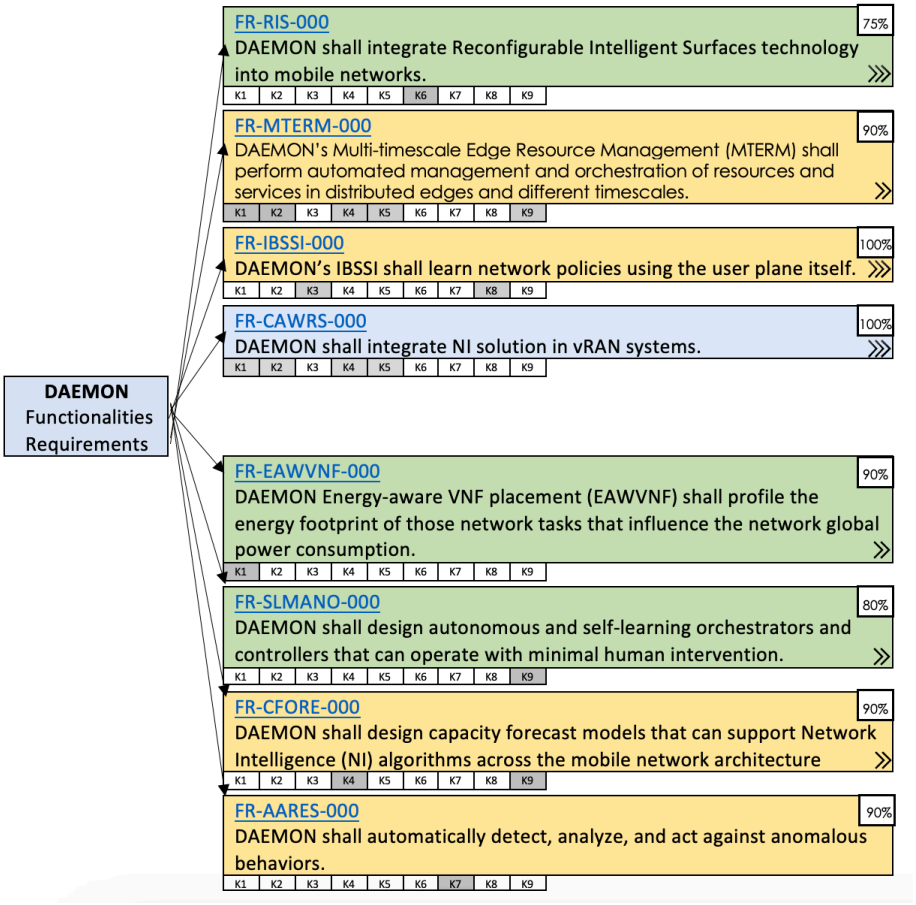


Figure 2. *First-level structure of the functionalities' requirements tree of the DAEMON project.*

Table 3. *Overall progression and risk management.*

Functionality group	Percent complete	Risk management	Explanation
RIS control (RIS)	75%	Successful	Reconfigurable Intelligent Surfaces (RIS) enable programming the wireless channel, conventionally considered a passive component, towards specific needs, e.g., focusing electromagnetic radiation towards specific locations. However, such a new dimension further complicates the intelligence in charge of optimizing wireless links. The ambition of DAEMON was to assess RIS-controlling NI empirically, which involved developing a complete RIS prototype. This involved severe risks of failing as RIS prototypes were nonexistent at the beginning of the project. Nevertheless, all the risks have been successfully mitigated, as will be described in D3.3 (design of NI) and D5.3 (empirical evaluation).
Multi-timescale Edge resource management (MTERM)	90%	Effective	The MTERM functionality aims at automatically manage and orchestrate resources and services in distributed edges and in different timescales. The solutions designed throughout the project's development fulfilled or are about to fulfill such a goal. For example, the solution presented in Section 3.1 of D3.2 [2] performs management and orchestration of services across multiple edges, as indicated by requirements FR-MTERM-004 and FR-MTERM-020. Similarly, the solution in Section 3.5 of the same deliverable manages resources in multiple timescales, as indicated by FR-MTERM-020. Finally, the risks were estimated as low-risk given the

			expertise of the consortium, which led to effective risk mitigation. Next, deliverables (D3.3 and D4.3) will present the final results on these solutions and a 100% completion is expected at the end of the project.
In-backhaul support for service intelligence (IBSSI)	95%	Successful	DAEMON developed solutions that operate in the user plane to support NI. Such solutions allow learning network policies using the user plane itself as well as performing inference at line-rate. All the risks were correctly managed during the project. For example, for FR-IBSSI-002, which aims at integrating Network Intelligence within programmable switches, risks were estimated as intermediate at the start of the activity, due to the limitation of the computing environment offered by programmable switch ASICs; such risks were avoided by using models that are relatively simple and mappings of such models that are tailored to the target hardware. Performance has been shown to achieve the envisioned accuracy. Similarly, for FR-IBSSI-002.01, which indicates that DAEMON shall handle both packet-level and flow-level inference, and which suffers from the same risks, these risks were avoided by designing novel approaches to feature representation that suited the target hardware.
Compute-aware radio scheduling (CAWRS)	100%	Successful	This functionality takes care of embedding Computing Awareness in the Wireless Access Network functions such as the ones running in the gNB. All the requirements related to this group have been successfully fulfilled, from both the algorithmic and the performance perspective. The main design principles and results are already available in D3.2 [2] and D5.2 [4], while further improvements are going to be introduced in D3.3 and D5.3.
Energy-aware VNF placement (EAWVNF)	90%	Effective	Energy-aware VNF placement functionality aims to monitor and measure energy efficiency. We already identified the factors that significantly impact energy consumption (D3.1 [5], Section 5.3.2.1, D3.2 [2], Section 3.3.3, and journal [9]) requested by requirement FR-EAWVNF-002 and their derived ones. Those factors were used to estimate energy consumption (FR-EAWVNF-001 and its derived requirements) and to monitor the impact of hardware resources of VNFs (solutions presented in D4.2 [3]; FR-EAWVNF-2). The cost in terms of migration (FR-EAWVNF-003, FR-EAWVNF-004 and its derived requirements) and scaling (FR-EAWVNF-006 and its derived requirements) are partially solved, as presented in the energy-aware placement solution for VNFs (section D4.2 [3]). Next deliverable D3.3 will extend the empirical evaluation of SAVRUS, demonstrating that SAVRUS generates meaningful results sufficiently fast by analyzing SAVRUS validity and SAVRUS scalability. It will also update the SAVRUS algorithm with Inductive Transfer Learning and Scoring Functions to improve the creation and updating of the energy-aware rankings of features. Also, it will describe changes in the algorithm to reduce the Curse of Dimensionality and the Negative Transfer, two drawbacks of the algorithm. In the following deliverable, D4.3, the iTarea algorithm will be updated to incorporate the energy consumption prediction provided by a Gradient Boosted Regression Trees algorithm.

Self-learning MANO (SLMANO)	80%	Effective	DAEMON designed autonomous and self-learning orchestrators and controllers that operate with minimal human intervention. In particular, the set-up and life cycle management we investigated via learning placement/routing algorithms and self-tuning control loop respectively. While the design principles are introduced in D4.1 [6], updated in D4.2 [3] and will be completed in D4.3, the performance results have been reported in D5.1 [7] and D5.2 [4] and will be complemented in D5.3. All proposed tools were evaluated on artificially generated data, which was constructed to be as close to realistic data as possible (e.g., exposing diurnal patterns, random noise, flash crowd). Nevertheless, testing them on actual data (once they become available) is reserved for future work.
Capacity forecasting (CFORE)	90%	Effective	DAEMON designed Capacity Forecasting models capable of anticipating the amount of resources needed to accommodate future mobile service demands, so as to support Network Intelligence (NI) algorithms across the mobile network architecture. This largely achieved the targets set at the beginning of the action, including aspects such as the capability to operate at different timescales (FR-CFORE-001) and on streaming data (FR-CFORE-003), the awareness of monetary costs for decision-making FR-CFORE-002), or the possibility to self-learn the objective loss function (FR-CFORE-005). Details on the design and evaluation of the models are provided in D2.2 [1], D4.2 [3] and D5.2 [4], in addition to refinements that are developed in the last iteration of the project, as presented in Section 7.1.3 of the present document and later complemented in D4.3 and D5.3.
Automated anomaly response (AARES)	90%	Successful	DAEMON implements three different activities for real-time anomaly detection and automated anomaly response, namely, A9, A19 and A25, as reported in D5.2 [4]. We provide details on the solution and its implementation in D4.2 [3] and D3.2 [2], which we will complement with their final status in D4.3 and D3.3, respectively. The reported status in D5.2 [4] shows an average completion of approximately 70% towards collecting the corresponding KPIs, which we will further update in D5.3.
Network Intelligence Plane (NIP)	95%	Successful	The DAEMON's NIP shall manage, coordinate, and orchestrate network intelligence with a closed control loop to meet service KPIs in different micro-domains. All the risks were correctly managed during the project by ensuring that each iteration (first in D2.1, second in D2.2 [1], and final in D2.3) of the architectural design added or improved the required functionalities, thus avoiding the initially foreseen risks. An example is FR-NIP-002, which aims to define internal and external interfaces. This deliverable provides the functional blocks and procedures necessary to realize it. This is also confirmed by the percentage of completion of the set of requirements at this point, which is expected to achieve 100% by the end of the project once we provide a set of measurements of the NIP's performance in D5.3. These performance measurements are intended to be used as reference values for future implementations of the NIP beyond the lifetime of the project (see FR-NIP-003 and FR-NIP-005 for more details).

Some factors may affect a completion percentage. The first is the priority. A requirement with low priority will receive fewer resources to be completed than others with high priority. That factor can be controlled since more resources can be allocated. However, other factors, such as risk can affect the completion rate as some of them cannot be completely avoided or mitigated. Figure 3 shows the relation between the risk level (estimated during the elicitation) and its management (reported during the execution after two iterations). As expected, requirements with lower risk level estimates (1 or 2) have had better risk management (75% successful, 23% effective, 2% partial). Conversely, the requirements with an estimation of high-risk levels (4 or 5) reported difficulties during risk management (38% successful, 43% effective, 19% partial). Overall, it can be observed that risk management has been successful in DAEMON: 65% successful, 28% effective, and 7% Partial.

Figure 3. *Initial risks and their management.*

This success in identifying and managing risks has positively impacted the completion percentage. Figure 4 shows how requirements' risk levels (estimated in advance) are related to the completion percentage (reported currently). It can be observed that every requirement has some degree of completion, even the ones with the highest risk level, meaning that risk management has been greatly successful. It also shows that the risk level estimation was properly assessed during the requirements elicitation. Finally, the requirements with risk identified as low are all nearly complete, regardless of their priorities.

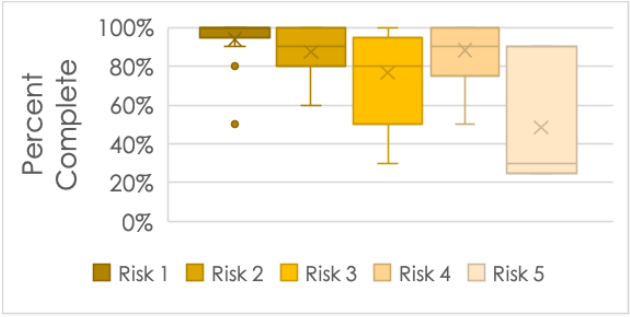


Figure 4. *Estimated risk versus percent completed.*

Notice also that although some KPIs were only covered by single NI functionality during the requirement definition phase (e.g., AARES - Automated anomaly response and RIS - Reconfigurable Intelligent Surfaces), most of them have achieved a high completion status at this point and it is expected that they continue its progress to achieve its 100% by the end of the project. This is also a result of the DAEMON's strategy of having multiple partners working on different algorithms under the same NI functionality (see Table 17 D5.2 [4]) or exceeding the expectations of some functionalities that were able to provide results to a KPI that was initially not planned as a requirement (see Table 24 in D5.2 [4] where K7 was evaluated IBSSI - In-backhaul support for service intelligence).

3 Network Intelligence Plane: Final architectural design

B5G and 6G networks set forth a vision for end-to-end NI coordination aimed at ensuring a conflict-free and synergic operation of the many NI algorithms running across schedulers, controllers, and orchestrators in the network. As a first step in the rigorous design of a complete framework for the joint operation of NI instances, we have identified a set of gaps in the current frameworks for network management that the main Standard-Defining Organization (SDO) entities, such as 3rd Generation Partnership Project (3GPP) and the European Telecommunications Standards Institute (ETSI), as well as by global industrial initiatives like O-RAN, are not currently delivering to support the native integration of NI and, subsequently, its practical adoption within 6G networks. These gaps can be the following: they do not provide (i) mechanisms to coordinate intelligence across different network micro-domains or (ii) solutions for decentralized and unified data management across NI instances. Also, their (iii) support for managing the NI lifecycle is minimal, and there is only an early consideration for (iv) methodologies for the defining and representing of NI models. Table 4 summarizes the main gaps, based on the detailed analysis presented in Section 10, Appendix B in D2.1 [8], and how the NIP contributes to fill such gaps.

Table 4. *Main gaps in SDOs and networking-related frameworks with respect to NI functionalities.*

Framework	Methodology to define NI	Mechanisms to manage the lifecycle of NI	Mechanisms to coordinate NI across different network segments	Decentralized and unified data management for NI instances
ETSI MEC	No	No	No	No
ETSI NFV	No	No	No	No
ETSI ENI	Yes	No	No	No
O-RAN	Yes	Partially	No	No
Open Source MANO (OSM)	No	No	No	No
3GPP	No	No	No	No
ONAP	No	No	No	No
Network Intelligence Plane	Yes [Addressed in D2.2 [1], Section 3.1.2]	Yes [Initially addressed in D2.2 [1], Sections 2.1, 3.1.3 and 3.1.4, and extended in D2.3, Sections 2 and 3]	Yes [Initially addressed in D2.2 [1], Sections 2.1, 3.1.3 and 3.1.4, and extended in D2.3, Sections 2 and 3]	Yes [Addressed in D2.1, Section 5.1 and 6.2]

To tackle the gaps mentioned above and remove the current barriers to fully support the aspects not necessarily covered by existing frameworks, DAEMON has outlined a clear set of functional and non-functional requirements, targeting the coordination of NI instances in an end-to-end fashion (see D2.2 [1], Section 2.2 and Section A in this document). This set of requirements includes developing synergies in terms of data management and handling the interaction with Machine Learning Operations (MLOps) platforms, managing the complete lifecycle of both complex NI instances and atomic NI functions, the maintenance of catalogs of NI models that ease de-composition and orchestration. Based on those requirements, ***The DAEMON project has proposed in D2.2*** [1] ***a novel NI Framework for 6G networks and proposed the Network Intelligence Plane (NIP),*** a collection of modules and interfaces responsible for managing NI within the network, as shown in Figure 5. However, ***this document introduces a refined architecture that has been adopted by the 5G Infrastructure Public Private Partnership (5GPPP) Architecture Working Group as the Network Intelligence Stratum*** [10], [11]. This term has been embraced as part of the comprehensive architectural framework that has been developed by the WG, as illustrated in Figure 6 and reported in the whitepaper "6G Architecture Landscape – European perspective" [11], released by the 5G Architecture Working Group in the 5GPP.

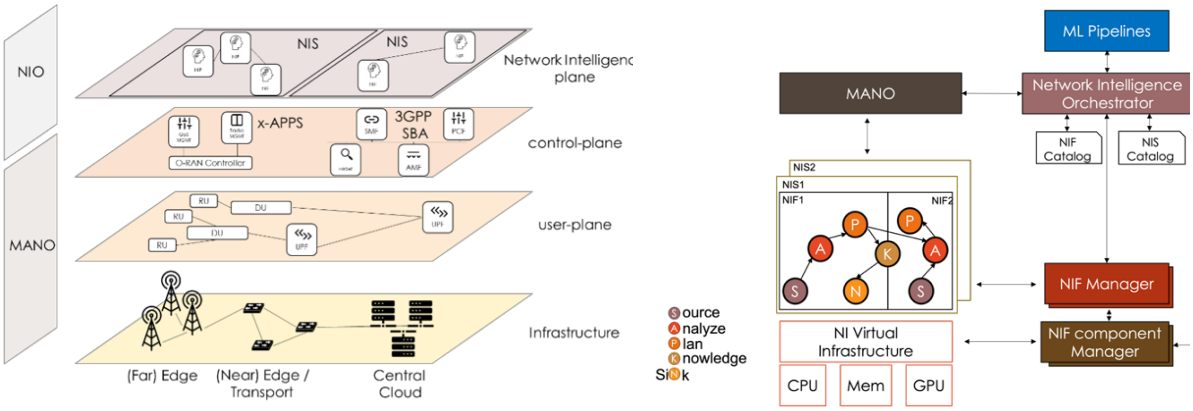


Figure 5. *A NI framework for 6G network (left) and the NIP functional blocks (right).*

The motivation behind this terminology shift is to align with the established usage in 3GPP, where the term "stratum" typically denotes a collection of elements that span various network domains. For example, the term "network access stratum" encompasses all the elements involved in user registration and authentication across the RAN and the Core. Considering that network intelligence components are distributed across multiple domains such as access, core, infrastructure, management, and orchestration, it was only natural to adopt this terminology in line with 3GPP standards. ***Moreover, this approach also moves the NIP design from a purely separate plane to a more orthogonal approach where NIFs and NISs can effectively be integrated into the traditional planes (data, control and management) for easy adoption in the industry.*** In order to maintain consistency and coherence with D2.2 [1], in this and future deliverables of the project, we will continue to use the term NIP.

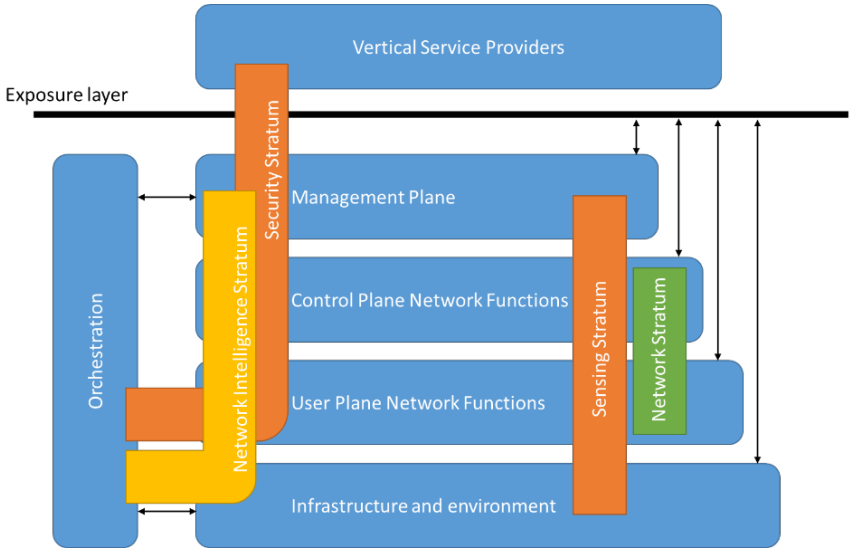


Figure 6. *The architectural framework proposed by the 5GPPP Arch WG* [11].

In the project effort to define the NIP organization and operations, we already introduced a reference representation of complex NI algorithms as a hierarchy of Network Intelligence Services (NISs) that can be broken down into one or more Network Intelligence Functions (NIFs), which, in turn, are composed of atomic NIF Components (NIF-Cs) [12], as represented in Figure 7. We also specified how NISs and NIFs can be managed by a Network Intelligence Orchestration (NIO) with a precise internal structure of fundamental building blocks.

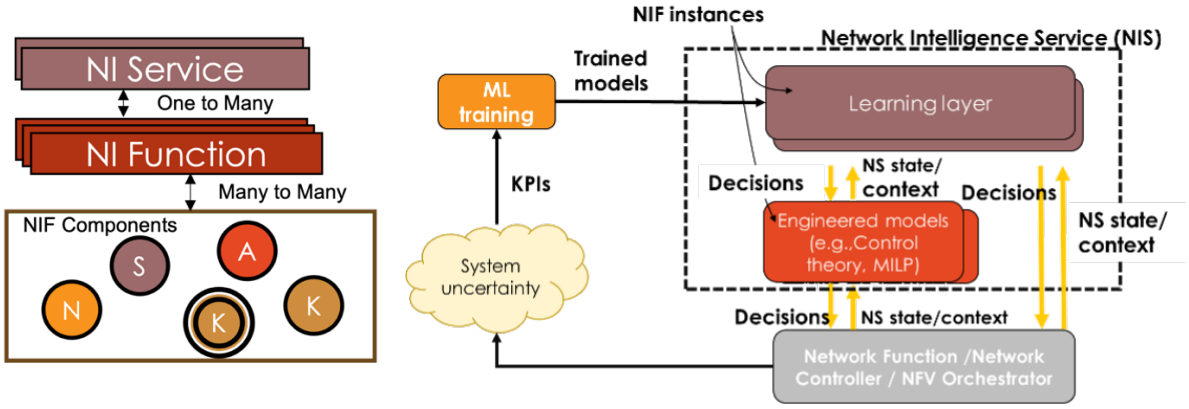


Figure 7. *The high-level hierarchical taxonomy of Network Intelligence algorithms. A NIF corresponds to an individual NI instance that assists a specific functionality: for example, it could capture the implementation of a capacity forecasting task, assisting an NI edge orchestration functionality.*

In addition, in a separate work, we defined a suitable reference representation to be adopted by the NIO to model any NI algorithm [13]. To that end, we adopted and adapted a popular model widely adopted for autonomous and self-adaptive systems, i.e., the Monitor-Analyze-Plan-Execute over a shared Knowledge (MAPE-K) feedback loop [14]. Building on top of the MAPE-K representation, we dissected NI algorithms into common elements that have different characteristics (e.g., a data-gathering probe or a Neural Network model) and introduced original training and closed control loops that a NIF may implement, which resulted in an extended Network MAPE-K (N-MAPE-K) model tailored to the NI environment, which is shown in Figure 8. The N-MAPE-K model allows capturing (i) the inference loop, (ii) a traditional supervised training loop, and (iii) a second training loop dedicated to online learning.

Mapping NI algorithm components into the N-MAPE-K representation allows highlighting the following fundamental classes of atomic NIF-Cs.

- Sensor NIF-Cs specify all the probes needed to gather the input measurement data.
- Monitors NIF-Cs specify how each NIF interacts with the Sensor NIF-Cs and gathers raw data from them.
- Analyze NIF-Cs include any pre-processing, summary, or data preparation for the specific NI algorithm implemented in the plan NIF-Cs.
- Plan NIF-Cs constitute the specific NI algorithm implemented by the NIF.
- Execute NIF-Cs specify how the algorithm is going to interact with the managed system and how to possibly change its configuration parameters.
- Effector NIF-Cs specify the configuration parameters updated in the Network Function (NF), and the Application Programming Interfaces (APIs) to be used to that end.

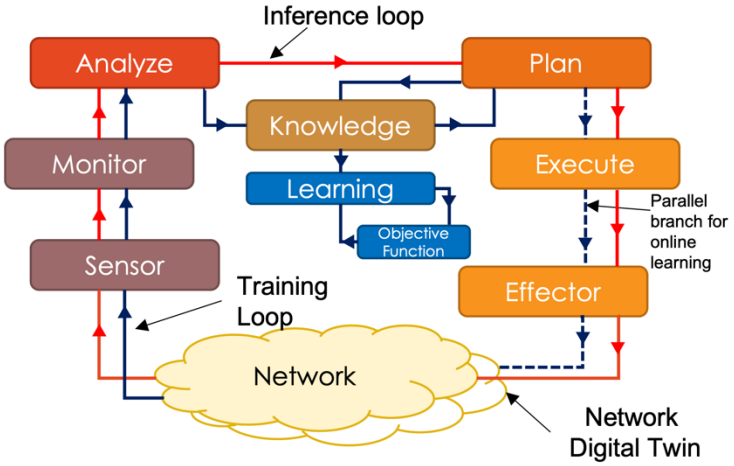


Figure 8. *Extended N-MAPE-K abstractions for NI algorithms.*

DAEMON's NIP is a unified framework that brings together our earlier proposals for (i) the operational hierarchy of NI components in the NIP and (ii) the N-MAPE-K representation of NIF-Cs. By doing so, we make a step forward toward the vision of a complete NIP anticipated above. An illustrative example of

the resulting integration is provided in Figure 9. There, each circle depicts a NIF-C and a double circle captures a NIF-C shared among multiple NI-assisted functionalities. For example, the circle in the O-Cloud rectangle captures the Forward Error Correction (FEC) decoder. Multiple united NIF-Cs constitute a NIF, e.g., Nuberu [15] or Henna [16], to mention two solutions developed in the project itself. Finally, by combining NIFs we get a NIS: as an example, the integration of different RAN-related algorithms can realize an overall reliable virtualized RAN (vRAN) service. Table 5 presents several examples of NIS and their respective NIF based on some NI-assisted functionalities developed in the DAEMON project [2], [3].

***Table 5**. Examples of several NISs, their NIFs, and their associated KPIs.*

NIS	KPIs	NIF
Reliable Virtualized RAN		Reliable distributed unit (DU) for virtualized RAN [15]
		Orchestration of radio and computing resources in vRANs [17]
Sustainable network operation	VNF Energy Savings	Cloud Acceleration for virtualized RAN[18], [19]
		Compute Aware scheduling analytics [20]
		AI-enhanced edge orchestration [21]
	Compute Resource Savings	Data-driven resource orchestration [22]
	OPEX Savings	Multi-timescale network slice reservation [23]
Network capacity management	Wireless Capacity Increase	Reconfigurable Intelligent Surfaces Control [24]
		Accurate WLAN performance prediction in dense environments [25]
Edge orchestration	OPEX Savings	Network Service Auto-scaling [26], [27]
		Capacity forecasting [28]

The variety of NIF and NIS that can be deployed at the network generates new challenges in the way they should be managed that are not presented in current management frameworks. In the rest of Section 3, we will present the final DAEMON architecture. We will first identify and present in detail the specific needs that NI algorithms pose on the NIP and understand their specificity in terms of challenges in the procedures for NI management at the Network Intelligence Orchestrator (NIO) level. Then, we will devise and describe the functionalities that the NIO shall provide to support such requirements (Section 3.1) and how they fit the whole architecture together (Section 3.2). The architectural design will be later complemented i) in Section 4 by presenting and discussing the interfaces that are required to allow communication between NIP components, and the NIP components with external entities such as the RAN controller, Core system, and local and end-to-end management systems, and ii) in Section 5 by designing the set of procedures that address the needs and challenges introduced in Section 3.1 and that motivate the functionalities presented in Section 3.2, together with some reference implementations as Proof of Concepts (PoC) in Section 5.3.

3.1 The need for specific NIP Procedures

The concurrent instantiation of many different NIFs/NISs raises challenges that the architecture we propose allows addressing. Next, we detail the management needs that such challenges create, and exemplify them with representative NI-assisted functionalities developed in the DAEMON project [2], [3].

3.1.1 Conflict resolution

DAEMON's NIO allows to efficiently re-use and combine different elements that can be shared across NIFs, by representing their split into atomic NIF-Cs that abide by the N-MAPE- K framework [14]. This eventually enables building in an effective way a NIS, analogously to the approach used by 3GPP SA5 to build the Network Slicing data model –where a Network Slice is decomposed into Network Slice subnets. However, while composing NIFs to build a NIS, through the sharing of different NIF-Cs, possible conflicts on operations and/or resources may arise. It is hence a task of the NIO to arbitrate the operation of such components, guaranteeing that the overall goal of the NIS is met.

Let us illustrate this issue by detailing the arrangement of Nuberu and Athena, two NIFs described in D3.2 [2] (Sections 2.1 and 2.5, respectively), that aim at improving the resiliency of a virtualized radio access network (vRAN) system by acting on the Medium Access Control (MAC) scheduling decision at the Distributed Unit (DU) of base stations. Nuberu [15] proposed a re-design of the full stack to be cloud-native and resilient, while Athena introduced a model that learns the limits of the infrastructure and takes scheduling decisions. Thus, both algorithms support the Radio MAC scheduler acquiring knowledge from similar input data (e.g., the information about the channel) and enforcing radio scheduling decisions,

optimizing the reliability of the system, at different timescales. This results in the sharing of two NIF-C, the sources and the sinks between these two NIFs, as also shown by the N-MAPE-K representation depicted in Figure 9.

A similar consideration applies when dealing with mixed user and control intelligence, as in the case of the algorithms in Section 5.1 of D3.2 [2], whose goal is (i) performing in-switch inference at line rate [16]and (ii) achieving optimal configuration of circuit switching by using real-time traffic demands. The NIF implementing in-switch inference, i.e., NIF1 in Figure 9, acts almost entirely in the user plane, directly classifying IP traffic and directly enforcing decisions into the NF that is classifying the traffic, the switch controller in this case. NIF2 in the figure generates the circuit switching configuration in the control plane instead and enacts it in the user plane. The configuration decision is taken based on information about traffic volume, which is available at each switch.

The previous two examples, in which different NIFs share sources and sinks, motivate the need for monitoring and coordination of policy enforcement. Here, different conflicts may arise, as follows.

- Conflicts when monitoring data. Algorithms may need data from the same source but with different granularity. Hence, the NIF Manager shall guarantee that the required information arrives from the Sources to the specific Plan/Analyze modules with the necessary granularity (e.g., at subframe or packet level) in an automated manner to, e.g., avoid duplicating the monitoring over IP packets.
- Conflicts in the policy enforcement. Different NI algorithms may act on the same network functions (in the proposed example, the DU MAC scheduler), configuring different parameters. Thus, the NIO shall deploy conflict resolution policies with the NIF-C of each NIF to guarantee that, e.g., the scheduled MAC frame never exceeds the available capacity or contrasting selected users.

Therefore, the NIO shall oversee and amend any suboptimal decision taken by individual NIFs by closely monitoring the access to data sources and the policies determined by decision-making algorithms.

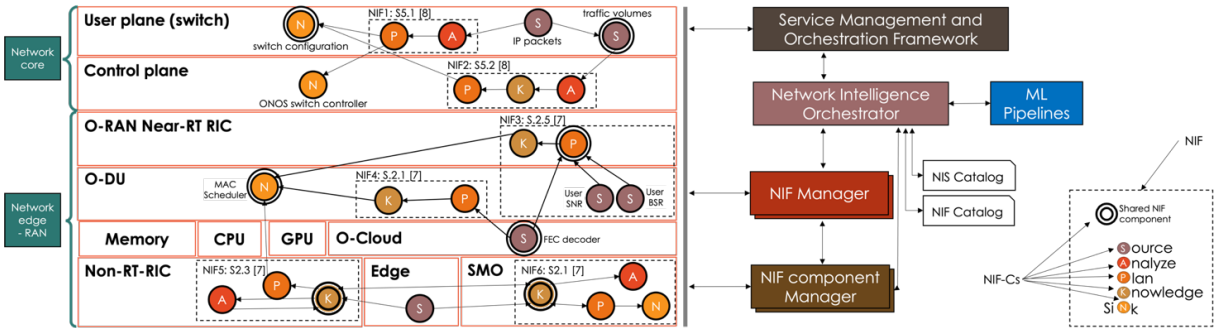


***Figure 9**. NI-native architectural concept proposed by the DAEMON project for the NIP. The diagram portrays the interactions between many different NIFs that implement two NI-assisted functionalities, or NIS, also developed in the project. The NIF-Cs that compose each NIF are categorized using our original N-MAPE-K representation. The hierarchies of NISs, NIFs, and NIF-Cs are managed all at once by the NIO framework, by avoiding conflicts and leveraging synergies among them.*

3.1.2 Knowledge sharing among NIFs

Figure 9 also illustrates the shared representation of two NIFs detailed in D4.2 [3] (Sections 2.3 and Sections 2.1, respectively): energy-driven vRAN orchestration, i.e., NIF5, and energy-aware VNF placement, i.e., NIF6. In the case of these two NIFs, energy consumption measurements from an edge cloud platform are required and a source node component is shared. Moreover, NIF5 generates knowledge about high-performing RAN control policies given a context and once virtualized instances of RAN components have been deployed. On the other hand, NIF6 is in charge of VNFs placement, which in this case, implements virtualized RAN functions. In this context, the NIO shall provide centralized coordination among multiple NIFs. Such centralized coordination would allow sharing of knowledge that fostered synergetic performance improvements between both NIFs. For instance, part of the knowledge learned by NIF6 can be used by NIF5 to make better placement decisions and, vice versa, NIF6 can use some knowledge learned by NIF5 to enforce informed (placement aware) RAN control policies.

Knowledge-sharing aspects should also be available cross-domain. For instance, in Section 4.2 of D4.2 [3], we describe an anomaly detection solution for IoT platforms. In that scenario, the user plane traverses multiple domains, bringing new challenges in running root-cause analysis of anomalies. Hence, the parties involved in building the user plane for the IoT devices suffering from anomalies should be integrated into the anomaly detection scheme, and such synchronization shall happen at the NI Orchestration.

3.1.3 Model selection, catalog, and re-training

Although this is not a condition directly stemming from the NI algorithms' design, NISs may need to build on the knowledge of the underlying environment. This calls for awareness of the software/hardware environment (e.g., as the performance of a specific FEC implementation depends on the target hardware [17]) or of the location of the device where they are executed (e.g., as reconfigurable intelligent surfaces may have different behaviors according to their geographical position and surrounding environment [29]). When executed in the context of a pure ML environment, these tasks are natively tackled by several MLOps frameworks such as Kubeflow[2] and MLflow[3]. In the context of an NI-native architecture, however, this requires tight interaction with the underlying orchestration environment. To guarantee that the deployed NIF can operate in the right context, NI models must match the specific hardware-software-environmental characteristics of the network functions deployed in a network service. Thus, the NIO shall exchange execution context information with the sibling Management and Orchestration (MANO) operating in the network to select the proper model to be used for inference within a NIF. This incidentally calls for the need for a model catalog from which the NIO can select the most appropriate model depending on the specific infrastructural status operated by the network at a certain point in time. If no model is available for the specific execution environment, the NIO shall be able to invoke the training of a new model, fetching the required data as required by the target algorithm.

3.2 NI Orchestrator functionalities to enable NI-Native architectures

As described in the previous section, several considerations and challenges emerge while concurrently deploying multiple NIFs providing the same or different NISs. Building on the NIO organization and N-MAPE-K representation of NIF-Cs, we next define processes that answer such needs.

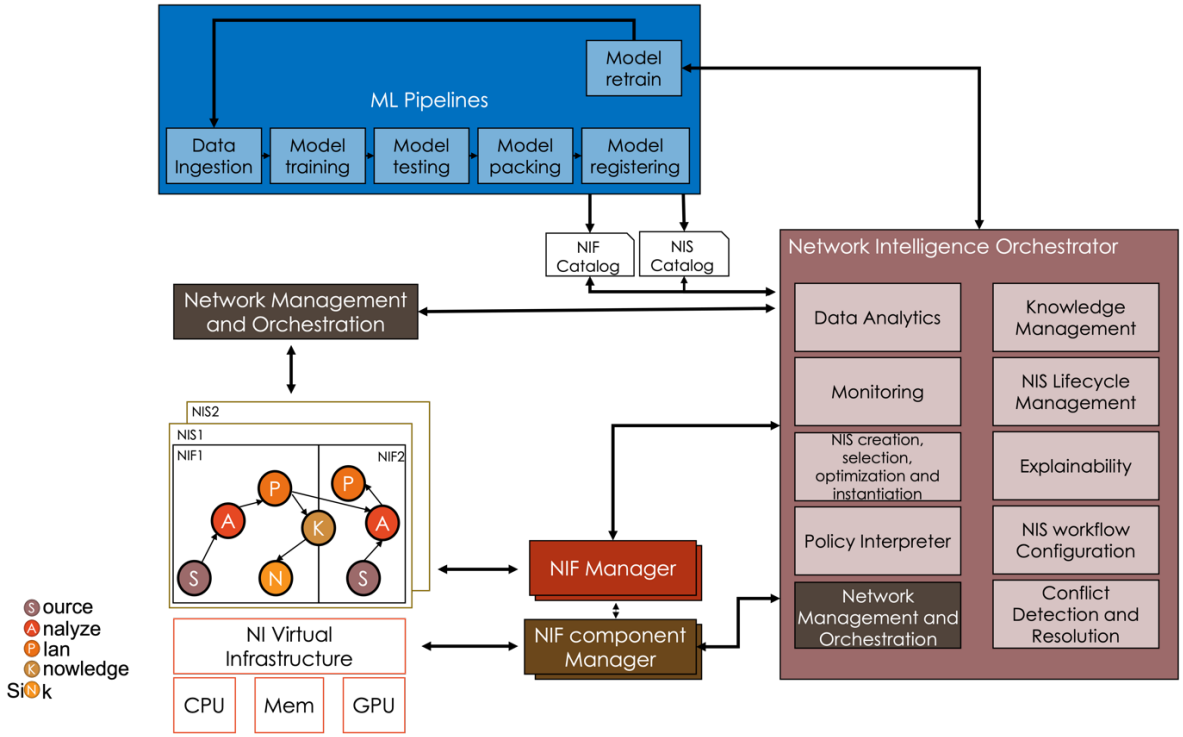


Figure 10. *The NIP and the functional blocks of the Network Intelligence Orchestrator and ML pipelines.*

3.2.1 Rationale

When used outside the network domain, the set of solutions that deal with the lifecycle management of intelligent algorithms is usually referred to as MLOps [30]. Items such as Feature Engineering, Model Training, Model Engineering, as well as their integration in a Continuous Integration (CI)/Continuous Deployment (CD) system are usually encompassed in this definition. When transferring this view into the mobile network realm, however, these items cannot be transferred as is, mostly because of the very different timescales that are usually involved in network environments, which may go down to sub-ms levels. Therefore, we split the items into elements that are only related to pure ML tasks and are commonly

[2] https://www.kubeflow.org/

[3] https://mlflow.org/

executed offline, either only once or very rarely. We mark them as Machine Learning Pipeline in Figure 10. Instead, other elements need to directly interact with the NIFs in the network, continuously evaluating the quality of the NIS and performing fine-grained lifecycle management of the NIF-Cs, including their coordination. These are the most interesting in the context of the NIP, and we discuss them next.

3.2.2 Overall description

As mentioned in the previous section, the NIO should incorporate multiple functionalities to support the described challenges and beyond. Some key functionalities are shown in Figure 10. Their main purposes are as follows.

- ***Data analytics.*** This block includes any pre-processing or preparation of the data (e.g., averages, autoencoders, filtering, or clustering algorithms).
- ***Knowledge management.*** A critical component of the NIO, the knowledge management block provides all the mechanisms required to plan, organize, act, and control the knowledge across all the deployed NIS.
- ***Monitoring.*** This block processes the NIS's information. As NIS can be composed of both non-ML (e.g., traditional VNFs) and ML-based functionalities, the monitoring information can also be of both types: ML-related (e.g., model-specific metrics and detection of data drift for essential features), and non-ML-related (e.g., QoE, QoS, etc.). In addition, this block will monitor NIs in both training and inference deployments.
- ***NIS lifecycle management.*** This functional block handles the deployment and maintenance of working ML models, aligned with MLOps practices. This includes the creation of new ML pipelines to re-train ML models.
- ***NIS creation/selection, optimization, and instantiation***. Before any deployment, the NIO has to select (e.g., based on hardware constraints), optimize (e.g., compress a Neural Network (NN)-based NIS to achieve a given tradeoff between model size and performance), and instantiate the selected NIS. If a given NIS is unavailable in the catalog, the NIO should be able to create it based on the available data and execution context information.
- ***Model explainability***. This block provides the methods that help human experts understand NIS composed of black-box (e.g., deep neural network) ML algorithms. This is a fundamental capability to understand the cause of a decision from a NIS such that a human can consistently review/correct its results.
- ***Policy interpreter and configuration***. This functional block interprets high-level user intent objectives, e.g., high-level QoE targets and business KPIs, that are associated with different NIS. If needed, it also performs changes in the policy.
- ***NIS workflow configuration.*** This block puts together data engineering, ML, and DevOps in a more straightforward, efficient, and effective fashion. In a general perspective, the NIO uses NIS workflow configuration to operationalize the deployment, monitoring, and lifecycle management in a modular and flexible way.
- ***Network MANO framework.*** This functional block manages the lifecycle management of the traditional Virtual Network Functions (VNF) that communicate with a NIS/NIF. In addition, it provides the context execution information from the network. Notice that in Figure 10, there are two MANO functional blocks, one internal and one external. The main reason is to show that MANO functional block can be an external or internal block of the NIO, depending on its implementation. We will discuss more details about the interactions between NIO and MANO functional blocks in Section 3.2.3.
- ***Conflict detection and resolution***. This block provides a mechanism to solve trade-offs that may emerge from conflicting objectives in the control and user planes, e.g., in establishing policies (at small timescales) versus enforcing such policies (at large timescales). This functionality allows the NIO to compare policies among different NIS to detect conflicts and perform conflict resolution based on comparison and resolution rules.

3.2.3 MANO and its interaction with the NIO

The NIO is crucial in coordinating and managing network intelligence. To fulfill its responsibilities effectively, DAEMON must seamlessly interact with the MANO framework in key areas.

Firstly, the NIO requires the synchronization of network slices and functions within the MANO framework. The NIO exchanges information with MANO components to track the state and health of network slices and the operational status of functions. This real-time synchronization enables the NIO to dynamically adapt decisions and optimize network operations. Secondly, the NIO relies on up-to-date information about available resources as available in the MANO framework, including computing power, storage,

and network characteristics (both in the wired and wireless parts). By maintaining a synchronized view of resource availability, the NIO can efficiently orchestrate the network resources that may allocate resources to network slices, functions, or specific vertical service requirements. This dynamic resource management optimizes resource utilization and enhances performance.

Additionally, the NIO may interact with vertical service providers, who have unique requirements for the network infrastructure. Effective communication channels should be established to exchange information, feedback, and service-specific instructions. This ensures that NIO aligns the orchestrated network intelligence with the objectives and needs of vertical service domains, enhancing overall service delivery and user's QoE.

In summary, the NIO collaborates with the MANO framework through various means. It can establish connections using an eastbound-westbound interface to enable seamless communication and integration. Alternatively, the NIO can directly extend modules within the MANO framework if applicable. For example, when paired with the ETSI Network Function Virtualization (NFV) MANO framework, specific mappings can be established between the NIO and corresponding MANO components. The NIO can align with the NFV Orchestrator (NFV-O) within the ETSI NFV MANO framework, ensuring coordination and cooperation between the two. Similarly, the NIF Manager of the NIO can be mapped with the Virtual Network Function Manager (VNFM) in the MANO framework, facilitating the management and control of virtual network functions. Additionally, the NIF-C Manager of the NIO can correspond to the Virtual Infrastructure Manager (VIM) within the MANO framework, enabling efficient management of the underlying virtualized infrastructure.

In the following sections, we will describe the internal and external interfaces that must be defined to allow communication between internal and external components and how the combination of some functional blocks in the architecture can help to address the challenges described in Section 3.1.

4 NIP Interfaces

As shown in Figure 10, the NIP is a composition of different functional blocks that aims for the native integration of NI in the network by providing the management and orchestration capabilities for NIF and NIS. Similar to other well-known frameworks for management and orchestration on specific domains, e.g., NFV-MANO [31] and O-RAN [32], the functional blocks of the NIP have their own set of internal interfaces. In the following sub-sections, we will provide a high-level definition of such interfaces and what is expected from them.

4.1 Internal Interfaces

To successfully orchestrate and manage NI, it is essential to establish seamless communication and coordination among the various functionalities of the NIP. In the subsequent section, we will outline and elaborate on the specific set of internal interfaces that are presented in Figure 11. These interfaces are the foundation for enabling effective communication and coordination among the different blocks within the NIP, ensuring a harmonized and cohesive NI management framework.

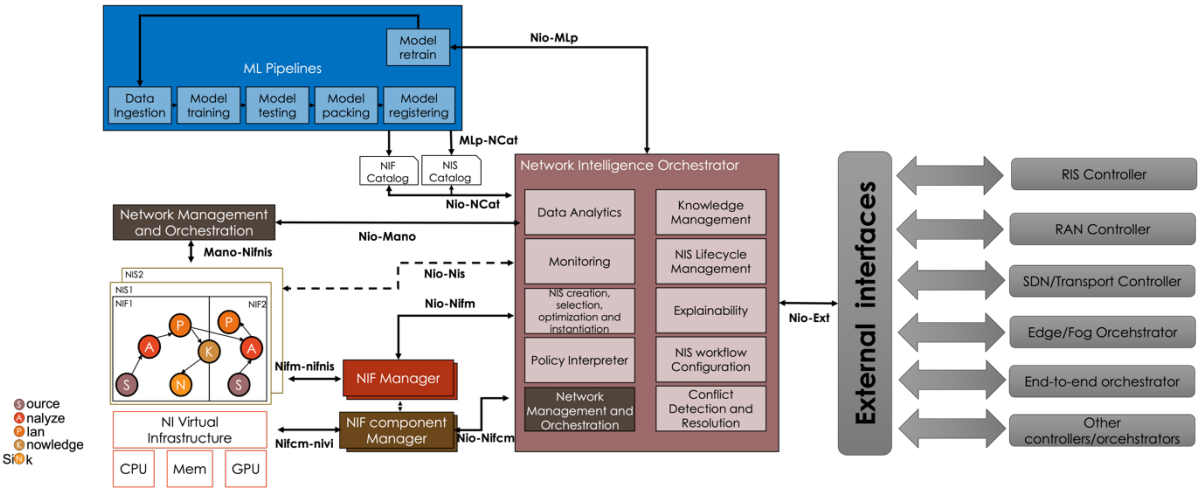


Figure 11*. Interfaces between functional block in the NIP.*

In the following, we present them according to their functional definition, although from an implementation perspective, they could be provided in a service-based fashion.

- ***Nio-Nifm.*** This interface allows communication between the NIO and the NIF Manager to effectively manage and orchestrate NIF instances within the NIP framework. It promotes efficient utilization of network resources, optimized network service delivery, and enhanced scalability and flexibility of virtualized NIF. Among life-cycle management, the NIO relies on the NIF Manager to perform operations related to NIF instances, including instantiation, scaling, healing, and termination.

  Via this interface, the NIF Manager can also provide monitoring information about the performance (reporting metrics related to both the learning process, e.g., the loss function when trained, or network performance indicators) and health, and trigger healing actions in case of failures, degradations, or conflicts. Moreover, the NIO can gather information related to the status of the NIFs so it can derive analytics to proactively optimize the NIFs (e.g., by changing the learning model data feeding speed/timescale to mitigate limitation on available computing resources) or control it (e.g., by adding a new input representation of the data or ML model to couple it with other NIF when instantiating a new NIS). Finally, the NIO can also gather information from the NIFs related to explainable capabilities and use it to take better orchestration and coordination actions among NIFs. Finally, this interface will allow the NIO to perform ML workload management.
- ***Nio-Nifcm.*** This interface allows the NIO to request the NIF-C for the allocation, placement, and lifecycle management of virtualized infrastructure resources. These resources include computing (GPU, FPGA, CPU, memory), storage, and networking components required to host and run NIF instances. It will also allow for gathering information about the utilization and performance of virtualized infrastructure resources. This includes monitoring the availability, capacity, and performance metrics of the allocated resources, providing visibility into resource usage and potential bottlenecks. In case of the need for infrastructure policy enforcement, this interface allows the NIO to enforce policies and constraints on the virtualized infrastructure resources such as security policies, learning and QoE/QoS requirements, or specific compliance regulations that

need to be applied to the infrastructure hosting the NIFs (e.g., data privacy, data anonymity, model isolation/federation, etc.).

- ***Nifm-Nifnis.*** This interface enables the NIF Manager to manage the lifecycle of NIF instances. It allows the NIF manager to perform operations such as NIF instantiation, scaling, healing, termination, and update. In the case of configuration and monitoring, this interface allows the NIF Manager to provide configuration parameters and policies to the NIF through the interface. Additionally, it can collect monitoring data and performance metrics from the NIF instances to ensure their proper functioning and adherence to Service-Level Agreements (SLAs) in terms of both networking (e.g., QoS and QoE) and learning (e.g., accuracy). This NIF Manager can also perform fault and performance management. The NIF Manager receives fault notifications and performance data from the NIFs through the interface, allowing it to detect and handle any issues that may arise based on policies defined by the NIO. This includes fault localization, fault resolution, performance optimization, and ensuring the desired performance of the NIF. Finally, the NIF Manager can manage the state and context of the NIF instances. It allows the NIF Manager to retrieve and update the state information of the NIF Manager, including their operational status, configuration parameters, and runtime data. This information is crucial for maintaining the consistency and continuity of the NIF operations.

  This interface can also provide the capabilities to monitor, manage and orchestrate NIS based on abstract data information such as model knowledge (e.g., Neural Network weights, expert knowledge encapsulated in rule-based systems) and explainable model data. Moreover, it will gather information about the NIF composition in NIS to detect possible conflicts in NIS before deployment, given its topological structure, or after re-orchestration of the NIS when NIF are added/removed/changed. Through this interface, the NIO can also configure the NIS (e.g., adding a new NIF in the NIS). In some implementations, the interaction between NIO and NIS can be done via a specific interface, e.g., a ***Nio-Nis interface.***
- ***Nifcm-Nivi.*** This interface allows the NIFs to interact with the NI virtualized infrastructure, which includes virtual machines, containers, storage resources, and networking components. This interface allows NIFs to utilize the underlying infrastructure to perform their designated functions efficiently. For example, allowing an ML model to switch among different computing hardware (e.g., CPU/GPU/TPU/FPGA) and modes (training vs. inference).
- ***Nio-MLp.*** This interface enables the NIO to provide ML model (re-)training features.
- ***MLo-Ncat.*** Via this interface, the ML pipeline framework in the NIP can access the model register, which serves as a critical connection point in managing and organizing ML models empowering NIF/NIS within the pipeline framework. This interface enables seamless integration and coordination between the pipeline framework and the model register, facilitating efficient model versioning, storage, retrieval, and tracking. This interface streamlines the integration of ML models within the pipeline, enabling seamless collaboration, reusability, and scalability of models across the ML workflow.
- ***Nio-Ncat.*** This interface allows the NIO to access the catalog of NIF/NIS available to deploy in the network. By accessing the catalog, the NIO can effectively discover, select, compose, onboard, and manage the lifecycle of NIF/NIS within the NIP. The interface enhances the agility, flexibility, and automation capabilities of the NI orchestration system, enabling seamless deployment and efficient management of NIS/NIF within the NI virtual infrastructure.
- ***Nio-Mano.*** When the MANO is deployed as an external functional block of the NIO, this interface provides the communication mechanism to exchange real-time information to track network slices, function states, and resource availability. This synchronization allows the NIO to dynamically adapt decisions and efficiently allocate resources based on the current network characteristics. By maintaining an up-to-date view of available resources, including computing power, storage, and network capabilities, the NIO can orchestrate network resources effectively and optimize resource utilization, thereby improving performance.
- ***Nio-Ext.*** This interface provides communication between the NIO and external orchestrators/controllers in the network in the same or across multiple domains. This interface enables resource coordination, NIF/NIS orchestration/coordination, policy management, event handling, and information exchange. Firstly, it allows for efficient coordination of resources by exchanging information about available resources and their utilization across different domains. This promotes optimal resource allocation and utilization. Secondly, the interface enables collaboration in NIF/NIS deployments across multiple domains by facilitating the exchange of NIF/NIS-level information and dependencies between the NIO and external orchestrators/controllers. This enables the instantiation, management, and scaling of complex NIS across multi-domain and heterogeneous environments. Additionally, the interface supports policy management by facilitating the exchange of policy information between the NIO and

external orchestrators/controllers. This ensures consistent policy implementation and governance across different domain systems. Moreover, the interface enables the exchange of event and alarm information, allowing for proactive event handling, correlation, and remediation across domains. Finally, the interface facilitates information exchange and federation by enabling the sharing of network topologies, hardware capabilities, NIS/NIF catalogs, and other relevant data (e.g., monitoring information, model weighs, etc.), improving decision-making and coordination capabilities among different orchestration systems.

In order to promote industry deployment, validation, and widespread adoption of standardized APIs, it is recommended that in the future, an OpenAPI[4] representation in YAML[5] and JSON[6] is available (e.g., via ETSI or IEEE). An example of it is NFV-MANO core APIs[7]. Moreover, tools to navigate the specifications and report bugs should also be provided to enhance the usability and effectiveness of the OpenAPI representation.

4.2 External Interfaces

The NIO-Ext interface will allow the NIO to communicate with external orchestrators/controllers to achieve efficient collaboration, resource coordination, and NIF/NIS orchestration across heterogeneous network environments (far edge, edge, RAN, Transport, core, cloud, etc.). The interface enhances interoperability, scalability, and flexibility, allowing for the effective management and orchestration of resources and NIF/NIS in complex network ecosystem. In this section, we will describe two specific cases of such interfaces.

4.2.1 O-RAN

The O-RAN Alliance (Open Radio Access Network Alliance) is a global community of mobile network operators, vendors, and research institutions, established in February 2018. Its primary goal is to drive the development of open, intelligent, and interoperable RAN technologies. Founded by AT&T, Orange, Deutsche Telekom, Docomo, and China Mobile, O-RAN now has the support of over 300 organizations, including major operators and vendors. Analysts predict that open vRANs could surpass the conventional RAN market by 2028,[8] generating revenues close to $20 billion.

The O-RAN architecture is a new approach to building mobile networks that aims to increase flexibility, interoperability, and innovation. It is designed to enable multi-vendor deployments, reduce costs, and improve network performance. Key aspects of the O-RAN architecture are presented in D2.1 [8], Section 10.4. A very important aspect of O-RAN is the integration of AI/ML workflows, i.e., Network Intelligence that may be managed by DAEMON's NIP, with the following principles [33]:

- Offline Learning: In O-RAN, even for reinforcement learning scenarios, some amount of offline learning (where a model is trained with offline data before deployment) is always recommended.
- Pre-training and Testing: Any model deployed within the network needs to be trained and tested beforehand. No completely untrained model should be deployed in the network.
- Modularity in ML Applications: As a best practice, ML applications should be designed in a modular fashion, with the capability to share data without knowledge of each other's data requirements. They should not be bound by the location or nature of a data source.
- Service Provider's Deployment Choice: The criteria for determining where an ML application should be deployed (Non-RT RIC or Near-RT RIC) may vary between service providers. Therefore, it should be the service provider's choice to decide the deployment scenario for a given ML application.
- Optimization of ML Model for Efficiency and Performance: To improve execution efficiency and inference performance, the ML model should be optimized and compiled considering the hardware capabilities of the inference host. There should be a balance between efficiency and inference accuracy, with acceptable accuracy loss as one of the optimization goals. The optimization parameters should be determined based on this threshold.

Figure 12 illustrates the general framework of AI/ML procedures and interfaces and its integration into DAEMON's NIP, including the potential mapping between ML components and O-RAN components.

[4] https://www.openapis.org/

[5] https://yaml.org/

[6] https://www.json.org/json-en.html

[7] https://nfvwiki.etsi.org/index.php?title=API_specifications

[8] ABI Research. 2020. Open RAN. Market Data Report.

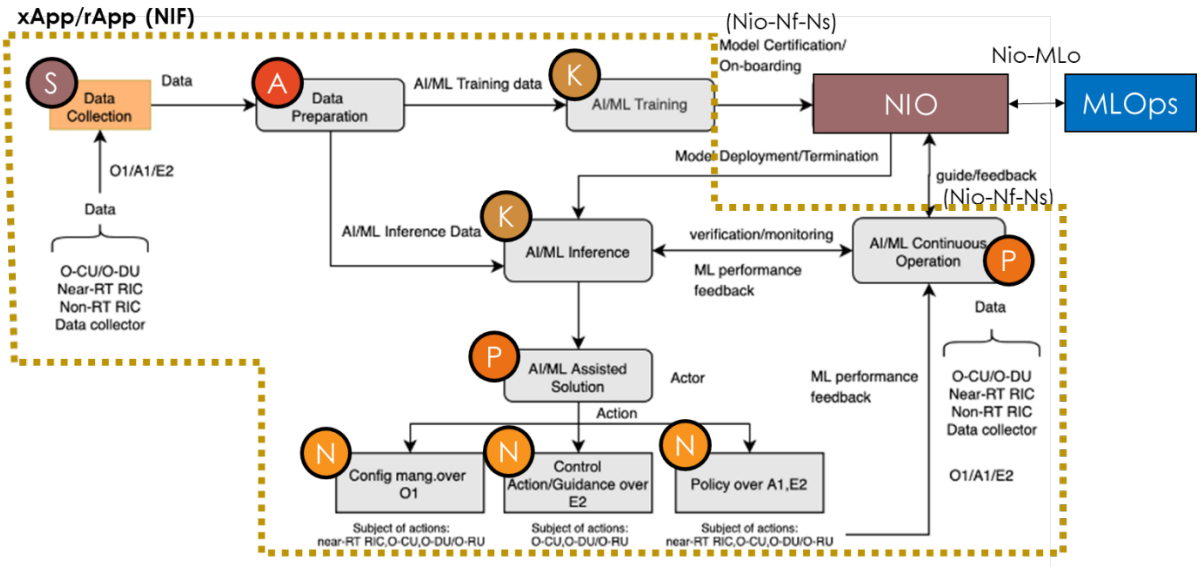


Figure 12. *Integration of DAEMON NIP and O-RAN AI/ML Lifecycle Procedures and Interface Frameworks.*

Relevant to DAEMON's NIP are the AI/ML deployment scenarios of O-RAN, summarized as follows:

- Deployment Scenario 1.1: In this case, AI/ML Continuous Operation, AI/ML Model Management, Data Preparation, AI/ML Training, and AI/ML Inference all take place within the Non-RT RIC (Non-Real-Time Radio Intelligent Controller).
- Deployment Scenario 1.2: Here, AI/ML Continuous Operation, Data Preparation for training, and AI/ML Training are located in non-RT RIC. However, AI/ML Model Management is outside non-RT RIC (either within or outside the SMO, Service Management and Orchestration). Data Collection for inference, Data Preparation for inference, and AI/ML Inference are in the Near-RT RIC (Near-Real-Time Radio Intelligent Controller).
- Deployment Scenario 1.3: AI/ML Continuous Operation and AI/ML Inference are within non-RT RIC. Data Preparation, AI/ML Training, and AI/ML Model Management are located outside the non-RT RIC (either within or outside SMO).
- Deployment Scenario 1.4: In this scenario, the non-RT RIC acts as the ML training host for offline model training, and the Near-RT RIC acts as the ML training host for online learning and also as the ML inference host.
- Deployment Scenario 1.5: Continuous Operation, Model Management, Data Preparation, and ML Training Host are in non-RT RIC. However, the O-CU/O-DU (Open Central Unit/Open Distributed Unit) acts as the ML inference host.

Please note that the deployment of "AI/ML Continuous Operation" outside of non-RT RIC is still under study.

Table 6. *O-RAN AI/ML deployment models.*

Scenario	Data preparation (A)	AI/ML training (K)	AI/ML inference (K)	Model management (NIO/MLOps)	Continuous operation (P)	Subject of action (N)	Action from inference host to subject (N)		Enrichment data for inference (S)
							Configuration management	Policy/ Control	
1.1	Non-RT RIC	Non-RT RIC	Non-RT RIC	Non-RT RIC	Non-RT RIC	Near-RT RIC	O1	A1 (policy)	SMO internal
						O-CU, O-DU, O-RU	O1	N/A	SMO internal
1.2	Non-RT RIC and Near-RT RIC	Non-RT RIC	Near-RT RIC	Out of Non-RT RIC	Non-RT RIC	Near-RT RIC	Near-RT RIC internal	Near-RT RIC internal	A1
						O-CU, O-DU, O-RU	N/A	E2 (control / policy)	E2 (if relevant)

1.3	Out of Non-RT RIC	Out of Non-RT RIC	Non-RT RIC	Out of Non-RT RIC	Non-RT RIC	Near RT-RIC	Near-RT RIC internal	Near-RT RIC internal	A1
						O-CU, O-DU, O-RU	Not defined	Not defined	Not defined
1.4	Non-RT RIC and Near-RT RIC	SMO/Non-RT RIC for offline training or Near RT RIC for online learning	Near RT RIC	Non-RT RIC and Near RT RIC	Near RT RIC	Near RT RIC	Near-RT RIC internal	Near-RT RIC internal	A1
						O-CU, O-DU, O-RU	N/A	E2 (control / policy)	E2 (if relevant)
1.5	Non-RT RIC	Non-RT RIC	O-CU / O-DU	Non-RT RIC	Non-RT RIC	O-CU, O-DU, O-RU	Not defined	Not defined	Not defined

4.2.2 5G Core

The 5G Core (5GC) is one of the most important domains in a 3GPP mobile system, hence it is part of the DAEMON framework.

The imperative of network automation drove the design of the 3GPP system in R15, marking a significant departure from previous releases. In earlier iterations, data generation and analytics in the network primarily relied on proprietary interfaces for exchanges between network elements and their respective managers. However, with R15 and subsequent consolidations, the architecture underwent a comprehensive overhaul to incorporate native support for collecting analytics. These analytics can be effectively utilized, as explained below, to establish feedback loops through standardized or proprietary solutions. At the heart of this system lies the Network Data Analytics Function (NWDAF), which performs three key functions: (i) aggregating data, encompassing metrics that reflect the current state of the network, sourced from other producer network functions (NFs); (ii) conducting analytics, involving the computation of refined statistics based on the gathered data; and (iii) sharing the computed analytics with other consumer functions across the network.

The generated analytic reports serve as outputs that either present statistics based on historical data or provide predictions for specific metrics, depending on whether the requested timeframe is in the past or future, respectively. These outputs play a crucial role in optimizing the operation of NFs. Additionally, the output may include a confidence parameter, ranging from 0 to 100, which conveys information about the reliability of the prediction made. Factors determining this confidence parameter may include the volume of data utilized in generating the prediction, the age of the AI model employed, and other relevant considerations.

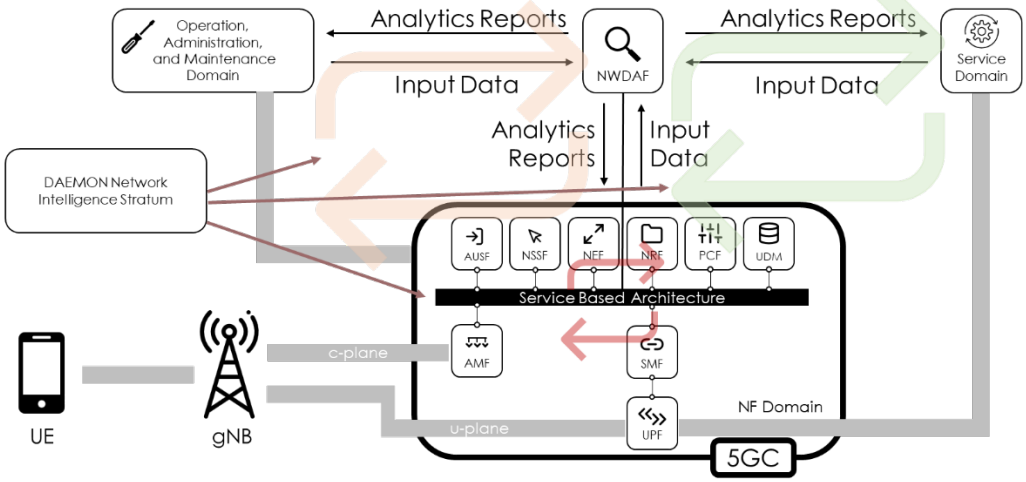


Figure 13. *The architectural framework proposed by the 5GPPP Arch WG* [11].

In the context of this framework, the NIP is depicted in Figure 13 above. The Figure presents the interconnections among various components. The framework is divided into three distinct domains and shows where the DAEMON NIP takes a role.

The first domain, referred to as 5GC, is where the Network Data Analytics Function (NWDAF) resides. Within this domain, other network functions (NFs) of the core act as the primary producers and consumers of

data and analytics. These NFs utilize the data and analytics to drive network operations in a data-driven manner. Thanks to the NWDAF, consumer NFs no longer need to directly communicate with every potential producer to compute analytics, as they can efficiently leverage the shared information. NWDAF is a specific (and very important) NIF, that can leverage on a number of NIF-C according to the analytics that are served.

The second domain encompasses Operations, Administration, and Maintenance (OAM) activities, which involve modules such as Element Managers or Network Elements in pre-5G networks. Starting from R15, OAM effectively enforces network slicing through the service-based management architecture. The OAM domain can also supply the NWDAF with data from the RAN and 5G NFs, such as resource consumption. Unlike the pre-5G 3GPP RAN architecture, which lacks an analytics hub like the NWDAF, alternative architectures like O-RAN feature dedicated analytics modules. The Management Data Analytics Function (MDAF) serves as the module responsible for interacting with the NWDAF and provides Management Data Analytics Services (MDAS). As discussed, the MDAF collaborates with the NWDAF and other core NFs to generate management analytics information, which is subsequently consumed by other NFs or management procedures like the self-organizing network. For the DAEMON NIP, the MDAF is a NIF, that in turn, can be further split into a number of NIF-C which i) interact with the NWDAF, effectively closing the loop with the core and ii) allows the internal interaction within the management domain.

The third domain encompasses the service domain, facilitated through the Application Function (AF). These functions, residing outside the 3GPP trust domain, play a crucial role in facilitating close interaction between service providers and network operators. This interaction is achieved through enriched service layers, which aid in commoditizing the network and enhancing the interplay between the service and network intelligence. Given the criticality of authorization and security, it becomes essential to verify whether AFs are appropriately authorized to interact with the NWDAF and engage in data exchange with third parties. Authentication can be managed in three different ways, such as basic user-password authentication, where the configuration of credentials is done via a configuration file. Support of Transport Layer Security (TLS) protocol where there is a server-side authentication or mutual TLS (mTLS) authentication, where both server-side and client-side authentication is required. In this case, the AF can be seen as a specific NIF-C (either Sink or Source, depending on the context). Overall, any NF deployed within the 5GC, the OAM system, or any AF can contribute input to the NWDAF and request analytic reports from it. This establishes a feedback loop where any NF, OAM component or AF can provide input data to the NWDAF and receive analytic reports generated from the collective data obtained by the NWDAF. Through these feedback loops, the majority of automated network operations can be executed, as exemplified by the ones already provided by the NWDAF in the standard and by the solutions proposed by DAEMON in WP3 and WP4.

5 NI Orchestration procedures

In this section, we show how the architectural building blocks described in Section 3.2 will interact with the blocks external to the Network Intelligence Orchestrator (NIO) and with the blocks inside the NIO. We will explore how the different components work together to create a cohesive system that can effectively orchestrate intelligence across multiple domains. Through these interactions, the NIO will be able to address the challenges that can emerge when NISs are deployed across different network domains and operating in multiple timescales, as described in Section 3.1.

Notice that all the procedures mentioned below are depicted using a process view. This view answers how the system behaves addressing concurrency and synchronization aspects. Unified Modeling Language sequence diagrams[9] were selected as the most appropriate form. Next, we briefly describe how the combination of some functional blocks can help to address the challenges described in the previous section.

5.1 Inter NIO Procedures

One of the most essential management and orchestration capabilities is to handle the lifecycle of each of its entities. The NIO is not an exception. Regarding networking functionalities, NFV MANO [31] is the referent architectural framework to look up to. In general, lifecycle management is responsible for the following operations: creation, instantiation or deployment, management (e.g., model selection and optimization), and termination. However, given the intelligent nature of the functionalities proposed by DAEMON, there are several factors that must be considered while addressing their lifecycle management. In the following subsections, we will discuss the procedures required to perform lifecycle management with the NIO in detail.

5.1.1 Creation

When creating a new NIS, the Network Intelligence Orchestrator (NIO) should verify that all the NIFs from that NIS are available in the catalog. If a NIF is unavailable, new NIF training should be started, e.g., based on user-defined NIF/ NIS Descriptor (NIFD or NISD) as described in the next paragraph. This training is represented by triggering a new MLOps pipeline. The data ingestion for training this new NIF should be coordinated between the NIO and the MLOps pipeline. Notice that this procedure only contemplates the creation of the NIF and not its usage.

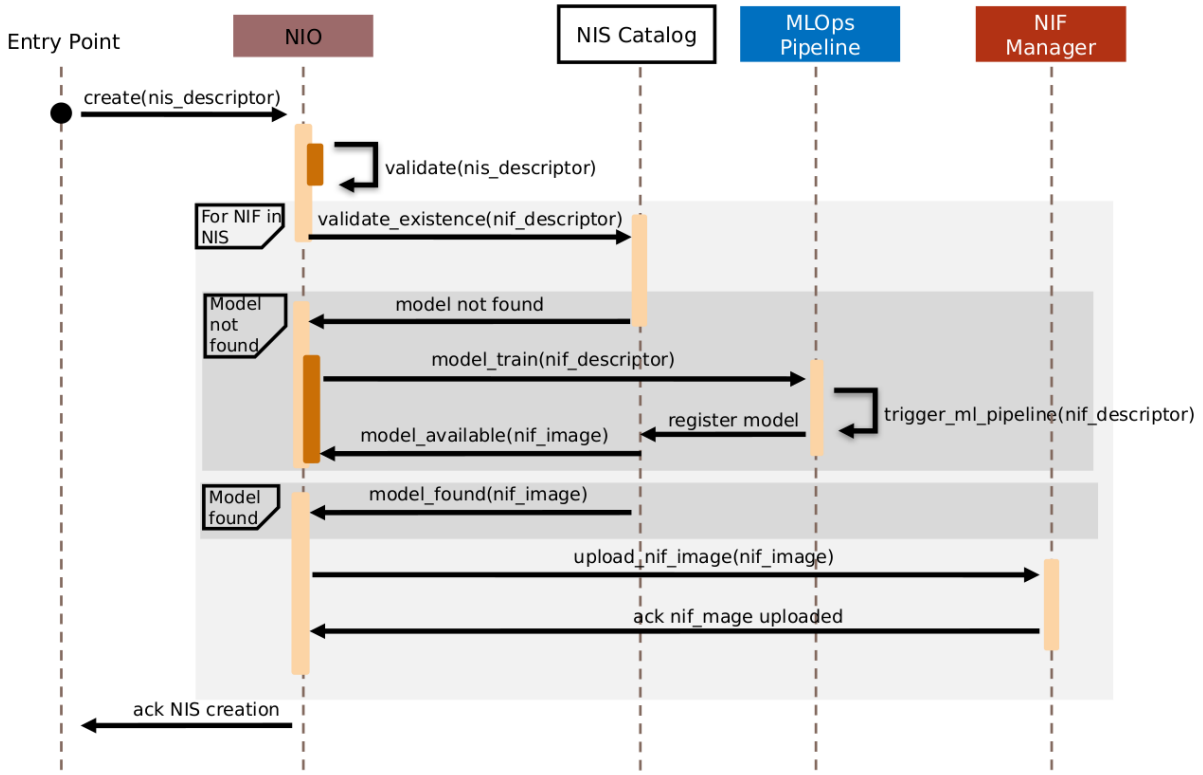


Figure 14. *NIS creation process flow.*

[9] https://developer.ibm.com/articles/the-sequence-diagram/

Figure 14 shows the required interactions to create a NIS/NIF. The main steps for NIS/NIF creation are:

- Through its API, the NIO should process a NIS/NIF creation request. A sender can submit this request, which could be a human, an AI, or another process with administration rights to trigger orchestration operations in the NIO. The sender identifies that a new NIS/NIF is needed to perform a given network operation and submits this request to the NIO. As input for this process, the NIO should receive a NIF/ NIS Descriptor (NIFD or NISD) which includes, but is not limited to:
    - Learning mode, if the ML model supports online learning or if the training is made offline.
    - Data on which the model is trained (whether the learning is online or offline). This field also specifies the format in which the input data is expected.
    - Learning metrics. This typically includes accuracy, cross-entropy, or a known loss function, e.g., Mean Squared Error (MSE).
    - Model performance upper and lower thresholds. Values on which the training can be concluded (upper threshold). It is assumed that once the upper threshold is met, the ML model is ready to be deployed in production. On the contrary, if the lower threshold is met, the ML model deployed in production should be updated. The definition of these thresholds could be different for different NI functionalities and should reflect a good performance.
    - Output format. This field specifies in which format the ML will communicate its output. For instance, a classification problem can produce a vector with the probability of a given sample belonging to a class or the class itself.
    - Last modification time. This field will indicate the age of the ML model. Given the constant evolution of network state and data, having an up-to-date ML model is crucial for network operation.
    - Dependencies required for operation. ML models are created using specific libraries (e.g., NumPy, pandas, etc.). The right versions of such libraries must be available when instantiating the ML model in production.
- NIO processes the NIFD/NISD, including but not limited to:
    - Checking for the existence of mandatory elements (network operation, data requirements, output format, accuracy).
    - Validating integrity and authenticity of the NIFD/NISD.
- For every NIF in the NIS, NIO verifies if the NIF model exists in the catalog.
    - If the NIF model is not present in the catalog, NIO triggers a train operation from ML pipelines.
        - ML pipeline model triggers a new pipeline (Data ingestion - Model training - Model Testing - Model packaging - Model registering) using the NIFD.
        - Once the NIF model is trained, the model is registered in the NIF catalog.
    - If the model is trained, it should be registered in the catalog and can be used in inference.
- NIO makes NIS/NIF images available to each applicable NIF Component Manager (NIF-C Manager).
- The NIF-C Manager acknowledges the successful uploading of the image.
- Finally, NIO acknowledges the NIS/NIF creation to the sender.

5.1.2 Instantiation or Deployment

Figure 15 shows the interactions required for instantiating or deploying a NIS/NIF. As in the previous step, NIO receives a request to instantiate a new NIS/NIF. Then, several variants might be possible:

- None of the NIFs belonging to the NIS is instantiated or deployed. Then, the NIS instantiation will include the instantiation of all the needed NIF instances.
- All the needed NIF instances have already been created. In this case, NIS instantiation would only deal with the interconnection of the corresponding NIF instances.
- A combination of the above where some NIF instances might exist, some might need to be created, and instantiated, and some network connectivity between the NIFs may already exist.

It is important to notice that if a NIF instance is already created, it can be shared between different NIS. In this case, the NIO should trigger the conflict resolution mechanism, because they may be deployed on the same node and/or accessing the same resources. If no conflict is produced, the same NIF can be

used to instantiate the current NIS. If a potential conflict is detected, the NIO should proactively address it by deploying specific policies implementing rules or priorities (c.f. Section 5.2.1) to effectively solve the aforementioned conflict.

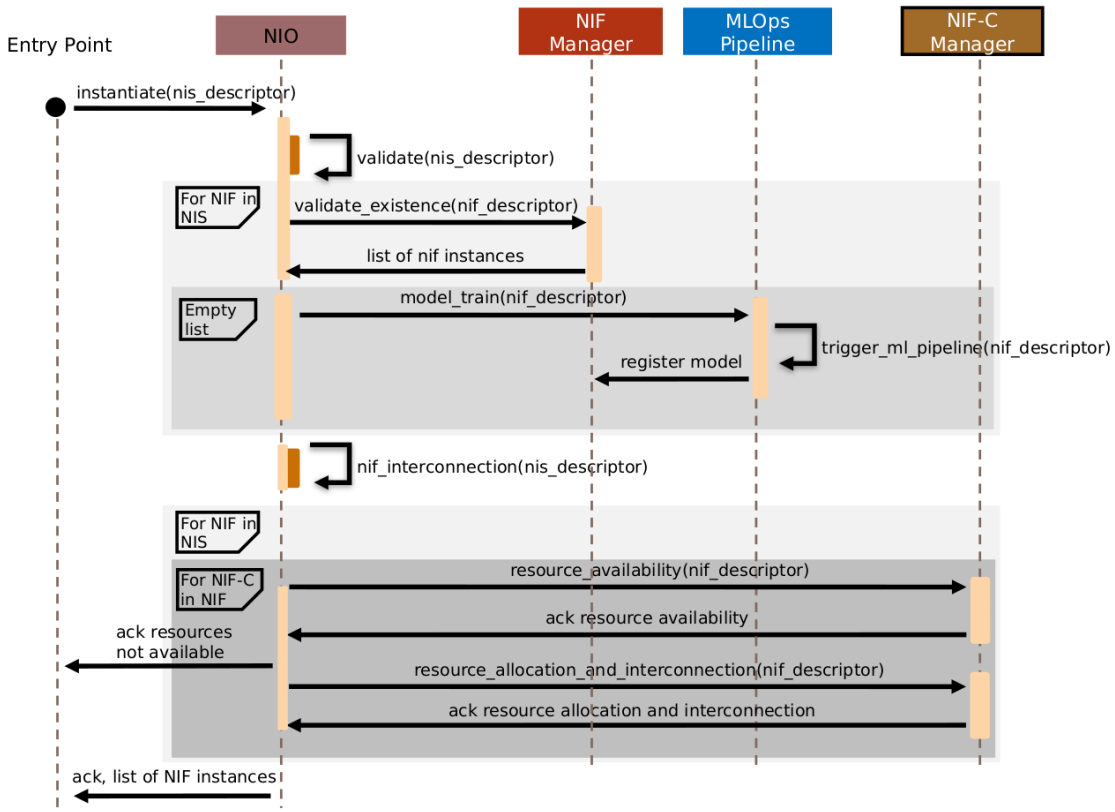


Figure 15. *NIS instantiation process flow.*

The main steps for NIS/NIF instantiation are:

- NIO receives a request to instantiate a new NIS/NIF.
- NIO validates the request, both validity of the request (including validating that the sender is authorized to issue this request) and validation of the parameters passed for technical correctness and policy conformance.
- For each NIF instance needed in the NIS, the NIO checks with the NIF Manager if an instance matching the requirements already exists. If such a NIF instance exists, it will be used as part of the NIS. If the NIF instance does not exist, the NIO triggers the Create NIF procedure.
- NIO runs a feasibility check of the NIF interconnection setup.
    - NIO requests to the NIF-C Manager the availability of resources needed for the NIF Interconnection and reservation of those resources.
    - The NIF-C Manager checks the availability of resources needed for the NIF Interconnection and reserves them.
    - The NIF-C Manager returns the result of the reservation back to NIO.
    - If the resources are not available, the NIS might not be instantiated. The result is given back to the Sender.
- Once the list of NIF instances to be provisioned is known, NIO requests the NIF-C Manager to allocate and interconnect the NIF instances.
    - The NIF-C Manager instantiates the connectivity network needed for the NIS.
    - The NIF-C Manager acknowledges completion.
- Finally, the NIO acknowledges the completion of the NIS instantiation.

5.1.3 Management

Several operations can be considered as management procedures, such as NIS/NIF update, optimization, scaling or migrating. NI solutions stored in the NIS/NIF catalog are inherently trained on hardware and software platforms that may not match the ones available in the new environment where

they need to be deployed. In such cases, the NIS creation/selection, optimization, and instantiation block will obtain networking and execution context information from its MANO block operating in the network and select the proper model to be used in inference within a NIF. Suppose a mismatch between trained and targeted hardware/software appears. In that case, the same block should perform the optimization/adaptation (e.g., compression of a neural network, change of inference library from GPU to CPU) to match the new environment. In case no model is available for the specific execution environment, the NIS creation/selection, optimization, and instantiation block will create a new NIS and then notify the NIS workflow configuration block to trigger a new training phase. Here we depict the NIS/NIF update with model selection as the most relevant and generic procedure that may involve optimization, re-train or selection.

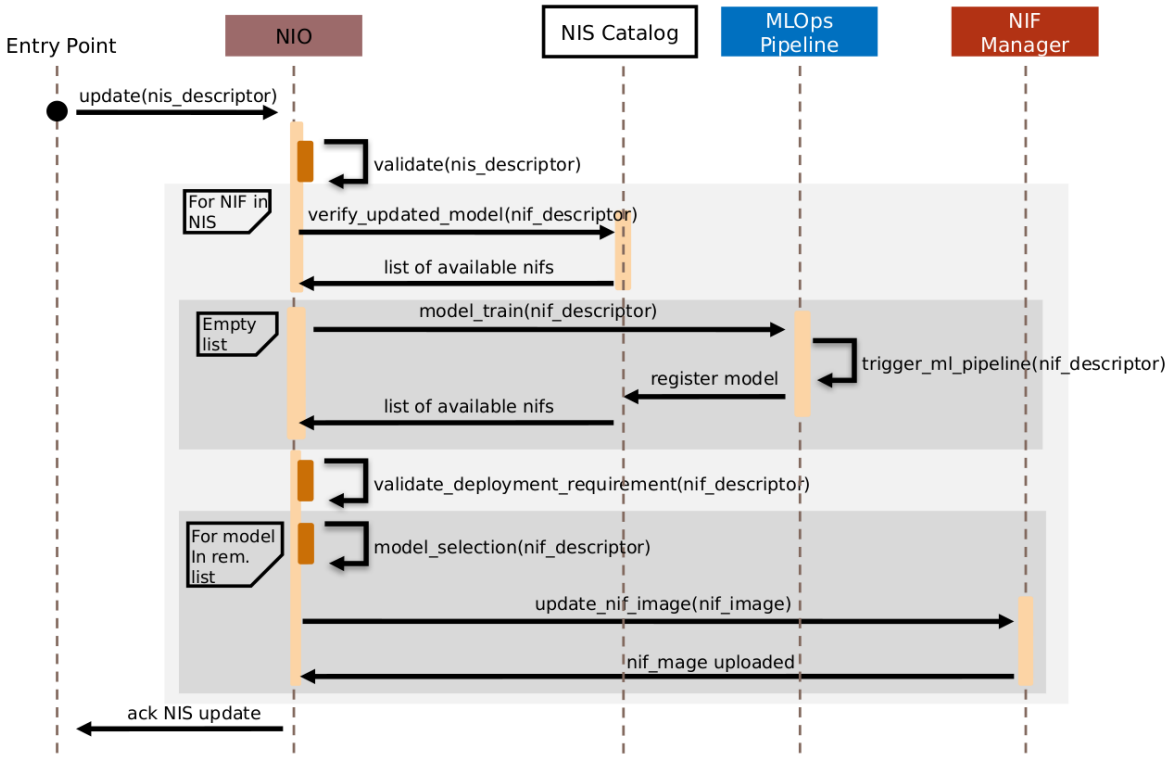


Figure 16. *NIS update process flow.*

Figure 16 shows the main steps for NIS/NIF updates. This procedure includes updating the parameters of the NIS/NIF. It is important to notice that the update process has similarities with the NIS/NIF creation:

- A request for NIS/NIF update is submitted from the sender, which could be a human, an AI, another process in the architecture (e.g., Data Analytics detecting a mismatch between the statistics of the input data), or the monitoring block from the NIO (e.g., current accuracy is lower than expected, the model is old, etc.). The sender identifies that a new NIS/NIF needs to be updated and submits its request to the NIO through the NIO API.
- NIO processes the NIFD/NISD, including, but not limited to:
    - Checking for the existence of mandatory elements (network operation, data requirements, output format, accuracy).
    - Validating the integrity and authenticity of the descriptor.
- For every NIF in the NIS, NIO verifies that an updated NIF model exists in the catalog.
    - If an updated model is needed but not in the catalog, NIO triggers a re-train operation from ML pipelines.
        - ML pipelines model triggers a new pipeline (Data ingestion - Model training - Model Testing - Model packaging - Model registering).
        - Once the NIF model is re-trained, the model can be registered in the NIF catalog. Update NIFD with new version and requirements (data format, hardware, software dependencies, etc.)

- If a model (or more than one model) is available, then the NIO verifies that the available models satisfy the deployment requirements in terms of data (e.g., input rate and format), computation runtime (e.g., CPU, GPU, or FPGA), dependencies (e.g., TensorFlow, PyTorch), and performance level. This process might return an empty list, meaning that there is no model that satisfies the deployment requirement and the creation of a new NIF is needed.
- In case the filtered list is not empty, and more than one model satisfies the deployment requirements, model selection should be carried out. In this phase, the component will compute an ML test score, and depending on arbitration policies, the best-performing model is selected to update the NIF image. The ML test score can contain learning-related metrics (e.g., loss/reward function) and non-learning-related metrics (e.g., QoE, QoS, stability in deployment, etc.). The arbitration policies are decision factors that the NIO considers primordial for model deployment, for instance, if model precision is preferred over energy consumption.
- If the model is updated, it should be registered in the catalog and can be used in inference.

- NIO makes NIS/NIF images available to each applicable NIF-C Manager.
- NIF-C Manager acknowledges the successful uploading of the image.
- NIO acknowledges the NIS/NIF update to the sender.

Other management operations include optimization, scaling in/out or migrating. The workflows are similar to those of NFV MANO, requiring an extra step to update the NIS/NIF, which was shown above.

- NIO receives a Manage NIS/NIF request. This could come even from the same NIO (e.g., forecasting model -in Data Analytics module- to scale because it expects an increase of demand, or migration of a NIS because the associated requester is moving).
- NIO validates this request, identifying if the management operation requires updating the NIS/NIF. If an update is needed (e.g., NIS/NIF optimization), NIO triggers the Update NIS/NIF procedure.
- NIO runs a feasibility check of the NIF interconnection setup.
  - NIO requests the NIF-C Manager the availability of resources needed for the NIF interconnection and reservation of those resources.
  - NIF-C Manager checks the availability of resources needed for the NIF Interconnection and reserves them.
  - NIF-C Manager returns the result of the reservation back to NIO.
  - If resources are not available, the NIS might not be managed. The result is given back to the Sender.
- Once the list of NIF instances to be provisioned is known, NIO requests the NIF-C Manager to allocate and interconnect the NIF instances. This can be done through:
  - Triggering the creation of a new NIS/NIF instance for scaling out the operation.
  - Triggering the deletion of a NIS/NIF for scaling in operation.
  - Triggering both the creation and the deletion of the NIS/NIF to migrate it.
- NIO acknowledges the completion of the NIS/NIF management.

5.1.4 Termination

The NIO receives a request to terminate a NIS/NIF instance. This request might come from a human, an AI process, or another process in the architecture. When terminating a NIS/NIF instance, several variants might be possible:

- All affected NIF instances contributing to the NIS that need to be terminated and were created when initiating the NIS. In this case, all these NIF instances need to be terminated, and the interconnectivity between these NIF instances must be removed.
- Some NIF instances are contributing to other NIS instances. In this case, only those NIF instances that do not contribute to other NIS instances must be terminated. The interconnectivity between them must be removed, leaving the other NIF instances in place and the interconnectivity between them intact.

The main steps for the termination of a NIS/NIF instance are:

- NIO receives a request to terminate a NIS/NIF instance using the NIS/NIF Lifecycle Management interface.
- NIO validates the request. It verifies the validity of the request (including the sender's authorization) and verifies that the NIS/NIF instance exists.
- NIO requests NIF Manager to terminate any NIF instances that were instantiated along with the NIS instantiation, provided they are not used by another NIS. This is done by calling the 1.3 Delete NIS/NIF (pending) request.
    - NIF Manager terminates the required NIF and sends a confirmation to the NIO that the NIFs are terminated.
- NIO requests the deletion (release) of resources for this NIS instance to the NIF-C Manager.
    - NIF-C Manager deletes (releases) the resources for this NIS instance.
    - NIF-C Manager acknowledges the completion of resource deletion back to NIO.
- NIO acknowledges the completion of the NIS instance termination.

5.1.5 Other operations

In addition to the operations presented above, operations such as deleting, querying, enabling, or disabling a NIS/NIF are also considered within the architecture defined by DAEMON. However, such operations are not different than those proposed in NFV MANO as they do not involve or require any interaction with blocks that are related to Network Intelligence (NI) and the MANO block can perform it. The implementation of such procedures is shown in [31].

5.2 Intra NIO Procedures

As introduced above, a NIS is usually composed of different NIFs and hence, some of the NIS management functionalities take place only within the NIO itself. These Intra NIO functionalities address the challenges that may emerge when NISs are deployed across different network domains and operating in multiple timescales, including conflict resolution and knowledge sharing among NIS.

5.2.1 Conflict Resolution

We introduced two specific conflict cases in Section 3.1.1: (i) when conflicts emerge when monitoring data, e.g., algorithms may need data from the same source but with different granularity, and (ii) when conflicts in the policy enforcement of different NI algorithms may act on the same network functions but configuring different values for the target parameters. In such situations, the policy interpreter and configuration block will gather information about the policy guiding the different NIS and pass their interpretation to the conflict detection and resolution module. In both cases, a conflict will be detected, and the NIO will identify and apply the conflict resolution rules associated with (i) multi-timescale coordination and (ii) parameter constraints and execution priority. After applying the rules, the outcome should provide a plan that will trigger a configuration modification of the NIS policies. In the case of NIS empowered by black-box ML algorithms, the Model Explainability block will interpret policies associated with such algorithms.

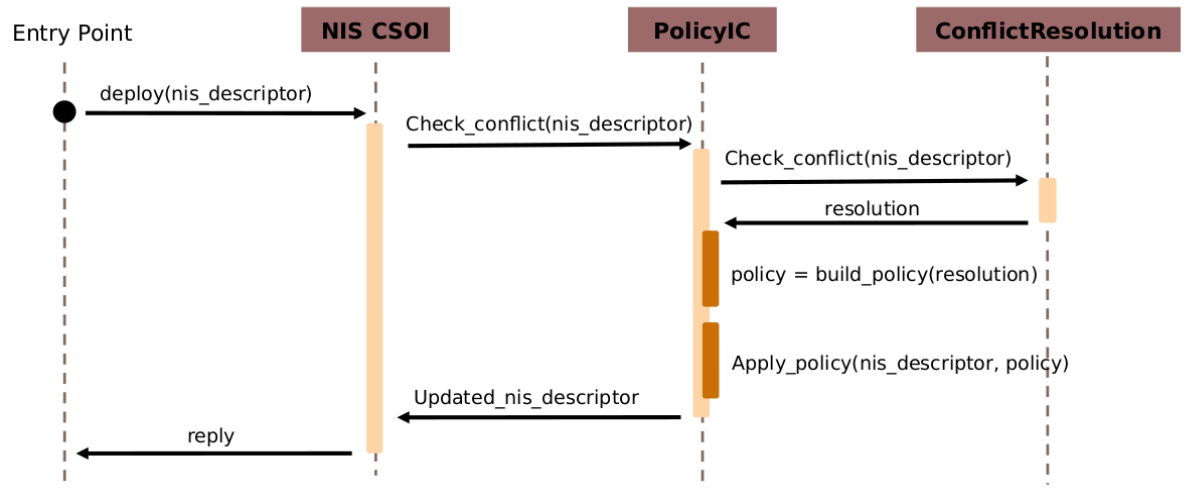


Figure 17. *NIS Conflict Resolution process flow.*

Figure 17 shows the main steps for the case of NIS Conflict Resolution. This procedure includes checking the parameters of the NIS against the Policy Interpreter and Configuration (PolicyIC) to arbitrate the

deployment of the NIS (e.g., if the NIS has different monitoring granularity in a shared source with other NIS, or requires controlling a network function that another NIS is already controlling with a different AI algorithm):

- A request for deploying a NIS is submitted from the sender (it could also be a NIS update), which could be a human, an AI, or another process in the architecture. The sender identifies that a new NIS needs to be deployed and submits its request to the NIO through the NIO API. It will be the NIS Creation Selection Optimization and Instantiation (CSOI) component that will receive the request and will validate the NIS internally, indicating if there is a conflict and updating and resolving in case any exists. Please note the validation command executed in the NIO for the creation, instantiation, and update processes described in Sections 5.1.1, 5.1.2, and 5.1.3 respectively, includes this procedure internally.
- NIO CSOI processes the NISD, including, but not limited to:
    - Checking for the existence of mandatory elements (network operation, data requirements, output format, accuracy).
    - Validating the integrity and authenticity of the descriptor.
- If the NIS request is correct and sound, the NIO CSOI verifies through the PolicyIC if there is any conflict by gathering information about the policy guiding the different NIS and passing their interpretation to the Conflict Resolution module.
- The Conflict Resolution component checks if the NIS to be deployed has any conflict with the existing NIS.
- The Conflict Resolution component globally solves trade-offs that may emerge from conflicting objectives in the control and user planes, e.g., in establishing policies (at small timescales) versus enforcing such policies (at large timescales). For the case of this NIS:
    - The Conflict Resolution component compares policies among different NIS to detect conflicts that may appear with the new/updated NIS.
    - It performs conflict resolution based on comparison and resolution rules, providing a NIS configuration. This configuration will result from a trade-off or priority mechanism that the Conflict Resolution component will execute to harmonize the NIS's coexistence. The resolution will contain the last valid configuration if no feasible solution exists.
- Once the PolicyIC receives the resolution, the new policy is built and applied to the specific NIS.
- The PolicyIC returns the NIS descriptor to the NIO CSOI. Consequently, the NIO CSOI further proceeds with the required NIS operation (i.e., creation, deployment, update, etc.).
- Eventually, the NIO acknowledges the NIS deployment to the sender.

5.2.2 Knowledge Sharing

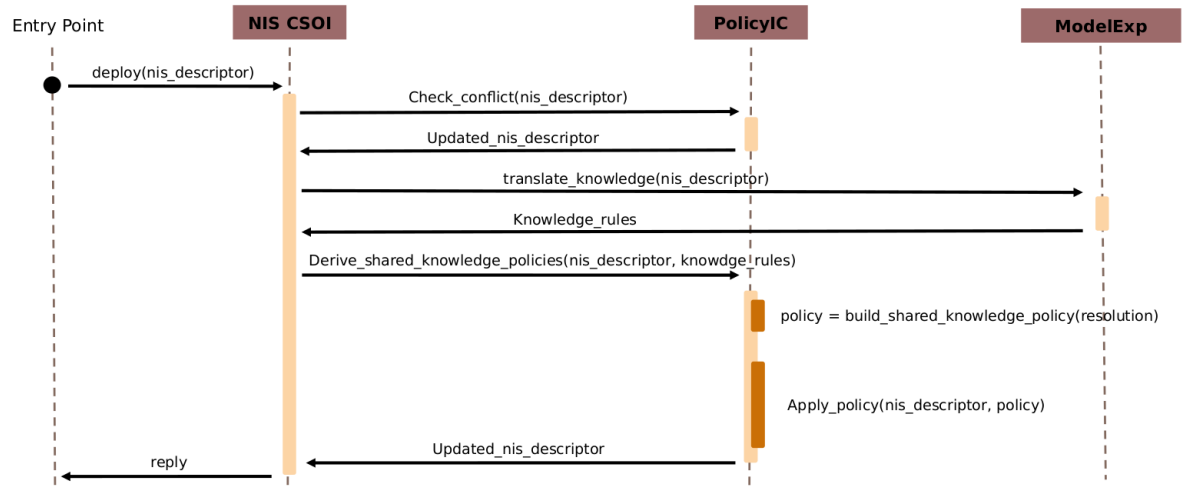


Figure 18. *NIS Knowledge Sharing process flow.*

NISs deployed in the same or across different domains use their knowledge to derive their execution plans. The knowledge management block will allow the NIO to understand the knowledge of each NISs, via the interaction with the Model Explainability block and derive new policies that represent the shared knowledge among NISs, by interacting with the PolicyIC block.

Figure 18 shows the main steps for the case of NIS Knowledge Sharing. This procedure includes checking the parameters of the NIS against the PolicyIC initially (and consequently also with the Conflict Resolution component internally). However, for the cases in which a NIS requires the use of knowledge coming from an external domain, the NIS CSOI will first translate such knowledge in the Model Explainability block before building and applying the shared knowledge policies:

- A request for deploying a NIS is submitted from the sender (it could also be a NIS update), which could be a human, an AI, or another process in the architecture. The sender identifies that a new NIS needs to be deployed and submits its request to the NIO through the NIO API. It will be the NIO CSOI component that will receive the request and proceed with the Conflict Resolution and Knowledge Sharing phases. Please note the validation command executed in the NIO for the creation, instantiation, and update processes described in Sections 5.1.1, 5.1.2, and 5.1.3 respectively includes this procedure internally.
- NIO CSOI processes the NISD, including, but not limited to:
    - Checking for the existence of mandatory elements (network operation, data requirements, output format, accuracy).
    - Validating the integrity and authenticity of the descriptor.
- If the NIS request is correct and sound, the NIO CSOI verifies against the PolicyIC if there is any conflict. As previously described in Section 5.2.1, the PolicyIC itself internally requests the Conflict Resolution component to check if the NIS has any conflict with the existing NIS.
- Once the PolicyIC receives the resolution, the new NIS domain-specific policies are built, and applied.
- With the updated NIS descriptor, the NIS CSOI requests the translation of the external domain knowledge to the Model Explainability block. As a result, the NIS CSOI receives the additional Knowledge rules.
- The NIS CSOI sends the NIS descriptor again to the PolicyIC but this time together with Knowledge rules in order to build and apply the shared knowledge policies:
    - The PolicyIC block builds the shared knowledge policies taking in account possible existing conflicts.
    - The PolicyIC block applies the shared knowledge policies and returns the NIS descriptor to the NIS CSOI.
- Eventually, the NIO acknowledges the NIS deployment to the sender.

5.2.3 Intra NIO Instantiation and deployment

The previously described intra NIO Conflict Resolution and Knowledge Sharing mechanisms are inherent to the NIS CSOI and Life Cycle Management components interacting with external components such as the NIS Catalog, the NIF Manager, or the NIF-C Manager. In order to illustrate how the external processes would occur inside the NIO, Figure 19 details the deployment interactions between the NIS CSOI with both the internal and external components.

As shown in Figure 19, the procedure includes the steps required to validate the NIS descriptor and identify and solve possible conflicts before deployment. Also, domain-specific policies are built and applied, followed by training new models in case there are no instances of them already in the catalog. Finally, the NIS Workflow Configuration block combines them to build the NIS and starts the instantiation and deployment. The detailed sequence of steps is described below:

- A request for deploying a NIS is submitted from the sender (it could also be a NIS update), which could be a human, an AI, or another process in the architecture. The sender identifies that a new NIS needs to be deployed and submits its request to the NIO through the NIO API. It will be the NIO CSOI component that will receive the request.
- NIO CSOI processes the NISD, including, but not limited to:

- Checking for mandatory elements (network operation, data requirements, output format, accuracy).
- Validating the integrity and authenticity of the descriptor.

- If the NIS request is correct and sound, the NIO CSOI will proceed with the validation of the NIS. Please note the validation command executed in the NIO for the creation, instantiation and update processes described in Sections 5.1.1, 5.1.2, and 5.1.3 respectively includes the following procedures of Conflict Resolution and Knowledge Sharing.
- First, the NIO CSOI verifies against the PolicyIC if there is any conflict. As described in Section 5.2.1, the PolicyIC internally requests the Conflict Resolution component to further check if the NIS has any conflict with the existing NIS.
- Once the PolicyIC receives the resolution, the new NIS domain-specific policies are built, applied, and an updated NIS descriptor is returned to the NIO CSOI.
- With the updated NIS descriptor, the NIS CSOI requests the translation of the external domain knowledge to the Model Explainability block. As a result, the NIS CSOI receives the additional Knowledge rules.
- The NIS CSOI sends the NIS descriptor again to the PolicyIC but this time together with Knowledge rules. Hence, shared knowledge policies are built and applied as described in Section 5.2.2.
- The NIS CSOI now iterates for every NIF in the NIS, and checks if an instance of the given NIF already exists in the NIF Manager, as described in more detail in Section 5.1.3. If no instance exists, a new model will be trained for that NIF in the MLOps Pipeline.
- Next, the NIS CSOI proceeds with the interconnection of all NIFs in the NIS (*nif_interconnection* as described in Section 5.1.3). This mechanism involves requesting the NIS Workflow Configuration block (NIS WConf) to virtually link the NIFs and define their interactions.
- Finally, the NIS CSOI starts the instantiation of every NIF in the NIS in the NIF Component. As previously described in Section 5.1.3, this involves:
  - Check the resource availability for that NIF in the NIF Component.
  - If resources are available, allocate the resources for that NIF and interconnect with the other NIFs instances through the NIF Component.
  - If resources are not available, notify accordingly to the entry point.
- Eventually, the NIO acknowledges the NIS deployment to the sender.

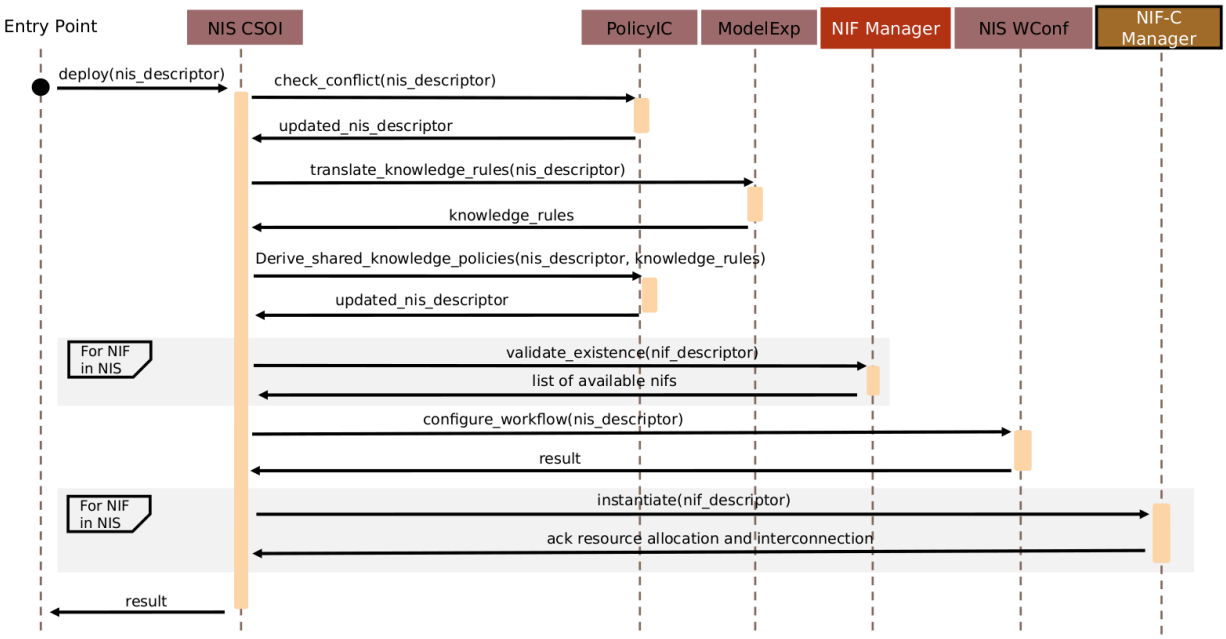


Figure 19. *Intra NIO NIS instantiation and deployment process flow.*

The following table summarizes the functionalities proposed in Section 3.2.1 for the NIO and which procedures are using them.

Table 7. *Summary of the procedures proposed to address the challenges described in Section 3.1 and the functionalities of the NIO that can be used to achieve it.*

Procedure	Procedure type	Functional blocks
Creation	Inter NIO Procedures	NIO, NIS Catalog, ML Pipelines, NIF Manager
Instantiation or Deployment	Inter NIO Procedures	NIO, ML Pipelines, NIF Manager, NIF-C Manager
Management	Inter NIO Procedures	NIO, NIS Catalog, ML Pipelines, NIF Manager, MANO
Termination.	Inter NIO Procedures	NIO, NIF Manager, NIF-C Manager
Other operations	Inter NIO Procedures	As in NFV-MANO
Conflict Resolution	Intra NIO Procedures	Policy Interpreter and Configuration, NIS Creation Selection Optimization and Instantiation, Conflict Detection and Resolution
Knowledge Sharing	Intra NIO Procedures	Policy Interpreter and Configuration, Explainability, Knowledge management
Intra NIO Instantiation and deployment	Intra NIO Procedures	Policy Interpreter and Configuration, Explainability, ML Pipelines, NIF Manager, NIF-C Manager

Notice that the procedures described in the previous sections are based on the challenges described in Section 3.1. However, further procedures can be defined based on other use cases, e.g., orchestration of NI in federated domains or intelligent orchestration of NI, where the decisions of the NIO are empowered by AI-based decision-making models). We expect that these procedures can be further extended to more complex cases or used as a reference to define new ones.

5.3 Reference Implementations

The DAEMON project has developed two reference implementations as its architecture's Proof of Concepts (PoC). In the following sub-sections, we will describe them.

5.3.1 DAEMON Orchestration of NIFs to build a NIS

One of the key features of the DAEMON NIP is to allow the creation, management, and deployment of NISs. This first proof of concept combines two network intelligence functions (NIF) to create a Network Intelligence Service (NS). The first NIF utilizes a federated learning algorithm, enabling anomaly detection at the edge (c.f. Section 5.1 of D4.1[6]). This means that the AI model for detecting anomalies is trained locally on individual devices, preserving data privacy while still benefiting from a collaborative learning process. The second NIF employs a service reallocation algorithm that leverages monitoring information from the edge [21]. This algorithm, based on a multi-criteria decision-making algorithm, dynamically reallocates services based on real-time data, ensuring optimal resource utilization and performance.

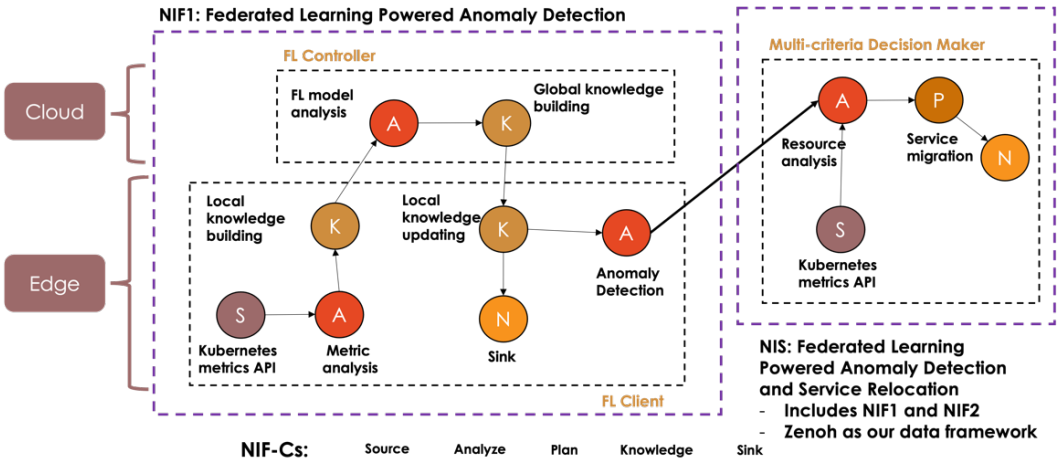


Figure 20. *A federated learning powered anomaly detection and service relocation NIS.*

Following the DAEMON's framework based on the N-MAPE-K to define NIS, we effectively combined the federated learning-powered anomaly detection with service relocation capabilities to realize Network Intelligence Service (NS). Figure 20 shows the N-MAPE-K based diagram of the resulting NIS. By integrating the federated learning approach, the NS ensures that the detection of anomalies is performed securely and efficiently across the network's edge devices. The NS also leverages the monitoring information collected from the edge to make informed decisions about service reallocation, maximizing the network's overall performance and responsiveness.

The proof of concept relies on the Eclipse Zenoh data communication framework, which provides a reliable and scalable solution for exchanging data between devices and components within the network. Additionally, the implementation utilizes Kubernetes functionalities to realize the NIF component manager and NIF manager. Kubernetes helps manage the deployment, scaling, and orchestration of the NIF components, ensuring smooth operation and efficient resource allocation. The Smart HighWay[10] testbed located on top of the E313 highway in Belgium served as the edge environment for testing and validating the effectiveness of the proposed NIS, providing a real-world scenario to assess its performance and potential benefits. Figure 21 shows a high-level representation of the deployment and how the components were deployed at the cloud[11] (centralized server) and edge (road-side units at the Smart Highway).

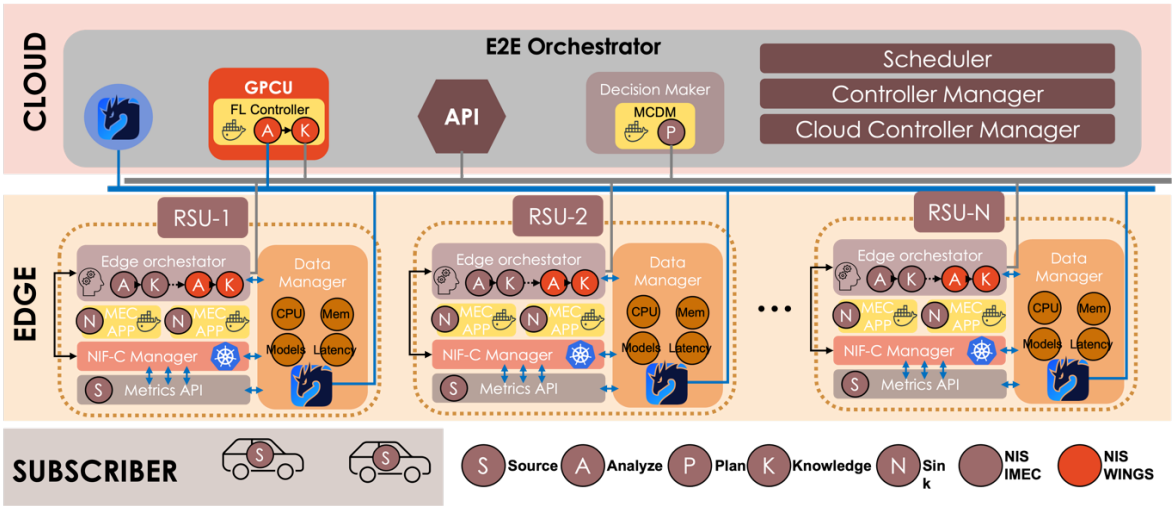


Figure 21. *Cloud-to-edge deployment of the proposed NIS and its different components.*

5.3.2 DAEMON Orchestration of NIS with support of ML pipelines

Our second implementation of NI-native architecture is presented above as a PoC using Kubernetes[12] as the main deployment environment. In addition, Kubeflow[13] is used to perform MLOps and as the baseline for developing some of the NIO functionalities. Furthermore, selected functionalities of the NIO are developed from scratch. The Eclipse Zenoh framework[14] is used for data flow programming among the NIF-Cs and for metric collection and aggregation, such as the ones coming from the sources NIF-Cs. A visual representation of the prototype implementation is in Figure 22. In the prototype, Kubernetes serves as the main deployment environment taking care of the MANO functionalities on top of a virtualized infrastructure. The Kubeflow deployment is realized as a Kubeflow cluster with one controller and 3 worker nodes, in which the NIF-C components are deployed as pods. The management of NIF-Cs is realized by the NIF component manager in Figure 10 through the Kubernetes API. As described in previous sections, a set of interconnected NIF-Cs following the N-MAPE-K representation compose a NIF. This is realized by a pipeline of pods managed by the NIF Manager utilizing the Kubeflow Pipelines Software Development Kit (SDK). The generated pipeline of NIF-Cs is defined in Python, translated in YAML and then deployed in Kubernetes (both pods and connectivity) using the developed service which utilizes the Kubeflow pipeline service. In the same fashion, the NI Orchestrator manages the NISs (using Kubeflow) at a higher hierarchical level.

Following the described approach, we can provide a set of NIO functionalities including (i) NIS composition, (ii) NIS lifecycle management, (iii) NIS workflow configuration, (iv) NIS selection, and (v)

[10] https://www.uantwerpen.be/en/research-groups/idlab/infrastructure/smart-highway/
[11] https://doc.ilabt.imec.be/ilabt/virtualwall/
[12] https://kubernetes.io/
[13] https://www.kubeflow.org/
[14] https://zenoh.io/

Monitoring, which are realized by building on the functionality already available in the Kubeflow framework. The developed monitoring service of the NIO provides monitoring of (i) NIF-C/NIF/NIS deployment status, (ii) NIF/NIS pipeline progress, (iii) MLOps progress, (iv) resource utilization, and (v) performance KPIs. The MLOps operations responsible for the model retraining, at the top-left corner of Figure 22, are realized as ML pipelines in the Kubeflow environment, while the NIF/NIS catalog is created using a Docker repository linked to the Kubernetes environment. Finally, it is important to stress that the NIF-C taxonomy (i.e., Analyze, Plan), as well as the adopted communication paradigm (Eclipse Zenoh) were adopted in all components of the architecture including NIF/NIS Catalogs (Dockers with different prebuilt libraries per NIF-C type) and during the NIF/NIS creation process (different preconfigured attributes per NIF-C type). Notice also that this PoC is also a realization of (part of) the procedures described in Sections 5.1 and 5.2.

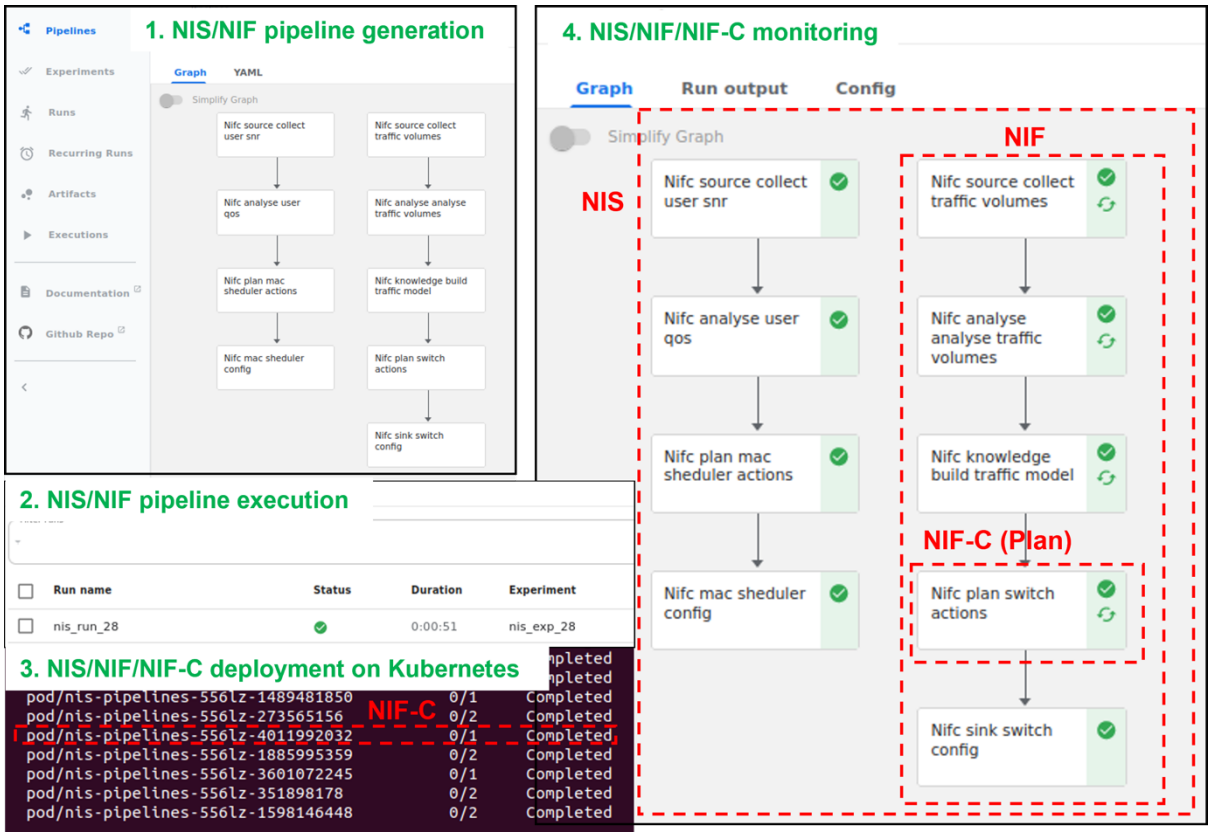


***Figure 22**. Prototype demonstrating NIS/NIF/NIF-C pipeline generation, deployment and monitoring.*

6 Updated state-of-the-art and final taxonomy of intelligent network management

In Section 5 of D2.2 [1], we presented the initial results of the literature review carried out in DAEMON. This literature review was guided through a protocol explained in D2.2 [1]. Following the protocol, we identified major trends in current research work. Through the analysis of the state-of-the-art research, we were able to identify areas where the research done in DAEMON stands out from the rest. Here, we present a final report including the updated numbers and concluding results.

6.1 Updated literature analysis

In Section 5.2 of D2.2 [1], we proposed a methodology for surveying the current state of the art on the research topics we are interested. Following this approach, we were able to review 39 papers in total. The results of that analysis were presented in Section 5.3 of the same deliverable. ***However, for this deliverable, we made slight modifications to the methodology. We came to the realization that some of the tasks we accomplished in DAEMON represent the cutting-edge in various subjects, including meta-learning, RAN virtualization, resource allocation, online learning, and more.*** Therefore, in this report we also include our own works and complement them with related works and in the cases that are applicable, works that improve our solutions and functionalities. The detailed results from the complete literature review can be found in the Annex B of this document.

In total, we reviewed 78 papers, adding 39 new papers with respect to D2.2 [1], from which 12 belong to the research made in DAEMON. Table 8 shows that most of the reviewed papers focus on the area of network optimization and control. Naturally, there is a bias due to the modification of the methodology. Most of our own works focus on this area, which explains the increase in reviewed works in such an area. Also, the preferred location for such algorithms is in the control and orchestration plane, as per Table 9.

Table 8. *Count of publications per Network Micro-Domain and Application Areas.*

	Network Application Areas				
Network Micro-Domain	**Network Diagnostics and Security**	**Network Optimization and Control**	**Network Planning**	**Network slicing**	**Grand Total**
Transport	7	3	0	0	10
Subscriber	1	0	1	0	2
Edge/core	0	10	0	0	10
Edge/Client	0	1	0	0	1
Edge	1	15	0	0	16
Cross-domain	1	13	1	4	19
Core, Transport	0	2	0	0	2
Core	1	5	0	0	6
Access	2	9	1	0	12
Grand Total	**13**	**58**	**3**	**4**	**78**

Table 9. *Algorithm Location.*

Algorithm Location	NI Solutions
Control Plane	40
Control and Orchestration Plane	18
Orchestration Plane	11
Data Plane	9
Grand Total	**78**

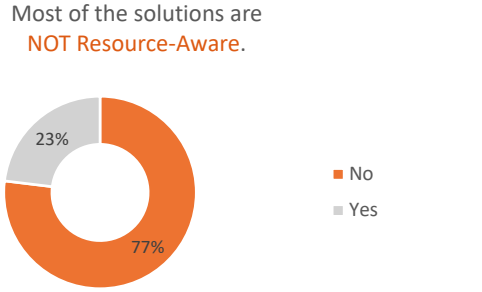


Figure 23. *Resource-awareness of reviewed works.*

A NI solution tackling the resource optimization problem is a good example of a cross-domain approach. The solution can be applicable at the edge, at the core or at the network access. The same applies to anomaly detection. Therefore, classifying such papers in absence of explicit indication is complicated.

However, unlike D2.2 [1], ***the most used ML method is Supervised Learning, followed by Reinforcement Learning***, as shown in Figure 24. Supervised learning is powerful in predicting future network states and resource usage, which can be used later to take decisions, hence, its popularity. Nevertheless, most of the algorithms are not resource-aware, which hinders the applicability of such models, considering that most of them are deployed at the network access and edge (cf. Table 8). ***ML quantization and pruning are relatively new techniques; they are not often applied in published papers***. However, recent efforts from Xilinx [34] are inspiring in achieving ML training and inference in resource-constrained devices. Exploiting this research line, DAEMON proposes the design of a standardized methodology to determine the correct level of quantization of Deep Learning (DL) models for each specific NI functionality, as it will be shown in Section 7.1.8.

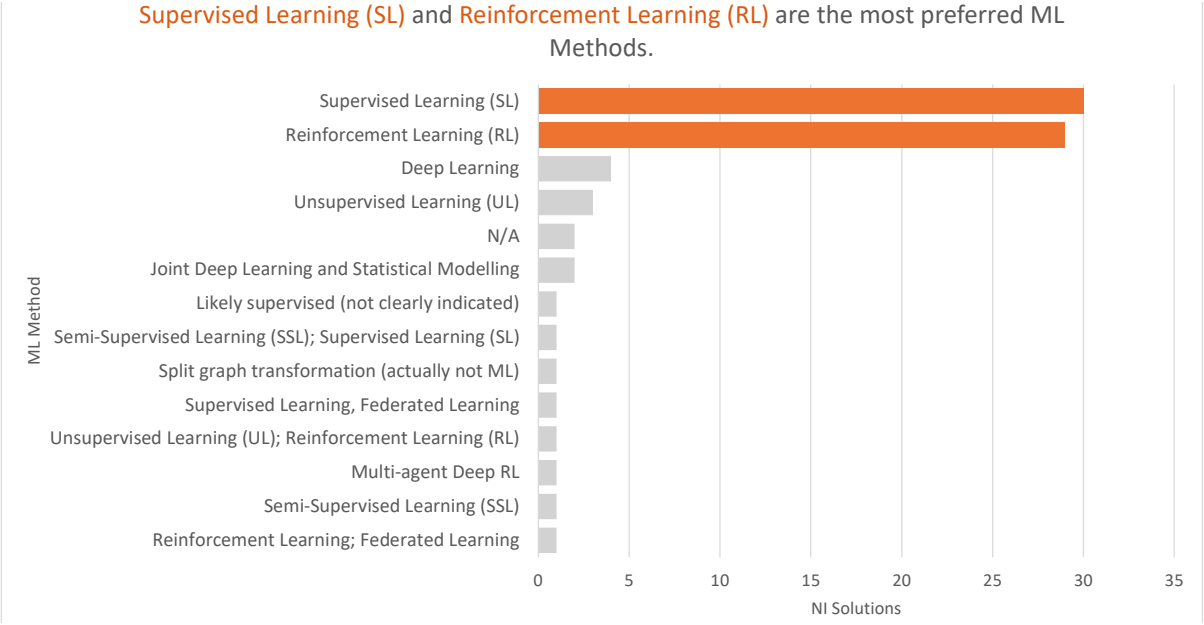


Figure 24. *Most common ML methods in the literature review.*

Finally, ***real and synthetic datasets are equally used in the reviewed papers.*** As shown in Figure 25, most of the published papers use real and synthetic datasets. Alternatively, the authors are using a combination of both or do not provide enough information regarding the dataset they are using. However, the real dataset setup is limited to a few nodes, which might not be representative of the expected density of B5G networks. We are aware of the difficulties of obtaining a real dataset, though we emphasize the importance of training and testing ML models on real, high-quality, and large data since it will facilitate the adoption of ML models in the field. As an alternative, researchers should focus on the creation of ML models that are robust yet generalized so that they can be trained in synthetic data and perform well when deployed on production.

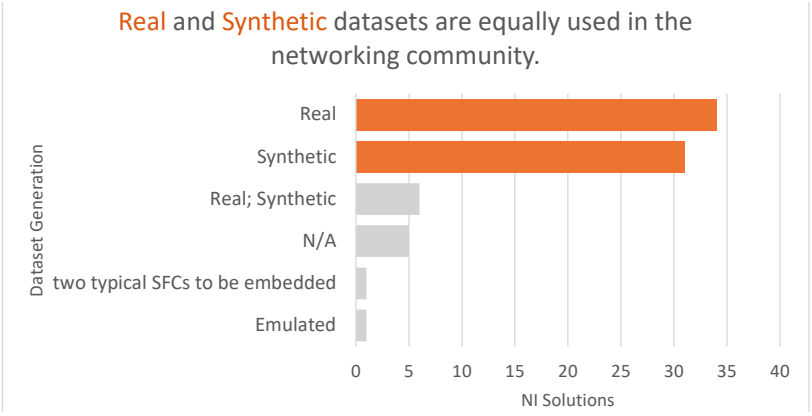


Figure 25. *Dataset generation of the reviewed works.*

6.2 Concluding remarks literature review

The previous section analyzed over 90 works that proposed ML and hybrid techniques to solve several known problems in network management. Given the approach we follow for surveying the current state of the art, we recognize that the surveyed papers are biased towards certain topics, such as resource management and network optimization and control. However, DAEMON functionalities cover several network microdomains, and therefore, the conclusions that we draw from the previous analysis are equally valuable.

Something that we cannot deny is how ***the AI/ML hype has vastly penetrated the networking community***. Proof of that, beyond the papers that we surveyed, is the creation of the brand-new IEEE Transactions on Machine Learning in Communications and Networking[15] journal, focusing on high-quality manuscripts on advances in ML and AI methods and their application to problems across all areas of communications and networking. Another effort from IEEE is the first International Conference on Machine Learning for Communication and Networking[16], aiming at promoting fundamental and applied research of ML for designing, analyzing and optimizing communication systems. These efforts mentioned above are the response to the ever-increasing interest of researchers in applying ML in networking and we are sure that they will foster more endeavors in the field.

AI and ML bring benefits to the telecommunications industry in two dominant fields, namely, data processing and automation. Given the amount of information it is produced on current networks, ML excels at finding hidden patterns in such data and meaningful features. Those patterns can be leveraged later by controllers and orchestrators, optimizing other network processes, which emulate human intelligence. In this setup, ML models look for correlations in multi-dimensional data to gain insights into, e.g., resource utilization, to forecast future system states and adapt accordingly. Regarding adaptivity, thanks to the generalization properties of supervised learning methods, online learning, or adapting a model during runtime as in the case of reinforcement learning, ML will allow communication systems to change dynamically accordingly to the system dynamics. This is not possible with current methods since the operational parameters are valid for a given network configuration and traffic load. Once they change, the operational points must be calculated again, which implies an interruption of the service or manual configuration.

During the literature review, we could find evidence of one of the statements we advocate in DAEMON, i.e., that most ML models for networking are designed to work in isolation. That is, they do not interact with other ML methods, and if they do, the solution is designed in such a way that the data pipeline and the interactions between the ML models are fixed to work in the conditions given. In principle, this goes against future workflows where: 1) the outputs of different ML models could be aggregated to take better-informed decisions and share knowledge among themselves; 2) the input data to such algorithms can come from different sources and in different formats; and 3) ML models belonging to different administrative domains can act over the same infrastructure which can cause conflicts.

Undoubtedly, data is a big part of the ML workflow. However, ***data is often overlooked in the networking community.*** Most of the reviewed papers do not pay much attention to the data they are using for training and validation. Synthetic datasets are generated under some specific conditions that, if not shared, other scientists cannot reproduce. On the other hand, real datasets may be subject to privacy regulations, which make them hard to replicate. As a result, two models that solve the same networking problem, e.g., traffic classification, are not comparable because they were trained using different data or in different network conditions. This will obstruct, enormously, the adoption of ML in networking. Ideally, the networking community should push towards high-quality, large, open datasets, hopefully standardized, so ML models can be trained on the same data and the same network conditions, so a fair model comparison is possible. Think for instance how the computer vision community has several open-source datasets, e.g., ImageNet[17], CIFAR[18] and COCO[19], and how that promoted the creation of powerful models [35], which were the stepping stones of computer vision as we know it today. Or in the reinforcement learning community where they have Gym[20], a standard API with a diverse collection of reference environments. In gym, multiple RL algorithms can be trained and tested, solving the same environment. Having a common networking environment and common data generation would facilitate the comparison between intelligent and non-intelligent models, which is a weak point spotted during the literature review.

[15] https://www.comsoc.org/publications/journals/ieee-tmlcn
[16] https://icmlcn2024.ieee-icmlcn.org/
[17] https://www.image-net.org/
[18] https://www.cs.toronto.edu/~kriz/cifar.html
[19] https://cocodataset.org/
[20] https://www.gymlibrary.dev/

Finally, we noticed that ***most of the proposed ML models are not tailored for networking.*** The loss and the reward function are defined in terms of known error functions, e.g., MSE, cross-entropy, or minimization/maximization functions. Notice that the loss and reward functions optimize the algorithm's learning, e.g., minimizing the cross-entropy, according to the learning problem, e.g., classification. However, the learning problem is not necessarily related to the optimal solution of the networking problem, e.g., selecting the best modulation scheme to minimize interference. The same is valid for RL approaches, where two different models will produce different learning metrics, e.g., cumulated reward, but their behavior in terms of the network metrics, e.g., minimizing the latency, needs to be evaluated. Therefore, we need to better understand how a model will impact the network stability and reliability, not only by improving their learning metrics.

7 Final guidelines on the pragmatic design of Network Intelligence and Limits of AI

We have presented the evolved architectural design of the project, which has been designed to support a native integration of NI in the network, and where DAEMON has focused on eight groups of such NI functionalities (see Section 2 of this document). In this section, we present an updated version of the guidelines proposed by DAEMON for incorporating machine-learning-based functions in the design and implementation of each of these NI functionalities. These guidelines are based on the experimental results and outcomes obtained from the project's research. We provide two sets of guidelines: The first focuses on the modifications required to adapt AI/ML solutions into specific networking applications (i.e., on *Tailored AI*), whereas the second set comprises insights on whether AI/ML solutions are the best choice for different network use cases (i.e., on *Limits of AI*). We build upon the preliminary guidelines provided in Section 4 of D2.2 [1]. For that, we recall the guidelines provided in [1], while adding new guidelines and additions to already existing guidelines generated from last year's development. A comprehensive description of the evolution of each guideline is provided in Table 10, where we show, for each guideline, (i) the related functionalities, (ii) its evolution during the last year's iteration of the project (stable, updated, or new), and (iii) its category (whether it relates to tailored AI or to the limits of AI).

Table 10. *Evolution of the DAEMON project's guidelines from the previous deliverable.*

Guideline	Related Functionality								Evolution from D2.2	Category
	RIS	MTERM	IBSSI	CAWRS	EAWVNF	SLMANO	CFORE	AARES		
Incorporating prior knowledge in decision making schemes						✓			Stable	**Tailored AI**
Avoiding the loss-metric mismatch							✓		**Updated**	
Loss function meta-learning							✓		**Updated**	
Self-learning models based on dataflow programming		✓							Stable	
Adapting a known reward function to networking		✓				✓			**Updated**	
Low inference time and energy consumption	✓			✓	✓				**Updated**	
Explainable NI				✓			✓		**New**	
No "one-size-fits-all" in Neural Network Quantization				✓	✓				**New**	
Traffic classification		✓							**Updated**	**Limits of AI**
Wireless Network performance inference		✓							Stable	
Self-learning MANO						✓			Stable	
Forecasting in mobile networks							✓		Stable	
In-backhaul inference			✓						**Updated**	
Federated learning powered NI functionalities						✓		✓	Stable	
Predictive HARQ				✓					Stable	
Hard constraints					✓				**New**	
Anticipatory decision-making in mobile networks		✓				✓	✓		**New**	

After illustrating the evolution of the guidelines from previous documents, we describe in detail all newly added guidelines, as well as the new content for the ones that have been updated. We keep a brief description of the stable guidelines for the sake of completeness, while the detailed description can be found in Section 4 of D2.2 [1]. We separate the guidelines in two main categories, as previously mentioned, depending on whether they are related to tailoring AI design for NI or to the limits of AI.

For each of these guidelines (stable, updated, and new), we provide a critic view of the provided solutions by briefly describing the future challenges and the potential limitations of the proposed guidelines, so as to offer a complete picture of the State of the Art and the current NI situation for future research projects.

7.1 Tailored AI design for NI

One of the main goals of DAEMON is to define methodologies to adapt legacy modern deep-learning-based AI models to the particularities of real-world NI problems. This objective is crucial because networking operation, optimization, and management conform to create a complex and singular framework that greatly differs from other fields. Because of that, top-level solutions with unmatched performance in other less constrained fields may fail to achieve a similar operability in networking applications.

In light of these considerations, the DAEMON project challenges the current practice of addressing NI problems by directly adopting general-purpose AI models or models that have been successfully employed in other domains, without significant modifications. Instead, a sensible integration of AI models into NI calls for substantial customization and contextualization. In this section, ***we provide a detailed list of the research outcomes obtained within the DAEMON project and aimed at adapting and tailoring AI/ML solutions for network intelligence***. We link these adaptations to the requirements of the target functionalities described in Section 2 of D2.2 [1] and in Section 2 of the current document. These guidelines are connected to those requirements in a twofold manner: on the one hand, the derived solutions build on the constraints and the goals set by those functional requirements; on the other hand, the developed solutions allow us to unveil the limitations of the requirements from the outcomes of the research work, thus triggering requirements updates. This last interconnection has been crucial to steer, improve and evolve each project year iteration. Table 11 summarizes all the guidelines related to tailoring AI for NI that have been produced by DAEMON. We indicate which requirements are related to each guideline, and we also describe the main take-away message for each guideline. The guidelines that were already presented in D2.2 [1] are briefly commented in the remainder of this section, complemented with a further detail or follow-up guideline formulated during the last project's iteration, while fully new guidelines are described in detail. For all these guidelines, we provide the main current limitations and future challenges that have been identified during the project development, such that future research can extract meaningful guidelines for future steps and open problems. Note that the focus there is on the extrapolation of the design guidelines of AI for NI, guidelines that arise from the activities carried out during the DAEMON project. Therefore, when applicable, we also link guidelines to their implementation for some specific NI-assisted functionality that are presented in other deliverables of the project.

Table 11. *Summary of the DAEMON project's guidelines on tailoring AI for NI.*

Guideline	Requirements	Description	DAEMON related work
Incorporating prior knowledge in decision-making schemes	FR-SLMANO-002 FR-SLMANO-005	AI models for NI shall incorporate prior knowledge about the network system by design, e.g., as restrictions on the coefficients of the neural network, or as simplifications to the training data. This reduces the amount of data needed for training without impairing AI performance.	[36]
Avoiding the loss-metric mismatch	FR-CFORE-002 FR-CFORE-005	AI models for NI shall be trained using customized loss functions that are carefully developed based on expert system knowledge. Unlike legacy loss functions that are designed to be generic enough to work well in a wide range of scenarios, task-tailored losses can capture the specific performance targets and dramatically improve results. **Update:** When it is not feasible to obtain or design a customized loss function, the DAEMON project advocates the use of loss function meta-learning, which enables the customization of loss functions for unknown	[37] [38]

		relationships between decisions/predictions and the system performance.	
Loss function meta-learning	FR-CFORE-002 FR-CFORE-005 FR-CFORE-006 FR-CFORE-007	AI models for NI may adopt when relevant a design that meta-learns the loss function that best suits the network management objective at hand. This is the case, e.g., when the performance metric to be optimized by anticipatory MANO actions is not known a priori by the network operator. **Update:** Anticipatory MANO must take into account intertwined forecasts. The structure and configuration of the AI/ML methods designed for loss meta-learning and anticipatory MANO must incorporate the characteristics needed to handle such complex problems to facilitate scalability and modularity.	[37] [38]
Self-learning models based on dataflow programming	FR-MTERM-004	AI models for NI shall be informed by tailored data feeds. The input to AI models for NI requires decentralized and distributed data management, unification of data patterns, support for heterogeneous devices, support for eventual consistency models, or support for different timescales and real-time communications. In turn, these call for both decentralized data pipelines as well as the ability to declare deadlines for real-time operations and the reusability of components.	[39]–[42]
Adapting a known reward function to networking	FR-SLMANO-003 FR-MTERM-007.00	AI models for NI that are based on RL may adapt known rewards instead of defining new ones. Contrary to most of the RL applications in networking, where the states, actions, and reward function are defined using a networking rationale, the DAEMON project commends that the many different reward expressions used in well-known applications of RL can be leveraged and adapted to suitable rewards that drive NI decisions in specific network functionalities. **Update:** The AI solutions for networking shall be integrated into control frameworks such as the MAPE-K. Furthermore, those general-use frameworks shall be adapted to the particular structure of the network. We defined a new reward function that optimizes a multi-objective function regarding the number of replicas and a target delay. Additionally, we framed the solution into the N-MAPE-K framework, going beyond the state-of-the-art, where several scaling solutions can be swiftly integrated as NIFs in future network infrastructures.	[26], [43]
Low inference time and energy consumption	NFR-RIS-001	AI models of NI may be designed for extremely low inference latency and energy consumption. This requirement	[17], [44], [45]

	NFR-RIS-002 NFR-EAWVNF-003 NFR-EAWVNF-004 NFR-CAWRS-000 NFR-CAWRS-001 NFR-CAWRS-003	applies to a number of mobile network applications such as traffic classifiers or load balancers in multi-gigabit-per-second backhaul segments, or in baseband processing operations in the radio interfaces, where the processing latency budget for inference is well below 100 microseconds. Techniques for AI design that meet such specifications include (i) distribution of complexity across simple and fast models, e.g., via multi-actor-critic RL, (ii) in-subsystem inference that avoids time-consuming communication with a GPU, e.g., by running AI directly in the network interface card (NIC), or (iii) use of low-complexity AI models, e.g., Binarized Neural Networks (BNN). **Update:** The usage of Digital Twins (DT) shall be fostered to obtain faster and more resilient models, while avoiding the need to deploy in real hardware and take real measurements. DTs can speed up the design phase and at the same time reduce design costs.	
Explainable NI (New guideline)	NFR-CAWRS-000 NFR-CAWRS-001 NFR-CAWRS-003 FR-CFORE-000	Network management and orchestration require accountability and verification, but most of learning-based solutions are opaque blocks that are not designed with the objective of transparency. DAEMON advocates for the use of explainable AI for developing intelligent solutions in the network's core, since this conflict may preclude the ubiquitous use of AI for networking. To generalize explainability for any network problem, DAEMON proposes the use of the classification of explanations developed by the Machine Reasoning community: attributive, contrastive, and actionable explanations. Furthermore, in order to extract the most from the system-independent standard explainable AI methods, DAEMON proposes the use of specific explainable blocks that provide a compact, human-friendly, network-aware representation of the otherwise verbose complex explanations that Explainable AI (XAI) techniques provide.	[46]
No "one-size-fits-all" in Neural Network Quantization (New guideline)	NFR-EAWVNF-004 NFR-CAWRS-002	Next-generation communication systems will face new challenges related to efficiently managing the available resources. DL is one of the optimization approaches to address and solve these challenges. However, there is a gap between research and industry. Most AI models that solve communication problems cannot be implemented in current communication devices due to their high computational capacity requirements. New approaches seek to reduce the size of DL models through quantization techniques, which provides	[47]

		the means to change the traditional method of using operations with 32 (or 64) floating-point representation to a fixed point (usually small) one. However, the recent works using quantization techniques apply the one-size-fits-all approach: all layers are quantized equally. DAEMON proposes a methodology to determine the level of quantification that is required to obtain the best trade-off between the reduction of computational costs and an acceptable accuracy in a specific problem.	

This set of guidelines, as a whole, addresses all the items related to the tailored design of AI for NI as presented in the Description of the Action (DoA) of the DAEMON project. We make these links explicit as follows.

- The guidelines on (i) incorporating prior knowledge in decision-making schemes, (ii) avoiding the loss-metric mismatch, and (iii) adapting a known reward function to networking address the issue of ***"closing the loss-metric mismatch, by deriving general guidelines for the design of dedicated loss functions that are perfectly aligned with the actual performance metrics of interest".***
- The guidelines on (i) loss function meta-learning and (ii) self-learning models based on dataflow programming address the problem of ***"designing a methodology for self-learning AI models that dynamically and automatically balance costs and efficiency, by learning the loss function indirectly from the feedback of the end-customers, without requiring them to explicitly identify their objectives".***
- The guideline on (i) low inference time and energy consumption address the problem of ***"developing elastic NI models capable of adapting their own complexity to the context, trading off (computational) complexity for accuracy, responsiveness or energy efficiency as needed".***

7.1.1 Incorporating prior knowledge in decision-making schemes

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1[8], updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- FR-SLMANO-002, FR-SLMANO-005 for Self-learning MANO.

In many networking problems tackled via ML, the probability of taking an action a when a certain observation ω is made, i.e., the policy $\pi(\omega, a)$, is modeled as a neural network. In such problems, ***there often exists some prior knowledge inherent to the problem that constrains the action space****;* for instance, it may be known a priori that, for two observations ω_1, ω_2 in different regions of the observation space, a certain action a should be more likely for observation ω_1 than for observation ω_2. The simplest example of this feature is a monotonicity constraint: $\pi(\omega_1, a) > \pi(\omega_2, a)$ if $\omega_1 > \omega_2$. Plain vanilla neural networks do not possess such a property; however, they can only learn this property after being trained on a sufficiently large set of data. Incorporating this prior knowledge in the neural network modeling can be achieved in various ways: (i) by putting adequate restrictions on the coefficients of the neural network; (ii) by preprocessing the training data such that pairs of data that do not expose the desired behavior are suitably altered or removed. In both cases, by incorporating prior knowledge, less data is needed to train the neural network to achieve a reasonable performance. In Section 4.1.1 of D2.2 [1] we have demonstrated how the inclusion of prior knowledge in a model, which estimates the acceptance probability of a network service, enhances the performance of the said model.

Limitations and future challenges

The incorporation of prior knowledge, in the form of unfeasible subspaces, structure of the data or the solution, etc., is a complex task that requires a lot of tailoring and must be tackled differently for each problem. The related techniques developed in the DAEMON project, which were described in D2.2 [1], all are restricted by the fact that the prior knowledge is under the form of monotonicity constraints. At the same time, the techniques can be readily used in all problems where such a (set of) monotonicity constraints can be identified. Some other ad hoc techniques will have to be developed to cope with problems where the prior knowledge is presented in a different form.

7.1.2 Avoiding the loss-metric mismatch in network intelligence

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this document:

- FR-CFORE-002, FR-CFORE-005 for Capacity Forecasting NI.

Loss functions drive the training process of supervised machine learning models. In most cases, loss functions are designed to be generic enough to work well in a wide range of scenarios. In regression problems, including forecasting tasks, Mean Absolute Error (MAE), Mean Square Error (MSE), or Mean Squared Logarithmic Error (MSLE) are common choices for expressing the loss.

However, in many practical cases in network management, such traditional losses do not characterize well the target performance metric of forecasting tasks. For instance, in anticipatory resource allocation problems, the goal is anticipating a capacity that is *sufficient* to accommodate future traffic demand. Indeed, underprovisioning of capacity leads to the disruption of the offered service and violations of the Service-Level Agreements (SLAs) with the service providers. There, it is critical that the predictor learns to forecast a minimum quantity that is always above the demand.

Using a traditional loss function to perform forecasts in cases such as those outlined above results in a so-called loss-metric mismatch, where the regression objective (i.e., the loss to be minimized) does not correspond to the optimization of the actual performance metric. As a result, the AI model's predictions are not aligned with the expected network management objective.

As part of its guidelines for the tailored design of AI for networking, ***the DAEMON project supports the use of customized loss functions that are carefully developed based on expert system knowledge***, i.e., a deep understanding of the network engineering or management task at hand, as well as of the variables that affect it and how they do so. Figure 26 illustrates how the tailored design of loss functions for NI shall occur. In the left plot (a), a pure traffic predictor is trained using a legacy loss, e.g., MAE or MSE for regression. The resulting forecast serves as an input to the actual decision block, which is manually designed by human experts. In the right plot (b), the novel approach proposed by the DAEMON project is outlined: expert knowledge is used to directly design a dedicated loss that encodes the relationship between the prediction and the performance objective. As a result, the predictor directly forecasts the management decision to drive the MANO actions. Importantly, the action decision is now aware of the unavoidable prediction error (e.g., lower accuracy in predicting small traffic volumes), and automatedly compensates for it. More details about the techniques that implement this guideline can be found in D2.2 [1].

Limitations and future challenges

Tailoring the loss functions to the networking metric of interest will be crucial to achieve many of the envisioned applications of AI for networking and will be necessary to accomplish a general network intelligence. However, the sheer number of different problems with completely diverse characteristics that can be found in the network makes unfeasible a detailed tailoring of the loss functions for each task that we desire to implement by means of NI. It would require enormous resources of expert knowledge, and it would probably require a continuous adaptation. An alternative to solve this problem and being able to generalize NI for any network-related task is to substitute the human-designed, manually tailored implementation by autonomous learning that allows the network to self-adapt to new/unknown problems, as we will describe in the following guideline.

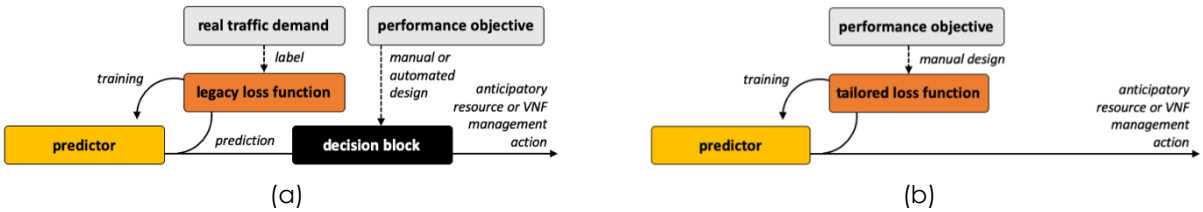


Figure 26. *Different approaches for solving the loss-metric mismatch.*

7.1.3 Enhanced Loss meta-learning for network intelligence

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- FR-CFORE-002, FR-CFORE-005, FR-CFORE-006, FR-CFORE-007 for Capacity Forecasting NI.

The performance metric to be optimized by anticipatory MANO actions is not always known a priori by the network operator. This is the case, for instance, when the performance must be measured at the

application layer (i.e., in the service provider domain), or when it concerns end user satisfaction (e.g., if it relates to mean opinion scores or quality of experience). In these situations, designing tailored loss functions as presented in Section 7.1.2 is not possible, since the human expert (e.g., network manager or system engineer) does not know the exact relationship between the forecast and target performance.

DAEMON sets forth innovative guidelines to deal with NI design in the complex situations described above. Specifically, instead of imposing a predefined expression of the loss function used to train the predictor, ***the DAEMON project advocates a design of forecasting models that is free to meta-learn the loss function that best suits the network management objective at hand***. In practice, this is realized by combining a loss-learning block with the actual predictor, as shown in Figure 27. This block is responsible for learning the loss function or, equivalently, capturing the relationship between the forecast produced by the predictor and the target management objective. Once ready, the loss-learning block can operate as a tailored loss function: it receives the output of the predictor and determines its quality for the precise management task. Therefore, it can be employed to train the predictor so as to steer the optimization of its parameters toward minimizing the actual MANO objective.

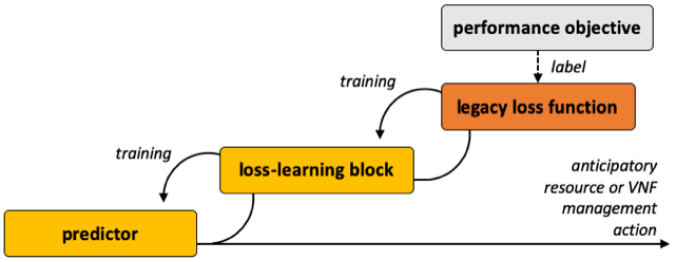


Figure 27. *Loss meta-learning for NI. The network management objective is learned and encoded into a loss-learning block. This block then serves as the loss function to train the predictor, so that it directly outputs the anticipatory action.*

This model, coined *Loss-Learning Predictor*, solves a regression problem and outputs a continuous-valued action, but does so by learning from experience, similarly to Reinforcement Learning (RL) approaches. It is worth noting that such loss training can use performance measurements collected in the target system as a direct representation of the objective, without any need to formalize it as a mathematical function.

Previous guidelines in D2.2 [1] advocated for an implementation based on two Deep Neural Networks (DNN), one for the decision-making in cascade with a second DNN that implements the loss learning block. However, such a structure is limited and cannot operate in scenarios with intertwined variables. Unfortunately, networking problems usually depend on intertwined variables that need all to be forecast in order to deliver the anticipatory MANO decision. Based on the knowledge and expertise acquired during the last year of the project, we propose a new structure, which is generic, can be applied to any general problem with multiple intertwined variables, is scalable, and modular. From this knowledge, ***DAEMON advocates the use of modular and scalable structures that are jointly trained but structurally independent, such that the neural network structure mimics the logical shape of the problem.***

This vision is realized in the DAEMON project through a specific and simple solution. The crucial aspect is to split the previously mentioned predictor/decision-making block (cf. Figure 27) into two different blocks. A first part is composed of separate parallel neural networks, each one receiving as input one of the variables. All these parallel blocks are fed into an *Assembler* block, which finally outputs the decision, as shown in Figure 28. The main guideline and idea is to logically separate the learning of the temporal correlation of each variable (carried out by the first block) and the inter-variable relationship (carried out by the aggregator). This allows for much faster and stable training, and it facilitates transfer learning as each sub-block can be extracted and applied to different scenarios.

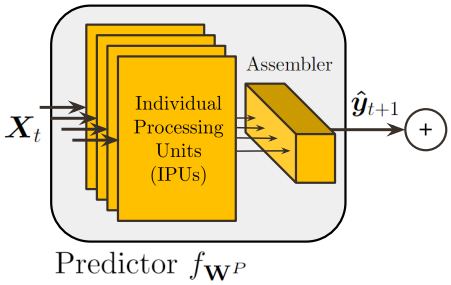


Figure 28. *Proposed architecture of the predictor for loss meta-learning model set forth by DAEMON.*

This design has several key advantages:

- The loss-learning DNN can learn the relationships between the prediction and the objective from measurement data, without the need for human intervention.
- Without any need for prior knowledge of the system, the loss-learning DNN can model tangled non-linear and multivariate objectives that may characterize practical MANO decisions.

Full details on the design and operation are available in [37] and [38], and a preliminary performance evaluation showing the advantages of loss meta-learning over legacy DNNs is presented in Section 4.6.2 and Section 4.6.3 of D5.1[7] of the DAEMON project. Overall, this guideline paves the road to the design of more adapted and automated NI models for MANO operations.

Limitations and future challenges

The development of loss meta-learning solutions is still in its infancy, and there are many open questions with respect to its limitations. One of the main limits is the need to scale the complexity of the neural network depending on the complexity of the problem, which would be unknown for the problems here considered. This could be solved through some controlled iteration, with the inherent increase in the required time for correct training. Moreover, the fact that these solutions are intended for complex environments with unknown performances implies that it is difficult to verify the optimality or correction of the offered solutions. These approaches would require a strong and well-defined AI lifecycle management to correct and adapt the developed algorithms.

7.1.4 Self-learning models based on dataflow programming.

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- FR-MTERM-004 on Multi-timescale edge resource management.

Designing an NI-native architecture for B5G systems requires clear requirements and specifications, related to data-driven features such as: decentralized and distributed data management, unification of data patterns, support for heterogeneous devices, support for eventual consistency models, or support for different timescales and real-time communications. Based on these considerations, ***the DAEMON project advises that we need both decentralized data pipelines as well as the ability for declaring deadlines for real-time operations and the reusability of components***.

Eclipse Zenoh-Flow[21] provides the mechanism for simplifying and structuring (i) the declaration, (ii) the deployment, and (iii) the writing of complex applications that can span from the Cloud to the Edge or beyond the edge, offering flexibility and extensibility for data flow programming structures. The main benefit of this approach is that it enables us to decorrelate applications from the underlying infrastructure: data are published and subscribed to without the need to know where they are actually located, e.g., cloud, edge, or beyond edge.

We tackled the challenge of integrating NI algorithms into the overall DAEMON's architecture presented in the previous sections. We adopted the mentioned N-MAPE-K feedback loop methodology (Network Monitor-Analyze-Plan-Execute over a shared Knowledge) to handle the fundamental point of understanding which are the needed interfaces. With N-MAPE-K, the algorithms that run at NI instances can be classified in a unified manner, according to how they interact with the other network elements. Based on these activities, ***DAEMON advocates the adaptation of standard methodologies to the specific characteristics of the network.*** More details are presented in the other deliverables of the project.

Limitations and future challenges

There exist several methodologies that are widely adopted or generally known to enact the feedback loop control. It is yet not analyzed the particular advantages that each of them may offer, or if some of them are equally valid for their application in network and NI lifecycle management. This study requires a detailed analysis that is expected to be done in the future.

7.1.5 Adapting a known reward function to networking

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- FR-SLMANO-003 for Self-learning MANO, and FR-MTERM-007.00 for Multi-timescale resource allocation.

[21] https://github.com/eclipse-zenoh/zenoh-flow

In D2.2 [1], ***the DAEMON project commended that the many different reward expressions used in known applications of Reinforcement Learning can be leveraged to identify suitable rewards that drive NI decisions in specific network functionalities***, contrary to most of the RL applications in networking where the states, actions, and reward function are defined using a networking rationale. We exemplified this guideline for autonomous service scaling, which was mapped to the well-known Cart-pole environment, showing how adapting a reward function from one domain (e.g., *Cart-Pole* environment) to another (e.g., networking environment) can be leveraged to identify suitable rewards that drive NI decisions in specific network functionalities.

Yet, the standard reward functions and environments cannot often be directly applied to networking problems, and they may require different levels of adaptation. ***In this regard, DAEMON commends the adaptation of standard reward functions and methods to networking and orchestration frameworks, such that the N-MAPE-K framework***. In order to realize this guideline, in [43] we defined a new reward function that optimizes a multi-objective function regarding the number of replicas and a target delay. Additionally, we framed the solution into the N-MAPE-K framework, going beyond the state-of-the-art, ***where several scaling solutions can be swiftly integrated as NIFs in future network infrastructures.***

More specifically, in every time step, the agent pays an immediate cost depending on how good or bad the action it took is. The cost of taking action when the environment moves from one state to another can be defined as a weighted function, including the following contributions.

- If the agent cannot fulfill the SLA, it incurs a performance cost, c_{perf} with an associated w_{perf}, which is paid every time the perceived peak latency (d) exceeds a predefined threshold (d_{max}). The cost is zero otherwise.
- If the agent must deploy a new replica, a resource cost c_{res} is paid, with an associated w_{res}; this can be seen as a rental cost in cloud environments or the consumed energy of the replica while it is running.

These two contributions are combined into a weighted function, where the respective non-negative weights define an optimization profile, $w_{perf} + w_{res} = 1$. The weights ($w_{perf}$and w_{res}) multiply an indicator function ($\mathbb{1}\{\cdot\}$) that varies between 1 and −1 depending on whether a condition is met. For instance, if the perceived peak latency is above a threshold, the indicator function is 1 or 0 otherwise; if a new replica is instantiated, the indicator function is 1, or −1 if the replica is removed. Finally, the reward function is defined as the negative cost function, since the main objective is to minimize the total cost.

$$r = -c_{total} = c_{perf} + c_{res} = w_{perf} \cdot \mathbb{1}perf + w_{res} \cdot \mathbb{1}res$$

$$\mathbb{1}perf = \begin{cases} 1 \; if \;\; d \leq d_{max} \\ 0 \quad otherwise \end{cases}$$

$$\mathbb{1}res = \begin{cases} -1 \quad if\ remove\ replica \\ 0 \quad if\ maintain\ replica \\ 1 \quad if\ add\ replica \end{cases}$$

With this reward function we trained a Proximal Policy Optimization (PPO) agent, which showed very different behavior depending on the optimization objective. When optimizing the resources over the performance, the average amount of replicas is always low; however, there is no guarantee of the achievement of the SLA. On the contrary, when the scaler is trained with a reward function that optimizes the performance over the resources the violations are reduced to their minimum at the expense of creating more replicas. For more details, please refer to [43].

Limitations and future challenges

RL-based algorithms in general are non-deterministic, meaning that different outputs can be obtained for the same parameter configuration due to the random initialization of the neural network's weights used in the state-action approximation. Therefore, since the learning behavior of an RL-based scaler heavily depends on the reward function definition, the reward function must be carefully designed, and its effects on the stability of the RL algorithm and its impact on network reliability must be studied. The orchestration frameworks will play an important role in controlling such reliability.

7.1.6 Low inference time and low energy-consuming NI

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- NFR-RIS-001 and NFR-RIS-002 for Reconfigurable Intelligent Surfaces Control NI;
- NFR-EAWVNF-003 and NFR-EAWVNF-004 for Energy-aware VNF Orchestration NI;
- NFR-CAWRS-000, NFR-CAWRS-001, and NFR-CAWRS-003 for Compute-aware Radio Scheduling NI.

Extremely low inference latency and energy consumption is a requirement for NI models in a number of mobile network applications such as traffic classifiers or load balancers in multi-gigabit-per-second backhaul segments, or in baseband processing operations in the radio interfaces, where the processing latency budget for inference is well below 100 microseconds. Most of the existing AI/ML solutions are resource-demanding and do not consider so stringent constraints for the inference task***. In DAEMON, we advocate for the development of tailored highly efficient ML solutions that focus on the said limiting processing latency while minimizing the loss of performance.***

In D2.2 [1], we provided three different approaches to meet such tight requirements: **Low complexity** (also by means of distributed and multi-agent learning)[48]; ***In-subsystem inference*** (directly in the CPU or Network Interface Card (NIC) that collects the input data); and ***Binarized Neural Networks*** (BNN) (i.e., quantized weights and activations)[49].

The design of solutions that fulfill these tight constraints are costly, since we need to evaluate the performance in the real hardware and obtain the results via measurements. ***In order to avoid long design processes, with difficult design loops, DAEMON advocates for the use of Digital Twins (DT) of the actual systems,*** which allows us to speed up the design and verification phases, reduce costs, and improve the transferability of the solutions.

Limitations and future challenges

The different approaches suggested above are not the ultimate solution for the problem of low inference time and low energy consumption, and each one suffers from different problems: "*Low complexity*" approach suffers from the loss in performance due to the reduced complexity, while distributed learning approaches require great efforts to avoid stability or fairness issues; "*In-subsystem inference*" leads to the need of simple models due to the reduced capabilities of NIC and CPU with respect to GPU, and "*binarized neural networks*" reduce precision and require special tools for training (since usual stochastic gradient descent does not generally work for them). However, there are very promising results for the three approaches, which motivates the research in these fields to advance in the achievement of the envisioned goals. For example, [48] has demonstrated an 18x increase in latency performance when using a common pipeline of NIC+CPU for data collection and ML inference, compared to performing both steps directly on the NIC; and, compared to an equivalent 8-bit quantized network, BNNs require 8 times smaller memory size and 8 times fewer memory accesses, with drastic gains on optimized hardware, e.g., exploiting SIMD extensions on intel or AMD CPUs.

7.1.7 Explainable NI

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- NFR-CAWRS-000, NFR-CAWRS-002, NFR-CAWRS-003, for Computation Aware RAN.
- FR-SLMANO-004 for self-learning MANO.

Network management and orchestration are based on reliability and fast response time. It is crucial for network operators and any involved stakeholder to be able to obtain a clear explanation and justification of any of the processes applied in the network. Unfortunately, most of the top-performing learning-based solutions (e.g., deep neural networks) are opaque blocks that offer little explainability and which are not designed with the objective of transparency. This conflict may preclude the ubiquitous use of AI for networking as stakeholders will not support losing accountability capabilities. Based on different activities within the project, ***DAEMON advocates for the use of explainable AI for developing intelligent solutions in the core functionalities of the network.*** This can be achieved by considering explainability as one of the objectives during the design phase.

In order to ensure a comprehensive and transparent system, it is very important to incorporate interpretability and explainability features. Significant enhancements to address the issue of explainability within the model were explained in Section 2.5 of [2], drawing upon machine reasoning techniques from existing literature in the field, in a solution called ATHENA. Within ATHENA, Section 2.5 of D3.2 [2], the Machine Reasoning component comprises the second block, which plays a crucial role in interpreting the outputs of the machine learning (ML) module and generating actionable decisions for the network, albeit at a slower pace. To achieve this, our model needs to offer insights into its internal functionality and decision-making process, which we refer to as explanations.

A relevant guideline for NI is the need for ***providing explanations falling into three distinct categories coming from the research in Machine Reasoning: attributive, contrastive, and actionable explanations***, as defined by [50]. Attributive explanations aim to provide an understanding of the attributes and features that contribute to a particular decision. Contrastive explanations highlight the factors that differentiate one decision from another. Lastly, actionable explanations offer insights into the steps or actions that can be taken based on the model's output. These three types of explanations shall be

adapted according to the specific context of the model, in the case of ATHENA, to the neural network-based actor-critic architecture. By doing so, the system not only produces accurate results but also provides meaningful explanations that enable the expert running the system to comprehend the reasoning behind its decisions and take appropriate actions based on those insights.
Besides this, we also encountered that many of the well-known X-AI techniques natively provide a verbose explanation, which is not human-friendly and is based on the fields they are based on (usually an image or natural language processing). Based on this fact, ***DAEMON proposes the use of specific explainable blocks that provide a compact, human-friendly, network-aware representation of the otherwise verbose complex explanations that Explainable AI (XAI) techniques provide***. The proposed architecture is shown in Figure 29.

This guideline was implemented in a project activity that analyzed the explainability of anomaly detection for traffic forecasting [46]. Generally, the state-of-the-art only considers stealthy perturbation techniques applicable to all the input to assess model vulnerabilities. We contribute a new way of assessing vulnerabilities that is specific to the problem of spatio-temporal mobile traffic forecasting. We first pinpoint with Explainable AI the most relevant gNBs to the forecast from a spatial perspective at each point in time. Next, we show that traffic injected only in those gNBs (perturbation) causes the model to under-estimate the prediction while SotA techniques lead to overestimation.

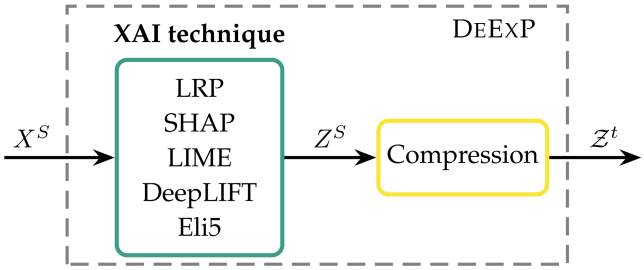


Figure 29. *Proposed comprehensible explainable AI set forth by DAEMON.*

Limitations and future challenges

Explainable AI methods have not yet been deployed and implemented in networks. These methods, although they provide some sort of explainable answer to the question of why the output of the algorithm has happened, they do not offer human-friendly outputs, neither network-related nor network-based outputs that understand the inherent structure and concept of the network. There are a lot of open problems in this topic, and the fundamental limits of explainability are still to be discovered. The project is still advancing this topic in different directions, i.e., defining a vulnerability score that is a combination of XAI scores and statistics, or quantify the damage to the predictor of newly defined traffic injection techniques based on the vulnerability score.

7.1.8 No "one size fits all" in Neural Network Quantization

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- NFR-EAWVNF-004 for Energy-aware VNF Orchestration NI
- NFR-CAWRS-002 for Compute-aware Radio Scheduling NI.

As seen in several NI solutions, large Deep Learning (DL) models are typically used to solve complex problems. Nevertheless, due to the size/complexity of such models, the inference must be performed in machines with high computational power, which is not characteristic of the devices composing the radio access, edge, or far-edge networks. Neural Network Quantization [51] helps to reduce the computational cost of implementing and deploying such DL models. However, the recent works that apply quantization to reduce the complexity of the DL models apply the "one-size-fits-all" approach, where all the layers are quantized using the same value [52]–[54]. Although it may work in some cases, this approach does not allow for finding quantization configurations that provide a desired trade-off level between model learning performance, e.g., accuracy, and the model's complexity.

Based on the previous problem, ***DAEMON advocates for the design of a standardized methodology to determine the correct level of quantization of DL models for each specific NI functionality.*** The objective is to automate the selection of the models and quantization in a zero-touch manner, ***such that the system can select the appropriate quantization choice in an automated manner***, which can ***be integrated into the generic NI lifecycle management*** presented in the previous sections.

To describe the proposed methodology, let us consider an experiment with three design factors, where each element has three possible levels. Then, using a complete factorial design, we would need to run the same experiment $3^3-1=26$ times to determine which design factor impacts the response variable (i.e., the outcome) the most. Translating this small example to the field of DL model quantization, if we have three quantifiable parameters (e.g., the input, the activation layers, and the weights), each in the range of 1 to 32 bits, it gives $32^3-1=32{,}767$ possible combinations. Evaluating the impact of each parameter's quantization level regarding the accuracy and inference cost is prohibitively time- and resource-consuming.

Based on a fractional factorial design, our methodology allows for reducing the number of experiments concerning quantifiable parameters. We divide the methodology into four stages. After each stage, we measure the trade-off between the model's performance metric (e.g., accuracy) and its inference cost (e.g., space in memory, number of operations) regarding the unquantized model. It is worth mentioning that it is possible to include a preprocessing of the input signals to reduce their size before applying this methodology, such as dimensionality-reduction methods or averaged filters. Figure 30 illustrates a detailed block diagram of the developed methodology.

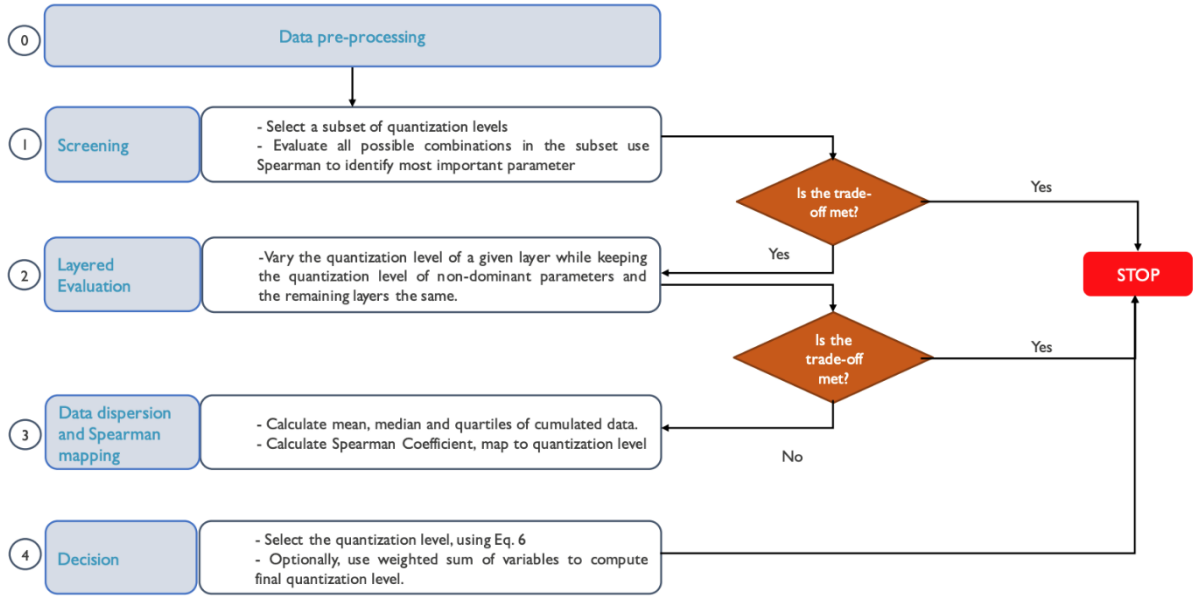


Figure 30. *Proposed methodology based on fractional factorial design.*

This methodology is proposed in [47] and applied to the Automatic Modulation Classification and Recognition (AMR/AMC) problem, as an example. However, this approach can be applied to any other problem. During the first stage, we identify the dominant parameter in the quantization. To perform this, we select a subset of representative levels and evaluate all the possible combinations over that subset (i.e., screening). That evaluation is made regarding the model accuracy and the Normalized Inference Cost Score (NICS), i.e., the outputs. Accuracy is the ratio between the number of correct and the total number of predictions in all classes. The NICS calculation is obtained by comparing the weight bits, total activation bits, and Bits Operations (BOPS) against the reference model, as per the following equation.

$$NICS = 0.5 * \left(\frac{bops}{bops_{baseline}}\right) + 0.5 * \left(\frac{w_{bits}}{w_{bit\_baseline}}\right)$$

Where $bops$ and $bops_{baseline}$ are BOPS of the evaluated (quantized) and the reference (non-quantized) model, respectively. Similarly, w_{bits} and $w_{bits\_baseline}$ are the total bits used by the weights in the evaluated and the reference models, respectively. Note that previous information can help refine the selection of subsets. For instance, [55], [56] showed that an 8-bit quantized Convolutional Neural Network (CNN) model achieves an accuracy close to that of an unquantized model for AMR. Then, a reduced subset of quantization levels can be used (i.e., from 1-bit to 8-bit quantization). Once we obtain the performance and the inference cost for each combination of the subset, we apply Spearman's correlation coefficient (see following equation) to identify which quantized parameter (i.e., input, activations, and weights) has the highest impact on the output, i.e., the dominant parameter. We use Spearman's correlation, where n is the number of observations and D is the variable of interest, since it allows for correlation variables that bear a nonlinear relationship. If, during the screening, a combination that meets the expected trade-off is found, e.g., by creating the Pareto front using the resulting accuracy

and NICS metrics from the quantized models, then we can conclude our search. Otherwise, we can move to the second stage.

$$\rho = 1 - \frac{6\sum_{i=1}^{n} D^2}{n(n^2-1)}$$

Since modern DL models are composed of several layers, if the dominant parameters are activation or weights, we can evaluate in a layered way which layer has the highest effect on the trade-off (stage 2). Notice that the weights are more likely to significantly impact the accuracy and inference cost outputs than activations and inputs since there are more hidden units and connections among them than layers. During the second stage, we measure the degree of quantization per layer required to meet a given trade-off. In addition, notice that if the input is the parameter that most impacts the outputs, then the second stage is the same, but the only layer to alter is the input one. A good starting point is to take the same quantization subset as in stage one. In this stage, we vary the quantization level of a given layer while keeping the quantization level of the non-dominant parameters and the remaining layers the same. Our second stage differs from previous works such as [52]–[54] since they typically apply the same quantization level to all the model's layers. Suppose the trade-off between the model performance metric and the inference cost is met, then we conclude our search by obtaining an architecture in which we have identified which layer of the model has the highest impact. Otherwise, we can continue with the third stage.

So far, we have analyzed the impact of only one layer in the trade-off. However, we may obtain a better model configuration by quantizing different layers using different quantization levels. Using the results from the previous stage, we analyze the data dispersion using the mean, the median, and the main quartiles per layer per variable of interest (i.e., model performance metric and inference cost). At this point, the layer with higher dispersion is the layer that influences the trade-off the most. This allows us to analyze the behavior of each layer, but we still need to determine its quantization level. To answer this question, we must correlate the information using Spearman. Since Spearman's correlation ranges from 1 to −1, we can obtain an equivalent scale for the quantization level. During the search, we identify the quantization level that, in general terms, offers a better trade-off. This quantization level is regarded as the highest Spearman's correlation coefficient. Then, it is possible to obtain the quantization level and the direction, e.g., a 1 as correlation coefficient means that the layer must be quantized at the highest quantization level possible. In contrast, a −1 correlation coefficient means the layer must be quantized with the lowest possible level.

In the last stage (stage four), we can select the level of quantization that every model layer should have. Thus, having Spearman's correlation results per layer, we map the correlation coefficient with the quantization level described above and apply the following equation, where M is the median and D is the variable of interest. Suppose there is more than one variable of interest. In that case, the equation should be applied per variable, and the quantization level of each model layer can be selected as a weighted sum of the quantization levels per variable of interest. In this way, the experimenter can choose which variable of interest to care for the most.

$$quantization_{layer} \begin{cases} 4 + \dfrac{(4 * M) + 4}{2} & if\ \rho \geq 1 - \dfrac{6\sum_{i=1}^{n} D^2}{n(n^2-1)} \\ -4 + \dfrac{(4 * M) + 4}{2} & Otherwise \end{cases}$$

When validating with a concrete set of experiments on a well-known DNN architecture for AMR, the results demonstrate that our methodology finds quantized architectures (red dots in Figure 31) better than the "one-fit-all" approach. The solutions obtained in phases 1 and 2 are also shown. Notice that by varying the weight of the two objective functions α_1 (accuracy) and α_2 (inference cost), different configurations can be obtained that were not found in the initial experimentation analyzed with the Pareto optimum.

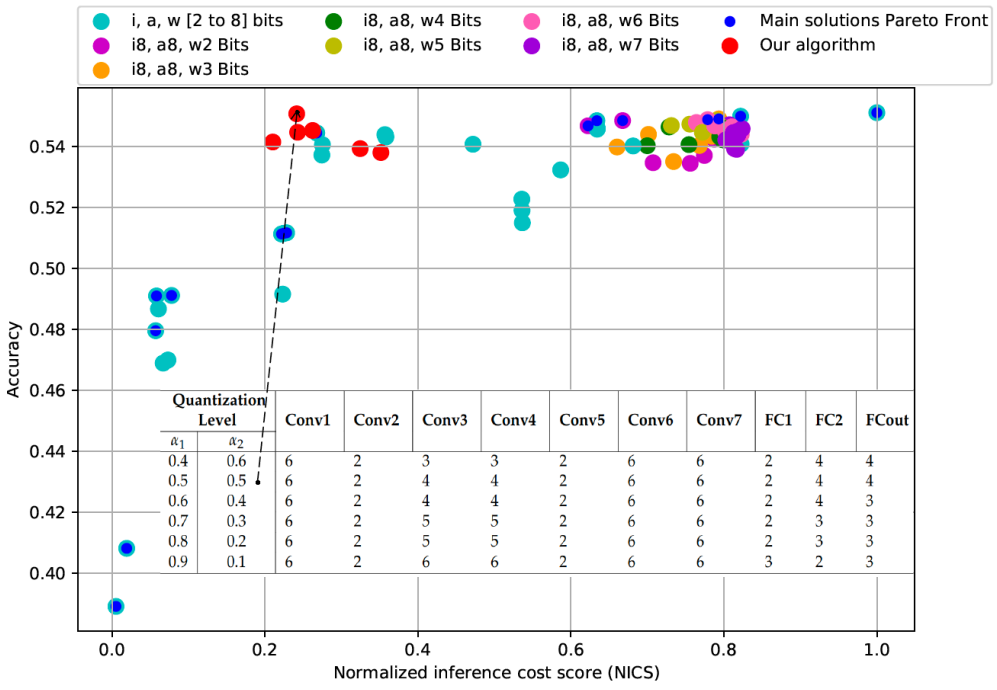


***Figure 31**. Solutions that were obtained using the proposed methodology for a DL model solving the AMR problem.*

If we select the solution that balances the two objectives, Figure 32 shows the same accuracy as the models quantized with 10 and 8 bits while providing the lowest inference cost.

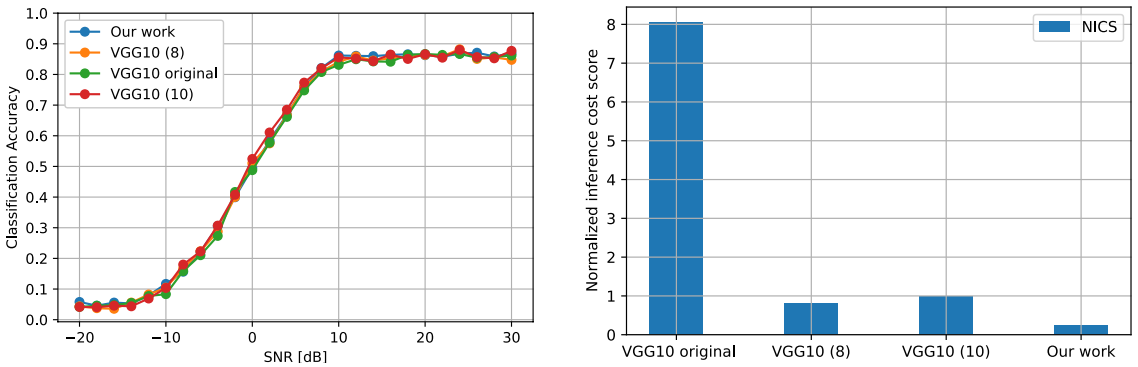


***Figure 32.** Modulation classification accuracy of the original (unquantized) model and the quantized versions with different Signal-to-Noise Ratio(SNR) values (left) and the comparison of the quantized VGG10 1D-CNN model versus the non-quantized model in inference cost (right).*

Limitations and future challenges

Although the proposed methodology is very generic, it has only been tested on one NI problem type of architecture. Therefore, applying this methodology to find the appropriate quantization level using other DNN architectures for AMR and extending it to other NI problems (e.g., traffic classification) is essential. In addition, quantization is not the only method to reduce the model size. Therefore, further experimentation with our methodology with other quantization techniques or in combination with pruning remains to be performed, which could provide an even more significant reduction in the inference cost and validation of the generality of the proposed approach. Finally, performance evaluations of some of the resulting models as a part of a wireless communication system running on an FPGA have to be done to provide further quantitative results of the trade-off between model accuracy and other metrics related to the model size, such as energy consumption and processing speed.

7.2 Limits of AI for NI

One of the cornerstones of the DAEMON project is its critical approach toward AI, intended as complex data-hungry black-box models based on deep learning and as a sliver bullet to solve any task in network management. Following this stance, the project is exploring the limits of AI in the case of the eight NI-

assisted network functionalities targeted in the DoA, so as to identify potential limitations of AI in such practical tasks. At the same time, we are investigating alternative methods that allow to broaden the spectrum of learning and optimization tools that are best suited to concrete networking problems, including classical statistical models, simple ML techniques, optimization tools, or heuristics. Such tools can be employed in stand-alone approaches or jointly in hybrid approaches if the latter are found to work better for the functionality at hand.

In this section, we summarize the results of the activities in the project that aim at responding to the question: "*When should AI be preferred to (or combined with) other approaches in order to maximize the efficiency and performance of NI?*" We therefore provide the insights and conclusions that outcome from the research derived in the context of DAEMON about whether/when AI is the most appropriate solution to network management problems. Next, ***we provide a list of the current outcomes of the research conducted within the DAEMON project and aimed at understanding the limitations of AI/ML solutions by demonstrating that other classes of models are better suited to empower the NI-assisted network functionalities we target***. We link these adaptations to the requirements of the target functionalities proposed in DAEMON. As in the case of the guidelines for a tailored design of AI for NI, the connection between requirements and these new guidelines is bidirectional, as (i) the functional requirements set the constraints that the guidelines fulfill, and (ii) the evaluation of solutions built on the guidelines allows for revealing limitations of the requirements, which shall be updated accordingly.

Table 12 summarizes the seven guidelines produced by the project to date, indicating the requirements they relate to and providing a brief description of their key message. Full details on each guideline are then presented in the remainder of this section. Note that the focus there is on the extrapolation of the design guidelines of AI for NI, which arises from the activities carried out during the first iteration of the DAEMON project. Therefore, when applicable, we also link guidelines to their implementation for some specific NI-assisted functionality that are presented in other deliverables of the project.

Table 12. *Summary of the DAEMON project's guidelines on the limits of AI for NI.*

Guideline for	Requirements	Description	DAEMON related work
Models for Traffic classification	FR-MTERM-006	For unencrypted traffic, in DAEMON we propose the use of simple statistical algorithms for traffic classification of unencrypted data, as we have shown that they perform as well as complex AI/ML approaches. In such situations, the statistical approaches are preferred due to the huge difference in complexity. **Update:** When performing traffic classification at spectral-level packets (i.e., physical layer packets represented as a time series using IQ samples or any other spectrum-level representation), DAEMON proposes to use DL models over statistical ML models, as the former outperforms the latter. Moreover, CNN is more suitable than RNN in terms of input size (CNN can manage larger times-series sequences), accuracy, and faster inference time.	[57], [58]
Wireless Network performance inference	FR-MTERM-006	While ML approaches outperform classic mathematical approaches that rely on simplified assumptions, the former suffers from the limitations on the fixed size of input and scalability. It has been shown that hybrid approaches based on machine learning algorithms that make use of graph theory clearly improve the performance over both standard ML and mathematical solutions.	[25]
Self-learning MANO	FR-SLMANO-000	In auto-scaling of virtual resources, it has been proven that classical control theory approaches outperform RL-based controllers in terms of the trade-off between resource requirements and QoE. It turns out that the flexibility that the RL approach brings incurs the cost of having a lower performance.	[26], [43]

Forecasting in mobile networks	FR-CFORE-000	While there has been extensive research on ML approaches for forecasting, showing that such approaches usually outperform more classical statistical solutions, in DAEMON we have proposed a truly hybrid approach that takes the best from both paradigms, and which improves the results of state-of-the-art predictors. The concept is simple: instead of applying a global normalization of the traffic time series before it is input to the DNN predictor, a dynamic normalization is performed at each time step; the level used for such a dynamic normalization is decided by a statistical model. Both the DNN and the statistical model's parameters are trained through the same gradient descent mechanism. From this and the second point of this table, DAEMON advocates for the use of hybrid solutions that provide synergistic gains.	[28] [59], [60] Not yet published results appearing in D4.3
In-backhaul inference	FR-IBSSI-002 NFR-IBSSI-000 NFR-IBSSI-001	The feasibility of realizing inference in programmable user planes at line rate is a challenging network environment for NI, because of the strong limitations of the programmable switch's hardware. In such applications, highly-elaborated, complex non-interpretable deep learning models for the user-plane tasks analyzed provide a similar performance as much simpler and interpretable tree-based approaches. The DAEMON project advocates the use of Random Forest models instead of other approaches, including those based on deep learning, for in-backhaul inference. Indeed, apart from not achieving a better performance, neuron-based approaches are challenging to implement in resource-constrained programmable switches. **Update:** The DAEMON project advocates the use of hierarchical inference models for in-switch/in-line classification, which is a resource-constrained scenario. Hierarchical inference leads to better accuracy while reducing the required resources.	[16]
Federated learning powered NI functionalities	FR-SLMANO-000 FR-SLMANO-003 FR-AARES-000 FR-AARES-001	While the main question is whether ML should be preferred to non-ML-based approaches for some NI applications, another related question is which ML framework should be considered, which also falls within the questions about the best practices and limits of each of the AI frameworks. For example, in DAEMON we recommend the use of Federated Learning (FL) over centralized or distributed learning for applications that require several intelligent agents acting cooperatively, in cases where the decisions taken at distant parts of the network are intertangled and impact each other. FL allows for a low response time due to the existence of the local module, and a high scalability due to the exchange of limited traffic between FL clients and the FL controller.	[61]
Predictive HARQ	NFR-CAWRS-000 NFR-CAWRS-001 NFR-CAWRS-003	Predictive HARQ is a network application that requires extremely low latency, while maintaining both ultra-high accuracy and low false positive rate. Complex ML-based algorithms fail to provide performance guarantees, and they consume excessive time in the inference task. In DAEMON,	[15]

		we have found clearly identifiable patterns that distinguish decodable and non-decodable code blocks, which can be detected through simple algorithms with minimum computation delay. Hence, in DAEMON we suggest the use of simple statistical or control-theory approaches to implement predictive HARQ and other ultra-low-latency-inference applications.	
Hard constraints (New guideline)	NFR-EAWVNF-005 NFR-EAWVNF-006	Conventional AI models based on neural networks struggle to satisfy hard constraints. The usual approach is to build solutions that guarantee constraint satisfaction only on average. DAEMON advocates the use of Bayesian learning and expansive safe sets to overcome this limitation.	[18] [62]–[64]
Anticipatory decision-making in mobile networks (New guideline)	FR-SLMANO-000 FR-CFORE-001 FR-MTERM-000	Taking anticipatory decisions in network management requires solutions that are able to operate at different time-scales and require the fulfillment of hard constraints. Based on these conditions and characteristics, DAEMON proposes the use of cascaded hybrid methods that include both ML-based elements and optimization-based block, and the use of replicated similar-but-not-equal methods for different time-scales, which update and operate the same function at different timescale and with different accuracy requirements. Based on different activities within the project, DAEMON also advocates for the use of yield management strategies such as overbooking of services, which can be exploited thanks to the bursty and non-stable nature of mobile traffic.	[65], [66] Not yet published results appearing in D4.3

7.2.1 Traffic classification

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- FR-MTERM-006 for Multi-timescale resource allocation.

Initial guidelines about NI for Traffic Classification (TC) have been presented in Section 4.2.1 of D2.2 [1], where we indicated that when dealing with encrypted traffic at packet-level (byte) representation [58], the automatic feature extraction procedure of DL models can help in the TC task, given that the features used in simple ML models based on statistical ML are not enough to properly identify among traffic classes. On the contrary, ***in unencrypted traffic, DL models behave as simple statistical IP/port-based architecture and can be replaced by simpler ML models***.

However, traffic classification is traditionally performed at packet-level (byte) representation. This is based on the assumption that the traffic flows on a wired network under the same network management domain. This assumption limits the capabilities of TC systems in wireless networks since users' traffic on one network domain can be negatively impacted by undetected users' traffic from other network domains or detected ones but with no traffic context in a shared spectrum. To solve this problem, we introduce a novel framework to achieve TC at any layer in the radio network stack [57], which uses a DL-based classifier that can process L1 (physical) layer packets as input, represented as a time series of In-Phase and Quadrature (IQ) samples, and provide as output a label representing the type of traffic that is transported by the L1 packet at a given layer. ***Therefore, DAEMON proposes to perform traffic classification at L1 using Deep Neural networks (DNN) architectures such as CNN and Recurrent Neural Network (RNN), where CNN are preferred as they can manage large time-series data (more than 3K IQ samples) with lower computational complexity compared to RNN. Moreover, it was demonstrated that statical ML models were not suitable for dealing with limited number of features due to the encrypted nature of wireless transmissions.***

Limitations and future challenges

Traffic Classification (TC) systems allow inferring the application that is generating the traffic being analyzed. State-of-the-art TC algorithms are based on Deep Learning (DL) and have outperformed

traditional methods in complex and modern scenarios, even if traffic is encrypted. Nowadays, internet traffic that is generated by regular users is being encrypted for privacy and security reasons. In such cases, DL models are required to outperform the limitations of feature-based statistical ML models. However, other type of networks, e.g., industrial/sensor networks, can still take advantage of simpler models since its traffic can be encrypted in the access link (e.g., to avoid transmitting in plain text) but its gateway can decode it.

With respect to TC at the spectrum level, the proposed models are complex and require a high-end hardware accelerator to run them, making their deployment unfeasible on constraint environments like the edge/far-edge. Further research on energy efficiency and reduction of the model complexity must be carried on allowing them to run in traditional wireless environments. It is important to also recognize that there is a lack of understanding of what the models are learning when using DL models, which requires further research on representation learning and explainability for this kind of classification tasks. Finally, there is a big need for the creation of new datasets (synthetic but even more important with real data) and make them open and available for further benchmarking and validation of the research as done in this research[22].

7.2.2 Inferring wireless networks performance using Graph Neural Networks

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- FR-MTERM-006 for Multi-timescale resource allocation.

The limits of ML models for wireless performance prediction were explored in Section 4.4.2 of D2.2 [1], where it was shown that ***traditional ML models face challenges when learning from structured data represented as graphs, as the relationships among nodes are not captured or must be represented differently***. To overcome this limitation, ***DAEMON commends the use of Graph Neural Networks (GNNs)***. We refer to D2.2 [1] for further details.

Limitations and future challenges

GNNs are promising ML models for solving networking problems, especially because there is a 1:1 mapping of the network infrastructure with the graph definition. However, as shown in [67], they lack generalization capabilities to operate with large graphs. Ideally, we should be able to produce ML models that can be trained in small-size testbeds and make sure that the ML model is able to operate with guarantees in real-size networks.

Moreover, besides contextual and structural information, temporal information (e.g., dynamically changing parameters such as location or channel allocation) can increase the prediction performance of GNNs as shown in [68].

7.2.3 Self-learning MANO – reinforcement learning

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- FR-SLMANO-000 for Self-learning MANO.

For autonomous service scaling (a key component of self-learning MANO), the number of VNFs needs to be scaled according to the work that is offered. The decision to add or remove a VNF often needs to be made based on the observed QoE, e.g., the latency incurred in processing the work. Scaling algorithms can rely on Reinforcement Learning (RL) or on Control Theory (CT). In contrast to the Deep Reinforcement Learning (DRL) approach, which has a neural network at its core with as many parameters as there are synapses, the CT approach has only a few parameters to tune.

As explained in D2.2 [1], it turns out that ***the CT approach outperforms an RL-based controller in terms of the trade-off resource requirements versus QoE*** but needs to be tuned manually. In other words, the flexibility that the RL approach brings comes at the cost of having a lower performance.

In the third project year, we have improved the CT controller in the following way: it reacts in a different way when the KPI is above an upper threshold than when the KPI is below a lower threshold. This doubles the number of parameters (which makes it more cumbersome to tune) but increases the performance drastically. The details of this new contribution, algorithm and results, will be presented in the deliverables D3.3-D4.3 and D5.3, respectively.

[22] https://github.com/miguelhdo/tc_spectrum

Limitations and future challenges

Automatic tuning of the CT algorithm remains challenging, while for tuning of the parameters of an RL algorithm, a well-known technique (relying on the theory of Markov decision processes) exists. If the statistics of the workloads do not change drastically over time, it is worth spending enough computation resources to tune a CT algorithm, while in case there are frequent statistical changes in the workloads one has the rely on the plasticity of an RL based approach.

7.2.4 Forecasting in mobile networks

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- FR-CFORE-000 for Capacity Forecasting NI.

In deliverable D2.2 [1], we introduced a guideline for forecasting in mobile networks. This guideline is related to the current trends in forecasting for anticipatory networking, which lean towards the systematic adoption of models that are purely based on deep learning approaches (see Section 4.2.4 in D2.2 [1] for a detailed description of the different approaches and the corresponding references). However, very recent results from the machine learning community suggest that hybrid engines integrating statistical modelling and DNN can, in fact, substantially outperform pure DNN approaches in time series forecasting tasks [69]. Based on the demonstrated superior performance over pure DNN solutions, ***in DAEMON, we advocate for a hybrid statistical-learning paradigm to the problem of forecasting for Network Intelligence (NI)***. This superiority is also demonstrated over state-of-the-art dedicated DNN-based predictors from the literature. The specific architecture proposed, named Thresholded Exponential Smoothing with Recurrent Neural Network (TES-RNN), is a general-purpose network traffic forecasting technique that can be tailored to perform predictions for different NI functions. The concept behind TES-RNN is simple: instead of applying a global normalization of the traffic time series before it is input to the DNN predictor, a dynamic normalization is performed at each time step; the level used for such a dynamic normalization is decided by a statistical model, whose parameters are optimized jointly with those of the DNN during training. The architecture incorporates Auto-ML mechanisms for the selection of hyperparameters. Further details on the proposed structure can be found in Section 4.2.4. in D2.2 [1]. Overall, by proposing this very first hybrid approach to forecasting for NI, DAEMON paves the way for a different strategy for the design of predictors for mobile network environments. This guideline motivated further studies that led to the guideline presented in Section 7.2.9 for anticipatory decision-making.

Limitations and future challenges

Fully hybrid approaches, where the parameters of the statistical model are jointly trained with the weights of the neural networks, are a novel approach that still has many open questions and challenges. One of the main challenges is understanding the dependency of one model (statistical or learning-based) with the other since the performance one the first may be intertwined with the updates of the other. Achieving a correct convergence is key in such models. Integrating both approaches is expected to be the by-default strategy for many complex problems, since in this way we can take advantage of the strengths of the two approaches. However, we need a better understanding of the fundamental limits of statistical and learning-based models in order to prevent us from suffering the disadvantages of both approaches.

7.2.5 In-backhaul inference

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- FR-IBSSI-002, NFR-IBSSI-000, NFR-IBSSI-001 for In-backhaul Support for Service Intelligence NI.

As part of the project activities, we investigate the feasibility of realizing inference in programmable user planes at line rate. This is a challenging network environment for NI, given the strong limitations of the programmable switch hardware, as detailed in Section 7.1 of D3.1 [5] of DAEMON.

Based on the results of extensive tests with multiple real-world use cases for network traffic classification and anomaly detection, and as already mentioned in Section 4.2.5 of D2.2 [1], ***the DAEMON project advocates the use of Random Forest models instead of other approaches, including those based on deep learning, for in-backhaul inference***. Indeed, we did not identify any significant advantage in relying on complex non-interpretable deep learning models for the user-plane tasks analyzed: simpler approaches based on multiple decision trees achieve an accuracy that is similar or even superior in such tasks. Instead, we found that deep learning models are much more challenging to implement in resource-constrained programmable switches, which dramatically limits their internal complexity (e.g., in

terms of layer depth or number of neurons per layer) and thus an inference potential that is classically largely dependent on their architectural complexity.

Results supporting the guideline above are available in Section 4.7.1 of D5.2 [4] of the DAEMON project.

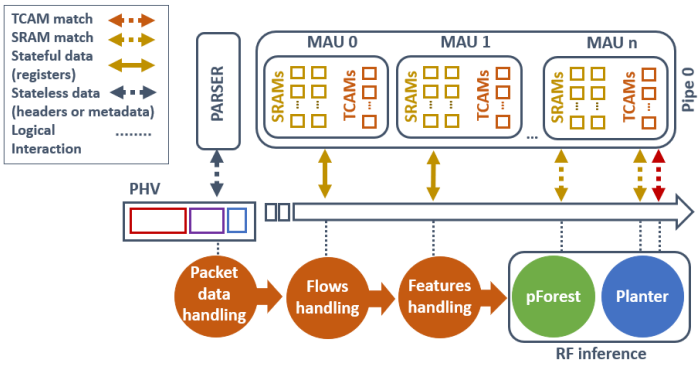


Figure 33. *Proposed framework for in-backhaul inference.*

Furthermore, ***the DAEMON project advocates the use of hierarchical paradigms for in-switch inference.*** In this scenario, the amount of available resources in the switch is one of the limiting dimensions for implementing inference algorithms in-line. Here is where hierarchical paradigms come to the rescue by splitting the overall target task into simpler ones that are themselves easier to handle; then, smaller classifiers can be trained to solve the sub-tasks, collectively yielding a better accuracy while being able to fit within the limited switch capabilities. In this manner, we are able to both improve performance in terms of accuracy but also in terms of resources. The results showcasing the benefit of applying this guideline will be reported in D3.3 and D5.3. The general hierarchical framework is illustrated in Figure 34.

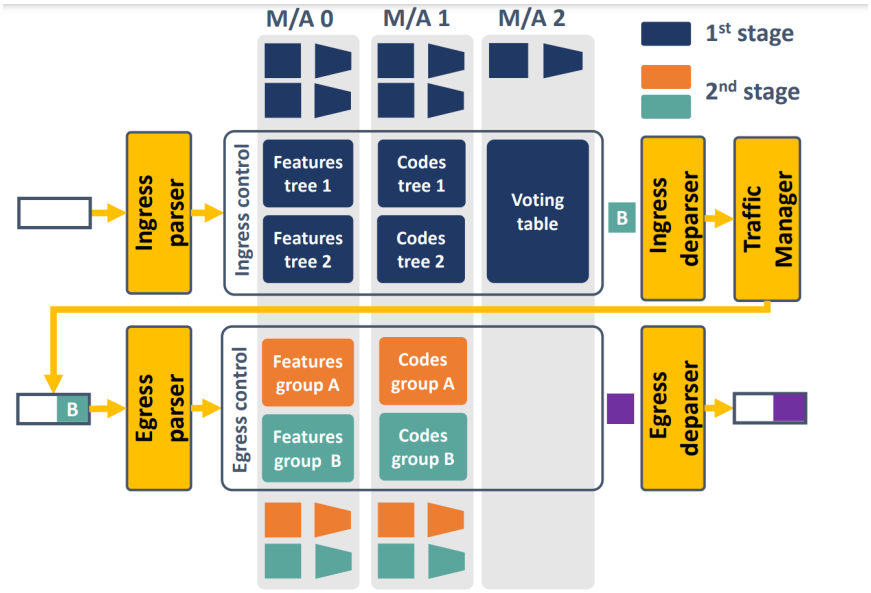


Figure 34. *Proposed framework for hierarchical in-backhaul inference.*

Limitations and future challenges

In-switch inference is a very demanding scenario, due to strongly limited capabilities of the hardware and the stringent delays to be satisfied to maintain line rates. The guidelines proposed by DAEMON prove that it is feasible to implement this approach while improving the performance of the functions and satisfying all the required constraints. Yet, these initial results are just a first step towards comprehending the full potential of in-backhaul inference. In the future, the community would need to extend the analysis to other applications, as well as measuring how the inclusion of inference in the switch impacts any other simultaneous task that is run in the switch. This functionality requires resource- and hardware-aware inference algorithms, and thus is very related to the fundamental limits of resource-limited AI; because of that we will require tailored algorithms that are able to extract the most from the few available resources.

7.2.6 Federated learning powered NI functionalities

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- FR-SLMANO-000 and FR-SLMANO-003 for Federated Learning powered Controller.
- FR-AARES-000 and FR-AARES-001 for anomaly detection.

Most of the advances in ML approaches are based on the idea of a single intelligent agent that computes and executes the learning process. When we consider multi-agent environments, we can consider (i) the *distributed* version of the single-agent approach, where each agent acts independently and attempts to learn a selfish (or common) goal at the same time as all the other agents, or (ii) Federated Learning (FL))[70]; ***the FL approach can be used for reasons of scalability and data protection***. Scalability is crucial for many DAEMON functionalities and applications. In the FL approach, the distributed knowledge bases contain local information regarding the performance of the model. Certain information is communicated to a centralized controller of the FL model, and such controller will be tuning the algorithm in accordance with the FL paradigm.

In general, ***FL is the recommended solution over RL and other centralized and distributed choices,*** since it provides the following advantages:

- Low fault tolerance in anomaly detection process due to FL enhancement.
- Low response time due to the existence of the local anomaly detection module.
- High scalability due to the exchange of limited traffic between FL clients and FL controller.
- Access to local database for the execution of anomaly detection process, while keeping a small central database in the FL controller.
- Low network traffic exchanged between FL clients and FL controller.

We refer to D2.2 [1] for a detailed explanation of the advantages of FL.

Limitations and future challenges

Federated learning is known to have several challenges that may prevent a fast development of FL-based solutions. It is crucial to ensure reliability and synchronization, in the sense that each agent must be treated in a similar way, and the performance may be impacted by convergence issues. These challenges affect FL in general and solutions from other fields can be implemented to realize FL-based NI functionalities.

7.2.7 Predictive HARQ

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- NFR-CAWRS-000, NFR-CAWRS-001, NFR-CAWRS-003 for Compute-aware Radio Scheduling NI.

In D2.2 [1], we provided a detailed analysis of the guidelines for implementing Hybrid Automatic Repeat Query (HARQ), which is an essential operation at the physical layer of a 5G Distributed Unit. Predictive HARQ enables the inference of the decodability of Uplink data essentially using feedback from the decoder, minimizing situations where uplink subframes are discarded because they cannot be processed in time. The produced prediction allows the subsequent tasks to be performed without waiting for the decoding process to be finished. Due to the extremely fast-inference constraints of this task, ***complex data-driven models based on neural networks are not necessarily the best tool for HARQ operations.*** This was illustrated in D2.2 [1], where results evidenced clearly identifiable patterns between decodable and non-decodable code blocks, which could be detected via simple, non-ML approaches. Indeed, to effectively take advantage of the decodability forecasting, the inference time of the proposed solution must be extremely low as stated in the design constraint NFR-CAWRS-001. With the information observed in the previous results and the time requirements imposed by design, ***rule- or threshold-based algorithms will better fit the design of predictive HARQ mechanisms rather than ML-based techniques.***

Limitations and future challenges

While some tasks may take advantage of simple patterns of clear correlations, some others may lack such property, thus making impossible to obtain at the same time the required extremely high reliability and the equally exacting latency constraints. Further research is required on the fundamental limits of the trade-off between speed and accuracy for a plethora of diverse applications.

7.2.8 Satisfaction of hard constraints

A number of NI problems require satisfying hard network constraints, e.g., minimizing energy consumption while ensuring minimum QoS metrics. Though these constraints may be relaxed, e.g., ensure them only on average, there are some other problems that cannot afford to violate such constraints even while learning. Problems of this type include, but are not limited to, reliability network problems or physical system limitations.

To address this type of problems, ***DAEMON proposes using a non-parametric Bayesian learning approach that can address these scenarios with the concept of expansive safe action sets.*** This approach models the cost to minimize and the hard constraints to satisfy as samples of Gaussian Processes (GPs) over the joint context-control space. ***This non-parametric estimator deals with the system non-linearities and correlations, and quantifies the function estimation uncertainty, effectively addressing the exploration vs. exploitation trade-off***. In the following, we detail the key components of the approach, which can then be applied to many different problems that face the same hard constraints.

Function approximator. In order to estimate the cost and constraint functions we propose using GPs, which consist of a collection of random variables that follow joint Gaussian distributions. Let $z \in \mathcal{Z} = \mathcal{C} \times \mathcal{X}$ denote a context-control pair. We model each of the unknown functions as a sample from $GP(\mu(z), k(z, z'))$, where $\mu(z)$ is its mean function and $k(z, z')$ denotes its kernel or covariance function. W.l.o.g., we assume $\mu := 0$ and $k(z, z') < 1$, which we refer to as the *prior distribution*, not conditioned on data. Given the prior distribution and a set of observations, the *posterior distribution* can be computed using closed-form formulas.

The sets of observations of the cost and constraint functions at points $Z_T = [z_1, \dots, z_T]$ up to time period T are denoted by $y_T^{(0)} = [u_1, \dots, u_T]$, $y_T^{(1)} = [d_1, \dots, d_T]$, $y_T^{(2)} = [\rho_1, \dots, \rho_T]$, respectively, assuming i.i.d. Gaussian noise $\sim N(0, \zeta_{(i)}^2)$. The posterior distribution of these functions follows a GP distribution with mean $\mu_T^{(i)}(z)$ and covariance $k_T^{(i)}(z, z')$:

$$\mu_T^{(i)}(z) = k_T^{(i)}(z)^\top (K_T^{(i)} + \zeta_{(i)}^2 I_T)^{-1} y_T^{(i)}$$

$$k_T^{(i)}(z, z') = k^{(i)}(z, z') - k_T^{(i)}(z)^\top (K_T^{(i)} + \zeta_{(i)}^2 I_T)^{-1} k_T^{(i)}(z')$$

where $k_T^{(i)}(z) = [k^{(i)}(z_1, z), \dots, k^{(i)}(z_T, z)]^\top$, $K_T^{(i)}(z)$ is a kernel matrix defined as $[k^{(i)}(z, z')]_{z,z' \in Z_T}$, I_T is the T-dimension identity matrix, and $\zeta_{(i)}^2$ the variance of noise in observations. Index i denotes the objective function, with $i = 0$ for the cost function, $i = 1$ for the delay, and $i = 2$ for the mAP. The distribution of unobserved values of $z \in \mathcal{Z}$ for function i is computed from the prior distribution, vector Z_T and the observed values $y_T^{(i)}$.

Kernel selection. The kernel function shapes the GP's prior and posterior distributions having an impact on the learning rate. It encodes the correlation of the function values for every pair of context-control points. That is, the kernel characterizes the *smoothness* of the functions.

The properties of the kernel function should be thoroughly selected for each specific application and the underlying functions that must be learned. Two common properties for these functions are *stationarity* and *anisotropicity*. This means that the kernel $k(z, z')$ is invariant to translations in $\mathcal{Z}$ but not invariant to rotations in $\mathcal{Z}$. The smoothness of the kernel for each dimension of function i is encoded in the length-scale vector $\mathcal{L}^{(i)} = [l_1^{(i)}, \dots, l_N^{(i)}]$, where N indicates the number of dimensions of $\mathcal{Z}$. The distance between two points based on the length-scale vector is given by:

$$d^{(i)}(z, z') = \sqrt{(z - z')^\top (L^{(i)})^{-2} (z - z')},$$

where $L^{(i)} = \mathrm{diag}(\mathcal{L}^{(i)})$ is a diagonal matrix of the length-scale vector. In order to satisfy the properties stated above, we propose a Matérn kernel on its anisotropic version. Moreover, following standard practice, we propose to particularize it with parameter $\nu = \frac{3}{2}$, indicating that the function is at least once differentiable. Thus, the expression of the kernel can be particularized as follows:

$$k^{(i)}(z, z') = (1 + \sqrt{3} d^{(i)}(z, z')) \exp(-\sqrt{3} d^{(i)}(z, z')).$$

Note that although we propose using the same kernel for all the functions (cost and constraints), their hyperparameters will vary depending on its shape. In fact, the hyperparameters $\mathcal{L}^{(i)}$ and noise variance $\zeta_{(i)}^2$ should be optimized for each function i before running the algorithm by maximizing the likelihood estimation over prior data. During execution, the hyperparameters shall remain constant. This is because when the hyperparameters are optimized using newly acquired data, it is not guaranteed that the GP's confidence interval will cover the true function within, causing the optimization to fall into poor local optima.

Safe set. It is crucial to identify first which controls or actions satisfy the constraints, which, however, depends also on the context. We define the *safe set* as the set of policies that satisfy all the constraints for a given context c:

$$S(c) = \{x \in \mathcal{X} \mid d(c,x) \leq d^{max} \wedge \rho(x) \geq \rho^{min}\}$$

Unfortunately, the computation of the safe set is very challenging for several reasons. Firstly, the observations of the performance indicators are noisy due to the stochastic nature of the system. And secondly, the number of available controls $|\mathcal{X}|$ is usually very large in practice, making it unfeasible to explore all controls for all possible contexts. For that reason, we use the GPs to compute an estimation of the safe set:

$$S_t = \{x \in \mathcal{X} \mid \mu_{t-1}^{(1)}(c_t,x) + \beta\sigma_{t-1}^{(1)}(c_t,x) \leq d^{max}$$
$$\wedge\ \mu_{t-1}^{(2)}(c_t,x) - \beta\sigma_{t-1}^{(2)}(c_t,x) \geq \rho^{min}\}$$

where $\left(\sigma_t^{(i)}(z)\right)^2 = k_t^{(i)}(z,z)$ and β is a weighting parameter. Note that at each time period t the point z_t is observed and the vectors Z_t and $y_t^{(i)} \forall i$ are updated consequently. Due to their correlation, the posterior distribution of points near z_t will vary having an impact on the controls that will be included in the safe set in $t+1$.

Acquisition function. It indicates, at each time period t, which control x_t shall be used in the system given context c_t. This task is crucial for the convergence of the algorithm and needs to interleave an exploration process in order to expand the safe set while seeking a safe control with high performance. Many previous works have proposed acquisition functions for constrained Bayesian optimization, but they do not consider contexts. We hence propose the contextual lower confidence bound as an acquisition function, but *constrained to the safe set*:

$$x_t = \underset{x \in S_t}{\operatorname{argmin}}\ \mu_{t-1}^{(0)}(c_t,x) - \sqrt{\beta}\sigma_{t-1}^{(0)}(c_t,x).$$

Limitations and future challenges

As previously mentioned, this approach requires a profound knowledge of the scenario that is being managed. In particular, kernel functions should be thoroughly selected for each specific application and the underlying functions that have to be learned. This aspect is a limitation in as much as it limits the generalization and applicability to a broad set of problems, and connect with the guidelines on tailored AI design presented in Section 7.1, and in particular with 7.1.2 and 7.1.3. One solution, as mentioned in Section 7.1.3, would be to include Auto-Machine Learning algorithms that are able to search on the space of kernel functions and learn the best one for each application (or group of applications).

Another limitation is the inference time of Bayesian learning models, which is usually large. Although the learning rate of these models is extremely high, the rather high inference time incurred in computing Bayesian updates makes this approach unsuitable for very fast control loops (below second-level granularity.

7.2.9 Anticipatory decision-making in mobile networks

The NI design guidelines set forth next are relevant to the following requirements of DAEMON, which were presented in Section 4 of D2.1, updated in Section 2 of D2.2 [1], and which are reported in full in Appendix A of this same document:

- FR-SLMANO-000 for self-learning MANO.
- FR-CFORE-001 for capacity forecasting.
- FR-MTERM-000 for Multi-timescale Edge Resource Management.

One of the main purposes of the development of intelligence is to be able to take anticipatory decisions based on accurate future forecasts that allow the network operator to act before the change or problem appears. This decision-making often covers different time-scales, such that the orchestration, management, and coarse resource allocation can be updated with long interval periods, while the same decision is refined in a much smaller timescale to leverage fresher data and system updates. This requires of *duplicated* algorithms that act in a similar way but at different time-scales, with probably different parameters and complexity. Furthermore, these actions usually require the fulfillment of hard constraints, as stated in the previous Section 7.2.8. Based on these conditions and characteristics, ***DAEMON proposes the use of cascaded hybrid methods that include both ML-based elements and optimization-based blocks***, so as to better handle the hard constraints while exploiting all the advantages of ML methods for regression and forecasting, and ***the use of quasi-similar systems for different time-scales***, which update and operate the same function at different timescale and with different accuracy requirements.

This guideline has been implemented for admission control and resource allocation for network slicing in a solution that will be fully described in deliverables D3.3 and D5.3. Motivated by this application, ***DAEMON also advocates for the use of yield management strategies such as the overbooking of services***, which can be exploited thanks to the bursty and non-stable nature of mobile traffic.

In this activity, we provide a first assessment of overbooking gains in the presence of real-world demands generated by multiple service providers, as measured in a metropolitan-scale production network. We investigate advantages for the mobile network operator in terms of net profit along diverse dimensions that include the resource orchestration flexibility, the cost of allocated resources to slices, or the over-dimensioning strategy of the operator.

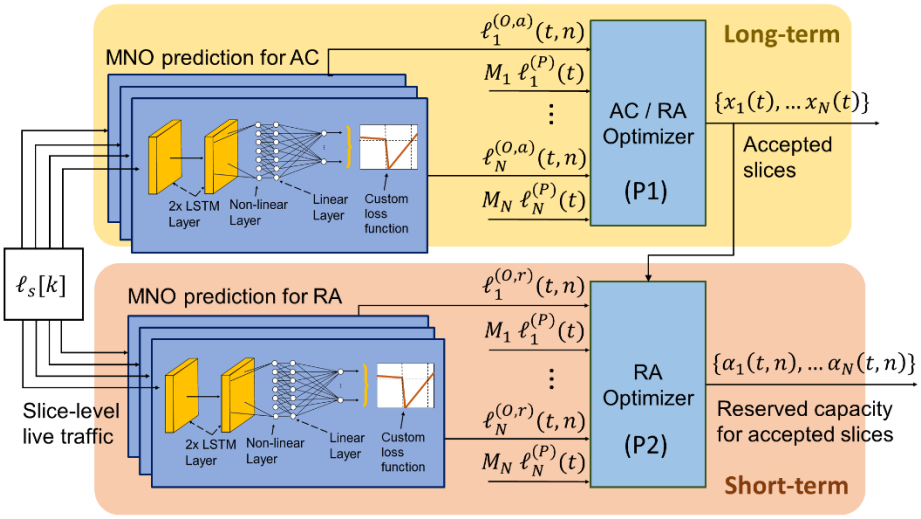


Figure 35. *Proposed framework for hierarchical hybrid anticipatory MANO decisions.*

The proposed architecture is shown in Figure 35, where we can observe how there are operations at two different timescales: (i) First, on top, the Admission Control (AC) is performed in a long-term timescale; (ii) once the AC decision is decided for the whole long-term interval (i.e., which slices are accepted and which ones are rejected), the Resource Allocation (RA) block below is enacted at a much shorter timescale, and receives the AC decisions as part of the state of the system. Each one of the two block (AC and RA) are composed of a hybrid structure tailored to NI: (i) the input data is first introduced in a deep learning block that predicts the future capacity required to serve the traffic of each slice *without violating the SLA* (for that, it makes use of an expert-defined loss function, as recommended by the guideline in Section 7.1.2). Then, this capacity prediction is used as input for a knapsack-type optimization problem that decides the specific resources allocated to each slice. This architecture allows us to break the complex problem into smaller pieces, easing the operation and understanding of the proposed solution.

Furthermore, the use of overbooking strategies for network slicing resource allocation is illustrated in Figure 36, where we can see the intervals of time in which the overbooking strategy provides increased benefit for the operator due to a higher number of accepted slices in green, and the benefit from reducing the amount of resources reserved in blue.

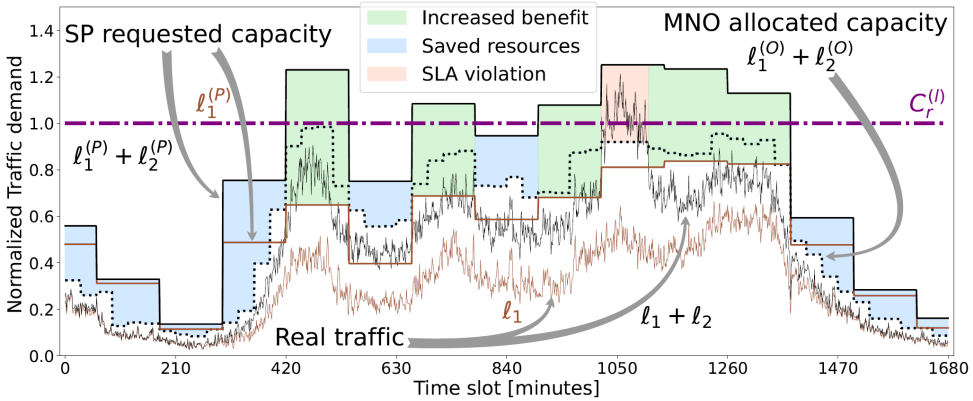


Figure 36. *Illustration of the overbooking concept proposed for network slicing resource allocation.*

Limitations and future challenges

Dealing with anticipatory decisions always implies a certain level of uncertainty. However, this uncertainty and the corresponding risks and possible loss of performance can be compensated with the use of hybrid, well-defined algorithms and the joint operation at different timescales for coarser and finer decisions. Yet, the anticipatory decisions shall measure and control the confidence intervals required to satisfy the imposed conditions. For example, in Figure 36, we highlight red the interval of time in which an accepted slice cannot be served because the overbooking approach is too aggressive and there are not enough resources to serve all the accepted services. Thus, fully characterizing the accuracy and distribution of the forecast is crucial to leverage the maximum potential of the proposed architectures.

8 Conclusions

This deliverable serves as a crucial link between the second and third iterations of the DAEMON project, providing the necessary groundwork for the final phase of the project. The document encompasses various key aspects of the project, starting with the final updates on the functional and non-functional requirements for the eight NI-assisted functionalities, includes the evaluation of the risks associated with meeting these requirements, and provides their current completion status. It also outlines specific actions required for the successful finalization of unresolved requirements, indicating the corresponding deliverable for presenting the results.

Furthermore, the document presents the final updates of the Network Intelligence Plane (NIP), which has undergone significant development. The NIP now serves as a unified framework, incorporating the operational hierarchy and orchestration of NI components, as well as the N-MAPE-K representation. This progress aligns with the vision previously described in D2.2 [1]. Moreover, we motivated that the NIP has evolved towards a NI Stratum, which typically denotes a collection of elements that span various network domains. Considering that network intelligence components are distributed across multiple domains such as access, core, infrastructure, management, and orchestration, it was only natural to adopt this terminology in line with 3GPP standards. Moreover, this approach also moved the NIP design from a purely separate plane to a more orthogonal approach where NIFs and NISs can effectively be integrated into the traditional planes (data, control and management) for an easy adoption in the industry.

The document thoroughly identifies and discusses the specific needs and challenges that NI algorithms pose to the NIP, particularly in terms of NI management procedures at the NIO level. It outlines the necessary functionalities that the NIO should provide to address these needs and highlights their integration within the overall architecture. Additionally, the document delves into the interfaces required for communication between NIP components and external entities, such as the RAN controller and the 5G Core systems. These interfaces enable the design of procedures that address the introduced needs and challenges.

A comprehensive literature review on integrating machine learning and NI in mobile network management is presented, showcasing the unique contributions of the DAEMON project and highlighting key trends in current research. The findings from this analysis further support the final updates to the project guidelines, which aim to achieve a pragmatic design of NI. These guidelines focus on two main directions: designing NI tailored to the needs of B5G network management, orchestration, and control, and emphasizing the utilization of more traditional, simpler, or interpretable models to avoid overburdening the system with data-heavy models.

The content of this document successfully provides the final version of the DAEMON framework and toolset for NI, which continues to provide the foundation for the subsequent stages of the third and final iteration. It will guide the updated design implementation of NI-assisted functionalities, ensuring the fulfillment of all requirements, verifying performance against the project's Key Performance Indicators (KPIs), and delivering a final version of the NI functionalities that aligns with the detailed architecture, interfaces, and NIP procedures presented. By following this roadmap, the DAEMON project aims to achieve its objectives and contribute to the advancement of NI in mobile network management.

9 References

[1] A. Bazco-Nogueras *et al.*, "DAEMON Deliverable 2.2: Initial DAEMON Network Intelligence framework and toolsets." Zenodo, Aug. 2022. doi: 10.5281/zenodo.6970839.

[2] A. Garcia-Saavedra *et al.*, "DAEMON Deliverable 3.2: Refined design of real- time control and VNF intelligence mechanisms." Zenodo, Nov. 2022. doi: 10.5281/zenodo.7525876.

[3] L. Fuentes *et al.*, "DAEMON Deliverable 4.2: Refined design of intelligent orchestration and management mechanisms." Zenodo, Nov. 2022. doi: 10.5281/zenodo.7573188.

[4] M. Fiore *et al.*, "DAEMON Deliverable 5.2: Report on evaluation results and initial proof-of-concept demonstrations." Zenodo, Mar. 2023. doi: 10.5281/zenodo.7818319.

[5] M. Gramaglia *et al.*, "DAEMON Deliverable 3.1: Initial design of real- time control and VNF intelligence mechanisms." Zenodo, Nov. 2021. doi: 10.5281/zenodo.5745433.

[6] G. Iosifidis *et al.*, "DAEMON Deliverable 4.1: Initial design of intelligent orchestration and management mechanisms." Zenodo, Nov. 2021. doi: 10.5281/zenodo.5745456.

[7] M. Gucciardo *et al.*, "DAEMON Deliverable 5.1 Preliminary evaluation results and plan for proof-of-concept demonstrations." Zenodo, Jul. 2022. doi: 10.5281/zenodo.6801832.

[8] I. Paez *et al.*, "DAEMON Deliverable 2.1: Initial report on requirements analysis and state-of-the-art frameworks and toolsets." Zenodo, Jun. 2021. doi: 10.5281/zenodo.5060979.

[9] D.-J. Munoz, M. Pinto, and L. Fuentes, "Detecting feature influences to quality attributes in large and partially measured spaces using smart sampling and dynamic learning," *Knowl Based Syst*, vol. 270, p. 110558, 2023, doi: https://doi.org/10.1016/j.knosys.2023.110558.

[10] L. E. Chatzieleftheriou *et al.*, "Orchestration Procedures for the Network Intelligence Stratum in 6G Networks," in *Accepted for Publication at 2023 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2023.

[11] M. K. Bahare *et al.*, "The 6G Architecture Landscape - European perspective." Zenodo, Feb. 2023. doi: 10.5281/zenodo.7313232.

[12] M. Camelo *et al.*, "DAEMON: A Network Intelligence Plane for 6G Networks," in *2022 IEEE Globecom Workshops (GC Wkshps)*, 2022, pp. 1341–1346.

[13] M. Gramaglia *et al.*, "Network intelligence for virtualized ran orchestration: The daemon approach," in *2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2022, pp. 482–487.

[14] IBM, "An architectural blueprint for autonomic computing," *IBM White Paper*, vol. 31, no. 2006, pp. 1–6, 2006, [Online]. Available: https://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf

[15] G. Garcia-Aviles, A. Garcia-Saavedra, M. Gramaglia, X. Costa-Perez, P. Serrano, and A. Banchs, "Nuberu: Reliable RAN virtualization in shared platforms," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 749–761.

[16] A. T.-J. Akem, B. Bütün, M. Gucciardo, and M. Fiore, "Henna: hierarchical machine learning inference in programmable switches," in *Proceedings of the 1st International Workshop on Native Network Intelligence*, 2022, pp. 1–7.

[17] J. A. Ayala-Romero, A. Garcia-Saavedra, M. Gramaglia, X. Costa-Perez, A. Banchs, and J. J. Alcaraz, "vrain: Deep learning based orchestration for computing and radio resources in vrans," *IEEE Trans Mob Comput*, vol. 21, no. 7, pp. 2652–2670, 2020.

[18] J. A. Ayala-Romero, A. Garcia-Saavedra, X. Costa-Perez, and G. Iosifidis, "Orchestrating energy-efficient vrans: Bayesian learning and experimental results," *IEEE Trans Mob Comput*, 2021.

[19] R. Singh, C. Hasan, X. Foukas, M. Fiore, M. K. Marina, and Y. Wang, "Energy-efficient orchestration of metro-scale 5g radio access networks," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[20] M. Kalntis and G. Iosifidis, "Energy-Aware Scheduling of Virtualized Base Stations in O-RAN with Online Learning," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*, 2022, pp. 6048–6054.

[21] N. Slamnik-Kriještorac, M. C. Botero, L. Cominardi, S. Latré, and J. M. Marquez-Barja, "An ML-driven framework for edge orchestration in a vehicular NFV MANO environment," in *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, 2023, pp. 728–733.

[22] S. Tripathi, C. Puligheddu, S. Pramanik, A. Garcia-Saavedra, and C. F. Chiasserini, "VERA: Resource Orchestration for Virtualized Services at the Edge," in *ICC 2022-IEEE International Conference on Communications*, 2022, pp. 1641–1646.

[23] J.-B. Monteil, G. Iosifidis, and L. DaSilva, "No-regret slice reservation algorithms," in *ICC 2021-IEEE International Conference on Communications*, 2021, pp. 1–7.

[24] M. Rossanese, P. Mursia, A. Garcia-Saavedra, V. Sciancalepore, A. Asadi, and X. Costa-Perez, "Designing, building, and characterizing RF switch-based reconfigurable intelligent surfaces," in *Proceedings of the 16th ACM Workshop on Wireless Network Testbeds, Experimental evaluation & CHaracterization*, 2022, pp. 69–76.

[25] P. Soto *et al.*, "ATARI: A graph convolutional neural network approach for performance prediction in next-generation WLANs," *Sensors*, vol. 21, no. 13, p. 4321, 2021.

[26] P. Soto *et al.*, "Towards autonomous VNF auto-scaling using deep reinforcement learning," in *2021 Eighth International Conference on Software Defined Systems (SDS)*, 2021, pp. 1–8.

[27] A. Cañete, K. Djemame, M. Amor, L. Fuentes, and A. Aljulayfi, "A proactive energy-aware auto-scaling solution for edge-based infrastructures," in *2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)*, 2022, pp. 240–247.

[28] L. Lo Schiavo, M. Fiore, M. Gramaglia, A. Banchs, and X. Costa-Perez, "Forecasting for network management with joint statistical modelling and machine learning," in *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2022, pp. 60–69.

[29] W. Xia *et al.*, "Generative neural network channel modeling for millimeter-wave UAV communication," *IEEE Trans Wirel Commun*, vol. 21, no. 11, pp. 9417–9431, 2022.

[30] D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine learning operations (mlops): Overview, definition, and architecture," *IEEE Access*, vol. 11, pp. 31866–31879, 2023, doi: 10.1109/ACCESS.2023.3262138.

[31] ETSI, "Network Functions Virtualisation (NFV); Management and Orchestration," 2022. [Online]. Available: https://www.etsi.org

[32] O-RAN Alliance, "O-RAN Working Group 1 Slicing Architecture," 2023.

[33] O-RAN Alliance, "O-RAN Working Group 2 AI/ML workflow description and requirements," Oct. 2020. Accessed: Jun. 09, 2023. [Online]. Available: https://wiki.lfaidata.foundation/download/attachments/24281098/O-RAN.WG2.AIML-v01.02.02%20%282%29.docx?version=1&modificationDate=1609811670000&api=v2

[34] F. Jentzsch, Y. Umuroglu, A. Pappalardo, M. Blott, and M. Platzner, "RadioML Meets FINN: Enabling Future RF Applications With FPGA Streaming Architectures," *IEEE Micro*, vol. 42, no. 6, pp. 125–133, 2022.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[36] C. Shih-Huan Hsu, D. De Vleeschauwer, and C. Papagianni, "SLAs Decomposition for Network Slicing: A Deep Neural Network Approach," Zenodo, Jun. 2022. doi: 10.5281/zenodo.6685244.

[37] A. Collet, A. Banchs, and M. Fiore, "Lossleap: Learning to predict for intent-based networking," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*, 2022, pp. 2138–2147.

[38] A. Collet, A. Bazco-Nogueras, A. Banchs, M. Fiore, and others, "AutoManager: a Meta-Learning Model for Network Management from Intertwined Forecasts," in *IEEE International Conference on Computer Communications*, 2023.

[39] M. Camelo *et al.*, "Requirements and Specifications for the Orchestration of Network Intelligence in 6G," in *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, pp. 1–9.

[40] J. A. Ayala-Romero, A. Garcia-Saavedra, X. Costa-Perez, and G. Iosifidis, "Bayesian online learning for energy-aware resource orchestration in virtualized rans," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[41] G. Baldoni, J. Loudet, L. Cominardi, A. Corsaro, and Y. He, "Zenoh-based Dataflow Framework for Autonomous Vehicles," in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2021, pp. 555–560.

[42] F. Giarré, L. Cominardi, and P. Casari, "Realizing Flat Multi-Zone Multi-Controller Software-Defined Networks using Zenoh," in *2022 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2022, pp. 45–51.

[43] P. Soto *et al.*, "Network Intelligence for NFV Scaling in Closed-Loop Architectures," *IEEE Communications Magazine*, vol. 61, no. 6, pp. 66–72, 2023, doi: 10.1109/MCOM.001.2200529.

[44] N. Apostolakis, L. E. Chatzieleftheriou, D. Bega, M. Gramaglia, and A. Banchs, "Digital Twins for Next-Generation Mobile Networks: Applications and Solutions," *IEEE Communications Magazine*, pp. 1–7, 2023, doi: 10.1109/MCOM.001.2200854.

[45] V. Valls, G. Iosifidis, and L. Tassiulas, "Birkhoff's decomposition revisited: Sparse scheduling for high-speed circuit switches," *IEEE/ACM Transactions on Networking*, vol. 29, no. 6, pp. 2399–2412, 2021.

[46] S. Moghadas Gholian *et al.*, "Spotting Deep Neural Network Vulnerabilities in Mobile Traffic Forecasting with an Explainable AI Lens," in *IEEE International Conference on Computer Communications*, 2023.

[47] D. Góez, P. Soto, S. Latré, N. Gaviria, and M. Camelo, "A Methodology to Design Quantized Deep Neural Networks for Automatic Modulation Recognition," *Algorithms*, vol. 15, no. 12, p. 441, 2022.

[48] L. Buşoniu, R. Babuška, and B. De Schutter, "Multi-agent reinforcement learning: An overview," *Innovations in multi-agent systems and applications-1*, pp. 183–221, 2010.

[49] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," *Adv Neural Inf Process Syst*, vol. 29, 2016.

[50] K. Cyras *et al.*, "Machine reasoning explainability," *arXiv preprint arXiv:2009.00418*, 2020.

[51] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4820–4828.

[52] S. Budgett and P. de Waard, "Quantized neural networks for modulation recognition," in *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications IV*, 2022, pp. 397–412.

[53] M. Blott *et al.*, "FINN-R: An end-to-end deep-learning framework for fast exploration of quantized neural networks," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 11, no. 3, pp. 1–23, 2018.

[54] S. Kumar, R. Mahapatra, and A. Singh, "Automatic modulation recognition: An FPGA implementation," *IEEE Communications Letters*, vol. 26, no. 9, pp. 2062–2066, 2022.

[55] Q. Ducasse, P. Cotret, L. Lagadec, and R. Stewart, "Benchmarking Quantized Neural Networks on FPGAs with FINN," *arXiv preprint arXiv:2102.01341*, 2021.

[56] P. Bacchus, R. Stewart, and E. Komendantskaya, "Accuracy, training time and hardware efficiency trade-offs for quantized neural networks on FPGAs," in *International symposium on applied reconfigurable computing*, 2020, pp. 121–135.

[57] M. Camelo, P. Soto, and S. Latré, "A general approach for traffic classification in wireless networks using deep learning," *IEEE Transactions on Network and Service Management*, 2021.

[58] K. Ismailaj, M. Camelo, and S. Latré, "When deep learning may not be the right tool for traffic classification," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2021, pp. 884–889.

[59] J.-B. Monteil, G. Iosifidis, and L. A. DaSilva, "Learning-based Reservation of Virtualized Network Resources," *IEEE Transactions on Network and Service Management*, vol. 19, no. 3, pp. 2001–2016, 2022.

[60] C. Fiandrino, G. Attanasio, M. Fiore, and J. Widmer, "Traffic-driven sounding reference signal resource allocation in (beyond) 5G networks," in *2021 18th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, 2021, pp. 1–9.

[61] A. Pentelas, D. De Vleeschauwer, C.-Y. Chang, K. De Schepper, and P. Papadimitriou, "Deep Multi-Agent Reinforcement Learning with Minimal Cross-Agent Communication for SFC Partitioning," *IEEE Access*, 2023.

[62] J. A. Ayala-Romero, A. Garcia-Saavedra, X. Costa-Pérez, and G. Iosifidis, "EdgeBOL: A Bayesian Learning Approach for the Joint Orchestration of vRANs and Mobile Edge AI," *IEEE/ACM Transactions on Networking*, 2023.

[63] J. A. Ayala-Romero, A. Garcia-Saavedra, X. Costa-Pérez, and G. Iosifidis, "EdgeBOL: Automating energy-savings for mobile edge AI," in *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, 2021, pp. 397–410.

[64] A. Galanopoulos, J. A. Ayala-Romero, D. J. Leith, and G. Iosifidis, "AutoML for video analytics with edge computing," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[65] N. Mhaisen, G. Iosifidis, and D. Leith, "Online Caching with Optimistic Learning," in *2022 IFIP Networking Conference (IFIP Networking)*, 2022, pp. 1–9.

[66] N. Mhaisen, A. Sinha, G. Paschos, and G. Iosifidis, "Optimistic No-regret Algorithms for Discrete Caching," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 6, no. 3, pp. 1–28, 2022.

[67] M. Ferriol-Galmés, J. Suárez-Varela, K. Rusek, P. Barlet-Ros, and A. Cabellos-Aparicio, "Scaling graph-based deep learning models to larger networks," *arXiv preprint arXiv:2110.01261*, 2021.

[68] H. Zhou, R. Kannan, A. Swami, and V. Prasanna, "HTNet: Dynamic WLAN Performance Prediction using Heterogenous Temporal GNN," *arXiv preprint arXiv:2304.10013*, 2023.

[69] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The M4 Competition: 100,000 time series and 61 forecasting methods," *Int J Forecast*, vol. 36, no. 1, pp. 54–74, 2020.

[70] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[71] P. Mursia, V. Sciancalepore, A. Garcia-Saavedra, L. Cottatellucci, X. C. Pérez, and D. Gesbert, "RISMA: Reconfigurable intelligent surfaces enabling beamforming for IoT massive access," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 4, pp. 1072–1085, 2020.

[72] D.-J. Munoz, M. Pinto, and L. Fuentes, "Detecting feature influences to quality attributes in large and partially measured spaces using smart sampling and dynamic learning," *Knowl Based Syst*, vol. 270, p. 110558, 2023.

[73] D.-J. Munoz, M. Pinto, and L. Fuentes, "Quality-aware analysis and optimisation of virtual network functions," in *Proceedings of the 26th ACM International Systems and Software Product Line Conference-Volume A*, 2022, pp. 210–221.

[74] W.-Y. Liang, Y. Yuan, and H.-J. Lin, "A performance study on the throughput and latency of zenoh, MQTT, Kafka, and DDS," *arXiv preprint arXiv:2303.09419*, 2023.

[75] 3GPP, "5G System; Network Data Analytics Services; Stage 3," 2021. [Online]. Available: https://www.3gpp.org/DynaReport/29520.html

[76] A. Kaloxylos, A. Gavras, D. Camps Mur, M. Ghoraishi, and H. Hrasnica, "AI and ML – Enablers for Beyond 5G Networks." Zenodo, Dec. 2020. doi: 10.5281/zenodo.4299895.

[77] A. T.-J. Akem, M. Gucciardo, and M. Fiore, "Flowrest: Practical Flow-Level Inference in Programmable Switches with Random Forests," 2023. Accessed: Jun. 25, 2023. [Online]. Available: https://dspace.networks.imdea.org/handle/20.500.12761/1649

[78] X. Zhu, Y. Luo, A. Liu, N. N. Xiong, M. Dong, and S. Zhang, "A Deep Reinforcement Learning-Based Resource Management Game in Vehicular Edge Computing," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2021, doi: 10.1109/TITS.2021.3114295.

[79] M. Nakanoya, Y. Sato, and H. Shimonishi, "Environment-adaptive sizing and placement of NFV service chains with accelerated reinforcement learning," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2019, pp. 36–44.

[80] Y. Xiao *et al.*, "NFVdeep: Adaptive online service function chain deployment with deep reinforcement learning," in *Proceedings of the International Symposium on Quality of Service*, 2019, pp. 1–10.

[81] P. T. A. Quang, Y. Hadjadj-Aoul, and A. Outtagarts, "A deep reinforcement learning approach for VNF forwarding graph embedding," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1318–1331, 2019.

[82] J. Pei, P. Hong, M. Pan, J. Liu, and J. Zhou, "Optimal VNF placement via deep reinforcement learning in SDN/NFV-enabled networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 263–278, 2019.

[83] J. Zheng *et al.*, "Optimizing NFV chain deployment in software-defined cellular core," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 248–262, 2019.

[84] R. Solozabal, J. Ceberio, A. Sanchoyerto, L. Zabala, B. Blanco, and F. Liberal, "Virtual network function placement optimization with deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 292–303, 2019.

[85] X. Foukas and B. Radunovic, "Concordia: teaching the 5G vRAN to share compute," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021, pp. 580–596.

[86] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Trans Wirel Commun*, vol. 18, no. 11, pp. 5141–5152, 2019.

[87] S. Bakri, B. Brik, and A. Ksentini, "On using reinforcement learning for network slice admission control in 5G: Offline vs. online," *International Journal of Communication Systems*, vol. 34, no. 7, pp. 1–12, 2021, doi: 10.1002/dac.4757.

[88] S. Tripathi, C. Puligheddu, C. F. Chiasserini, and F. Mungari, "A Context-aware Radio Resource Management in Heterogeneous Virtual RANs," *IEEE Trans Cogn Commun Netw*, 2021.

[89] F. B. Mismar, J. Choi, and B. L. Evans, "A framework for automated cellular network tuning with reinforcement learning," *IEEE Transactions on Communications*, vol. 67, no. 10, pp. 7152–7167, 2019.

[90] Z. Xiong and N. Zilberman, "Do switches dream of machine learning? toward in-network classification," in *Proceedings of the 18th ACM workshop on hot topics in networks*, 2019, pp. 25–33.

[91] C. Gijón, M. Toril, S. Luna-Ramírez, M. L. Marí-Altozano, and J. M. Ruiz-Avilés, "Long-Term Data Traffic Forecasting for Network Dimensioning in LTE with Short Time Series," *Electronics (Basel)*, vol. 10, no. 10, 2021, doi: 10.3390/electronics10101151.

[92] C. Gutterman, E. Grinshpun, S. Sharma, and G. Zussman, "RAN resource usage prediction for a 5G slice broker," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Catania, Italy, Jul. 2019, pp. 231–240.

[93] B. Yang, X. Cao, O. Omotere, X. Li, Z. Han, and L. Qian, "Improving medium access efficiency with intelligent spectrum learning," *IEEE Access*, vol. 8, pp. 94484–94498, 2020.

[94] X. Liu, J. Yu, J. Wang, and Y. Gao, "Resource allocation with edge computing in IoT networks via machine learning," *IEEE Internet Things J*, vol. 7, no. 4, pp. 3415–3426, 2020.

[95] M. Camelo *et al.*, "An ai-based incumbent protection system for collaborative intelligent radio networks," *IEEE Wirel Commun*, vol. 27, no. 5, pp. 16–23, 2020.

[96] Z. Yan, J. Ge, Y. Wu, L. Li, and T. Li, "Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1040–1057, 2020.

[97] L. Wang, W. Mao, J. Zhao, and Y. Xu, "DDQP: A double deep Q-learning approach to online fault-tolerant SFC placement," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 118–132, 2021.

[98] J. Jia, L. Yang, and J. Cao, "Reliability-aware dynamic service chain scheduling in 5g networks based on reinforcement learning," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, 2021, pp. 1–10.

[99] K. Nakashima, S. Kamiya, K. Ohtsu, K. Yamamoto, T. Nishio, and M. Morikura, "Deep reinforcement learning-based channel allocation for wireless lans with graph convolutional networks," *IEEE Access*, vol. 8, pp. 31823–31834, 2020.

[100] Y. Xu, P. Cheng, Z. Chen, Y. Li, and B. Vucetic, "Mobile collaborative spectrum sensing for heterogeneous networks: A Bayesian machine learning approach," *IEEE Transactions on Signal Processing*, vol. 66, no. 21, pp. 5634–5647, 2018.

[101] A. Manousis, H. Shah, H. Milner, Y. Li, H. Zhang, and V. Sekar, "The Shape of View: An Alert System for Video Viewership Anomalies," in *Proceedings of the 21st ACM Internet Measurement Conference*, in IMC '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 245–260. doi: 10.1145/3487552.3487819.

[102] D. Perino, X. Yang, J. Serra, A. Lutu, and I. Leontiadis, "Experience: Advanced Network Operations in (Un)-Connected Remote Communities," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, in MobiCom '20. New York, NY, USA: Association for Computing Machinery, 2020. doi: 10.1145/3372224.3380893.

[103] A. Padmanabha Iyer, L. Erran Li, M. Chowdhury, and I. Stoica, "Mitigating the Latency-Accuracy Trade-off in Mobile Data Analytics Systems," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, in MobiCom '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 513–528. doi: 10.1145/3241539.3241581.

[104] J. M. Navarro, A. Huet, and D. Rossi, "Human readable network troubleshooting based on anomaly detection and feature scoring," *CoRR*, vol. abs/2108.11807, 2021, [Online]. Available: https://arxiv.org/abs/2108.11807

[105] J. M. Navarro and D. Rossi, "HURRA! Human readable router anomaly detection," *International Teletraffic Congress (ITC32)*. Sep. 2020.

[106] C. Kattadige, A. Raman, K. Thilakarathna, A. Lutu, and D. Perino, "360NorVic: 360-Degree Video Classification from Mobile Encrypted Video Traffic," in *Proceedings of the 31st ACM Workshop on*

Network and Operating Systems Support for Digital Audio and Video, in NOSSDAV '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 58–65. doi: 10.1145/3458306.3460998.

[107] T. Mangla, E. Halepovic, E. Zegura, and M. Ammar, "Drop the Packets: Using Coarse-Grained Data to Detect Video Performance Issues," in *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies*, in CoNEXT '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 71–77. doi: 10.1145/3386367.3431294.

[108] T. Subramanya and R. Riggio, "Centralized and federated learning for predictive VNF autoscaling in multi-domain 5G networks and beyond," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 63–78, 2021.

[109] S. Rahman, T. Ahmed, M. Huynh, M. Tornatore, and B. Mukherjee, "Auto-scaling VNFs using machine learning to improve QoS and reduce cost," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.

[110] H. Huang *et al.*, "Scalable orchestration of service function chains in NFV-enabled networks: A federated reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2558–2571, 2021.

[111] C. Zhang, X. Costa-Pérez, and P. Patras, "Tiki-taka: Attacking and defending deep learning-based intrusion detection systems," in *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, 2020, pp. 27–39.

[112] J. Prados-Garzon, T. Taleb, and M. Bagaa, "LEARNET: Reinforcement learning based flow scheduling for asynchronous deterministic networks," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[113] H. Zhu, V. Gupta, S. S. Ahuja, Y. Tian, Y. Zhang, and X. Jin, "Network planning with deep reinforcement learning," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021, pp. 258–271.

[114] S. Yan, X. Wang, X. Zheng, Y. Xia, D. Liu, and W. Deng, "ACC: Automatic ECN tuning for high-speed datacenter networks," in *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021, pp. 384–397.

[115] S. Wang, X. Zhang, H. Uchiyama, and H. Matsuda, "HiveMind: Towards cellular native machine learning model splitting," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 2, pp. 626–640, 2021.

[116] L. He, L. Li, and Y. Liu, "Towards chain-aware scaling detection in nfv with reinforcement learning," in *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, 2021, pp. 1–10.

[117] F. Rossi, M. Nardelli, and V. Cardellini, "Horizontal and vertical scaling of container-based applications using reinforcement learning," in *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, 2019, pp. 329–338.

[118] A. A. Khaleq and I. Ra, "Intelligent autoscaling of microservices in the cloud for real-time applications," *IEEE Access*, vol. 9, pp. 35464–35476, 2021.

[119] V. Zalokostas-Diplas, N. Makris, V. Passas, and T. Korakis, "Experimental Evaluation of ML Models for Dynamic VNF Autoscaling," in *2022 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2022, pp. 157–162.

[120] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-Air Deep Learning Based Radio Signal Classification," *IEEE J Sel Top Signal Process*, vol. 12, no. 1, pp. 168–179, 2018, doi: 10.1109/JSTSP.2018.2797022.

[121] J. Rosa *et al.*, "BacalhauNet: A tiny CNN for lightning-fast modulation classification," *ITU Journal on Future and Evolving Technologies*, vol. 3, no. 2, pp. 252–260, 2022.

[122] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, "Service Function Chain Embedding for NFV-Enabled IoT Based on Deep Reinforcement Learning," *IEEE Communications Magazine*, vol. 57, no. 11, pp. 102–108, Jun. 2019, doi: 10.1109/MCOM.001.1900097.

[123] J. Koo, V. B. Mendiratta, M. R. Rahman, and A. Walid, "Deep Reinforcement Learning for Network Slicing with Heterogeneous Resource Requirements and Time Varying Traffic Dynamics." 2019.

[124] A. Dalgkitsis *et al.*, "SCHE2MA: Scalable, Energy-Aware, Multidomain Orchestration for Beyond-5G URLLC Services," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2022, doi: 10.1109/TITS.2022.3202312.

[125] G. L. Santos, T. Lynn, J. Kelner, and P. T. Endo, "Availability-aware and energy-aware dynamic SFC placement using reinforcement learning," *J Supercomput*, pp. 1–30, 2021, doi: https://doi.org/10.1007/s11227-021-03784-7.

[126] M. A. Khan, R. Hamila, N. A. Al-Emadi, S. Kiranyaz, and M. Gabbouj, "Real-time throughput prediction for cognitive Wi-Fi networks," *Journal of Network and Computer Applications*, vol. 150, p. 102499, 2020.

[127] D. Minovski, N. Ogren, C. Ahlund, and K. Mitra, "Throughput prediction using machine learning in lte and 5g networks," *IEEE Trans Mob Comput*, 2021.

[128] D. Teixeira, R. Meireles, and A. Aguiar, "Wi-fi throughput estimation for vehicle-to-network communication in heterogeneous wireless environments," in *2023 18th Wireless On-Demand Network Systems and Services Conference (WONS)*, 2023, pp. 24–31.

[129] C. Busse-Grawitz, R. Meier, A. Dietmüller, T. Bühler, and L. Vanbever, "pForest: In-Network Inference with Random Forests," *CoRR*, vol. abs/1909.05680. 2019. [Online]. Available: http://arxiv.org/abs/1909.05680

[130] G. Xie, Q. Li, Y. Dong, G. Duan, Y. Jiang, and J. Duan, "Mousika: Enable General In-Network Intelligence in Programmable Switches by Knowledge Distillation," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 1938–1947. doi: https://doi.org/10.1007/s11227-021-03784-7.

[131] C. Zheng *et al.*, "Automating In-Network Machine Learning," *arXiv preprint arXiv:2205.08824*, 2022.

[132] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "DeepCog: Cognitive Network Management in Sliced 5G Networks with Deep Learning," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019, pp. 280–288. doi: 10.1109/INFOCOM.2019.8737488.

[133] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Aztec: Anticipatory capacity allocation for zero-touch network slicing," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 794–803. doi: 10.1109/INFOCOM41043.2020.9155299.

[134] C. Zhang and P. Patras, "Long-Term Mobile Traffic Forecasting Using Deep Spatio-Temporal Neural Networks.," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc '18)*. , 2018, pp. 231–240. doi: https://doi.org/10.1145/3209582.3209606.

[135] C. Zhang, M. Fiore, and P. Patras, "Multi-Service Mobile Traffic Forecasting via Convolutional Long Short-Term Memories," in *Proceedings of 2019 IEEE International Symposium on Measurements & Networking (M&N).*, 2019, pp. 1–6. doi: 10.1109/IWMN.2019.8804984.

[136] L. G. H. D. Trinh and P. Dini, "Mobile Traffic Prediction from Raw Data Using LSTM Networks," in *Proceedings of the 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018, pp. 1827–1832. doi: 10.1109/PIMRC.2018.8581000.

[137] T. J. O'Shea, S. Hitefield, and J. Corgan, "End-to-end radio traffic sequence recognition with recurrent neural networks," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016, pp. 277–281. doi: 10.1109/GlobalSIP.2016.7905847.

[138] M. Camelo *et al.*, "A semi-supervised learning approach towards automatic wireless technology recognition," in *2019 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2019, pp. 1–10. doi: 10.1109/DySPAN.2019.8935690.

[139] M. Camelo, T. De Schepper, P. Soto, J. Marquez-Barja, J. Famaey, and S. Latré, "Detection of traffic patterns in the radio spectrum for cognitive wireless network management," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6. doi: 10.1109/ICC40277.2020.9149077.

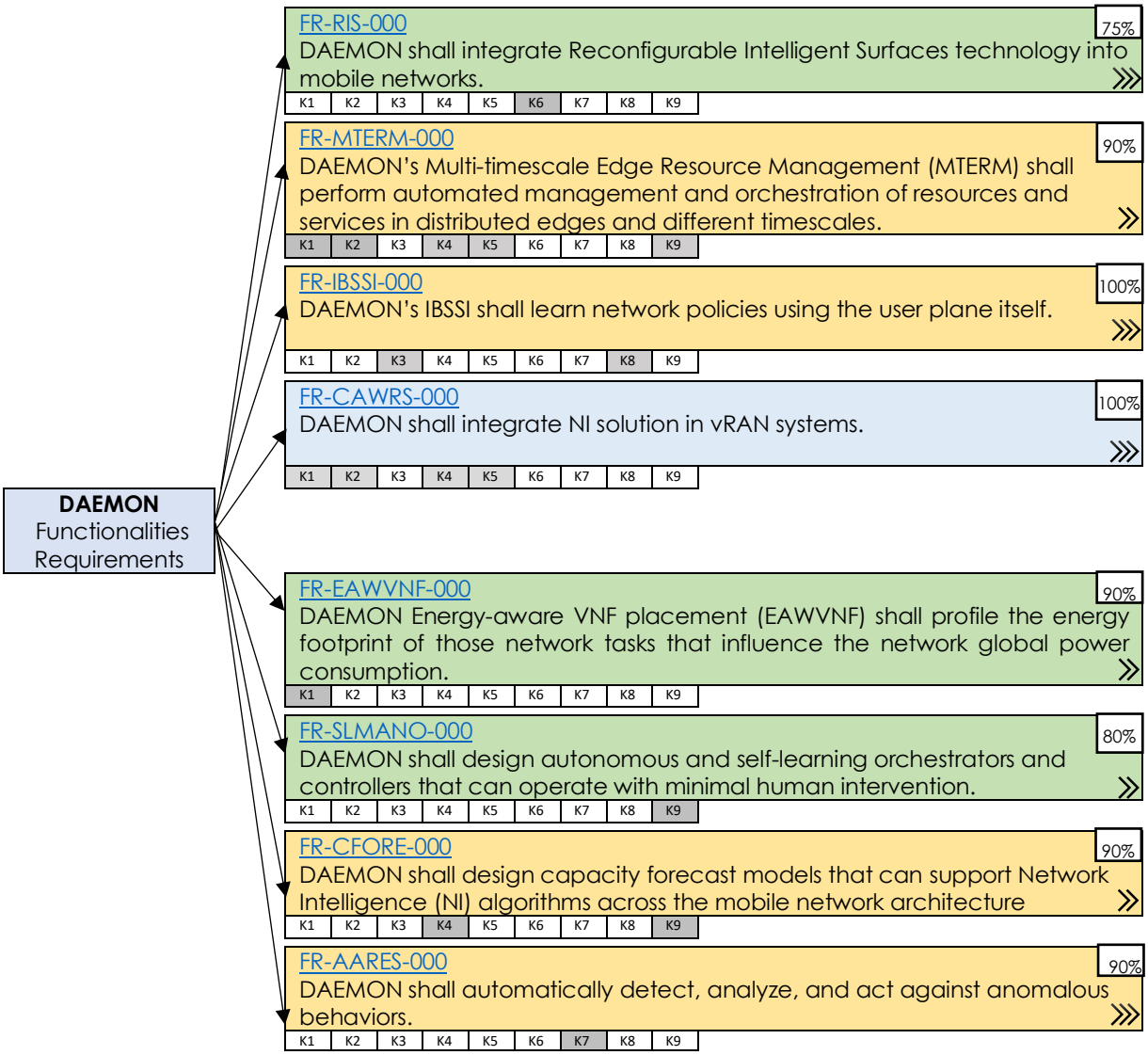
[140] A. Dalgkitsis, P.-V. Mekikis, A. Antonopoulos, and C. Verikoukis, "Data Driven Service Orchestration for Vehicular Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4100–4109, 2021, doi: 10.1109/TITS.2020.3011264.

[141] H. Ma, Z. Zhou, and X. Chen, "Leveraging the Power of Prediction: Predictive Service Placement for Latency-Sensitive Mobile Edge Computing," *IEEE Trans Wirel Commun*, vol. 19, no. 10, pp. 6454–6468, 2020, doi: 10.1109/TWC.2020.3003459.

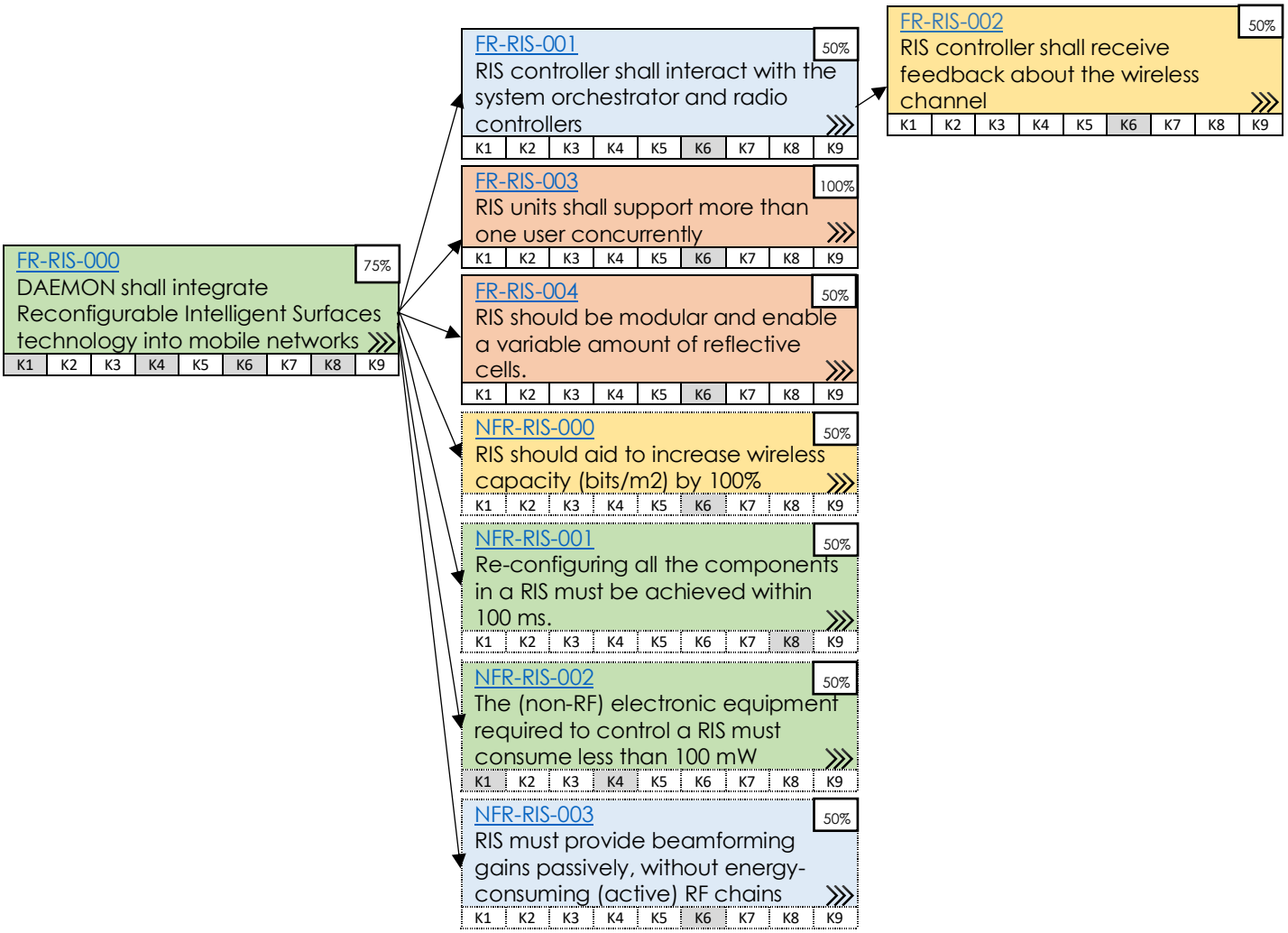
[142] C. Grasso, R. Raftopoulos, and G. Schembra, "Smart Zero-Touch Management of UAV-Based Edge Network," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4350–4368, 2022, doi: 10.1109/TNSM.2022.3160858.

A Appendix: NI Use Cases Functional Requirements

In the following figure, the top-level requirements of DAEMON are represented tree shaped. Each requirement is colored considering the risk assessment, and the KPIs addressed by each requirement are shown at the bottom of each box. We represent either the functional and non-functional requirements, which can be visually distinguished using a dotted or continuous line. The percent complete and the risk management are represented on the left side of each cluster.



A.1 RIS control



FR-RIS-000																		
Description	DAEMON shall integrate Reconfigurable Intelligent Surfaces (RIS) technology into mobile networks.																	
Version	001M1																	
Owner	NEC																	
Priority	High																	
Risk	2																	
Risk Description	There is a mild risk that the project will not be able to build a RIS prototype. Should this happen, the project will rely on simulations and mathematical models.																	
Rationale	RIS technologies will play a key role in increasing the wireless network capacity of next-generation networks, reducing energy consumption, and creating new privacy and security applications. However, optimal RIS operation can only be achieved in coordination with the radio access network controller. To this end, native support by DAEMON platform and open interfaces that integrate RIS controllers into the rest of the mobile network control ecosystem is required.																	
K1		**K2**		**K3**		**K4**		**K5**		**K6**	X	**K7**		**K8**		**K9**		
Parents	None																	
Current Status																		
Percent complete	75%																	
Risk management	Successful																	
Rationale	Risks were low. There was an initial risk that a RIS prototype may not have been available for experimentation, in such a case, analytical data would have been used. But the risk has not materialized. The initial design of a RIS control NI was presented in D3.2 [2], Section 4, and the final design will be presented in D3.3. The detailed design has been presented in [71]. Moreover, the initial design of an experimental prototype was presented in D5.2 [4], Section 4.6, and the final prototype will be presented in D5.3.																	

FR-RIS-001																		
Description	RIS controller shall interact with the system orchestrator and radio controllers																	
Version	002M5																	
Owner	NEC																	
Priority	High																	
Risk	1																	
Risk Description	No risk																	
Rationale	An interface between the mobile network orchestrator, the gNB controllers, and the RIS controller shall enable joint optimization of gNBs, UEs and surfaces.																	
K1		**K2**		**K3**		**K4**		**K5**		**K6**	X	**K7**		**K8**		**K9**		
Parents	FR-RIS-000-001M1																	
Current Status																		
Percent complete	50%																	
Risk management	Successful																	
Rationale	A RIS prototype is being built along with an interface to interact with an external controller. The initial prototype design was presented in D5.2 [4], Section 4.6, and the final design will be presented in D5.3.																	

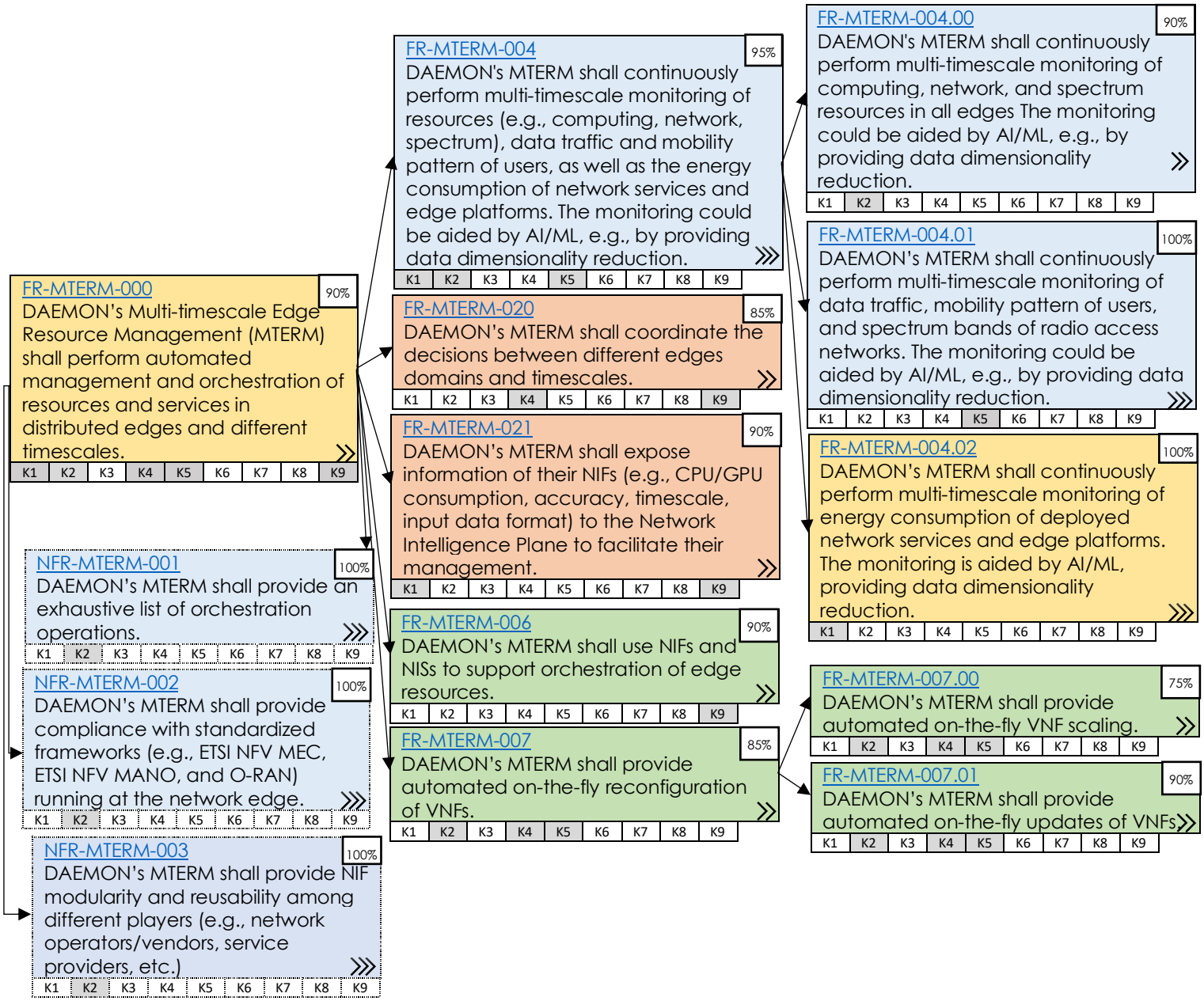
FR-RIS-002	
Description	RIS controller shall receive feedback about the wireless channel
Version	002M4
Owner	NEC
Priority	Medium
Risk	3
Risk Description	Channel feedback may not be received in a timely manner or with the required accuracy so as to be useful information.
Rationale	Reconfigurable Intelligent Surfaces modify the propagation properties of impinging wireless signals in a controllable manner. To this end, good estimations

	about the wireless environment based upon feedback from users and gNBs, i.e., channel information, are required to perform optimal RIS operation.
K1	K2	K3	K4	K5	K6	X	K7	K8	K9
Parents	FR-RIS-001-002M5
Current Status	
Percent complete	50%
Risk management	Successful
Rationale	A RIS prototype is being built along with an interface to receive feedback from an external controller. The initial prototype design was presented in D5.2 [4], Section 4.6, and the final design will be presented in D5.3.

FR-RIS-003	
Description	RIS units shall support more than one user concurrently
Version	003M17
Owner	NEC
Priority	Medium
Risk	4
Risk Description	Tight and timely coordination between gNB MAC schedulers may be required
Rationale	This enables increasing the system capacity for multiple users.
K1	K2	K3	K4	K5	K6	X	K7	K8	K9
Parents	FR-RIS-000-001M1
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	There was a risk that optimizing the RIS configuration for multiple concurrent users would be overly hard to achieve. We solved this complicated problem as will be explained in D3.3. The details can be found in [71].

FR-RIS-004	
Description	RIS should be modular and enable a variable amount of reflective cells.
Version	002M17
Owner	NEC
Priority	Low
Risk	4
Risk Description	Modularity may be overly hard to achieve when designing a RIS.
Rationale	The ability to change the amount of reflective surface would enable a RIS to adapt itself to the surface, which may be highly irregular.
K1	K2	K3	K4	K5	K6	X	K7	K8	K9
Parents	FR-RIS-000-001M1
Current Status	
Percent complete	50%
Risk management	Successful
Rationale	Though initially was believed that modularity would be overly hard to achieve, we have solved this problem as will be reported in D5.3.

A.2 Functional requirements: Multi-timescale Edge resource management



FR-MTERM-000	
Description	DAEMON's Multi-timescale Edge Resource Management (MTERM) shall perform automated management and orchestration of resources and services in distributed edges and different timescales.
Version	003M18
Owner	IMEC
Priority	High
Risk	3
Risk Description	The decisions made by Network Intelligent Functions (NIFs) and network Intelligent Services (NIS) distributed across the edge networks might be out of sync, since they can make decisions in different timescales. We might need to assign the level of priority to decision-making entities in different tiers and have a control loop that will track the effect of these decisions on the service KPIs.
Rationale	Services are deployed in a distributed fashion, due to the high mobility of users, and an uneven distribution of resources across the edge networks. Thus, proper management and orchestration of these distributed services need to be achieved. The network intelligence in the form of AI-based NIFs and NISs needs to be distributed to different edges in the management and orchestration architecture in order to treat different service dynamics in coarse/fine granular timescale. This should be done in an automated way. Unfortunately, current management frameworks do not provide automation in the form of flexible and dynamic NFV management and orchestration and therefore, this gap should be addressed. Moreover, management frameworks should be able to coordinate intelligence or resources across different network segments and timescales. **Services**: MEC application services (specific to use cases, i.e., vertical services), Value-added services (e.g., location services, Radio Network Information Service), NISs and NIFs (e.g., traffic classifiers), energy consumption analyzers, etc. **Resources**: CPU, memory, spectrum, storage, and network

K1	**X**	**K2**	**X**	**K3**		**K4**	**X**	**K5**	**X**	**K6**		**K7**		**K8**	**X**	**K9**	X

Parents	None
Current Status	
Percent complete	90%
Risk management	Effective
Rationale	Since this requirement is the root, by fulfilling the child requirements, DAEMON can effectively perform automated management and orchestration of resources and services in distributed edges and different timescales. All the risks were low and effectively or successfully managed.

FR-MTERM-004	
Description	DAEMON's MTERM shall continuously perform multi-timescale monitoring of resources (e.g., computing, network, spectrum), data traffic and mobility pattern of users, as well as the energy consumption of network services and edge platforms. The monitoring could be aided by AI/ML, e.g., by providing data dimensionality reduction.
Version	002M18
Owner	IMEC
Priority	High
Risk	1
Risk Description	The amount of data that is being collected might burden the resource-constrained edge nodes. Thus, we need to assess the resource requirements of monitoring services that will be running along with other services on the edge platforms, and to perform a corresponding management of these services in order to produce meaningful and credible results.
Rationale	Monitoring is one of the main pillars of any automated and adaptative system. By monitoring, any system can verify that its decisions were correctly applied, achieving closed-loop control. However, given the diversity of network

	operators/vendors/infrastructure/providers/service providers, monitored data stems from multiple sources. In that sense, AI/ML techniques could help to pre-process and reduce such data's dimensionality. However, current frameworks do not incorporate real-time data analytics, making difficult the monitoring of data.

K1	X	K2	X	K3		K4		K5		K6		K7		K8	X	K9	

Parents	FR-MTERM-000
Current Status	
Percent complete	95%
Risk management	Successful
Rationale	Several NI solutions implemented a monitoring method. For instance, in the AI-enhanced edge orchestration (Section 3.1 of D3.2 [2]), NIFs are constantly collecting data from various nodes. This data is composed of computing resources, network metrics (latency, bandwidth) and application services. Similarly, the model in Section 3.2 of D3.2 [2] continuously consumes the monitored spectrum usage. Moreover, the model in Section 3.4 of D3.2 [2], leverages big data to collect information about different devices. Despite the energy footprint was not measured in any of the activities mentioning this requirement, the solution presented in FR-EAWVNF-001 can be used for this purpose.

FR-MTERM-004.00	
Description	DAEMON's MTERM shall continuously perform multi-timescale monitoring of computing, network, and spectrum resources in all edges. The monitoring could be aided by AI/ML, e.g., by providing data dimensionality reduction.
Version	001M18
Owner	IMEC
Priority	High
Risk	1
Risk Description	Risk FR-MTERM-004
Rationale	The constant monitoring input of computing, network and spectrum resources will feed the orchestration entities that perform orchestration operations, to control and provide an on-the-fly reconfiguration of deployed virtualized network functions, to migrate them, and to identify anomalies in service and/or framework operation. These metrics shall be monitored at different timescales, depending on the granularity required by the service consuming the data and the available resources at the edge.

K1		K2	X	K3		K4		K5		K6		K7		K8		K9	

Parents	FR-MTERM-004
Current Status	
Percent complete	90%
Risk management	Effective
Rationale	Same as parent. Detailed in Sections 3.1 and 3.4 of D3.2 [2].

FR-MTERM-004.01	
Description	DAEMON's MTERM shall continuously perform multi-timescale monitoring of data traffic, mobility patterns of users, and spectrum bands of radio access networks. The monitoring could be aided by AI/ML, e.g., by providing data dimensionality reduction.
Version	002M18
Owner	IMEC
Priority	Low
Risk	2

Risk Description	The value-added services that collect and parse data from the network traffic, the UE mobility, and the spectrum bands impose additional burdens on the resource-constrained edge nodes. Thus, we need to assess the resource requirements of those services that will be running along with other services on the edge platforms, and to perform a corresponding management of these services in order to produce meaningful and credible results.
Rationale	The constant monitoring input of data traffic, mobility patterns and spectrum bands will provide input about the UEs to the orchestration entities that perform orchestration operations, to proactively deploy additional VNFs when and where needed, to migrate them, and to reconfigure existing VNFs to meet demands of all UEs in the system. These metrics shall be monitored at different timescales, depending on the granularity required by the service consuming the data and the available resources at the edge.

K1		**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**	X	**K9**	

Parents	FR-MTERM-004
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	Same as parent. Detailed in Section 3.2 of D3.2 [2].

FR-MTERM-004.02	
Description	DAEMON's MTERM shall continuously perform multi-timescale monitoring of energy consumption of deployed network services and edge platforms. The monitoring is aided by AI/ML, providing data dimensionality reduction.
Version	002M18
Owner	IMEC
Priority	Low
Risk	3
Risk Description	The energy consumption calculation of isolated services might be a complex task, while at the same time, an aggregated energy consumption per edge platform might severely affect the accuracy of energy-aware NIFs. Furthermore, although those NIFs that manage energy consumption in the whole system run in cloud, they still need to have probes installed on the edges, and a proper assessment of their energy consumption and resource requirements needs to be obtained.
Rationale	The constant monitoring of energy consumption per service/per edge platform is needed to make an optimal decision on VNF placement and VNF migration from one edge to another. With such an energy consumption footprint in the whole system, the cloud orchestrator can perform load balancing between edge platforms, and accordingly turn off certain NIFs and deployed services if energy consumption needs to be decreased.

K1	X	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**	

Parents	FR-MTERM-004
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	Same rationale as FR-EAWVNF-001.

FR-MTERM-020	
Description	DAEMON's MTERM shall coordinate the decisions between different edge domains and timescales.
Version	001M18

Owner	IMEC
Priority	Low
Risk	4
Risk Description	Depending on the number of decision-making engines, the coordination can be cumbersome.
Rationale	Several management and orchestration operations are based on the decisions of different decision-making engines. Such engines can be based on AI/ML. To guarantee service continuity, coordination between distributed orchestrators in different edge domains is almost mandatory. However, current frameworks do not coordinate intelligence or resources across different network segments and timescales.

K1		**K2**		**K3**		**K4**	X	**K5**		**K6**		**K7**		**K8**		**K9**		

Parents	FR-MTERM-000
Current Status	
Percent complete	85%
Risk management	Effective
Rationale	The solutions presented in Sections 3.1 and 3.5 of D3.2 [2] coordinate the decisions between different edges (§3.1) and timescales (§3.2). The risks were mitigated by considering two levels of priorities. When taking decisions among multiple edges, decisions at the edge affect only the local resources, while cloud decisions affect the global resources and, therefore, have higher priority. Regarding the timescales, decisions made at longer timescales (e.g., placement) are performed less frequently than decisions made at shorter timescales (e.g., adjustments to respond to the evolution of the data).

FR-MTERM-021	
Description	DAEMON's MTERM shall expose information of their NIFs (e.g., CPU/GPU consumption, accuracy, timescale, input data format) to the Network Intelligence Plane to facilitate their management.
Version	001M18
Owner	IMEC
Priority	Low
Risk	4
Risk Description	The huge amounts of collected data from surrounding infrastructure might represent a risk, since that data might be incomplete or inconsistent. This lack of sufficient and consistent input data leads to inefficiencies in decision-making, e.g., when to replace a NIF.
Rationale	Information about NIFs, like CPU/GPU consumption, accuracy, timescale, and input data format, should be exposed to the Network Intelligence Plane. Based on this information, the intelligent orchestrator(s) should take a decision (e.g., change NIFs because of their poor performance). This would facilitate the lifecycle management of AI/ML-based functions, which current frameworks do not support.

K1	X	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**	X	

Parents	FR-MTERM-000
Current Status	
Percent complete	90%
Risk management	Effective
Rationale	The solution presented in Section 3.4 of D3.2 [2] performs a classification, then, based on that, a resource allocation decision is made in the orchestrator. The results from the NIF, a classification model in this case, are exposed through open APIs as mentioned in Section 3.3 of D2.3.

FR-MTERM-006																			
Description	DAEMON's MTERM shall use NIFs and NISs to support orchestration of edge resources.																		
Version	002M28																		
Owner	IMEC																		
Priority	Low																		
Risk	2																		
Risk Description	The interfaces created to support the instantiation of NIFs and NISs could be tightly coupled which difficult their integration with existing management and orchestration frameworks.																		
Rationale	Current research has shown that management and orchestration operations can be improved by using Network Intelligence Functions (ML-based solutions). However, existing management and orchestration frameworks that operate at the network edge (e.g., NFV MANO, OSM, ETSI MEC) do not fully integrate and support the instantiation of such intelligent functions. These frameworks do not provide the necessary interfaces to enable services and applications to be data-driven.																		
K1		**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**	X	**K9**			
Parents	FR-MTERM-000																		
Current Status																			
Percent complete	90%																		
Risk management	Effective																		
Rationale	The solutions in Sections 3.1, 3.2 of D3.2 [2] and 5.2 of D4.2 [3] use a model which improves the quality of the decisions made by orchestrators. For instance, the solution proposed in Section 4.1 of D4.2 [3] (Federated Anomaly Detection) can be used by the orchestrator in Section 3.1 of D3.2 [2] to receive the indication that the current node has an abnormal behavior, so the service can be migrated to a healthy node (Synergic Integration of Network Intelligences Demo).																		

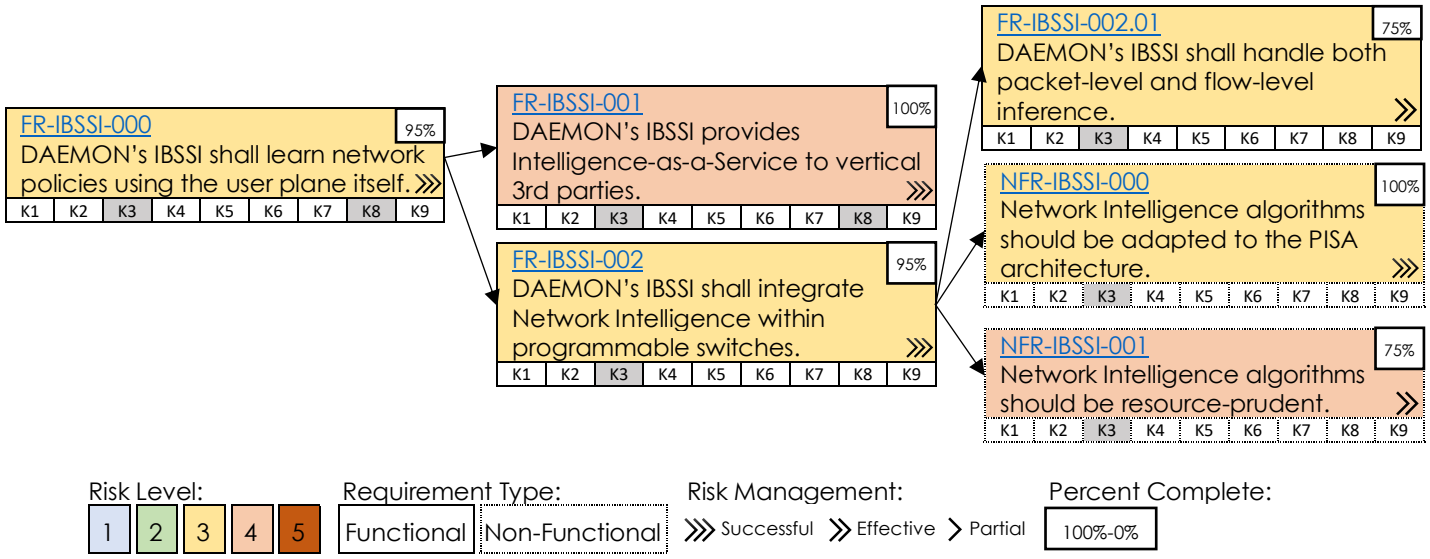
FR-MTERM-007																			
Description	DAEMON's MTERM shall provide automated on-the-fly reconfiguration of VNFs																		
Version	003M18																		
Owner	IMEC																		
Priority	High																		
Risk	2																		
Risk Description	The reconfiguration of VNFs in the service function chain might impose a risk of service unavailability during the reconfiguration.																		
Rationale	Following the cloud-native service design, the service function chains consist of loosely-coupled VNFs that can be replaced and separately configured. Orchestration entities can make decisions to scale up/down/out/in any of these VNFs, and to replace the faulty ones, while maintaining service continuity.																		
K1		**K2**	X	**K3**		**K4**	X	**K5**	X	**K6**		**K7**		**K8**		**K9**			
Parents	FR-MTERM-000																		
Current Status																			
Percent complete	85%																		
Risk management	Effective																		
Rationale	The solutions in Section 3.1 of D3.2 [2] and 5.2 of D4.2 [3] perform on-the-fly reconfiguration of VNFs. For example, the model proposed in §5.2 od D4.2 [3] changes the number of replicas of a service whose processing is subject to a delay constraint. Then, scaling is performed accordingly with the workload																		

	changes. Similarly, the orchestrator in §3.1 of D3.2 [2] performs scaling and service relocation, which are considered as methods to perform VNF updating.

FR-MTERM-007.00																		
Description	DAEMON's MTERM shall provide automated on-the-fly VNF scaling.																	
Version	002M18																	
Owner	IMEC																	
Priority	High																	
Risk	2																	
Risk Description	The VNF scaling in the service function chain might impose a risk of service unavailability during the reconfiguration.																	
Rationale	Rationale FR-MTERM-007																	
K1		**K2**		**K3**		**K4**	X	**K5**		**K6**		**K7**		**K8**		**K9**		
Parents	FR-MTERM-007																	
Current Status																		
Percent complete	75%																	
Risk management	Effective																	
Rationale	Same as parent. The solution described in Section 5.2 of D4.2 [3] performs autonomous VNF scaling.																	

FR-MTERM-007.01																		
Description	DAEMON's MTERM shall provide automated on-the-fly update of VNFs																	
Version	002M18																	
Owner	IMEC																	
Priority	Low																	
Risk	2																	
Risk Description	The update of VNFs (e.g., change of VNF image, VNF descriptor, IP address, etc.) in the service function chain might impose a risk of service unavailability during the reconfiguration.																	
Rationale	Following the cloud-native service design, the service function chains consist of loosely-coupled VNFs that can be replaced and separately configured. Orchestration entities can make decisions to update VNFs, e.g., if an updated image or descriptor is needed.																	
K1		**K2**	**X**	**K3**		**K4**	X	**K5**	**X**	**K6**		**K7**		**K8**		**K9**		
Parents	FR-MTERM-007																	
Current Status																		
Percent complete	90%																	
Risk management	Effective																	
Rationale	Same as parent. The solution proposed in Section 3.1 of D3.2 [2] performs relocation of services deployed as VNFs from one edge to the other, to avoid service quality degradation (e.g., because the users are driving away from a given RSU).																	

A.3 Functional requirements: In-backhaul support for service intelligence



FR-IBSSI-000																		
Description	DAEMON's IBSSI shall learn network policies using the user plane itself.																	
Version	003M18																	
Owner	UC3M																	
Priority	Low																	
Risk	3																	
Risk Description	To ensure fast reaction times for orchestration mechanisms upon network changes, the network shall learn directly from data-plane network functions, providing triggers for the required re-orchestrations or re-configurations of the network functions.																	
Rationale	Besides monitoring of KPIs, the network shall already understand and detect malfunctioning already from the analysis of specific traffic patterns or control-plane interactions. This is especially important for operations such as anomaly detection.																	
K1		**K2**		**K3**	X	**K4**		**K5**		**K6**		**K7**		**K8**	X	**K9**		
Parents	None																	
Current Status																		
Percent complete	95%																	
Risk management	Successful																	
Rationale	The procedures described in Section 5 of this document allow the online monitoring of the traffic to support the online learning of policies.																	

FR-IBSSI-001																		
Description	DAEMON's IBSSI shall provide Intelligence-as-a-Service to vertical 3rd parties																	
Version	003M17																	
Owner	UC3M																	
Priority	High																	
Risk	4																	
Risk Description	Third parties will be allowed to be included in the network operation through specific APIs that are used to i) manage the kind of provided intelligence and ii) ensure that the resources are provided to them. Also, these interfaces shall accommodate different intelligence instances running in the third-party premises and in the network domain.																	
Rationale	DAEMON will provide algorithms for the execution of network intelligence directly related to the vertical service (e.g., video analytics directly in the u-plane) and allow efficient and secure resource provisioning through the usage of solutions based on, e.g., distributed ledger platform.																	
K1		**K2**		**K3**	X	**K4**		**K5**		**K6**		**K7**		**K8**	X	**K9**		
Parents	FR-IBSSI-000-003M18																	
Current Status																		
Percent complete	100%																	
Risk management	Successful																	
Rationale	The work performed and discussed in Section 4.2.2 defined the interfaces towards the management and orchestration and other core functions such as the AF, which can be used to interact with 3rd parties.																	

FR-IBSSI-002	
Description	DAEMON's IBSSI shall integrate Network Intelligence within programmable switches.
Version	002M17
Owner	IMDEA
Priority	Medium
Risk	3
Risk Description	Programmable switches have extremely limited computational capabilities and memory, which substantially constrains what they can do in terms of learning.

Rationale	Programmable user planes are starting to be leveraged for network telemetry functionalities. However, these are limited to data collection and pre-processing, which are then fed to NI located in the control plane to take network management decisions. DAEMON will investigate what portion of the decision process can be moved to the switches directly, at line rate and avoiding the delay of interacting with the control plane.

K1		K2		K3	X	K4		K5		K6		K7		K8		K9	

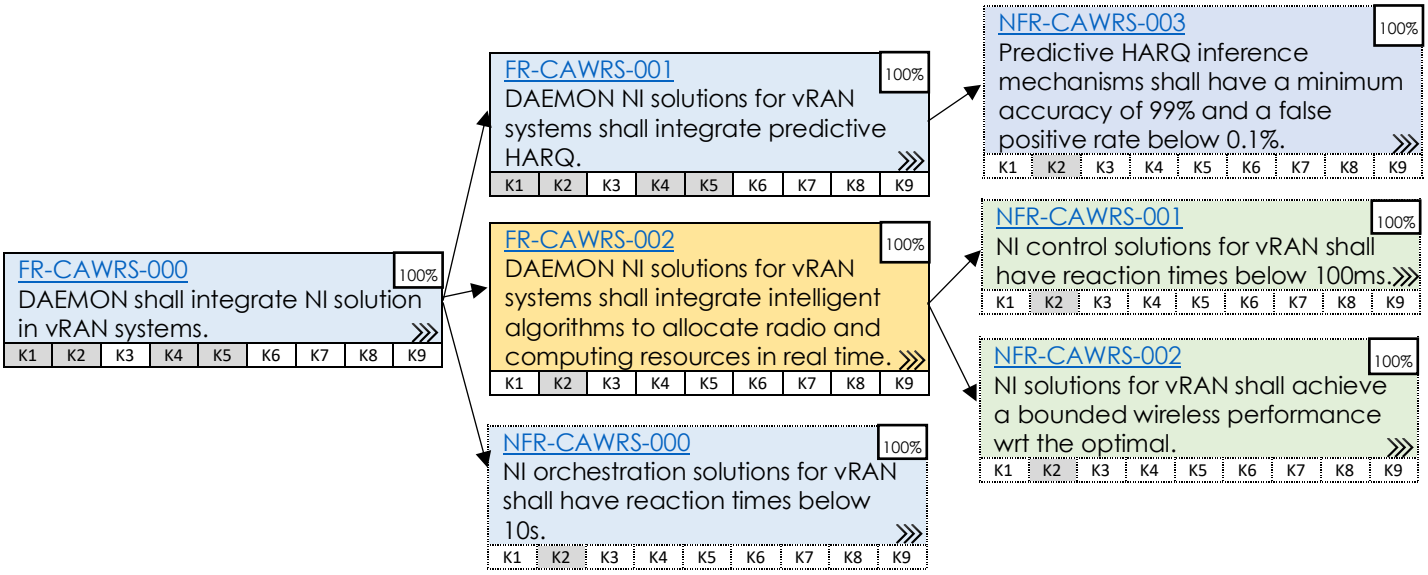
Parents	FR-IBSSI-000-003M18
Current Status	
Percent complete	95%
Risk management	Successful.
Rationale	DAEMON has developed Random Forest (RF) models that are tailored to the hardware of programmable switch ASICs, where they can extract flow-level features and use them for inference, as described in Section 5.1 of D3.2 [2]. The models have been evaluated in a real-world experimental platform with production-grade hardware, where they could achieve high accuracy (up to 99%) at line rate with ultra-low (~100 ns) latency, as per Section 4.7.1 of D5.2 [4]. Risks were estimated as intermediate at the start of the activity, due to the limitation of the computing environment offered by programmable switch ASICs; such risks were avoided by using models that are relatively simple and mappings of such models that are tailored to the target hardware.

FR-IBSSI-002.01	
Description	DAEMON's IBSSI shall handle both packet-level and flow-level inference
Version	001M18
Owner	IMDEA
Priority	Medium
Risk	3
Risk Description	Programmable switches have extremely limited storage and memory capabilities that constrain the possibility of computing and preserving features about the many individual packets or flows traversing the switch.
Rationale	Inference in programmable user planes can largely benefit from the availability of both packet-level (e.g., header fields) and flow-level (e.g., inter-arrival times, counters, etc.) input features. Indeed, these features offer different correlations with prediction variables (e.g., for classification, anomaly detection, intrusion detection, etc.). It is thus desirable that both types of features are available to a machine learning model deployed in the switch.

K1		K2		K3	X	K4		K5		K6		K7		K8		K9	

Parents	FR-IBSSI-002-002M17
Current Status	
Percent complete	75%
Risk management	Effective.
Rationale	The solutions developed by DAEMON, as indicated in the parent requirement, can gather and use both packet-level and flow-level features. The last remaining step toward meeting this requirement in a fully successful way is designing RF models that can operate on packet-level features for the first very few packets of each flow, i.e., before flow-level features can be reliably computed. Risks were estimated as intermediate at the start of the activity, due to the added complexity of computing and storing flow-level features in resource-constrained switch architectures; these risks were avoided by designing novel approaches to feature representation that suited the target hardware.

A.4 Functional requirements: Compute-aware radio scheduling



Risk Level: 1 2 3 4 5

Requirement Type: Functional | Non-Functional

Risk Management: Successful | Effective | Partial

Percent Complete: 100%-0%

FR-CAWRS-000	
Description	DAEMON shall integrate NI solution in vRAN systems
Version	001M3
Owner	UC3M
Priority	High
Risk	1
Risk Description	There is a low risk that DAEMON will not integrate NI solutions into vRAN systems, as DAEMON partners were already capable of integrating such kinds of solutions in Open Source vRAN environments.
Rationale	The mobile network industry is moving towards virtual network function solutions, and RAN Functions are not an exception. Being among the most resource-consuming functions (in terms of computation), thus allowing the re-design of such function by taking into account the computing resource optimization as further objective will improve the overall spending (both CAPEX and OPEX, for the resource provisioning) for the network operation.

K1	X	K2	X	K3		K4	X	K5	X	K6		K7		K8		K9	

Parents	None
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	Several solutions have been provided already in D3.2 [2] (Section 3) and will be extended in D3.3

FR-CAWRS-001	
Description	DAEMON NI solutions for vRAN systems shall integrate predictive HARQ.
Version	001M17
Owner	i2CAT
Priority	High
Risk	1
Risk Description	There is a low risk that DAEMON will not integrate predictive HARQ solutions, because they have been widely studied in other contexts before.
Rationale	Predictive HARQ mechanisms collect data from the subframe decoding process and make a prediction about the decodability of the corresponding transport blocks. This enables the usage of transport blocks that otherwise would have been dropped because they were not decoded on time.

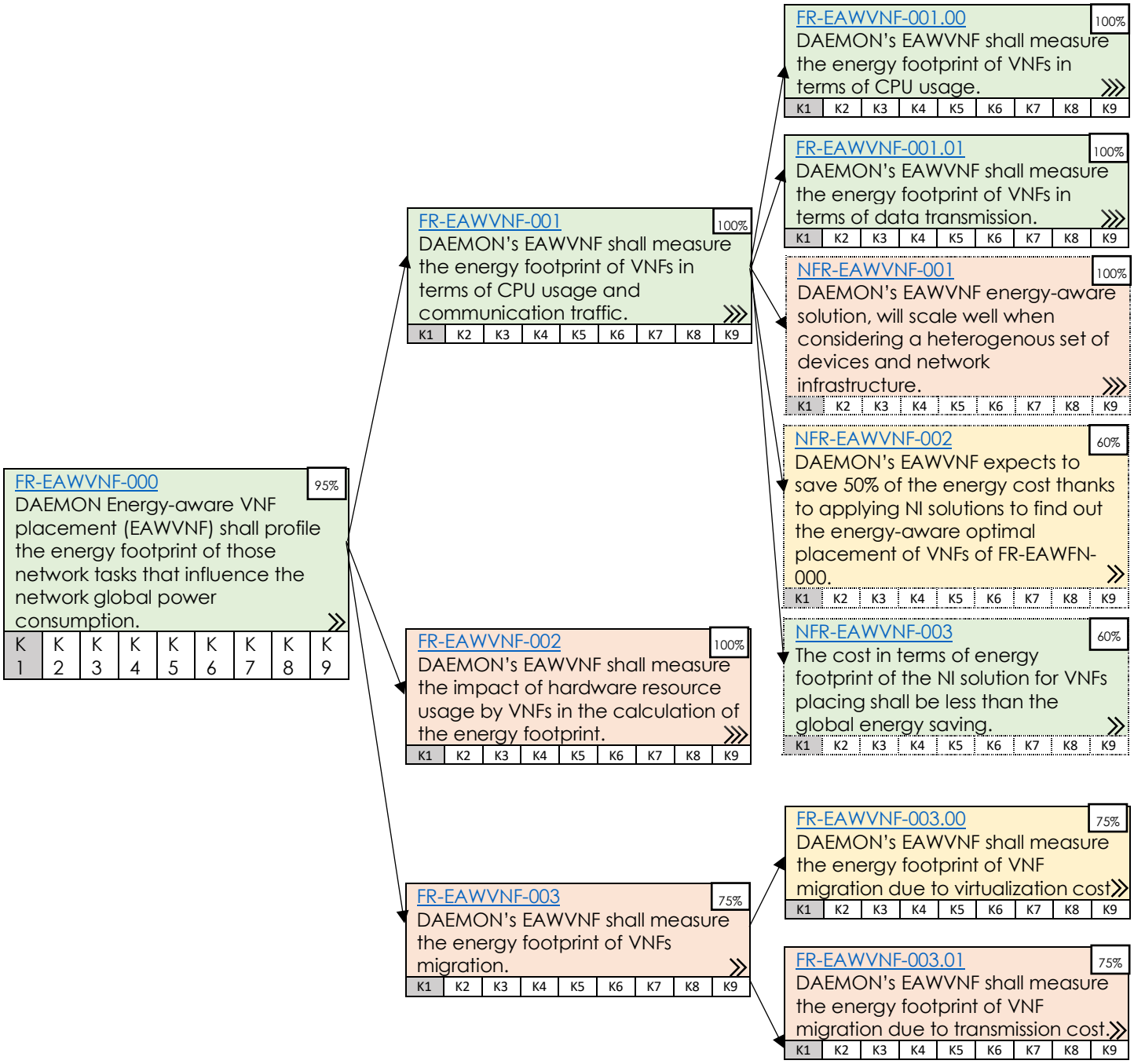
K1	X	K2	X	K3		K4	X	K5	X	K6		K7		K8		K9	

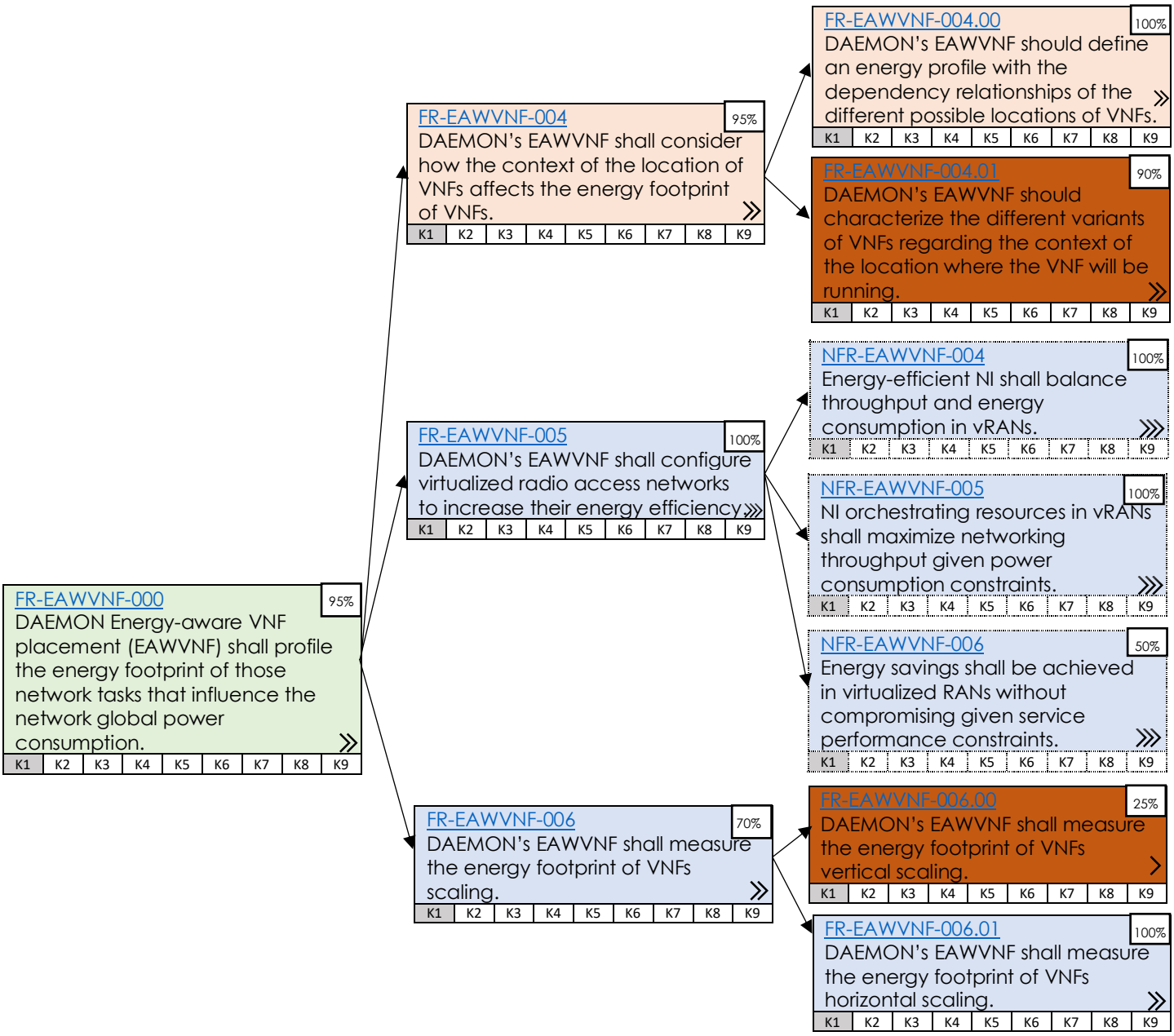
Parents	FR-CAWRS-000-001M3
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	The requirement has been fulfilled as already demonstrated in D5.2 [4], Section 4.1.1.2

FR-CAWRS-002	
Description	DAEMON NI solutions for vRAN systems shall integrate intelligent algorithms to allocate radio and computing resources in real-time.
Version	001M17
Owner	i2CAT
Priority	High
Risk	3
Risk Description	There is a medium risk in integrating intelligent algorithms for radio and computing resource allocation in real time because of the reduced operation timescale.
Rationale	Intelligent radio and computing allocation algorithms provide mechanisms to efficiently distribute the available radio and computing resources, being at the

				same time crucial for providing latency guarantees and for maximizing the performance of the overall system.															
K1		**K2**	**X**	**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents				FR-CAWRS-000-001M3															
Current Status																			
Percent complete				100%															
Risk management				Successful															
Rationale				The risk described above was mitigated by implementing light rule-based NI rather than complex neural network-based solutions.															

A.5 Functional requirements: Energy-aware VNF placement





Risk Level: 1 2 3 4 5

Requirement Type: Functional | Non-Functional

Risk Management: Successful | Effective | Partial

Percent Complete: 100%-0%

FR-EAWVNF-000																			
Description	DAEMON Energy-aware VNF placement (EAWVNF) shall profile the energy footprint of those network tasks that influence the network global power consumption.																		
Version	001M1																		
Owner	UMA																		
Priority	High																		
Risk	2																		
Risk Description	The reliability of the measurement depends on a complete identification of the external factors that affect the energy footprint (e.g., temperature, processor, or noisy neighbor problem), the accuracy of the energy measurement methods used, and the dependency on specific hardware. We should be able to estimate the energy consumption of VNFs both in simulated and real environments, obtaining possibly similar results.																		
Rationale																			
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	None																		
Current Status																			
Percent complete	90%																		
Risk management	Effective																		
Rationale	In order to meet the goal of energy efficiency we have to identify the list of elements that strongly affect the energy consumption in the edge context. We have implemented the SAVRUS algorithm that works with unknown-domain (in this case, VNFs foredge-based mobile networks) to identify and rank the main network features and their interactions by how much they affect the final energy consumption (D3.1 [5], Section 5.3.2.1, D3.2 [2], Section 3.3.3, and journal [72]). We evaluated the SAVRUS strategy with experiments that provide completely measured models, by properly degrading the completely measured spaces to represent incomplete measures space, which are the measures spaces under SAVRUS works. Regarding the construct validity of the data set, the degradation procedure was automatic and random and was independently applied to the original spaces several times. Consequently, we analyzed the same space many times but degraded it differently to minimize the collateral effects that the degradation procedure could have on the results. There is a risk to the internal validity, since the selected sampling and learning methods may not be the best choice for all systems. To mitigate this risk, we did not only validate SAVRUS outputs but individually reviewed each strategy component to avoid hidden errors. We repeated the analysis and presented average metrics to reduce a possible bias. Also, SAVRUS comprises a normality test within the process with a 95% confidence.																		

FR-EAWVNF-001	
Description	DAEMON's EAWVNF shall measure the energy footprint of VNFs in terms of CPU usage and communication traffic.
Version	001M2
Owner	UMA
Priority	High
Risk	2
Risk Description	The reliability of the measurement depends on the ability to identify and quantify the influence of external factors in the energy consumption calculation (e.g., noisy neighbor problem or distance to base station). Calculating the cost of executing code and transmitting and receiving information on specific hardware accurately is a complex task. It is possible to mitigate this risk by going through calculating an upper bound of its energy footprint.
Rationale	The main factors that influence energy consumption are CPU usage and the data sent and received by a given VNF. We need to identify what are the factors that should be considered in the formula that calculates the total energy footprint of the network, in terms of computation and communication. We should consider not only the internal factors, as we said, computation and communication, but also the external ones, such as the neighboring traffic. Since our main goal is not to report absolute energy footprint values, but relative ones, we need to find a

	sound method to quantify the revenue of placing a VNF in one or another location in terms of power saving.
K1 X **K2** **K3** **K4** **K5** **K6** **K7** **K8** **K9**	
Parents	FR-EAWVNF-000
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	In the placement solution presented in D4.2 [3] the energy model used to estimate energy consumption includes both CPU and data transmission.

FR-EAWVNF-001.00	
Description	DAEMON's EAWVNF shall measure the energy footprint of VNFs in terms of CPU usage.
Version	001M2
Owner	UMA
Priority	High
Risk	2
Risk Description	Calculating the cost of executing any kind of code, on specific hardware accurately is a complex task, since there are several factors that we need to quantify in order to calculate the energy footprint. The theoretical values given by CPU providers usually do not coincide with the real ones.
Rationale	We need to identify what are the factors that should be considered in the formula that calculates the global energy footprint of the VNFs instantiated for each application, in terms of computation. We know that the processor type of the device where a VNFs is running influences the energy footprint, but there are also other parameters that make the software provoke the hardware to consume more energy, like the size of VNF input.
K1 X **K2** **K3** **K4** **K5** **K6** **K7** **K8** **K9**	
Parents	FR-EAWVNF-001
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	In the placement solution presented in D4.2 [3] the energy model used to estimate energy consumption includes explicitly the energy cost of computation calculated from the CPU cycles and the CPU frequency along with other factors. In the placement and autoscaling solution presented in D4.2 [3] the energy consumption model calculates the energy footprint of VNFs in terms of CPU usage according to the node in which VNFs are going to be deployed.

FR-EAWVNF-001.01	
Description	DAEMON's EAWVNF shall measure the energy footprint of VNFs in terms of data transmission.
Version	001M2
Owner	UMA
Priority	High
Risk	2
Risk Description	Calculating the cost of data transmission over different types of network links, accurately is a complex task, since there are several factors that we need to quantify in order to calculate the energy footprint. The network throughput is something that varies a lot and depends on some external factors like the current traffic or transmitting neighboring devices.
Rationale	In the energy footprint calculation, we need to consider that some VNFs will produce some data that might need to be transmitted to other devices. We know that the transmission power, the payload and the transmission rate should be considered, along with other terms.
K1 X **K2** **K3** **K4** **K5** **K6** **K7** **K8** **K9**	
Parents	FR-EAWVNF-001

Current Status	
Percent complete	100%
Risk management	Successful
Rationale	In the energy-aware placement solution presented in D4.2 [3] the energy model includes explicitly the energy cost of data communication calculated in the amount of bytes sent and received along with other terms. In the autoscaling and placement solution in D4.2 [3] the energy consumption model calculates the energy footprint of VNFs in terms of the amount of data transmitted (sent and received) according to the node in which VNFs are deployed.

FR-EAWVNF-002																			
Description	DAEMON's EAWVNF shall measure the impact of hardware resource usage by VNFs in the calculation of the energy footprint.																		
Version	001M2																		
Owner	UMA																		
Priority	Low																		
Risk	4																		
Risk Description	The accuracy of the energy consumption measurement depends on specific hardware, including not only the computing device processor. Other hardware, such as memory use or access to HDD, could also influence the total energy footprint, but it is difficult to assess in which percentage. So, it is not easy to estimate it accurately.																		
Rationale	Measuring the hardware resources usage of VNFs and their energy footprint provides extra information to accurately estimate the overall energy footprint of a VNF. We are seeking to find additional factors to the energy consumption formula, to calculate more precisely the network energy footprint. Although the DAEMON approach does not need absolute values of energy consumption, we need to find out if there are certain situations where the excessive use of additional resources by a certain VNF strongly impacts the decision of, for example, migrating it to another location.																		
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-EAWVNF-000																		
Current Status																			
Percent complete	100%																		
Risk management	Successful																		
Rationale	The solution presented in D4.2 [3] monitors the impact of hardware resources of VNFs in the estimation of the energy consumption. The solution also considers the computation, communication, and storage resources as part of placement algorithm.																		

FR-EAWVNF-003																			
Description	DAEMON's EAWVNF shall measure the energy footprint of VNFs migration.																		
Version	001M2																		
Owner	UMA																		
Priority	High																		
Risk	4																		
Risk Description	The cost in terms of energy consumption of code migration in general, and in particular considering VNFs, depends on several factors that we need to identify. Also, there are different mechanisms to perform code migration and each of them requires a different formula for energy footprint calculation, affecting the accuracy of the final result.																		
Rationale	The migration of a certain VNF has an energy cost that should be analyzed. It is essential to understand this energy cost to prioritize migrations to other systems if needed.																		
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-EAWVNF-000																		
Current Status																			
Percent complete	75%																		

Risk management	Partial
Rationale	The energy consumption model used in the proactive autoscaling solution and VNF placement presented in D4.2 [3] can be used to calculate the energy footprint of VNF migration due to virtualization costs and data transmission cost as virtualization scaling cost is considered in this solution.

FR-EAWVNF-003.00	
Description	DAEMON's EAWVNF shall measure the energy footprint of VNF migration due to virtualization cost.
Version	001M2
Owner	UMA
Priority	High
Risk	3
Risk Description	The main risk is that we do not consider all the factors relative to virtualization that affect the energy consumption of migrating a certain VNF. Another risk is that even when we find a formula to calculate this energy footprint for a certain virtualization technology (or a few of them), later new technologies may appear.
Rationale	Virtualization has an energy cost and should be analyzed. We should find out if this cost depends on the device (mainly Edge devices and Cloud), and how we can calculate it for both simulated and real environments. It is essential to understand this energy cost to prioritize migrations to other systems if needed. Also, we need to choose the list of virtual machines we are going to consider in this requirement.

K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**	

Parents	FR-EAWVNF-003
Current Status	
Percent complete	75%
Risk management	Partial
Rationale	The energy consumption model used in the proactive autoscaling solution and VNF placement presented in D4.2 [3] can be used to calculate the energy footprint of VNF migration due to virtualization costs as virtualization scaling cost is considered in this solution.

FR-EAWVNF-003.01	
Description	DAEMON's EAWVNF shall measure the energy footprint of VNF migration due to transmission cost.
Version	001M2
Owner	UMA
Priority	High
Risk	4
Risk Description	The main risk is that we do not consider all the factors relative to virtualization that affect the energy consumption of migrating a certain VNF. Another risk is that even when we find out a formula to calculate this energy footprint for a certain virtualization technology (or a few of them), later new technologies appear.
Rationale	The main factors that affect the energy footprint of VNF migration are the code and data transmission. There are different mechanisms to move a VNF to a different location and each one implies transferring more or less data. So, the code migration mechanism strongly influences the energy footprint since it varies the amount of information to be transmitted. We should find out how we can calculate it for both simulated and real environments. It is essential to understand this energy cost to prioritize migrations to other systems if needed. Also, we need to decide on a single migration mechanism, if possible, to be able to calculate its energy footprint.

K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**	

Parents	FR-EAWVNF-003
Current Status	
Percent complete	75%

Risk management	Partial
Rationale	The energy consumption model used in the proactive autoscaling solution and VNF placement presented in D4.2 [3] can be used to calculate the energy footprint of VNF migration due to data transmission costs as data transmission cost is considered in this solution and can be used to calculate transfer cost in terms of data sent and receive.

FR-EAWVNF-004																			
Description	DAEMON's EAWVNF shall consider how the context of the location of VNFs affects the energy footprint of VNFs.																		
Version	001M3																		
Owner	UMA																		
Priority	High																		
Risk	4																		
Risk Description	The main risk is not modeling the context properly due to external and non-measurable artifacts. Moreover, DAEMON could not capture all the possible scenarios related within the location context to model the data's location to feed																		
Rationale	The VNF placement cost in terms of energy footprint should consider the execution context where a VNF will be running, and the location of the data that will feed this function. The goal is to adapt the energy footprint of the needed VNFs to the context of the location where they are running.																		
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-EAWVNF-000																		
Current Status																			
Percent complete	95%																		
Risk management	Effective																		
Rationale	D4.2 [3] presents an energy-aware placement solution for VNFs, considering the execution location context. This solution selects nodes based on an energy profile considering available RAM, storage, and hardware configurations. It aims to reduce energy consumption while meeting infrastructure needs. The solution includes an energy-aware orchestrator that assigns VNFs to the most energy-efficient nodes. It also considers the energy cost of computing VNF placement as part of the overall energy footprint, placing VNFs in an energy-efficient manner throughout the infrastructure.																		

FR-EAWVNF-004.00																			
Description	DAEMON's EAWVNF should define an energy profile with the dependency relationships of the different possible locations of VNFs.																		
Version	001M3																		
Owner	UMA																		
Priority	High																		
Risk	4																		
Risk Description	One possible risk is that we cannot capture all the possible scenarios related to the location context. The variability of execution location contexts and their relationship with the energy footprint could be so high that it is not possible to consider all the cases in the AI algorithms that compute the best solution to deploy a set of VNFs.																		
Rationale	To compute the energy footprint of a VNF we need to consider the energy cost depending on the location of the input data, and also the context of the execution location. One possible context could be the quality of the energy consumed, if it is green and renewable energy or polluting energy.																		
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-EAWVNF-004																		
Current Status																			
Percent complete	100%																		
Risk management	Effective																		

Rationale	The energy-aware placement solution in D4.2 [3] has the specificity of an energy profile considering requirements that are related to the execution location context, such as the available Random Access Memory (RAM), storage, or specific hardware/server configurations. Based on the different constraints above, the solution allows selecting the nodes where VNFs can be run to reduce energy consumption while meeting the needs of the infrastructure. In addition, one of the modules that are part of the proactive auto-scaling solution and VNF placement is the energy-aware orchestrator, which calculates the energy consumption according to the location of the VNFs and assigns the applications/VNFs to the most energy-efficient node.

FR-EAWVNF-004.01																		
Description	DAEMON's EAWVNF should characterize the different variants of VNFs regarding the context of the location where the VNF will be running.																	
Version	001M3																	
Owner	UMA																	
Priority	High																	
Risk	5																	
Risk Description	Sometimes the proposed solutions for energy saving cost about the same or sometimes even more than applying a non-energy aware policy. So, we need to assess the cost of computing NI solutions in terms of energy, by adding this cost to the global energy footprint of the solution proposed by DAEMON.																	
Rationale	The energy footprint of a VNF could depend on the energy cost of getting the input information depending on the location of the input data. Sometimes, the best solution could be to migrate the VNFs, but other times DAEMON could propose to adapt VNFs so that we can instantiate the most energy-efficient version.																	
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**		
Parents	FR-EAWVNF-004																	
Current Status																		
Percent complete	90%																	
Risk management	Effective																	
Rationale	A solution in D4.2 [3] considers the energy cost of computing VNF placement, which is considered part of the global energy footprint of the solution. The placement decision considers the computational and communication energy consumption of VNFs based on their location in the infrastructure to place them in an energy-efficient manner.																	

FR-EAWVNF-005																		
Description	DAEMON's EAWVNF shall configure virtualized radio access networks to increase their energy efficiency																	
Version	001M17																	
Owner	NEC																	
Priority	High																	
Risk	1																	
Risk Description	There is a risk that DAEMON will be unable to configure virtualized radio access networks. This risk is low because O-RAN specification shall permit this.																	
Rationale	RAN virtualization promises high flexibility and lower costs but current virtualization techniques render higher energy consumption in the RAN. Hence, it is of paramount importance to configure virtualized base stations with their energy consumption in mind																	
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**		
Parents	FR-EAWVNF-000																	
Current Status																		
Percent complete	100%																	
Risk management	Successful																	
Rationale	The risk was low. The design of NI to configure virtualized radio access networks with energy-driven goals was presented in D4.2 [3], Section 2.3, and additional NI																	

	will be presented in D4.3, with an NI that jointly controls virtualized radio access networks and edge services. Details can be found in [18], [40], [63].

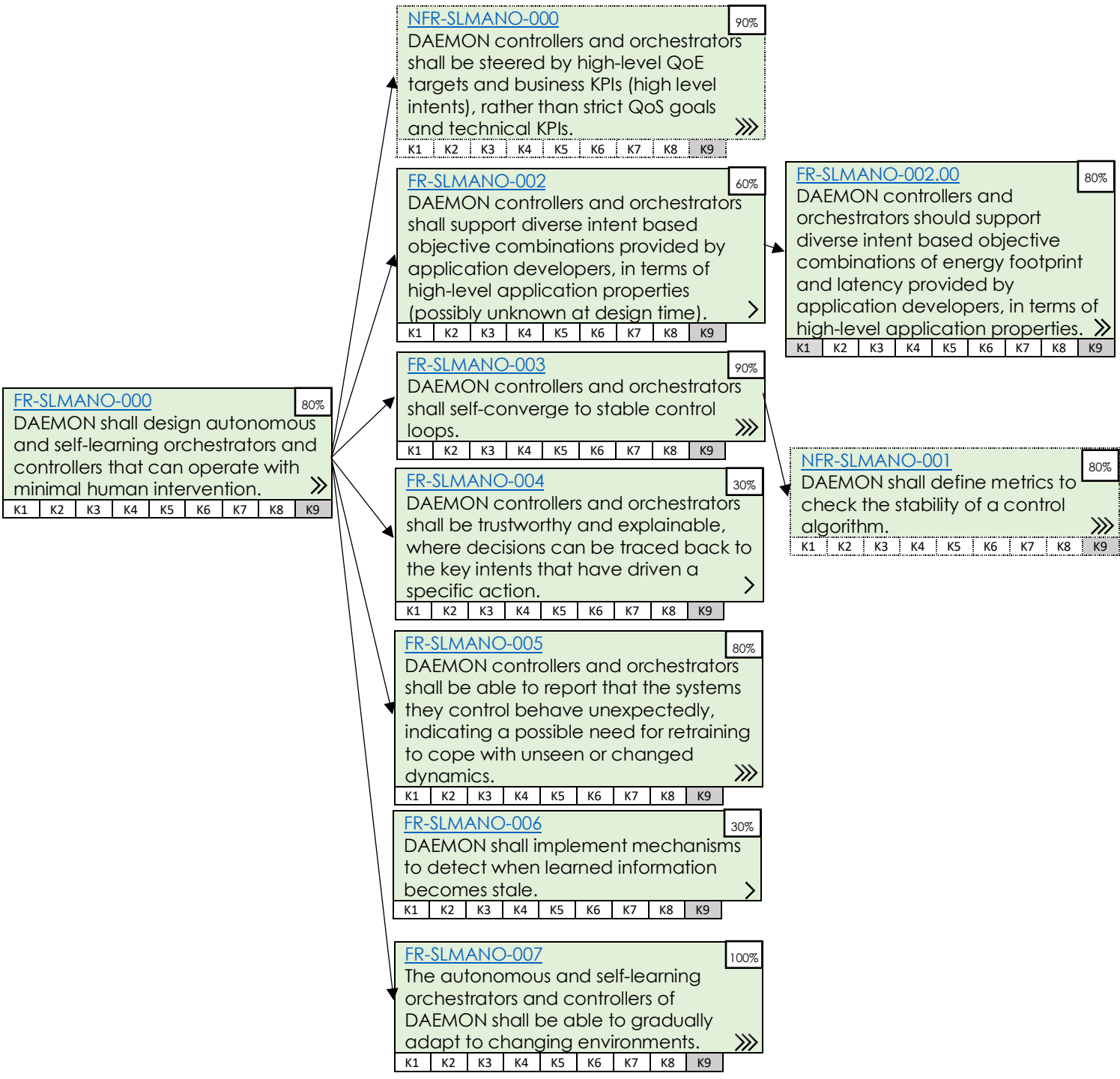
FR-EAWVNF-006																			
Description	DAEMON's EAWVNF shall measure the energy footprint of VNFs scaling.																		
Version	001M18																		
Owner	UMA																		
Priority	Medium																		
Risk	2																		
Risk Description	The cost in terms of energy consumption of VNFs scaling, depends on several factors that we need to identify. Depending on the approach used to calculate or estimate energy footprint the accuracy of the final result will be more or less adjusted to reality.																		
Rationale	there are different proposals to perform VNF scaling and each of them needs to incorporate an energy profile to calculate or estimate the energy footprint. Scaling up or down a certain VNF frequently according to a dynamic demand has an energy cost that should be analyzed. It is essential to understand this energy cost to prioritize if scaling up and down is needed.																		
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-EAWVNF-000																		
Current Status																			
Percent complete	70%																		
Risk management	Effective																		
Rationale	In D4.2 [3], a proactive autoscaling solution and VNF placement consider the energy consumption of horizontal scaling explicitly to optimize VNF placement. Also, it will be extended to consider vertical scaling.																		

FR-EAWVNF-006.00																			
Description	DAEMON's EAWVNF shall measure the energy footprint of VNFs vertical scaling.																		
Version	001M18																		
Owner	UMA																		
Priority	Low																		
Risk	5																		
Risk Description	The cost of VNF vertical (up/down) scaling to augment (i.e., scale up) the provision of VNF resources depends on several factors that we need to identify. Also, there are different approaches to perform VNF resource allocation and each of them requires a different formula for energy footprint calculation, affecting the accuracy of the final result.																		
Rationale	The cost of VNF vertical scaling to augment (i.e., scale up) the provision of VNF resources has an energy cost that should be analyzed. It is essential to understand the energy cost of resource provision to decide when VNF vertical scaling is needed, while taking into account the cost of resource provision actions.																		
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-EAWVNF-006																		
Current Status																			
Percent complete	25%																		
Risk management	Partial																		
Rationale	Our proactive energy consumption model presented in D4.2 [3] will be extended to calculate the energy footprint of VNFs' vertical scaling, but it has not been tested nor validated yet.																		

FR-EAWVNF-006.01	
Description	DAEMON's EAWVNF shall measure the energy footprint of VNFs horizontal scaling.
Version	001M18
Owner	UMA
Priority	Low
Risk	2

Risk Description																			
The main risk is that we do not consider all the factors relative to VNFs horizontal scaling due to virtualization that affects the energy consumption of scaling a certain VNF. Another risk is that, even when we find a formula to calculate this energy footprint for a certain virtualization technology (or a few of them), later new technologies may appear.																			
Rationale																			
The horizontal scaling (I.e., in/out) of a VNF has an energy cost due to the virtualization process, which should be analyzed. DAEMON should propose mechanisms to find out the elements that influence this energy cost, such as the HW of target devices (mainly Edge devices and Cloud). Also, DAEMON will propose mechanisms to estimate the energy footprint for both simulated and real environments. It is essential to understand this energy cost to prioritize horizontal scaling if needed.																			
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents																			
FR-EAWVNF-006																			
Current Status																			
Percent complete																			
100%																			
Risk management																			
Effective																			
Rationale																			
The proactive autoscaling solution and VNF placement presented in D4.2 [3] explicitly consider the energy consumption of horizontal scaling to optimize VNF placement. This solution considers both the base (idle) and dynamic (due to application execution) energy consumption of the nodes, as well as the energy consumption of node scaling.																			

A.6 Functional requirements: Self-learning MANO



FR-SLMANO-000	
Description	DAEMON shall design autonomous and self-learning orchestrators and controllers that can operate with minimal human intervention.
Version	002M17
Owner	NBL
Priority	High
Risk	2
Risk Description	Only regularly repeating patterns can be learned. Stochastic fluctuation on top of these regular patterns hamper learning and need to be filtered out. The learning rate, number of epochs (the number of times that is run through the data) and exploitation versus exploration balance need to be carefully chosen. Moreover, the behavior of the system can change either slowly (as the system evolves) or suddenly (when, e.g., new software is installed on some of the components. Both need to be handled.
Rationale	Any decision that the orchestration and control functions can be envisioned to be automized in the following way. First, the software agent taking the decisions needs to be provided (in a timely way) with the data necessary to take its decisions. The agent relies on the policy currently in force to take the appropriate action. With each action taken (given the provided data) the agent is provided with feedback that expresses how good that action was given the current data. Based on this feedback, the agent can change its policy to steer the system in the desired direction.

K1		K2		K3		K4		K5		K6		K7		K8		K9	X

Parents	None.
Current Status	
Percent complete	80%
Risk management	Effective
Rationale	We have studied the SLMANO components, i.e., placement/routing of requests for new network services and scaling/life-cycle management of existing network services under various traffic loads, as reported in Section 6.1.4 of D4.1 [6] and Section 4.4.7 of D5.1 [7]), respectively. Random noise was added to the regular patterns and flash crowds were introduced (see the talk on DAEMON's 1st industrial workshop). Under all circumstances the components behaved robustly. All these loads were artificially generated because, as far as we know, no real loads are available yet. Once real data becomes available an additional round of robustness checks with these real loads will be needed. We also studied the essential building block of an application-aware RAN (in D3.1 [5], D3.2 [2]), which classifies traffic in a finite number of classes (data, video, web, podcast, …), based on a labeled data captured in a testbed. The classifier, which is the essential component in the Application Aware RAN (AAR), operates without any human intervention (once the labeled data is captured): it is trained via a supervised learning technique (and interrupts automatically before it is overtrained), while the interference is operated without any human interaction.

FR-SLMANO-002	
Description	DAEMON controllers and orchestrators should support diverse intent based objective combinations provided by application developers, in terms of high-level application properties (possibly unknown at design time).
Version	002M17
Owner	NBL
Priority	High
Risk	2
Risk Description	Can a single algorithm fulfill this requirement for all use cases (e.g., URLLC (ultra-low latency reliable communication), EMBB (enhanced mobile broadband), and MMTC (massive machine type communication) defined in 5G)? Will it be too complex? Is it better to train multiple competing algorithms for each specific use case and select the best performing?

Rationale	Business KPIs will change frequently due to highly varying markets. The algorithms provided by DAEMON need to be flexible enough to self-learn and converge to sufficiently optimized behavior to avoid human intervention for retuning or redesigning the algorithms and mechanisms. If a classical algorithm still has parameters to tune a procedure, to tune (i.e., learn) these parameters need to be designed and investigated.

K1		**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**	X

Parents	FR-SLMANO-000
Current Status	
Percent complete	70%
Risk management	Partial
Rationale	We have tested the placement/routing and scaling algorithms with a variety of workloads (see FR-SLMANO-000) and under differentiated latency bounds (to be reported in D5.3) and noticed that the proposed algorithms perform well. Unfortunately, there was no real data available to us pertaining to ULLRC, eMMB and MMTC slices, so that beyond the test on a wide variety of artificially generated traces, tests with real traffic were not possible. We have also trained and (cross-)tested the classifier, which plays a central role in the application-aware RAN, on two different datasets, i.e., one captured with tail-drop buffers, and one captured with L4S activated (see D3.3 and D5.3), the former of which aims at maximizing throughput, while the latter of which respects a strict latency bound. Ideally, the AAR classifier allows fine-tuning the traffic differentiation based on class-specific intents, e.g., low latency for XR, high throughput for bulk downloads. Beyond latency and throughput, we did not consider high-level application properties.

FR-SLMANO-002.00	
Description	DAEMON controllers and orchestrators should support diverse intent based objective combinations of energy footprint and latency provided by application developers, in terms of high-level application properties.
Version	002M17
Owner	UMA
Priority	High
Risk	2
Risk Description	The energy consumption calculation of placement decision should take into consideration also the requirements in terms of the latency/time response/timescale needs of service function chains. The algorithm should find the best tradeoff between the most fitting timescale and the energy-saving requirements for the VNF placement.
Rationale	Following the cloud-native service design, the service function chains consist of loosely coupled VNFs that can be separately placed. Orchestration managers can make the VNFs placement decisions, while maintaining the service provision. The algorithms provided by DAEMON that make orchestration decisions should consider latency/time response/timescale needs of service function chains and at the same time the energy footprint. So, to make a tradeoff between time and energy footprint is desired.

K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**	X

Parents	FR-SLMANO-002
Current Status	
Percent complete	85%
Risk management	Effective
Rationale	The CQL framework [73] supports advanced optimization by supporting multi-objective combinations of quality attributes, such as latency and energy consumption. In the evaluation of the CQL framework and to control randomness we repeated the experiments 97 times and averaged the results for a confidence level of 95% with a 10% margin of error. External validity is that not all the evaluated systems are NFVs, but well-known models with registered complex quality measurements are rare in the SDN literature. Consequently, by choosing the real-

	world systems selected in the evaluation we pretended to cover a variety of properties, quality attributes and functions commonly found in VNF cases. Nonetheless, we are aware that they do not cover every possible casuistic individually. Moreover, while one could claim that larger systems should be tested, we should mention that larger spaces are very rare for VNF orchestrators. The problem in SDN systems is the complexity of the reasoning and not the size of it. Testing our algorithms with only one Category Theory reasoner could be another threat. The problem is that Category Theory tools are rare due to the intrinsic abstraction and knowledge requirement.

FR-SLMANO-003																		
Description	DAEMON controllers and orchestrators should self-converge to stable control loops.																	
Version	002M17																	
Owner	NBL																	
Priority	High																	
Risk	3																	
Risk Description	Can we capture realistic dynamic behavior in the systems, emulators or simulators that we will use? How can stability be verified?																	
Rationale	Different parts of the system will have different dynamics, potentially changing over time due to SW and HW upgrades. The algorithms provided by DAEMON need to be intelligent enough to self-learn and converge to a stable though responsive behavior without human intervention for retuning or redesigning the algorithms and mechanisms. In general, a system operating in a steady state is stable if after an infinitesimally short, small enough perturbation applied to it dies out exponentially fast so that it returns to that steady state working point. The perturbation needs to be short compared to the reaction time inherent to the system and small so that it does not jump to another working point.																	
K1		**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**	X	
Parents	FR-SLMANO-000																	
Current Status																		
Percent complete	90%																	
Risk management	Successful																	
Rationale	We used a practical definition of stability: as long as small perturbations on the input, did not lead to growing fluctuations at the output, we called the system stable. For the closed-loop scalers we noticed that in all cases we considered there was a setting of the parameters that yielded stable operation (see Section 4.4.7 of D5.2 [4]). However, during training of the reinforcement learning scaler and the tuning of the parameters of the proportional integral scalers, some parameter settings lead to unstable operation. Whenever such a situation occurred, we interrupted the learning process and restarted with new initial parameters. This procedure turned out to be sufficient to end up with a stable scaler.																	

FR-SLMANO-004	
Description	DAEMON controllers and orchestrators should be trustworthy and explainable, where decisions can be traced back to the key intents that have driven a specific action.
Version	003M17
Owner	NBL
Priority	Low
Risk	5
Risk Description	Often ML tools are black boxes that after training work well but do not give any indication of why they work. Human network operators might distrust such tools and hence, be reluctant to use them. Moreover, when decisions are taken at multiple layers at different timescales, conflicts may arise amongst agents operating at different timescales (possibly due to a human error when setting the goals).

Rationale	In a complex composition of multi-layer controllers, conflicts between different levels of intents need to be visualized, such that unexpected unwanted behavior can be analyzed and revised in terms of the active and potentially erroneously specified intents (due to human errors).

K1		**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**	X

Parents	FR-SLMANO-000
Current Status	
Percent complete	30%
Risk management	Partial
Rationale	We started this work in the 3rd year, and we are still working on it. If there is progress, we will update this later.

FR-SLMANO-005	
Description	DAEMON controllers and orchestrators should be able to report that the systems they control behave unexpectedly, indicating a possible need for retraining to cope with unseen or changed dynamics.
Version	001M2
Owner	NBL
Priority	High
Risk	3
Risk Description	There is a risk that spurious changes are seen by the system as changes in the environment, causing the system to retrain (doing a lot of exploration and making the associated wrong decisions) where it is not needed.
Rationale	If online retraining is prohibited, or if the algorithms are incapable of self-converging to a sufficient solution, the need for human intervention needs to be reported.

K1		**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**	X

Parents	FR-SLMANO-000
Current Status	
Percent complete	80%
Risk management	Successful
Rationale	The classifier associated with the application-aware RAN was trained on two different datasets (describing the evolution of the RLC buffer under various types of traffic), i.e., one dataset in which the RLC buffer was governed by tail drop and another dataset in which the RLC buffer was governed by L4S active queue management to keep the RLC buffer (and hence. The latency incurred over that buffer) small. It was shown that training the classifier on one data set and testing on another yielded a low performance (will be updated in D5.3), indicating that detecting a change in buffer acceptance mechanism can be easily detected when observing the performance of the traffic classifier.

FR-SLMANO-006	
Description	DAEMON shall implement mechanisms to detect when learned information becomes stale.
Version	002M17
Owner	NBL
Priority	Medium
Risk	3
Risk Description	The behavior of the system can change suddenly (when, e.g., a flash crowd arrives generating a lot of traffic, when new software is installed on some of the components, when there is an outage of part of the infrastructure), which makes that the policy learned on past system behavior is no longer applicable. Therefore, a system is needed to detect when learned information becomes stale indicating when retraining is required.
Rationale	In the framework defined under FR-SLMANO-000-001M2, where the software agent taking the decisions is provided (in a timely way) with the data necessary to take its decisions and where with each action taken (given the provided data),

	the agent is provided with quantitative feedback that expresses how good that action was, a change in behavior can be detected by observing the evolution of the feedback. If there is a drastic change, the balance between exploration and exploitation needs to be tilted in favor of exploration so that the system can be retrained to work properly in the new environment.

K1		K2		K3		K4		K5		K6		K7		K8		K9	X

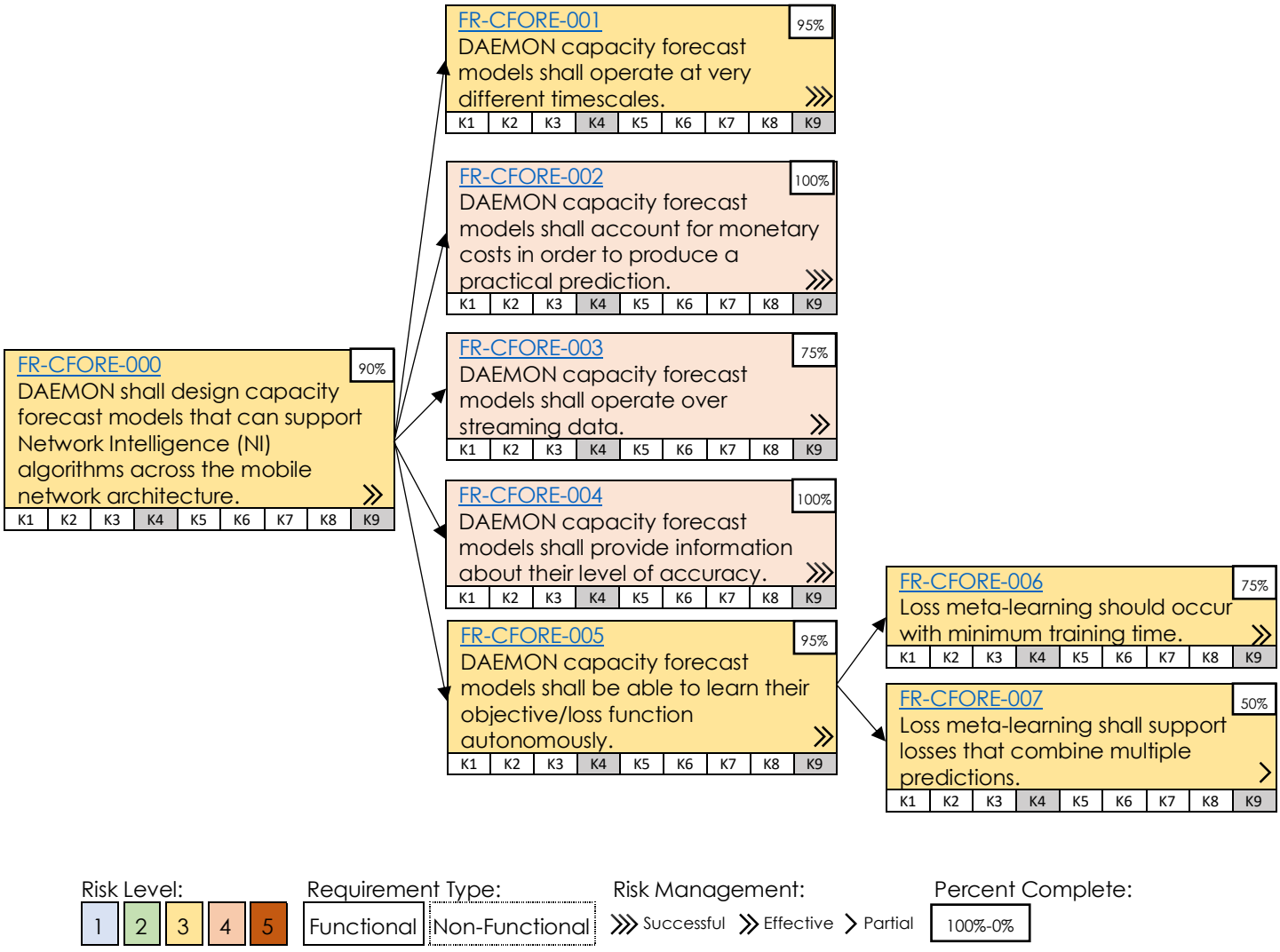
Parents	FR-SLMANO-000
Current Status	
Percent complete	30%
Risk management	Partial
Rationale	In the study of how to segment a service function chain (SFC), in which the segments are assigned to various datacenters, based on a distributed multi-agent reinforcement learning (DMARL) technique we gradually decreased the balance between exploitation and exploration (see Section 4.2.6 of D5.2 [4] and [61]). We chose the decrease in such a way that the system was able to learn the desired behavior. This turned out to be trickier than we initially expected, so not enough time remained in the scope of the project to investigate how this balance should be reinstated when the traffic load would drastically change.

FR-SLMANO-007	
Description	The autonomous and self-learning orchestrators and controllers of DAEMON shall be able to gradually adapt to changing environments.
Version	001M17
Owner	NBL
Priority	Medium
Risk	2
Risk Description	The behavior of the system can change slowly together with the usage patterns. DAEMON self-learning MANO needs to follow these changes otherwise the decisions it takes will gradually become worse. This can be achieved by setting a good balance between exploration and exploitation.
Rationale	A learning system that relies on feedback to improve its policy, can gradually learn by taking from time-to-time exploratory actions (i.e., random actions which are deemed not to be optimal by the current policy). Usually, the fraction of exploration actions is large (close to 100%) at the start of the learning process, and gradually reduces to 0 as the system learns. In order to be able to adapt to a changing environment, the exploration fraction is kept at, say, 10%.

K1		K2		K3		K4		K5		K6		K7		K8		K9	X

Parents	FR-SLMANO-000
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	In the study of how to segment a service function chain (SFC), in which the segments are assigned to various datacenters, based on a distributed multi-agent reinforcement learning (DMARL) technique we determined how to decrease the balance between exploitation and exploration (see Section 4.2.6 of D5.2 [4] and [61]). We have spent a lot of time investigating what the optimal decrease is, avoiding, on the one hand, that the system gets stuck in a suboptimal policy and, on the other hand, that it takes random decisions for too long a time.

A.7 Functional requirements: Capacity forecasting



FR-CFORE-000																		
Description	DAEMON Capacity Forecasting (CFORE) shall design models capable of anticipating the amount of resources needed to accommodate future mobile service demands, so as to support Network Intelligence (NI) algorithms across the mobile network architecture.																	
Version	001M2																	
Owner	IMDEA																	
Priority	High																	
Risk	3																	
Risk Description	The main risk is that the forecasting models do not achieve the accuracy needed to support efficient decision-making, hence limiting the effectiveness of NI.																	
Rationale	Many decisions to be taken by orchestrators and controllers deployed across different micro-domains of the mobile network must be taken in an anticipatory manner, i.e., proactively, with respect to the actual demand or requirements. Such decisions concern the capacity that orchestrators and controllers shall allocate in their micro-domain of competence. Predicting such capacity is thus a key enabler for the NI operating across the whole network.																	
K1		**K2**		**K3**		**K4**	X	**K5**		**K6**		**K7**		**K8**		**K9**	X	
Parents	None																	
Current Status																		
Percent complete	90%																	
Risk management	Effective																	
Rationale	DAEMON has developed models for capacity forecasting that can follow original design guidelines and allow effectively allocating resources in an anticipatory fashion. Extensive performance evaluations have demonstrated the accuracy of the models. Details on the design and evaluation are provided in D2.2 [1], D4.2 [3] and D5.2 [4], in addition to refinements that are developed in the last iteration of the project, as presented in Section 7.1.1 of the present document and later complemented in D4.3 and D5.3. Additional details are reported in the lower-level requirements, where we discuss how the developed solutions meet the requirements and what additional steps must be taken to meet them fully if not yet successfully completed. Risks were estimated as intermediate at the start of the activity, due to the limited amount of prior work on the topic of capacity forecasting; yet, all risks have been avoided or largely mitigated, as detailed in the children requirements, and the expectation is to reach a 100% completion of nearly all requirements by the end of the project, based on evaluations carried out in the last iteration and presented in D5.3.																	

FR-CFORE-001																		
Description	DAEMON capacity forecast models shall operate at very different timescales																	
Version	001M2																	
Owner	IMDEA																	
Priority	Medium																	
Risk	3																	
Risk Description	The risk of insufficient accuracy in the prediction is exacerbated as timescales become faster, as traffic demands are increasingly bursty, and the changes in requirements become more and more rapid.																	
Rationale	Orchestrators and controllers operate at very different timescales across the diverse network domains and take decisions over intervals that range from hours to seconds or less depending on the nature of the concerned resources (e.g., computing resources, transport capacity, spectrum, etc.). Capacity forecasting models must be adapted to such diverse settings.																	
K1		**K2**		**K3**		**K4**	X	**K5**		**K6**		**K7**		**K8**		**K9**	X	
Parents	FR-CFORE-000																	
Current Status																		
Percent complete	95%																	
Risk management	Successful																	
Rationale	The capacity forecasting models developed by DAEMON have been applied to very different use cases with heterogeneous timescales, as outlined in Section 3																	

	of D4.2 [3]. The models proved effective in addressing those different settings. In order to fully meet the requirement in a successful way, further tests at even more diverse timescales (e.g., seconds to days) may be needed. Risks were estimated as intermediate at the start of the activity, due to the inherent uncertainty of the flexibility of the models to different timescales; the experimental results obtained by the project across a variety of capacity forecasting use cases prove that such risks have been avoided.

FR-CFORE-002																		
Description	DAEMON capacity forecast models shall account for monetary costs in order to produce a practical prediction																	
Version	001M2																	
Owner	IMDEA																	
Priority	High																	
Risk	4																	
Risk Description	Considering a high number of cost sources makes the forecasting problem more involved and identifying the correct capacity prediction becomes harder in general.																	
Rationale	Predicting the sheer capacity needed to accommodate the traffic demand is not sufficient in many practical applications of capacity forecasting to network orchestration and control. Often, decisions on the allocation of resources and Virtual Network Functions (VNFs) must consider the costs incurred by the network operator (e.g., unnecessarily assigned resources that go unused, Service Level Agreement violations, VNF reconfiguration delays that determine subscriber churn, energy consumption generated by running VNFs at different network elements, etc.). Designing models that can capture such costs, and output a capacity that jointly reduces them, is critical to the economic sustainability of the network management process.																	
K1		**K2**		**K3**		**K4**	**X**	**K5**		**K6**		**K7**		**K8**		**K9**	**X**	
Parents	FR-CFORE-000																	
Current Status																		
Percent complete	100%																	
Risk management	Successful																	
Rationale	The capacity forecasting models developed by DAEMON were tested in settings that involved, in several cases, monetary costs. Specifically, the models were employed to solve tasks in anticipatory capacity allocation, as per Section 3.1 of D4.2 [3], or minimization of video streaming slice OPEX, as per Section 3.3 of D4.2 [3]: these are problems that inherently include economic costs in their formulation. As shown in Section 4.5 of D5.2 [4], the models proved effective in such tasks. Risks were estimated as high at the start of the activity, due to the absence of prior work on the topic; yet, the project was able to demonstrate how the developed capacity forecasting models can successfully operate in settings where the performance (hence the loss design or loss meta-learning process) depend on monetary costs incurred by the operator.																	

FR-CFORE-003																		
Description	DAEMON capacity forecast models shall operate over streaming data																	
Version	001M5																	
Owner	IMDEA																	
Priority	High																	
Risk	4																	
Risk Description	Adapting capacity forecasting to support a streaming model adds complexity and challenges to the design of the solution, which may reduce its efficiency.																	
Rationale	While many traffic forecasting models are trained offline and tested on historical data, the operation of such models in production calls for training and operation on traffic data as it is measured in the network. This implicitly means that capacity forecasting models must be adapted to work on streaming data.																	
K1		**K2**		**K3**		**K4**	**X**	**K5**		**K6**		**K7**		**K8**		**K9**	**X**	
Parents	FR-CFORE-000																	

Current Status	
Percent complete	75%
Risk management	Effective
Rationale	The models for capacity forecasting proposed by DAEMON can operate over streaming data, as proven by the performance evaluations carried out in Section 4.5 of D5.2 [4], which employed measurement data from production networks as input to the models. In order to meet the requirement in a way that is fully successful, additional testing over very long time periods (e.g., months) is needed, so as to verify the capability of the models to generalize and adapt to varying traffic conditions that may be very different from those observed during the training period. Risks were estimated as high at the start of the activity, due to the lack of prior testing of capacity forecasting models over streaming data; the results of the proposed models with a number of use cases involving streaming mobile network traffic data collected from real-world production systems demonstrate that the risk has been effectively mitigated.

FR-CFORE-004																		
Description	DAEMON capacity forecast models shall provide information about their level of accuracy																	
Version	001M17																	
Owner	IMDEA																	
Priority	Low																	
Risk	4																	
Risk Description	Anticipating not only the target variable but also the uncertainty of its estimate makes the prediction task sensibly more complex.																	
Rationale	Having information on the uncertainty of the prediction can help fine-tuning resource allocation, e.g., by including safety margins dimensioned on the level of expected accuracy of the forecasting model.																	
K1		**K2**		**K3**		**K4**	**X**	**K5**		**K6**		**K7**		**K8**		**K9**	**X**	
Parents	FR-CFORE-000																	
Current Status																		
Percent complete	100%																	
Risk management	Successful																	
Rationale	The capacity forecasting models developed by DAEMON can be naturally extended to leverage dropout layers during the inference phase so as to emulate the behavior of computationally complex Bayesian models. This strategy is adopted for instance in the models presented in Section 4.3 of D4.2 [3]. Risks were estimated as high at the start of the activity, due to the computational complexity of obtaining accurate information during inference via traditional Bayesian approaches; we avoided the risk by adopting computationally efficient approximations that make the operation possible at low cost, hence supporting the viability of the model in practical settings.																	

FR-CFORE-005																		
Description	DAEMON capacity forecast models shall be able to learn their objective/loss function autonomously																	
Version	001M17																	
Owner	IMDEA																	
Priority	High																	
Risk	3																	
Risk Description	Meta-learning the correct loss from scratches is a challenging task, for which no solution exists in the machine learning community.																	
Rationale	Many network management tasks involve situations where the relationship between the prediction (e.g., of resources to be allocated) and the performance (e.g., quality of experience of users) is unknown a-priori. In these settings, designing a correct loss function for machine learning is not possible, and meta-learning the loss is the only viable option.																	
K1		**K2**		**K3**		**K4**	**X**	**K5**		**K6**		**K7**		**K8**		**K9**	**X**	

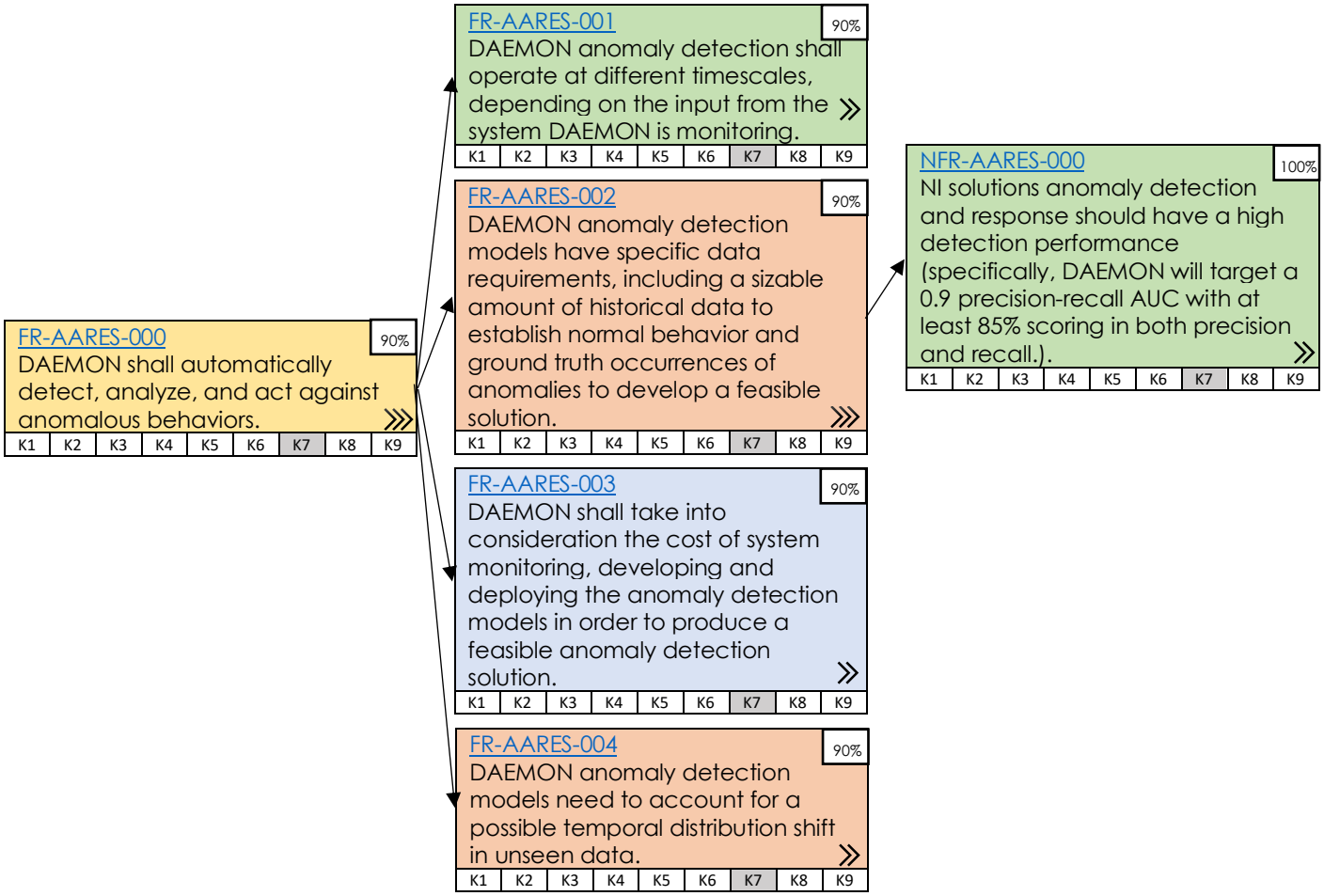
Parents	FR-CFORE-000
Current Status	
Percent complete	95%
Risk management	Successful
Rationale	DAEMON has designed loss-meta learning models as presented in Section 4.1.3 of D2.2 [1], which are also enhanced in Section 7.1.3 of the present document. These models can autonomously learn losses that are tailored to the network management task at hand. They have been applied to different practical use cases in Section 3.2 and Section 3.3 of D4.2 [3], and their effectiveness has been proven with real demands in Section 4.5.2 and Section 4.5.3 of D5.2 [4]. In order to achieve 100% success, we will need to evaluate the solutions for combined predictors, which will be done in D5.3 of the project. Risks were estimated as intermediate at the start of the activity, due to the extreme novelty of the approach, and to the lack of loss meta-learning solutions for forecasting; the risks were avoided by the introduction of a fully novel design that operates very well in a variety of practical use cases.

FR-CFORE-006																		
Description	Loss meta-learning should occur with minimum training time																	
Version	001M17																	
Owner	IMDEA																	
Priority	Medium																	
Risk	3																	
Risk Description	Meta-learning the loss inherently increases the time to convergence of a machine learning model, and reducing that time is challenging.																	
Rationale	In meta-learning models, the loss is learned (along with the model parameters) at runtime in the production system. Therefore, the initial lack of accuracy of the loss representation determines substantial errors in the predictions, hence significant costs for the operator. It is thus key to minimize the training time and the economic penalty for the operator of training the whole model from a cold start situation.																	
K1		**K2**		**K3**		**K4**	X	**K5**		**K6**		**K7**		**K8**		**K9**	X	
Parents	FR-CFORE-005																	
Current Status																		
Percent complete	75%																	
Risk management	Efficient																	
Rationale	The models developed by DAEMON have been trained on limited amount of real-world measurement data, i.e., 4-8 weeks, showing good performance in all cases, as demonstrated in the experiments described in D5.2 [4]. Meeting the requirement fully needs additional testing, e.g., with even shorter training datasets. Risks were estimated as intermediate at the start of the activity, due to the well-known needs for massive training data of deep neural network models; the risk was strongly mitigated thanks to a compute-prudent approach that traded off model complexity (e.g., depth) for a better (e.g., hybrid or meta-learning) design, which ultimately resulted in a reduced need for training data.																	

FR-CFORE-007	
Description	Loss meta-learning shall support losses that combine multiple predictions
Version	001M17
Owner	IMDEA
Priority	Medium
Risk	3
Risk Description	Having multiple forecasting models depend on the same loss implies correlations in the predictions, which are typically very complex to learn, making the problem more involved than single-input loss meta-learning.
Rationale	In many network management tasks, the performance does not depend on a single prediction but on a composition of multiple forecasting tasks. This is the case, for instance, in admission control problems over many predicted traffic flows, or in network slice brokering. Learning the correct loss function in those

			situations implies capturing the correlations among the different predictions and the performance metric, which calls for even more complex meta-learning tools.																
K1		**K2**		**K3**		**K4**	**X**	**K5**		**K6**		**K7**		**K8**		**K9**	**X**		
Parents			FR-CFORE-005																
Current Status																			
Percent complete			50%																
Risk management			Partial																
Rationale			Support for intertwined predictions in meta-learned loss is introduced by the original architecture described in Section 7.1.3 of the present document. The design allows handling loss meta-learning in the presence of multiple decisions that have a reciprocal influence on each other. The requirement is only partially met at the time of writing, as the effectiveness of the solution needs to be assessed in practical use cases. Such use cases will be defined in D4.3, and the performance of the proposed solution will be evaluated in D5.3 of the project. Risks were estimated as intermediate at the start of the activity, due to the high complexity of achieving accurate capacity forecasting in the presence of mutually dependent predictions; the risks were mitigated by introducing an appropriate neural network design that can manage intertwined predictions associated with a single loss function.														

A.8 Functional requirements: Automated anomaly response



FR-AARES-000	
Description	DAEMON shall automatically detect, analyze, and act against anomalous behaviors.
Version	003M35
Owner	TID
Priority	High
Risk	3
Risk Description	Anomaly detection tasks might not correctly capture new previously unseen anomalies.
Rationale	Most communication platforms use a reactive approach to deal with communication issues (i.e., operation teams react when incidents are severe only, and the service is often compromised already). DAEMON requires a proactive approach to anomaly detection that can detect both malicious and benign anomalies in the different systems it integrates.

K1		K2		K3		K4		K5		K6		K7	X	K8		K9	

Parents	None
Current Status	
Percent complete	90%
Risk management	Successful
Rationale	DAEMON implements three different activities for real-time anomaly detection and automated anomaly response, namely, A9, A19 and A25 as reported in D5.2 [4]. We provide details on the solution and its implementation in D4.2 [3] and D3.2 [2], which we will complement with their final status in D4.3 and D3.3, respectively. The reported status in D5.2 [4] shows an average completion of approximately 70% towards collecting the corresponding KPIs, which we will further update in D5.3.

FR-AARES-001	
Description	DAEMON anomaly detection shall operate at different timescales, depending on the input from the system DAEMON is monitoring.
Version	002M5
Owner	TID
Priority	Medium
Risk	2
Risk Description	Each anomaly detection task should take into consideration the requirements in terms of the timescale it needs to generate anomaly warnings. For finer granularities, the performance of the models might implicitly decrease, as the time available for the model to produce results also decreases. We will work to find the best tradeoff between the most fitting timescale and the performance requirements for the DAEMON anomaly detection tasks.
Rationale	Anomalies can become easier to spot depending on the timescale that fits to the particular system with which DAEMON interacts.

K1		K2		K3		K4		K5		K6		K7	X	K8		K9	

Parents	FR-AARES-000
Current Status	
Percent complete	90%
Risk management	Effective
Rationale	The DAEMON anomaly detection NI functionality (e.g., A9, A19 in D5.2 [4]) adapts the local anomaly detection process frequency, based on the incoming data volumes, as well as the needs of the engineers managing the systems we monitor.

FR-AARES-002	
Description	DAEMON anomaly detection models have specific data requirements, including a sizable amount of historical data to establish normal behavior and ground truth occurrences of anomalies to develop a feasible solution.
Version	002M5

Owner	TID
Priority	High
Risk	4
Risk Description	The lack of high-quality historical data to establish the baseline behavior of the system (i.e., anomaly-free state for training) poses a high risk to developing anomaly detection approaches for DAEMON. Similarly, the lack of ground truth anomalies that have been detected in the system will make the validation of any anomaly detection approach challenging. Finally, the lack of expert knowledge brings an extra risk when building data features to train the anomaly detection tools for DAEMON.
Rationale	The data quality is of paramount importance for the anomaly detection approaches we aim to integrate in DAEMON. Specifically, we aim to build on high quality ground truth for establishing the normal baseline for the system DAEMON monitors. Similarly, in order to validate the performance of our DAEMON anomaly detection solutions, we require a diverse set of ground-truth anomalies that operators previously captured in the systems DAEMON integrates. Furthermore, in order to craft data features that respond to the purpose of each system, we require expert knowledge and the operators' support in this process.

K1		K2		K3		K4		K5		K6		K7	X	K8		K9	

Parents	FR-AARES-000
Current Status	
Percent complete	90%
Risk management	Successful
Rationale	The solutions DAEMON proposes for anomaly detection rely on vast datasets of ground-truth anomalies that we collect from real-world systems (e.g., see the datasets supporting the evaluation of A19 in D5.2 [4]). DAEMON relies on the ticketing systems of the operational teams who manage the systems we monitor, which require continual updates.

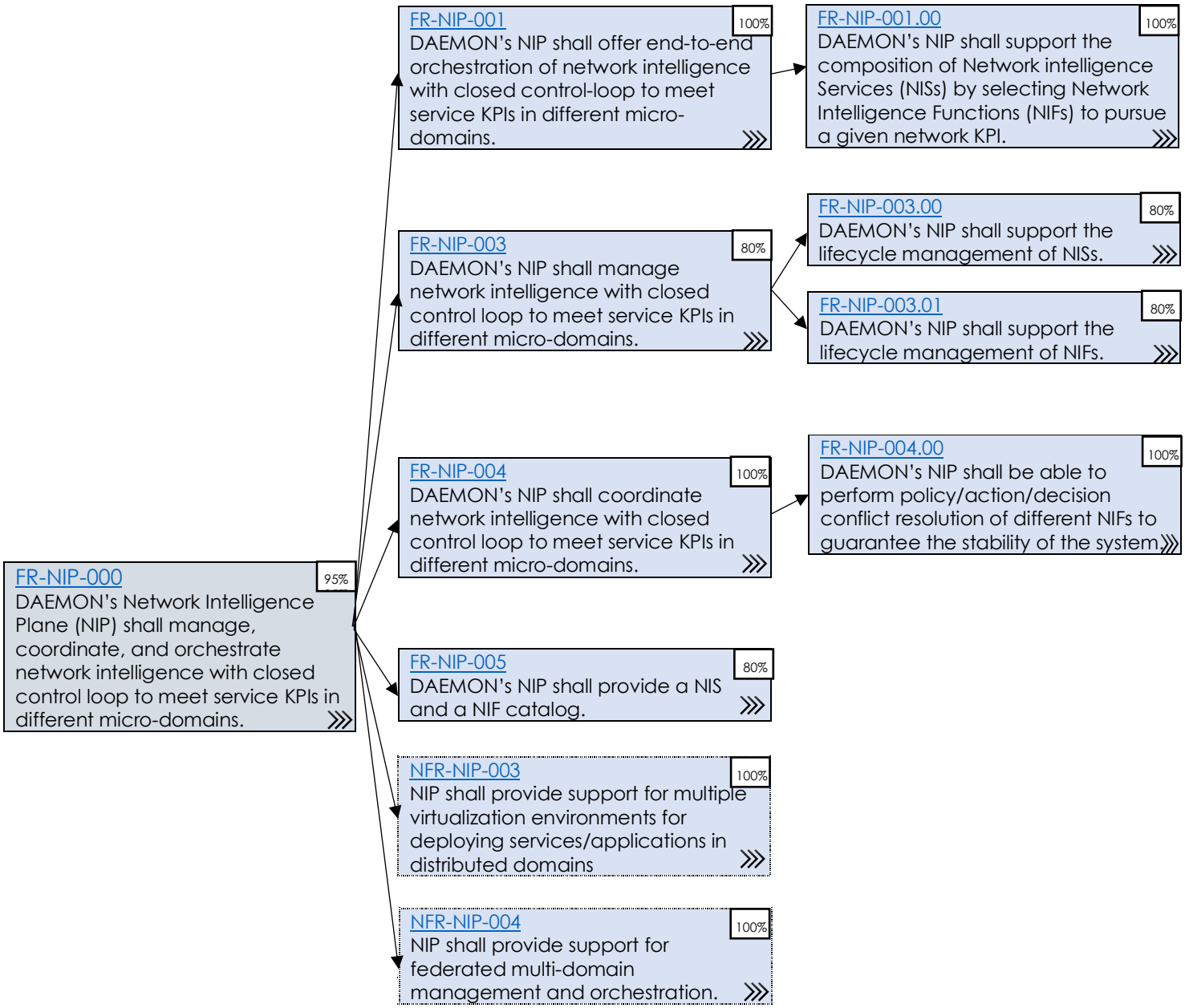
FR-AARES-003	
Description	DAEMON shall take into consideration the cost of system monitoring, developing and deploying the anomaly detection models in order to produce a feasible anomaly detection solution.
Version	003M5
Owner	TID
Priority	Low
Risk	1
Risk Description	The high cost of training and running DAEMON anomaly detection tools might suppose a high expenditure for the operators' of the system in question.
Rationale	DAEMON anomaly detection tools must run in real-world production systems, where we must also consider the actual monetary cost of running a state-of-the-art system for ML/DL tasks. We will work to produce solutions that adapt to different tiers of existing resources.

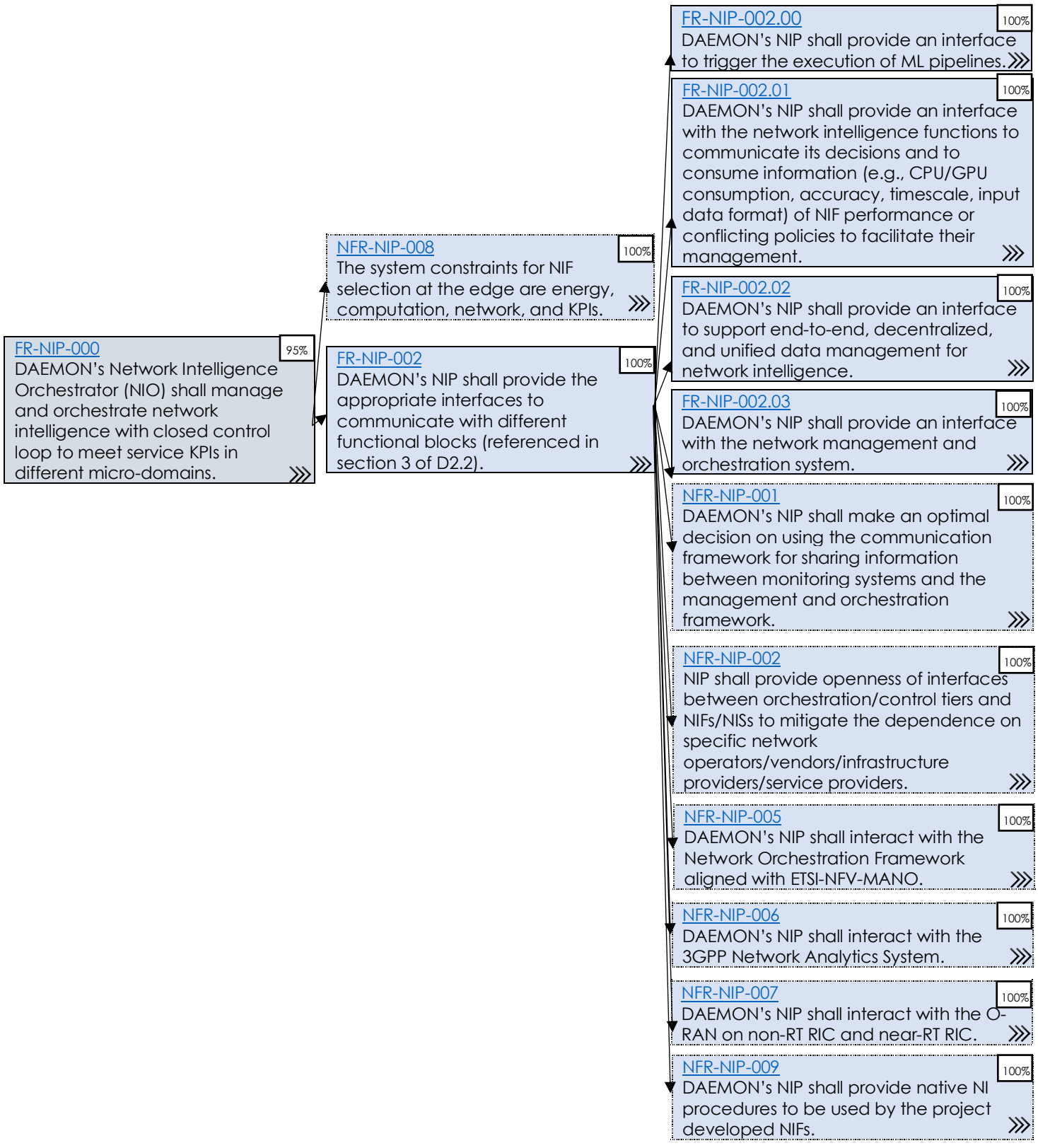
K1		K2		K3		K4		K5		K6		K7	X	K8		K9	

Parents	FR-AARES-000
Current Status	
Percent complete	90%
Risk management	Effective
Rationale	DAEMON's anomaly detection solutions have been developed in collaboration with engineering teams of real-world systems, enabling us to take into consideration their requirements in terms of monitoring, data transformation, model training and anomaly inference. For example, DAEMON solutions are in some cases to integrated in the cloud-based big data platform that engineering teams use internally.

FR-AARES-004																		
Description	DAEMON anomaly detection models need to account for a possible temporal distribution shift in unseen data.																	
Version	004M17																	
Owner	TID																	
Priority	High																	
Risk	4																	
Risk Description	The data captured in a network environment is indeed a temporal series that can have seasonal patterns or data can even be non-stationary. This issue poses a high risk for any anomaly detection approach that learns some normal behavior or statistics from the data. A possible distribution shift where features extracted from the captured data diverge too much over time will make the detection of anomalies in unseen data challenging.																	
Rationale	The data used for anomaly detection should cover historical data for an analysis of seasonal shifts and temporal distribution shifts. The features extracted from the features should be tested against stationarity and temporal covariance shift. Feature selection should select features that show high stability over time to avoid this issue. Nevertheless, anomaly detection models can age over time and new data should be captured regularly to update such models.																	
K1		**K2**		**K3**		**K4**		**K5**		**K6**		**K7**	**X**	**K8**		**K9**		
Parents	FR-AARES-000																	
Current Status																		
Percent complete	90%																	
Risk management	Effective																	
Rationale	DAEMON's solutions for anomaly detection are designed to be periodically re-trained to adapt to the shifts in the distributions of data. We tested this, for example, this is the solution we described in Section 4.2 in D4.2 [3], showing that a monthly time window is likely adequate.																	

A.9 Functional requirements: Network Intelligence Plane




FR-NIP-000
DAEMON's Network Intelligence Orchestrator (NIO) shall manage and orchestrate network intelligence with closed control loop to meet service KPIs in different micro-domains.
95%
NFR-NIP-008
The system constraints for NIF selection at the edge are energy, computation, network, and KPIs.
100%
FR-NIP-002
DAEMON's NIP shall provide the appropriate interfaces to communicate with different functional blocks (referenced in section 3 of D2.2).
100%
FR-NIP-002.00
DAEMON's NIP shall provide an interface to trigger the execution of ML pipelines.
100%
FR-NIP-002.01
DAEMON's NIP shall provide an interface with the network intelligence functions to communicate its decisions and to consume information (e.g., CPU/GPU consumption, accuracy, timescale, input data format) of NIF performance or conflicting policies to facilitate their management.
100%
FR-NIP-002.02
DAEMON's NIP shall provide an interface to support end-to-end, decentralized, and unified data management for network intelligence.
100%
FR-NIP-002.03
DAEMON's NIP shall provide an interface with the network management and orchestration system.
100%
NFR-NIP-001
DAEMON's NIP shall make an optimal decision on using the communication framework for sharing information between monitoring systems and the management and orchestration framework.
100%
NFR-NIP-002
NIP shall provide openness of interfaces between orchestration/control tiers and NIFs/NISs to mitigate the dependence on specific network operators/vendors/infrastructure providers/service providers.
100%
NFR-NIP-005
DAEMON's NIP shall interact with the Network Orchestration Framework aligned with ETSI-NFV-MANO.
100%
NFR-NIP-006
DAEMON's NIP shall interact with the 3GPP Network Analytics System.
100%
NFR-NIP-007
DAEMON's NIP shall interact with the O-RAN on non-RT RIC and near-RT RIC.
100%
NFR-NIP-009
DAEMON's NIP shall provide native NI procedures to be used by the project developed NIFs.
100%
Risk Level:
1 2 3 4 5
Requirement Type:
Functional
Non-Functional
Risk Management:
Successful
Effective
Partial
Percent Complete:
100%-0%

FR-NIP-000	
Description	DAEMON's Network Intelligence Plane (NIP) shall manage, coordinate, and orchestrate network intelligence with a closed control loop to meet service KPIs in different micro-domains.
Version	002M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	Given the wide range of NI solutions we need a common framework to map the most common features of NI algorithms, integrate them into a defined architecture, and design the necessary interfaces that algorithms use to interact with their environment.
Rationale	Network Intelligence (NI) is proposed to replace or assist network operators in their diverse set of network management tasks. However, current management frameworks (e.g., O-RAN, MANO) are not flexible enough or do not support the integration of NI instances. The DAEMON architectural framework enables the penetration of intelligence into both the user and control planes, thereby creating a hierarchical NI architecture that consists of distributed NI instances for network management, which altogether collaborate to improve their individual learning and decision-making processes.
Parents	None
Current Status	
Percent complete	95%
Risk management	Successful
Rationale	D2.3 presents the final updates of the architectural model for native orchestration in Beyond 5G (B5G) networks that was envisioned in Section 1 of D2.1. These updates contemplate the feedback from WP3 and WP4 during the second report period, plus the requirements defined in Section 2 of D2.2 [1]. Moreover, as shown in Sections 3, 4, and 5 of D2.3, the architectural model is able to fulfill the requirements imposed by the child, and consequently, is able to manage, coordinate and orchestrate network intelligence. The full competition of this requirement will be achieved once specific performance metrics related to orchestration and lifecycle management of NIF/NIS are measured and provided as reference values for NIP implementations.

FR-NIP-001	
Description	DAEMON's NIP shall offer end-to-end orchestration of network intelligence with closed control loop to meet service KPIs in different micro-domains.
Version	001M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	One of the main challenges in the orchestration of NI is to translate network requirements or KPIs to meet business needs.
Rationale	The NI Plane integrates the functions related to network intelligence. In several cases, these functions can be orchestrated to create end-to-end Network Intelligence Services. The creation of such services can be done in an automatic way, similarly as in network orchestrators.
Parents	FR-NIP-000
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	The architectural design finalized in D2.3 takes care of the end-to-end orchestration of intelligence with closed control-loop in different micro-domains.

FR-NIP-001.00	
Description	DAEMON's NIP shall support the composition of Network intelligence Services (NISs) by selecting Network Intelligence Functions (NIFs) to pursue a given network KPI.
Version	002M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	It is possible that the available NIFs do not address the system constraints. In this case, NIFs that try to fulfill system constraints as close as possible will be selected.
Rationale	Network Intelligence Functions (NIFs) are functional blocks that implement a decision-making functionality to be deployed in a controller. Similar to the information model specified for network management by, e.g., 3GPP, NIFs can be arranged to compose a Network Intelligence Service (NIS). Depending on the available resources and the business goals or SLAs, NIP will select the best NIF model that suits the system constraints. For example, in some cases it might be feasible to sacrifice accuracy at the expenses of a lower computational complexity.
Parents	FR-NIP-001
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	The composition of Network Intelligence Services can be achieved in a similar way as presented in [31]. Moreover, model selection was included in the NI Plane procedures in Section 5 of D2.3.

FR-NIP-002	
Description	DAEMON's NIP shall provide the appropriate interfaces to communicate with different functional blocks (referenced in section 3 of D2.2 [1])
Version	001M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	There are common communication patterns (e.g., pub/sub) that could be replicated here. However, we must select the most suitable communication system, considering that the decisions taken by the NIP might impact network behavior.
Rationale	Once a NIS is created/composed, training such models (NIFs) will be performed via the creation and deployment of MLOps frameworks. Once the models are trained, they will be registered in the NIF/NIS catalogue and will be ready to be deployed in a test/production environment. Currently there are several commercial frameworks that already do that for ML applications. The idea is not to reinvent the wheel, but to adapt such frameworks to the network domain. Once the NISs are deployed, the appropriate interfaces to manage the lifecycle management of their NIFs shall be used. Moreover, NIFs should be able to infer the network state/context as input. For that reason, the NIP should enable an interface with the corresponding management framework.
Parents	FR-NIP-000
Current Status	
Percent complete	100%
Risk management	Successful

Rationale	The results of this activity are reported in Section 4 of D2.3, where the interfaces (internal and external) with the main functional blocks of the architecture are defined.

FR-NIP-002.00	
Description	DAEMON's NIP shall provide an interface to trigger the execution of ML pipelines
Version	002M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	From the architectural point of view, we need to identify or to create the interaction points between the NIP and the MLOps framework.
Rationale	MLOps is a methodology that combines Machine Learning (ML) with software development operations (DevOps) and data engineering with the goal of building, training, deploying, and maintaining ML systems in productions with high reliability and efficiency guarantees. DAEMON architecture explicitly indicates that building ML models functionality (i.e., the ML pipelines) is delegated to an external platform, and MLOps frameworks are the de-facto platform to do this task. Currently there are several commercial frameworks that already do that for ML applications. The idea is not to reinvent the wheel, but to adapt such frameworks to the network domain.
Parents	FR-NIP-002
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	Same as parent.

FR-NIP-002.01	
Description	DAEMON's NIP shall provide an interface with the network intelligence functions to communicate its decisions and to consume information (e.g., CPU/GPU consumption, accuracy, timescale, input data format) of NIF performance or conflicting policies to facilitate their management
Version	001M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	Functionalities and NIFs can be very diverse. To ease the implementation of a NIP, information should be standardized (e.g., format) which can be cumbersome given the wide application domains of DAEMON functionalities.
Rationale	NIP decisions (replacement, retraining, execution, and termination) should be made based on the information coming from the Network Intelligence Functions (NIFs). This information should be enough to take a good decision. Furthermore, this decision must be communicated using the same channel, guaranteeing the stability of the system.
Parents	FR-NIP-002
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	Same as parent.

FR-NIP-002.02	
Description	DAEMON's NIP shall provide an interface to support end-to-end, decentralized, and unified data management for network intelligence.
Version	001M8
Owner	ZSC
Priority	High
Risk	1
Risk Description	Besides managing the lifecycle of different NISs, the burden of managing data can be too big as it involves a multiplicity of data sources and data types.
Rationale	The NI Multi-timescale Closed-loop AI Framework should provide end-to-end decentralized and unified data management to ease the development, operation, and management of any NI model. Such data is gathered with the purpose of training NI algorithms. The main characteristics are previously defined in FR-MTERM-001 in D2.1.
Parents	FR-NIP-002
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	Same as parent.

FR-NIP-002.03	
Description	DAEMON's NIP shall provide an interface with the network management and orchestration system.
Version	001M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	In some cases, the interaction point with network management and orchestration systems is evident (e.g., O-RAN architecture) but in other domains it can be hard to define (e.g., NFV MANO) since they are not developed to natively support NI.
Rationale	The NIP manages the connection towards the network management and orchestration to gather important information such as the expected network KPIs for the managed slice and service, as well as the information of the underlying network infrastructure.
Parents	FR-NIP-002
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	Same as parent.

FR-NIP-003	
Description	DAEMON's NIP shall manage network intelligence with closed control-loop to meet service KPIs in different micro-domains.
Version	001M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	Overall system stability should be achieved. However, it can be that some NISs span several domains which require extra coordination.

Rationale	Once a NIS is released for production, the NIP shall support its lifecycle management. By lifecycle management we refer to onboarding, instantiation, termination, scaling, and state retrieval. The same should happen with different NIFs that compose the NIS
Parents	FR-NIP-000
Current Status	
Percent complete	80%
Risk management	Successful
Rationale	The results of this activity are reported in Section 5 of D2.3, more specifically, subsection 5.1.3 described the procedures needed for managing NIFs and NISs. To complete this requirement, the measurement of performance metrics related to NIS/NIF/NIF-C composition and deployment will be provided as reference values to evaluate the performance of the lifecycle management capabilities provided by the NIP.

FR-NIP-003.00	
Description	DAEMON's NIP shall support the lifecycle management of NISs
Version	002M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	The metrics that measure the impact of a NIF in the overall performance of a NIS could be difficult to define.
Rationale	NISs are composed of one or more NIFs with a specific target, usually related with a specific set of targeted KPIs. They possibly span several network domains. Therefore, it is required to not only monitor the performance of a given NIF from the NIS, but also the impact of this NIF in the performance of the NIS.
Parents	FR-NIP-003
Current Status	
Percent complete	80%
Risk management	Successful
Rationale	The results of this activity are reported in Section 5 of D2.3, where the procedures for each step of the NIS lifecycle are defined using the building blocks proposed by the architectural design. To complete this requirement, the measurement of performance metrics related to NIS composition and its deployment will be provided as reference values to evaluate the performance of the lifecycle management capabilities provided by the NIP.

FR-NIP-003.01	
Description	DAEMON's NIP shall support the lifecycle management of NIFs
Version	001M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	NIFs themselves could be of different kinds: They could be learning models, based on, e.g., Deep Neural Networks or Engineered Models, or they could be built upon specific optimization algorithms such as the ones based on control theory or Mixed-Integer Linear Programming (MILP). Thus, it's necessary to define common strategies to proper manage both types of NIFs.
Rationale	According to the DAEMON architecture, the NIF manager is responsible for the lifecycle management of NIFs and monitoring the health of the intelligence

	functions. This includes typical diagnostic information, if the NIF is being used in inference or it is an online learning solution, or other metrics such as the loss and the training loops if the NIF is currently being trained. Moreover, the NIP needs to provide feedback on the NIFs performance so higher-level decisions can be made (e.g., that the model can be updated or replaced).
Parents	FR-NIP-003
Current Status	
Percent complete	80%
Risk management	Successful
Rationale	The procedures defined in Section 5 of D2.3 are valid for NISs as well as for NIFs. To complete this requirement, the measurement of performance metrics related to NIF and NIF-C composition and their deployment will be provided as reference values to evaluate the performance of the lifecycle management capabilities provided by the NIP.

FR-NIP-004	
Description	DAEMON's NIP shall coordinate network intelligence with closed control-loop to meet service KPIs in different micro-domains.
Version	001M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	To determine which action/policy/decision has priority on optimizing a given function is not trivial and it depends on multiple factors that need to be evaluated. Therefore, the initial selection of policies/actions/decisions is highly coupled with the use case.
Rationale	Coordination of NI can include, but is not limited to: • Sharing NIF-C among different NIFs (e.g., two NIFs that require the same input) • Arbitration policies in case of two NIFs that share the same sink, that is, the configuration APIs. • Guarantee system stability among conflicting policies/actions/decisions.
Parents	FR-NIP-000
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	The results of this activity are reported in Section 5 of D2.3, where the procedures for coordinating NIs is described. In particular, we've shown procedures for conflict resolution (§5.2.1) and knowledge sharing (§5.2.2) which are two of the main concerns of this requirement.

FR-NIP-004.00	
Description	DAEMON's NIP shall be able to perform policy/action/decision conflict resolution of different NIFs to guarantee the stability of the system.
Version	003M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	When a NIF/NIS performs an action that conflicts with the action of other NIF/NIS, it is required to solve the conflict in a coordinate manner. However, designing the conflict resolution mechanism may be a very hard problem as it will depend on the multiple factors tailored to specific use cases (e.g., centralized vs.

	decentralized vs. federate network domains, flat vs. hierarchical decision making, etc.).
Rationale	Optimizations will take place in different domains of the assisted system. Therefore, the decisions/policies/actions that are taken to optimize a certain objective function (e.g., business goal, SLAs) can be counterproductive to other policies/decisions/actions. Thus, a conflict resolution system is needed that can guarantee that the system is evolving towards a stable state.
Parents	FR-NIP-004
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	The results of this activity are reported in Section 5 of D2.3, specifically §5.2.1 show the steps necessary to perform conflict detection and resolution.

FR-NIP-005	
Description	DAEMON's NIP shall provide a NIS and a NIF catalog
Version	001M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	There must be some commonalities between NIFs and NISs, so they could be advertised in a general framework.
Rationale	The NIP has catalogs of already onboarded NIS and NIFs. In particular, NIFs may need to be (re)-trained to cope with changing or different conditions, or on a periodical basis. When a NIS is composed of NIF empowered by ML models, training such models will be performed via the creation and deployment of ML pipelines. Once the models are trained, they will be registered in the NIF/NIS catalog and will be ready to be deployed in a test/production environment.
Parents	FR-NIP-000
Current Status	
Percent complete	80%
Risk management	Successful
Rationale	The results of this activity are reported in Section 3 of D2.3, where the NIS/NIF catalog is included in the proposed architecture. The packaging of the diverse NIFs could be done as in [31]. However, the descriptor should include, beside the elements mentioned in the previous reference, the additional information mentioned in §5.1.1 of D2.3. To complete this requirement, the measurement of performance metrics related to NIS/NIF/NIF-C upload latency will be provided as reference values to evaluate the performance of the NIS/NIF/NIF-C register provided by the NIP.

A.10 Performance requirements

Specify both the static and the dynamic numerical requirements placed on the software or on human interaction with the software.

NFR-RIS-000																			
Description	RIS should aid to increase wireless capacity (bits/m^2) by 100%																		
Version	001M1g																		
Owner	NEC																		
Priority	High																		
Risk	3																		
Risk Description	There is a risk that the performance attained in realistic environments fall below 100%																		
Rationale	This will allow surfaces to adapt in a timely manner, following the channel dynamics.																		
K1		**K2**		**K3**		**K4**		**K5**		**K6**	X	**K7**		**K8**		**K9**			
Parents	FR-RIS-000-001M1																		
Current Status																			
Percent complete	50%																		
Risk management	Successful																		
Rationale	An experimental RIS prototype is being built and measurements will be collected in an anechoic chamber. The initial design steps were presented in D5.2 [4], Section 4.6, and the final design and results will be presented in D5.3.																		

NFR-RIS-001																			
Description	Re-configuring all the components in a RIS must be achieved within 100 ms.																		
Version	003M17																		
Owner	NEC																		
Priority	High																		
Risk	3																		
Risk Description	There is a risk that the electronic equipment required can only be re-configured in more than 100ms. For instance, nowadays shortages in electronic components may force us to resort to less performing designs.																		
Rationale	100 ms is the timescale of O-RAN near-real-time RAN Intelligent controller and a good trade-off between tracking fast wireless channel dynamics and high overhead.																		
K1		**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**	X	**K9**			
Parents	FR-RIS-000-001M1																		
Current Status																			
Percent complete	50%																		
Risk management	Successful																		
Rationale	An experimental RIS prototype is being built and measurements will be collected in an anechoic chamber. The initial design steps were presented in D5.2 [4], Section 4.6, and the final design and results will be presented in D5.3.																		

NFR-RIS-002																			
Description	The (non RF) electronic equipment required to control a RIS must consume less than 100 mW.																		
Version	001M17																		
Owner	NEC																		
Priority	High																		
Risk	3																		
Risk Description	There is a risk that the electronic equipment required to control a RIS consumes more than 100 mW. For instance, nowadays shortages in electronic components may force us to resort to more energy-consuming solutions.																		
Rationale	Provide smart RF reflectors that are very efficient in terms of energy consumption hence reducing OPEX.																		
K1	X	**K2**		**K3**		**K4**	X	**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-RIS-000-001M1																		
Current Status																			
Percent complete	50%																		
Risk management	Successful																		
Rationale	An experimental RIS prototype is being built and measurements will be collected in an anechoic chamber. The initial design steps were presented in D5.2 [4], Section 4.6, and the final design and results will be presented in D5.3.																		

NFR-CAWRS-000																			
Description	NI orchestration solutions for vRAN shall have reaction times below 10s																		
Version	002M17																		
Owner	UC3M																		
Priority	High																		
Risk	1																		
Risk Description	There is a low risk that DAEMON will not integrate and succeed in providing such timings for the NI-based network orchestration. In preliminary works, DAEMON partners were able to achieve computing resources orchestration within a 10s constraint. This constraint may be lowered with the usage of more complex orchestration solutions.																		
Rationale	In [62], DAEMON authors were able to orchestrate computing resources for vRAN by using the Docker API, with a 10 seconds granularity. This value is already enough to bring down the computing resource usage by more than 30% in some scenario. By using directly the cgroups API offered by the Linux system, we may achieve even lower values.																		
K1		**K2**	X	**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-CAWRS-000-001M3																		
Current Status																			
Percent complete	100%																		
Risk management	Successful																		
Rationale	As reported in [17] this requirement has been achieved.																		

NFR-CAWRS-001	
Description	NI control solutions to schedule computing and radio resources in real time for vRAN shall have an inference time below 500us
Version	002M17
Owner	UC3M
Priority	High
Risk	2
Risk Description	There is a mild risk that NI cannot achieve sub-second timings for the vRAN control algorithms such as the radio scheduling (which needs 1ms timings). If these timings

	cannot be achieved, DAEMON partners will use solutions such as slower scheduling patterns, enforced every 50-100 TTIs
Rationale	Ideally, scheduling decisions are taken every TTIs, thus in the ms range scale. This requirement is quite stringent and may require specialized hardware such as GPUs deployed at the edge if deep learning solutions shall be put in place. Alternatives could be the usage of mixed models between machine learning and traditional optimization

K1		K2	X	K3		K4		K5		K6		K7		K8		K9	

Parents	FR-CAWRS-002-001M17
Current Status	
Percent complete	90%
Risk management	Successful
Rationale	Validation results for this activity will be reported in D3.3 and D5.3

NFR-CAWRS-002	
Description	NI solutions for vRAN shall maximize spectral efficiency given computing capacity constraints.
Version	002M17
Owner	UC3M
Priority	High
Risk	2
Risk Description	There is a mild risk that NI cannot achieve bounded performance for the wireless performance (i.e., spectrum efficiency, leading to bandwidth and latency figures). In this case, specific boundaries to the achievable computing resource saving will be defined.
Rationale	With unbounded computing resource savings, the spectral efficiency may be unacceptably low. In [17], DAEMON partners were capable of achieving very good tradeoffs between achievable savings and pure performance, by correctly understanding the traffic patterns. Other solutions may have to be designed to guarantee that this tradeoff (the ratio between the best possible performance without computing resource optimization and the one obtained by DAEMON solutions never falls below certain thresholds) maximizing hence spectral efficiency given computing capacity constraints.

K1		K2	X	K3		K4		K5		K6		K7		K8		K9	

Parents	FR-CAWRS-002-001M17
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	As reported in [62] this requirement has been achieved.

NFR-CAWRS-003	
Description	Predictive HARQ inference mechanisms shall have a minimum accuracy of 99% and a false positive rate below 0.1%
Version	001M17
Owner	UC3M
Priority	High
Risk	1
Risk Description	There is a low risk that NI cannot integrate inference mechanisms whose accuracy is at least 99% and the false positive rate below 0.1%. There are multiple previous works applying this technique in other fields.
Rationale	It is critical that the inference mechanisms have a very high accuracy and a very low rate of false positives, because a wrong prediction (due to a prediction fail or a false positive result of the prediction) incurs substantially higher cost because the transport block has to be recovered by others.

K1		K2	X	K3		K4		K5		K6		K7		K8		K9	

Parents	FR-CAWRS-001-001M17
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	Validation results for this activity were reported in D5.2 [4] (Section 4.1).

NFR-EAWVNF-002																			
Description	DEAMON expects to save 50% of the energy cost, thanks to applying NI solutions to find out the energy-aware optimal placement of VNFs of FR-EAWFN-000.																		
Version	001M2																		
Owner	UMA																		
Priority	High																		
Risk	3																		
Risk Description	We cannot achieve the 50% of energy saving in all cases, only in some of them, or simply DAEMON solutions save an inferior percentage of energy.																		
Rationale	The performance in terms of energy consumption of the DAEMON solution should improve the current solutions by a 50%.																		
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-EAWVNF-001-001M1																		
Current Status																			
Percent complete	60%																		
Risk management	Effective																		
Rationale	The energy-aware placement solution reduces up to 51% of the energy consumption compared with the default deployment proposed by the existing MANO standard solutions, which follow non-energy-aware policies. In addition, the proactive autoscaling solution and VNF placement have a 92.5% decrease in energy consumption (a failed request rate of up to 0% and reasonable execution times of the auto-scaling process for different problem sizes).																		

NFR-EAWVNF-003																			
Description	The cost in terms of energy footprint of the NI solution for VNFs placing shall be less than the global energy saving																		
Version	001M2																		
Owner	UMA																		
Priority	High																		
Risk	2																		
Risk Description	The cost of the NI-assisted VNF placement could be not much less or even higher than the energy consumption savings of the proposed solutions.																		
Rationale	The energy saving obtained by applying energy profiling to the NI algorithms for the VNFs placement should be less than the global energy saving, to be worthy. So, the cost of the energy-awareness mechanism should be a lot less than 50% of the energy saving proposed in NRF-EAWVNF-002.																		
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-EAWVNF-001-001M1																		
Current Status																			
Percent complete	60%																		
Risk management	Effective																		
Rationale	Section 4.4.9 in D5.2 [4] presents the results in measuring the improvements in the Edge, and the energy cost reduction goes above 15% and, in some scenarios, up to 92%.																		

NFR-EAWVNF-004																			
Description	Energy-efficient NI shall balance throughput and energy consumption in vRANs																		
Version	001M17																		
Owner	NEC																		
Priority	High																		
Risk	1																		
Risk Description	There is no risk																		
Rationale	Tethered virtualized base stations may be interested in trade-off radio spectrum capacity for energy savings																		
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-EAWVNF-005-001M17																		
Current Status																			
Percent complete	100%																		
Risk management	Successful																		
Rationale	The risks were low. The design of the NI was presented in D4.2 [3], Section 2.3, and an empirical evaluation was presented in D5.2 [4], Section 4.2.1. The details can be found in [18], [40]. More specifically, a data-driven approach based on Bayesian Learning was designed to control different configuration parameters of virtualized base stations to balance throughput and power consumption.																		

NFR-EAWVNF-005																			
Description	NI orchestrating resources in vRANs shall maximize networking throughput given power consumption constraints																		
Version	001M17																		
Owner	NEC																		
Priority	High																		
Risk	3																		
Risk Description	There may be cases where power constraints cannot be satisfied.																		
Rationale	Respecting power consumption constraints, even while learning, it is of paramount importance for battery-powered small cells, solar-powered small cells or other types of power-constrained small cells.																		
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-EAWVNF-005-001M17																		
Current Status																			
Percent complete	100%																		
Risk management	Successful																		
Rationale	The design of the NI was presented in D4.2 [3], Section 2.3, and an empirical evaluation was presented in D5.2 [4], Section 4.2.1. The details can be found in [18], [40]. More specifically, a data-driven approach based on Bayesian Learning was designed to control different configuration parameters of virtualized base stations to maximize network throughput given hard power consumption constraints.																		

NFR-EAWVNF-006	
Description	Energy savings shall be achieved in virtualized RANs without compromising given service performance constraints
Version	001M17
Owner	NEC
Priority	High
Risk	3
Risk Description	It may be possible that energy savings can only be achieved when service performance constraints are not satisfied.
Rationale	Satisfying service-level agreements is the top priority of a mobile network. Hence, NI solutions should strive to meet service performance constraints with a minimum energy consumption toll.

K1	X	K2		K3		K4		K5		K6		K7		K8		K9	

Parents	FR-EAWVNF-005-001M17
Current Status	
Percent complete	50%
Risk management	Successful
Rationale	The design of NI is currently ongoing and will be presented in D4.3 (design) and in D5.3 (empirical results).

NFR-AARES-000	
Description	NI solutions anomaly detection and response should have a high detection performance (specifically, DAEMON will target a 0.9 precision-recall AUC with at least 85% scoring in both precision and recall.).
Version	001M5
Owner	TID
Priority	High
Risk	2
Risk Description	There is a mild risk that NI for anomaly detection cannot achieve its target performance. This highly depends on the quality and availability of ground-truth datasets from the systems DAEMON will monitor.
Rationale	It is important that NI solutions for anomaly detection in the different systems that DAEMON will monitor detect real (and important) anomalies, and do not flood the operators with false alarms for their systems.

K1		K2		K3		K4		K5		K6		K7	X	K8		K9	

Parents	FR-AARES-002-002M5
Current Status	
Percent complete	100%
Risk management	Effective
Rationale	We reported the performance results of A19 in D5.2 [4], showing that we were able to achieve the performance of the anomaly detection in DAEMON.

A.11 Design constraints

Specify constraints on the system design imposed by external standards, regulatory requirements, or project limitations.

NFR-RIS-003	
Description	RIS must provide beamforming gains passively, without energy-consuming (active) RF chains
Version	001M17
Owner	NEC
Priority	High
Risk	1
Risk Description	There is a small risk that beamforming gains can only be achieved with active RF chains that integrate RF amplifiers.
Rationale	Smart RF reflectors with active RF chains already exist and are called "relays". The main motivation for RIS is the possibility of attaining beamforming gains with minimal energy consumption and costly electronic equipment. Hence, a RIS must necessarily be passive.

K1		K2		K3		K4		K5		K6	x	K7		K8		K9	

Parents	FR-RIS-000-001M1
Current Status	
Percent complete	50%
Risk management	Successful

Rationale	Risks are low. The initial design steps, consisting of a patch antenna array without active RF chains, were presented in D5.2 [4], Section 4.6.

NFR-EAWVNF-001																			
Description	DAEMON energy-aware solution will scale well when considering a heterogenous set of devices and network infrastructure **FR-EAWVFN-001.**																		
Version	001M2																		
Owner	UMA																		
Priority	High																		
Risk	4																		
Risk Description	The variety of devices																		
Rationale	DAEMON should be able to consider the global footprint of VNFs placement solutions for a large number of different IoT and Edge devices with variable resources and networking infrastructure. The upper values of the devices' resources considered in DAEMON will be taken from the software and hardware network or device specifications of the underlying infrastructure.																		
K1	**X**	**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-EAWVNF-001-001M1																		
Current Status																			
Percent complete	100%																		
Risk management	Successful																		
Rationale	The auto-scaling solution includes an energy-aware orchestrator, which calculates the energy consumption according to the location of the VNFs and assigns the applications/VNFs to the most energy-efficient node. The energy-aware orchestrator, as well as the Essential Node Identifier module, includes the expected performance as a constraint, to reduce energy consumption without compromising throughput.																		

NFR-MTERM-001																			
Description	DAEMON's MTERM shall provide an exhaustive list of orchestration operations																		
Version	002M18																		
Owner	IMEC																		
Priority	Low																		
Risk	1																		
Risk Description	The list of orchestration operations might involve subgroups of operations, depending on the policies defined for specific types of applications (e.g., value-added services require different orchestration operations than services that are directly consumed by users).																		
Rationale	The NI-assisted management and orchestration framework needs to provide support for at least a basic set of orchestration operations, such as onboarding (I.e., preparation of application descriptors and images on all required edge platforms), instantiation (on all required edge platforms), scaling up/down/out/in depending on the resource (computing and network) requirements and current resource consumption, termination (i.e., releasing the allocated resources so they can be consumed by other applications, or save energy), and state/context migration (I.e., migrating the state/context of the application from one edge to another due to the UE mobility, resource availability, or energy saving purposes).																		
K1		**K2**	**X**	**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**			
Parents	FR-MTERM-000																		
Current Status																			
Percent complete	100%																		
Risk management	Successful																		

Rationale	The solution in Section 3.1, D3.2 [2] uses OSM on top of Kubernetes, which covers the most basic orchestration operations.

NFR-MTERM-002	
Description	DAEMON's MTERM shall provide compliance with standardized frameworks (e.g., ETSI NFV MEC, ETSI NFV MANO, and O-RAN) running at the network edge.
Version	002M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	The insufficient level of compatibility between different standardization tracks (ETSI MEC/ETSI NFV MANO & O-RAN) can potentially lead to complex and application-specific orchestration platforms, limiting their exploitability among research tracks.
Rationale	As the standardization plays a key role in ensuring that a software tool meets certain requirements that guarantee proper work in various conditions, and expanding the exploitability of such solution, NI-assisted management and orchestration framework needs to be designed and developed in accordance with the existing standardization efforts.

K1		**K2**	**X**	**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**	

Parents	FR-MTERM-000
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	All the solutions tackling the MTERM functionality use standardized frameworks.

NFR-MTERM-003	
Description	DAEMON's MTERM shall provide NIF modularity and reusability among different players (e.g., network operators/vendors, service providers, etc.)
Version	002M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	The lack of NIF complexity and an increased level of openness of I/O interfaces might decrease the accuracy of decision-making processes performed by those NIFs.
Rationale	Due to the heterogeneity in resource and service deployments across edge networks, the NIFs running in both framework tiers need to be application/service-agnostic, thus, no application-specific data should be considered apart from the resource requirements and KPIs stated in SLAs. With such a configuration, NIFs can be maintained and used by different stakeholders.

K1		**K2**	**X**	**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**	

Parents	FR-MTERM-000
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	All the solutions tackling the MTERM functionality are developed as containers following a cloud-based approach, facilitating modularity and reusability.

NFR-SLMANO-000

Description	DAEMON controllers and orchestrators should be steered by high-level QoE targets and business KPIs (high-level intents), rather than strict QoS goals and technical KPIs.
Version	002M4
Owner	NBL
Priority	Medium
Risk	2
Risk Description	The problem may be that it may be difficult to describe expected behavior in a concise way.
Rationale	Application developers should have an easy way of specifying intended behavior based on their application-level knowledge and requirements to guarantee QoE for their users.

K1		**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**	X

Parents	FR-SLMANO-000-002M17
Current Status	
Percent complete	90%
Risk management	Successful
Rationale	In the scaling work we have set a target latency and defined a reward function that penalizes latency violations and the (excessive) use of resources (Section 4.4.7 of D5.1 [7]), showing that high-level QoE targets can be used for MANO life-cycle management.

NFR-SLMANO-001	
Description	DAEMON shall define metrics to check the stability of a control algorithm.
Version	001M5
Owner	NBL
Priority	High
Risk	2
Risk Description	Although a rough definition of a stable control system is easy to understand, i.e., if after exciting the system with a short, small perturbation, it returns fast enough to the original equilibrium, it is hard to make that definition precise for nonlinear systems.
Rationale	It is well-known that closing the control loop may lead to unstable systems. In linear systems, instability stems from the fact that the closed loop transfer function has poles in the positive half plane, leading to an impulse response that exponentially increases. Although some ideas some chaos engineering may be applied, in non-linear systems there is no rigorous equivalent.

K1		**K2**		**K3**		**K4**		**K5**		**K6**		**K7**		**K8**		**K9**	X

Parents	FR-SLMANO-003-002M17
Current Status	
Percent complete	80%
Risk management	Successful
Rationale	We used a practical definition of stability: as long as small perturbations on the input, did not lead to growing fluctuations at the output, we called the system stable (see FR-SLMANO-003).

NFR-IBSSI-000	
Description	Network Intelligence algorithms should be adapted to the PISA architecture
Version	001M17
Owner	IMDEA
Priority	Medium
Risk	3
Risk Description	Programmable switches have specific internal architectural models that make some machine learning models more suitable than others for deployment.

Rationale	Modern programmable switches are compliant with the Protocol Independent Switch Architecture (PISA) model. The solutions for machine-learning-based inference implemented in such devices must thus be aligned with the internal organization into Match-Action Units (MAUs) of such a model.

K1		K2		K3	X	K4		K5		K6		K7		K8		K9	

Parents	FR-IBSSI-002-003M18
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	The RF models proposed by DAEMON are tailored to the PISA architecture by design, as detailed in Section 5.1 of D3.2 [2], hence are guaranteed to be fully compatible with the target programmable hardware in the user plane. Indeed, they have been implemented in real-world production-grade switches for the performance evaluation presented in Section 4.7.1 in D5.2 [4].

NFR-IBSSI-001	
Description	Network Intelligence algorithms should be resource-prudent
Version	001M17
Owner	IMDEA
Priority	Low
Risk	4
Risk Description	Programmable switches have extremely limited computational capabilities that are primarily intended to support forwarding-related policies.
Rationale	Decision-making is not a legacy or priority task in programmable user planes. Therefore, NI solutions deployed in programmable switches must consume as little resources as possible, in a way not to hinder the regular operation of the devices and the whole network. Ideally, NI models for programmable switches should not consume more than 1% of the different memory types available in these devices.

K1		K2		K3	X	K4		K5		K6		K7		K8		K9	

Parents	FR-IBSSI-002-003M18
Current Status	
Percent complete	75%
Risk management	Effective
Rationale	The performance evaluations carried out as described in Section 4.7.1 of D5.2 [4] show how the proposed solution for integration of NI in programmable user planes consumes a limited amount of resources in a production-grade switch. Depending on the use case, an RF model uses between 3% and 29% of the total memory resources in the switch and is thus compatible with other functionalities that the programmable switch must perform. In order to meet the requirement in a fully successful way, more resource-prudent mappings than the one adopted in the current implementation need to be devised and tested.

NFR-NIP-001	
Description	NIP shall make an optimal decision on using the communication framework for sharing information between monitoring systems and the management and orchestration framework
Version	002M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	Additional benchmarking of communication systems (e.g., message broker) that will be used for sharing information between framework entities is needed, and different systems might be suitable for different types of applications.

Rationale	As communication systems/platforms enable either synchronous or asynchronous communication between different orchestration components and NIFs, it is important to consider the complexity of using and managing the communication system (e.g., RabbitMQ is a simple and often used in most of the existing MANO solutions) for pub/sub purposes, but also the additional latency this entity involves in the communication (e.g., RabbitMQ inevitably generates additional latency because of message queuing on a central node, comparing to ZeroMQ).
Parents	FR-NIP-002
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	The prototype of the implementation of the NI Orchestrator given in Section 5.3 of D2.3, uses a pub/sub/query protocol implemented in Zenoh[23] as a communication system. In the context of messages communication protocols, a team from the National Taiwan University (NTU) completed a Performance Study on the Throughput and Latency of Zenoh, MQTT, Kafka, and DDS. The results showed that Zenoh outperforms the other communication protocols with impressive performance numbers [74].

NFR-NIP-002	
Description	NIP shall provide openness of interfaces between orchestration/control tiers and NIFs/NISs to mitigate the dependence on specific network operators/vendors/infrastructure providers/service providers
Version	001M18
Owner	IMEC
Priority	Low
Risk	3
Risk Description	The vulnerability of open interfaces between management and orchestration tiers, and between NIFs, might impose certain security risks that need to be properly handled.
Rationale	Distributed edge networks and cloud can be deployed by different vendors/infrastructure providers, belonging to different Mobile Network Operator (MNO) domains. Thus, it is of utmost importance to provide open interfaces between NIFs and management and orchestration tiers (i.e., edge and cloud) in order to facilitate orchestration operations, and to mitigate the dependence on the vendor-specific configuration of NIFs.
Parents	FR-NIP-002
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	All the interfaces defined in Section 4 of D2.3 are open and follow the standards.

NFR-NIP-003	
Description	NIP shall provide support for multiple virtualization environments for deploying services/applications in distributed domains
Version	002M18
Owner	IMEC
Priority	Low
Risk	1

[23] https://zenoh.io/

Risk Description	The diversity in virtualization environments needs specific maintenance, and virtualization-specific policies for orchestration operations, which significantly increases the complexity of orchestration operations in both tiers within NI-assisted management and orchestration framework.
Rationale	With regards to the limited resource availability within the edge platforms, comparing to the large and resourceful data-center, the lightweight virtualization, and orchestration solutions for small-size programmable devices are required. Thus, containerization proves to be a suitable candidate to deliver a lightweight deployment of services and applications suitable for network edge deployments.
Parents	FR-NIP-000
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	The results of this activity are reported in Sections 3.2.3 and 4.2 of D2.3, where we define the interactions between external entities such as O-RAN and MANO and the NI Orchestrator. External entities can be associated with different domains and use different virtualization environments.

NFR-NIP-004	
Description	NIP shall provide support for federated multi-domain management and orchestration.
Version	003M18
Owner	IMEC
Priority	Low
Risk	3
Risk Description	Management-level agreements are necessary for establishing collaboration between orchestration and management entities, and NIFs in different edge domains.
Rationale	Due to the high mobility of users in 5G and beyond 5G ecosystems, applications are deployed in distributed ways across different edge platforms. Thus, NI-assisted management and orchestration framework needs to support cross-domain/cross-edge service orchestration for achieving seamless service operation.
Parents	FR-NIP-000
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	The results of this activity are reported in Sections 3.2.3 and 4.2 of D2.3, where we define the interactions between external entities such as O-RAN and MANO and the NI Orchestrator. Thanks to the NI Orchestrator and its conflict resolution mechanism, the interaction with such external entities allows a federated approach in which each entity remains in control of its assets while allowing other NI to be implemented and achieving seamless service operation.

NFR-NIP-005	
Description	DAEMON's NIP shall interact with the Network Orchestration Framework aligned with ETSI-NFV-MANO
Version	002M18
Owner	UC3M
Priority	High

Risk	1
Risk Description	The NIP needs to understand what are the Network Services that are currently running in the system, in order to match the network intelligence to them.
Rationale	The Network Orchestrator (either based on an ETSI-NFV-MANO platform or an implementation using ONAP) is the element in the network architecture that keeps track of all the network services (and the network slices implementing them). Therefore, the NIO shall interact with the Network Orchestrator to (we use in the following the ETSI NFV MANO terminology): • The number and type of network slices/ services that are running (available at the NFV-O) • The number and extent of subnetwork slices that are running (available at the NFV-O) • The number and extent of VNFs that are running (available at the NFV-O and VNFM) • The network topology (available at the NFV-O and VIM) This information is required by the NIO to understand, e.g., where to run the NI and to match the already running network services (e.g., an eMBB Network Slice).
Parents	FR-NIP-002
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	The results of this activity are reported in Section 3.2.3 of this deliverable, where we define the interactions between the MANO and the NI Orchestrator.

NFR-NIP-006	
Description	DAEMON's NIP shall interact with the 3GPP Network Analytics System
Version	002M18
Owner	UC3M
Priority	High
Risk	1
Risk Description	The NIP needs to interact with the 3GPP Network Data Analytics System, as the network analytics defined by the standard are NIFs.
Rationale	The network analytics services, as defined by the 3GPP system in [75] are NIFs that need to be orchestrated and managed as the other NIF defined in the project. The producer/consumer NFs in the analytics systems are NIF-C in the DAEMON view. The NWDAF is a particular kind of NIF-C, that implements the model.
Parents	FR-NIP-002
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	The results of this activity are reported in Section 4.2.2 of this deliverable, where we define the interactions between the 5GC analytics and the NI Orchestrator.

NFR-NIP-007	
Description	DAEMON's NIP shall interact with the O-RAN on non-RT RIC and near-RT RIC
Version	002M18
Owner	NEC
Priority	High
Risk	1
Risk Description	The NIP needs to interact with the O-RAN RIC entities, namely non-RT RIC and near-RT RIC, through standard interfaces defined by O-RAN, e.g., via A1, E2, O2

	or O1. The interfaces may need to be extended or new interfaces need to be defined to communicate and interact with DAEMON NIFs.
Rationale	The Non-RT RIC entities (such as rApps) or Near-RT RIC entities (such as xApps), defined in the O-RAN reference architecture, can be provided by the NIFs in the project. The O-RAN RICs can be considered as the consumer of the NIFs for managing open RAN configuration and RAN related functionalities and resources.
Parents	FR-NIP-002
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	The results of this activity are reported in Section 4.2.1 of D2.3, where we define the interactions between the O-RAN RIC Controllers and the NI Orchestrator.

NFR-NIP-008	
Description	The system constraint for NIF selection at the edge are energy, computation, network, and KPIs
Version	002M18
Owner	IMEC
Priority	Low
Risk	1
Risk Description	
Rationale	Depending on the available resources and the business goals or SLAs, DAEMON will select the best NIF model that suits the assisted system. For example, in some cases it might be feasible to sacrifice accuracy at the expenses of a lower computational complexity.
Parents	FR-NIP-000
Current Status	
Percent complete	100%
Risk management	Successful
Rationale	In Section 5.1 of D2.3, we defined the parameters that compose a NIS/NIF Descriptor. Among them, parameters in learning metrics, data-related topics, such as data types, age and output format, and computation were defined. These parameters allow the NI Orchestrator to perform NIF selection, as indicated in Section 5.1.3 of the same deliverable.

NFR-NIP-009	
Description	DAEMON's NIP shall provide native NI procedures to be used by the project developed NIFs
Version	002M18
Owner	UC3M
Priority	High
Risk	1
Risk Description	The NIFs may require some advanced functionality that is provided by the NI Orchestrator, especially for their coordination and execution.
Rationale	As discussed in Section 7 of D3.2 [2], after the N-MAPE-K analysis of the different NIFs designed by the project, some additional features of the NIO are required: namely Knowledge Sharing, Conflict Resolution, and model deployment and re-training.
Parents	FR-NIP-002
Current Status	

Percent complete	100%
Risk management	Successful
Rationale	The results of this activity are reported in Section 5 of this deliverable, where we show the orchestration procedures that will be used by the project developed NIFs.

B Appendix: Literature Review – Final status

In this appendix, we present the final status of the literature review. We include all the reviewed papers and their data that support the findings listed in Section 6. We incorporate such information in this deliverable for completeness in a tabular format to improve readability.

We remind the reader that some of the research questions have a limited set of possible questions, which we detail in the following. Regarding the operation timescale, we distinguish the following cases:

- Very short timescale (us-ms)
- Short timescale (ms-s)
- Medium timescale (s-min)
- Long timescale (min-h)
- Very long timescale (h-days)

For algorithm location, we distinguish between

- Orchestration Plane
- Control plane
- Data plane

For the micro-domain of operation, we have:

- Subscriber
- Access
- Beyond Edge
- Far Edge
- Edge
- Transport
- Core
- Cross domain ("Cross")

With respect to the Application Area, we follow the latest 5GPPP white paper [76] where three major application areas were identified, namely i) Network Planning, ii) Network Diagnostics, and iii) Network Optimization and Control. We include Network Security inside the Network Diagnostics category, as they are correlated.

Like in D2.2 [1], we included two subsections, one dedicated to the questions regarding the network and dataset, and the second dedicated to the machine learning questions. We motivate this split mainly due to the size of the content. Each column shows the information of a given paper, while every row identifies the information related to one of the questions.

B.1 Research questions related to the network and data

	Bib key	**sotocamelo2021atari** [25]	**ayalagarcia2021** [40]	**sotocamelo2021**[26]	**sotocamelo2023** [43]	**goez2022quantizedmodclass** [47]	**akembutun2023** [16]	**akemgucciardo2023** [77]
Network Related	**Networking problem**	Link Evaluation, Throughput Prediction	Energy consumption and performance optimization in vRAN	Resource management	Resource management	Radio Resource Management	Traffic classification	Traffic classification
	Application Area	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control	Network Optimization and control	Network Diagnostics and Security	Network Diagnostics and Security
	Micro-domain	Edge	Edge	Cross-domain	Cross-domain	Access	Transport	Transport
	Algorithm Location	Control Plane	Control plane	Control and Orchestration Plane	Control and Orchestration Plane	Control Plane	Data Plane	Data Plane
	Operation Timescale	Short timescale (ms-s)	Medium timescale (s-min)	Medium timescale (s-min)	Medium timescale (s-min)	Short timescale (ms-s)	Very short timescale (us-ms)	Very short timescale (us-ms)
Dataset Related	**Dataset Generation**	Synthetic	Real	Synthetic	Synthetic	Synthetic	Real	Real
	Dataset Generation Setup	600 deployments with random number of APs and STAs (78078 devices in total). APs are fixed at the center of the cell while STAs are randomly placed around the AP's coverage area (10m)	2 nodes acting as UE and eNB	Analytically generated	Analytically generated	GNU radio was used to generate the waveforms	data collected from 28 IoT devices during a period of 6 months	Multiple datasets
	Dataset Availability	Open	Open	Open	Open	Public	Open	Open
	Data Velocity	End-to-end: around KB/seconds Only prediction: around KB/milliseconds	Second	1 sample per second	1 sample per second	Not provided	Velocity of the data through the switch in the order of MB/seconds	Variable
	Data Variety	Structured Data	Structured Data	Structured Data	Structured Data	Structured Data	Structured Data	Structured Data
	Data Volume	A deployment is given in a csv file with approximately 20 KB per file	3 floating numbers	Not provided	Not provided	20GB	652MB	Not provided
	Data Veracity	The Distributed Coordination Function operation and Dynamic Channel Bonding in Komondor, were validated against ns-3 and the well-known Bianchi and Markov models. However, there is no validation using real data.	Accurate	Accurate	Accurate	Accurate	Accurate	Accurate

	Bib key	**loschiavofiore2022** [28]	**colletbanchs2022** [37]	**colletbazco2023** [38]	**NinaSK2023CCNC** [21]	**Zhu2021**[78]	**ayalagarcia2020** [17]	**nakanoyasato2019** [79]
Network Related	**Networking problem**	Resource management	Resource management	Resource management	Resource management, Resource forecasting	Resource management	Radio and computing resource control	Resource management
	Application Area	Network Optimization and Control	Network Optimization and Control	Network Optimization and control	Network Optimization and control	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control
	Micro-domain	Transport	Cross-domain	Edge/core	Edge/core	Edge and Client	Edge	Cross-domain
	Algorithm Location	Control Plane	Control Plane	Control and Orchestration Plane	Control and Orchestration Plane	Control Plane	Control plane	Orchestration Plane
	Operation Timescale	Predictor: Short timescale (ms-s)	Predictor: Short timescale (ms-s)	Long timescale (min-h)	Short timescale (ms-s)	N/A	Very short timescale (us-ms); Short timescale (ms-s)	N/A
Dataset Related	**Dataset Generation**	Real dataset obtained from operator	Real dataset obtained from operator Synthetic dataset generated through network-simulation pipeline	Real dataset obtained from operator Synthetic dataset generated through network-simulation pipeline	Data generated on the testbed	Synthetic	Real	Real
	Dataset Generation Setup	Not provided	QoE pipeline simulated	Multiple datasets	Real data from Kubernetes clusters deployed on the Roadside units (Kubernetes metrics API)	Number of vehicles: 6, Computation power of VEC server: 6.3GHz, Computation power of vehicle: 1GHz, The data amount per task: [50, 600] kB, Initial price of VEC server: 0.3, Prices of VEC servers: [0, 1], Cost of vehicle: 1	2 nodes acting as UE and eNB	SFC1: Proxy - FW - IDS, SFC2: FW - IDS, SFC3: IDS - FW
	Dataset Availability	Private	Private	Private and public, depending on the evaluation	Yes	Not provided	Open	Not provided
	Data Velocity	GB/seconds	GB/seconds	GB/seconds	Not provided	Not provided	Second	Not provided
	Data Variety	Structured Data	Structured Data	Structured Data	Not provided	Not provided	Structured Data	Structured Data
	Data Volume	N/A	~10GBs	~10GBs	Not provided	Not provided	3xBxT floating numbers (B = number of Base Stations, T = nof monitoring samples per interval (100 in their case))	Not provided
	Data Veracity	Accurate	Accurate	Accurate	Not provided	Not provided	Accurate	Not provided

	Bib key	xiaozhang2019 [80]	quanghadjadj-aoul2019 [81]	peihong2019 [82]	zhengtian2019 [83]	solozabalceberio2019 [84]	foukasradunovic2021 [85]	zhaoliang2019 [86]
Network Related	Networking problem	Resource management	Resource management	Resource management	Resource management	Resource management	Execution times prediction	User association and resource allocation on heterogeneous cellular networks
	Application Area	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control	Network Diagnostics and Security	Network Planning
	Micro-domain	Cross-domain	Cross-domain	Cross-domain	Core	Cross-domain	Edge	UE
	Algorithm Location	Orchestration Plane	Orchestration Plane	Orchestration Plane	Orchestration Plane	Orchestration Plane	Control Plane	Control Plane
	Operation Timescale	Short timescale (ms-s)	Short timescale (ms-s)	Short timescale (ms-s)	Not provided	Not provided	Short timescale (ms-s)	Not provided
Dataset Related	Dataset Generation	Synthetic	Synthetic	Real	Synthetic	Synthetic	Real	Not provided
	Dataset Generation Setup	Not provided	Not provided	Not provided	Not provided	Not provided	Probes at the network deployment	Not provided
	Dataset Availability	Open	N/A	Open	Not provided	Not provided	Not provided	Not provided
	Data Velocity	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided
	Data Variety	Structured Data	Structured Data	Structured Data	Structured Data	Structured Data	Not provided	Not provided
	Data Volume	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided
	Data Veracity	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided

	Bib key	**Bakri2021** [87]	**tripathipuligheddu2021** [88]	**mismarchoi2019** [89]	**xiongZilberman2019** [90]	**gijon21_longterm** [91]	**gutterman19** [92]	**yangcao2020** [93]
Network Related	**Networking problem**	Network Slice Admission Control	Dynamic radio resource allocation in heterogeneous vRANs	Downlink SINR maximization problem given the worst case distribution of network fault predictability both indoors and outdoors	Traffic classification	Traffic Forecasting	Resource Forecasting	Spectrum Access
	Application Area	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control	Network Diagnostics and Security	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control
	Micro-domain	Core	Edge	Edge	Transport	Edge/core	Edge/core	Edge
	Algorithm Location	Control Plane	Control Plane	Control Plane	Data Plane	Control and Orchestration Plane	Control and Orchestration Plane	Control Plane
	Operation Timescale	Short timescale (ms-s; Medium timescale (s-min)	Short timescale (ms-s)	Short timescale (ms-s)	Very short timescale (us-ms)	Very long timescale (h-days)	Very long timescale (h-days)	Short timescale (ms-s)
Dataset Related	**Dataset Generation**	Not provided	Not provided	Not provided	Real	Real	Synthetic	Synthetic
	Dataset Generation Setup	Not provided	Not provided	Not provided	Testbed with 28 different IoT devices (e.g., cameras, sensors, etc.)	Data collected from January 2015 to June 2017 (i.e., 30 months) in a large live LTE network serving an entire country	LTE eNB configured with a 10 MHz bandwidth using 700 MHz wireless spectrum	Two USRP2 connected to respective PCs to generate and collect the RF traces. Authors varied the window size, the SNR levels and data payload size. All the experiments were perform in a 4-node star topology.
	Dataset Availability	Private	Not provided	Not provided	Open	Not provided	Not provided	Private
	Data Velocity	Not provided	Not provided	Not provided	Velocity of the data through the switch in the order of MB/seconds	Not provided	Not provided	Not provided
	Data Variety	Not provided	Not provided	Not provided	Unstructured Data	Not provided	Not provided	Unstructured Data
	Data Volume	Not provided	Not provided	Not provided	PCAP file	Not provided	Not provided	Not provided
	Data Veracity	Not provided	Not provided	Not provided	Accurate	Not provided	Not provided	RF traces are from real devices but might be limited to the ones used during training.

	Bib key	liuyu2020 [94]	camelomennes2020 [95]	yange2020 [96]	wangmao2021 [97]	jiayang2021[98]	nakashimakamiya2020 [99]	xucheng2018 [100]
Network Related	Networking problem	Resource management	Spectrum Sharing	Resource management	Resource management	Resource management	Channel Allocation	Spectrum Sensing
	Application Area	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control	Network Planning
	Micro-domain	Edge	Edge	Cross-domain	Cross-domain	Edge	Access	Access
	Algorithm Location	Orchestration Plane	Control and Orchestration Plane	Orchestration Plane	Orchestration Plane	Orchestration Plane	Control Plane	Control Plane
	Operation Timescale	Not provided	Short timescale (ms-s)	Short timescale (ms-s)	Short timescale (ms-s)	Not provided	Not provided	Not provided
Dataset Related	Dataset Generation	Synthetic	Synthetic	Synthetic	Synthetic	Synthetic	Synthetic	Synthetic
	Dataset Generation Setup	Task generation is modeled as a Poisson process; the distribution of the IoT devices is modeled by a Poisson cluster process.	5 Collaborative Intelligent Radios were sharing the spectrum; the incumbent was a doppler weather radar; radios were sharing 10MHz of bandwidth; the radios were connected to a collaboration network in Colosseum.	Not provided	Not provided	Not provided	Multiple APs were simulated using back-of-the-envelope (BoE) technique, this assumes each AP has a STA and the wireless link is saturated	Multiple PUs (2, 3) and multiple SUs (6, 9) were simulated; the transmission power of each PU is set to 50 mW; transmission signals are assumed to attenuate according to a free-space propagation model with pathloss exponent equal to 4.
	Dataset Availability	Private	Private	Not provided	Not provided	Not provided	Private	Private
	Data Velocity	Not provided	Sample rate of 23.04Mb/s; each scatter voxel contains 35 32-FFT samples	Not provided	Not provided	Not provided	Not provided	Not provided
	Data Variety	Structured Data	Unstructured Data	Structured Data	Structured Data	Structured Data	Structured Data	Structured Data
	Data Volume	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided
	Data Veracity	Good amount of considered parameters allowing to obtain a rich dataset	Not provided	Not provided	Not provided	Not provided	BoE is an easy computation method that produces very accurate results in modest-size networks, however it presents limitations in large-scale networks	The assumptions made might not hold in realistic scenarios (e.g., the coverage area of the PU is a perfect circle)

	Bib key	manousis2021 [101]	perinoyang2020 [102]	iyerli2018 [103]	navarrohuet2021arxiv, navarrorossi2020 [104], [105]	kattadigeraman2021 [106]	manglahalepovic2020 [107]	subramanya2021centralized [108]
Network Related	Networking problem	Anomaly Detection	Network Failure Management	RAN Performance Analysis	Anomaly Detection	Traffic classification	Quality of Experience Prediction	Resource management
	Application Area	Network Diagnostics and Security	Network Diagnostics and Security	Network Diagnostics and Security	Network Diagnostics and Security	Network Optimization and Control	Network Diagnostics and Security	Network Optimization and Control
	Micro-domain	Subscriber	Access	Access	Core	Transport	Transport	Edge/core
	Algorithm Location	Data Plane	Control Plane	Control Plane	Control Plane	Control Plane	Control Plane	Control Plane
	Operation Timescale	Short timescale (ms-s) Millisecond	Long timescale (min-h)	Long timescale (min-h)	Long timescale (min-h)	Long timescale (min-h)	Long timescale (min-h)	Not provided
Dataset Related	Dataset Generation	Private	Private	Private	Private	Public	Private	Private
	Dataset Generation Setup	Not provided	Not provided	6TB traffic per hour.	minute time scale 70K KPIs per router	Hours	Not provided	GB/seconds
	Dataset Availability	Not provided	Structured Data	Structured Data	Structured Data	Structured data	Structured Data	Structured Data
	Data Velocity	Not provided	Not provided	6TB traffic per hour.	Minute time scale 70K KPIs per router	Not provided	Not provided	Not provided
	Data Variety	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided	Accurate
	Data Volume	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided
	Data Veracity	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided

	Bib key	**rahman2018auto** [109]	**huang2021scalable** [110]	**zhang2020tiki** [111]	**prados2020learnet** [112]	**zhu2021network** [113]	**yan2021acc** [114]	**wang2022Hive** [115]
Network Related	**Networking problem**	Resource management	Resource Allocation	Anomaly Detection	Flow Allocation and Admission Control	Planning	ECN Tuning	ML Splitting
	Application Area	Network Optimization and Control	Network Optimization and Control	Network Diagnostics and Security	Network Optimization and Control	Network Planning	Network Optimization and Control	Network Optimization and Control
	Micro-domain	Transport	Core	Cross-domain	Core, Transport	Cross-domain	Core, Transport	Cross-domain
	Algorithm Location	Control Plane	Control and Orchestration Plane	Data Plane	Control Plane	Orchestration Plane	Data Plane	Control Plane
	Operation Timescale	Long timescale (min-h)	N/A	Medium timescale (s-min)	Short timescale (ms-s)	Very long timescale (h-days)	Short timescale (ms-s)	Long timescale (min-h)
Dataset Related	**Dataset Generation**	Private	Public	Public	Private	Public	Private	Private
	Dataset Generation Setup	Sample data in bits collected every 5-min interval for 1.5 month	10 epochs (~	Not provided	Not provided	Not provided	48KB/s bandwidth for one port for data collection	Not provided
	Dataset Availability	Structured Data	Structured Data	Structured Data	Not provided	Structured Data	Structured Data	Not provided
	Data Velocity	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided
	Data Variety	Accurate	Inaccurate	Not provided	Not provided	Not provided	Not provided	Not provided
	Data Volume	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided
	Data Veracity	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided

	Bib key	**he2021towards** [116]	**rossi2019horizontal** [117]	**khaleq2021intelligent** [118]	**zalokostas2022experimental** [119]	**oshea2018modulationclass** [120]	**Jentzsch2022quantizedmodclass** [34]	**rosa2022bacalhaunet** [121]
Network Related	**Networking problem**	Resource management	Resource management	Resource management	Resource management	Radio Resource Management	Radio Resource Management	Radio Resource Management
	Application Area	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control	Network Optimization and control	Network Optimization and control	Network Optimization and control
	Micro-domain	Core	Cross-domain	Core	Cross-domain	Access	Access	Access
	Algorithm Location	Control Plane	Control and Orchestration Plane	Control Plane	Control and Orchestration Plane	Control Plane	Control Plane	Control Plane
	Operation Timescale	Short timescale (ms-s)	Not provided	Not provided	Long timescale (min-h)	Short timescale (ms-s)	Short timescale (ms-s)	Short timescale (ms-s)
Dataset Related	**Dataset Generation**	Synthetic	Real	Synthetic	Real	Synthetic	Synthetic	Synthetic
	Dataset Generation Setup	They use MoonGen, a DPDK based packet generator	event-based system to process geo-spatial data of the taxi trips in New York city.	The traffic represents the amount of tweets that contain words related to disasters.	multi-source dataset of urban life in the city of Milan and the province of Trentino	GNU radio was used to generate the waveforms	GNU radio was used to generate the waveforms	GNU radio was used to generate the waveforms
	Dataset Availability	Private	Open	Private	Open	Public	Public	Public
	Data Velocity	MB/sec	2000 events per second	Not provided	~1200 events per day	Not provided	Not provided	Not provided
	Data Variety	Structured Data	Structured Data	Unstructured Data	Structured Data	Structured Data	Structured Data	Structured Data
	Data Volume	Not provided	130MB	Not provided	Not provided	20GB	20GB	20GB
	Data Veracity	Accurate	Accurate	Inaccurate	Accurate	Accurate	Accurate	Accurate

	Bib key	**fu2019** [122]	**koo2019deep** [123]	**DalgkitsisGarrido2022** [124]	**SantosLynn2021** [125]	**khan2020real** [126]	**minovski2021throughput** [127]	**teixeira2023wi** [128]
Network Related	**Networking problem**	Service function chain embedding	Service function chain embedding	Service Function Chain (SFC) orchestration for multi-domain networks	Service Function Chain placement considering availability and energy consumption	Link Evaluation, Throughput Prediction	Link Evaluation, Throughput Prediction	Link Evaluation, Throughput Prediction
	Application Area	Network slicing	Network slicing	Network slicing	Network slicing	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control
	Micro-domain	Not provided	Not provided	Not provided	Not provided	Edge	Edge	Edge
	Algorithm Location	Control Plane	Control Plane	Control Plane	Control Plane	Control plane	Control plane	Control Plane
	Operation Timescale	Long timescale (s-min)	Long timescale (s-min)	Short timescale (ms-s)	Long timescale (s-min)	Short timescale (ms-s)	Short timescale (ms-s)	Short timescale (ms-s)
Dataset Related	**Dataset Generation**	Two typical SFCs to be embedded	Artificial and real traces	Synthetic	Synthetic	Synthetic and Real	Real	Real
	Dataset Generation Setup	Not provided	Not provided	Not provided	Not provided	Real: Network traces obtained through Wireshark. 6 Wi-Fi stations and an access point. Synthetic: Using Mininet Wi-Fi. 10 Wi-Fi stations and an access point	Only one mobile phone is connected to the 5G network. They used active testing for labeling the dataset. The transmission alternates between UL and DL with intervals of 500ms being idle.	only one vehicle connected to one stationary access point
	Dataset Availability	Private	Not provided	Not provided	Not provided	Private	Private	Open
	Data Velocity	Not provided	Not provided	Not provided	Not provided	54Mbps	Not provided	1 sample per second
	Data Variety	Not provided	Not provided	Not provided	Not provided	Structured Data	Structured Data	Structured Data
	Data Volume	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided
	Data Veracity	Not provided	Not provided	Not provided	Not provided	Accurate	Accurate	Accurate

	Bib key	**busseGrawitz2019** [129]	**Xieli2022** [130]	**Zhengzang2022** [131]	**begagramaglia2019** [132]	**begagramaglia2020** [133]	**Zhangpatras2018** [134]	**Zhangfiore2019** [135]
Network Related	**Networking problem**	Traffic classification	Traffic classification	Traffic classification	Capacity prediction	Capacity prediction	Traffic Prediction	Traffic Prediction
	Application Area	Network Diagnostics and Security	Network Diagnostics and Security	Network Diagnostics and Security	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control	Network Optimization and Control
	Micro-domain	Transport	Transport	Transport	Edge/core	Edge/core	Edge/core	Edge/core
	Algorithm Location	Data Plane	Data Plane	Data Plane	Control and Orchestration Plane	Control and Orchestration Plane	Control and Orchestration Plane	Control and Orchestration Plane
	Operation Timescale	Very short timescale (us-ms)	Very short timescale (us-ms)	Very short timescale (us-ms)	Long timescale (min-h)	Long timescale (min-h)	Long timescale (min-h)	Not provided
Dataset Related	**Dataset Generation**	Real	Real	Real	Real	Real	Real	Real
	Dataset Generation Setup	Testbed network with 12 hosts attacked by 2 hosts	Multiple datasets	Multiple datasets	Real data from 470 4G eNBs of a mobile network deployed in a large metropolitan region of around 100 km^2 and collected at the gateway of an operational mobile network by monitoring the GPRS Tunneling Protocol (GTP).	Real data from 470 4G eNBs of a mobile network deployed in a large metropolitan region of around 100 km^2 and collected at the gateway of an operational mobile network by monitoring the GPRS Tunneling Protocol (GTP).	Real Data from Telecom Italia Dataset for Milan	large-scale mobile traffic dataset collected by a major operator in a large European metropolitan area during 85 consecutive days. 24,482 traffic snapshots for individual service. Each mobile traffic snapshot comprises the traffic demand accommodated by 792 antennas aggregated every 5 minutes.
	Dataset Availability	Open	Open	Open	Private	Private	Private	Private
	Data Velocity	MB/seconds	Variable	Variable	Each 5 minutes	Each 5 minutes	Not provided	Traffic aggregated every 5 minutes
	Data Variety	Unstructured Data	Structured	Structured	Structured Data	Structured Data	Structured Data	Structured Data
	Data Volume	PCAP file of about 8 GB	Not provided	Not provided	Not provided	Not provided	Not provided	36 services x 24,482 traffic snapshots x 792 antennas x 24*60/5 measurements x 85 days
	Data Veracity	Accurate	Accurate	Accurate	Accurate (but used for forecasting, future may not be known from current samples)	Accurate (but used for forecasting, future may not be known from current samples)	Accurate (but used for forecasting, future may not be known from current samples)	Accurate

	Bib key	Trinhgiupponi2018 [136]	camelo2022TrafficClassSpect [57]	oshea2016TraffClassSpec [137]	camelo2019TechClassSpect [138]	camelo2020TraffClassSpec [139]	Dalgkitsis2021TransactionsITS [140]	Ma2020TransactionsWC [141]	Grasso2022TransactionsNSM [142]
Network Related	**Networking problem**	Traffic Prediction	Radio Resource Management	Radio Resource Management	Radio Resource Management	Radio Resource Management	Resource management	Resource management	Resource management
	Application Area	Network Optimization and Control	Network Optimization and control	Network Optimization and control	Network Optimization and control	Network Optimization and control	Network Optimization and control	Network Optimization and control	Network Optimization and control
	Micro-domain	Edge/core	Access	Access	Access	Access	Edge	Edge	Edge
	Algorithm Location	Control and Orchestration Plane	Control Plane	Control Plane	Control Plane	Control Plane	Control and Orchestration Plane	Control and Orchestration Plane	Control and Orchestration Plane
	Operation Timescale	Not provided	Short timescale (ms-s)	Short timescale (ms-s)	Short timescale (ms-s)	Short timescale (ms-s)	Short timescale (ms-s)	Short timescale (ms-s)	Short timescale (ms-s)
Dataset Related	**Dataset Generation**	Real	Real traces of L2 were used to generated synthetic L1 Wi-Fi-compliance packets in an emulator	Real traces of L2 were used to generated synthetic L1 packets	Emulated	Synthetic	Online dataset (taxi traffic in San Francisco Bay Area)	Online dataset (real-world trace of mobile users using Twidere, an open-source Android Twitter)	Online dataset (CIFAR-10 dataset)
	Dataset Generation Setup	one-month of scheduling information gathered by monitoring different eNBs located in the city of Barcelona, Spain	A Wi-Fi access point and one client for generating real L2 and Matlab for L1	GNU radio was used to generate the waveforms	DARPA Colosseum	ns3+Matlab	Online dataset	Online dataset	Online dataset
	Dataset Availability	Private	Public	Private	Private	Private	Yes	Yes	Yes
	Data Velocity	1 ms	20 Mega samples per second (Msps)	1Mbps at L2	20 Msps	20 Msps	Not provided	Not provided	Not provided
	Data Variety	Structured Data	Structured Data	Structured Data	Structured Data	Structured Data	Not provided	Not provided	Not provided
	Data Volume	Not provided	20GB	Not provided	Not provided	Not provided	Not provided	Not provided	Not provided
	Data Veracity	Accurate	Accurate	Accurate	Accurate	Accurate	Not provided	Not provided	Not provided

B.2 Research question related to Machine Learning algorithms

Bib Key	**sotocamelo2021atari** [25]	**ayalagarcia2021** [40]	**sotocamelo2021** [26]	**sotocamelo2023** [43]	**goez2022quantizedmodclass** [47]
ML Method	Supervised Learning (SL)	Reinforcement Learning (RL)	Reinforcement Learning (RL)	Reinforcement Learning (RL)	Deep Learning
ML Problem	Prediction	Control	Decision Making	Decision Making	Classification and regression
Algorithm	Graph Neural Networks	Deep deterministic policy gradient	Deep Q-Learning	Proximal Policy Optimization	CNN
Resource Awareness	No	Yes	No	No	Yes
Model Description	Input: Node Type, Node positioning, Channel Configurations, RSSI, SINR, Airtime, Interference among BSSs, Distance between Nodes, Bandwidth per deployment Output: Throughput per Node per deployment	input: UL and DL channel quality indicator and the "new" bit presence output: configuration policies	State: Number of VNF Replicas, avg CPU usage, peak latency Actions: increase, decrease number of VNFs.	State: Number of VNF Replicas, avg CPU usage, peak latency Actions: increase, decrease number of VNFs.	Input: In-phase and Quadrature (IQ) samples of a modulated RF signal, Output: Modulation, Model: VGG10 CNN with different level of quantization at each layer., running platform: FPGA
Loss / Reward Function	Root Mean-Squared Error (RMSE)	Balance performance and cost in one case, being performance maximization the alternative	adapted from cartpole environment.	minimize the weighted sum of the performance and resource cost	Categorical Cross-entropy
Baseline Comparison	CNN, FNN and GB Heuristic: Truncated Normal Distribution	Only evaluation available	Threshold-based algorithm; PID controller	PI controller	VGG10 CNN with 32-bits floating point; VGG10 CNN with equal quantization value across all layers (2 to 8 bits)
Limitations of ML vs. benchmark	GNNs require more research when operating with edge features. Regularization and normalization do not work out-of-the-box.	Compute intensive inference	Slightly increased tradeoff between number of created VNFs and delay	slightly increased tradeoff between number of created VNFs and delay	Same quantization level is applied to all the layers, limiting the trade-off between model size and model performance.
Advantages of ML vs benchmark	GNNs exploit the graphs' topological information, independently of how many nodes the graph has.	Data efficient. Algorithm converges with a small number of samples	adaptability in defining the thresholds	adaptability in defining the thresholds	Lower energy consumption with higher accuracy since tailor-made selection of the quantization level finds solutions that are not possible with the traditional approach.
Optimality Gap	No optimal results. GNN outperformed ML methods and the Random guesser by 55% and 64%, respectively.	Not provided	No optimal results are available	No optimal results are available	No optimal results are available
Tradeoff ML vs. Benchmark	Not provided	Not provided	The ML method keeps the delay under control at expenses of creating more VNFs than the baseline methods. The SLA violations are reduced	The behavior of the RL algorithm can be tuned depending on the weights of the reward function. The PI controller turned out to be insensitive to those weights.	Selecting the best quantization level per layer obtains higher accuracy with the highest level of quantization possible.

Bib Key	**akembutun2023** [16]	**akemgucciardo2023** [77]	**loschiavofiore2022** [28]	**colletbanchs2022** [37]	**colletbazco2023** [38]
ML Method	Supervised Learning (SL)	Supervised Learning (SL)	Joint Deep Learning and Statistical Modelling	Forecasting/Regression Supervised Learning	Supervised Learning (SL)
ML Problem	Classification	Classification	Control	Loss-Metric Mismatch	Forecasting / Prediction
Algorithm	Random Forest	Random Forest	Recurrent Neural Networks with LSTM	Deep Neural Networks	Deep Neural Network
Resource Awareness	Yes	Yes	No	No	No
Model Description	Input: packet-level features (data extracted from ethernet, ip, tcp/udp headers). Output: classification of packets	Input: packet-level features (data extracted from ethernet, ip, tcp/udp headers) and flow-level features (mean, max, min values computed over interarrival time and packet length). Output: classification of flows of packets	Input: Past traffic Output: Traffic / Capacity / resources forecast	Input: Past states and decisions Output: Next action	Input: Past samples from several time series and/or other variables Output: Directly the MANO decision to realize in order to optimize a certain MANO objective
Loss / Reward Function	Precision, Recall, F1 score	Precision, Recall, F1 score	MSE of forecasted traffic or SLA violations	Self-learned	Self-learned
Baseline Comparison	Monolithic Random Forest classifier	State-of-the-art packet-level classifier (Zhengzang2022)	Not provided	MAE, MSE and ML with Expert-defined loss function	MAE, MSE and ML with Expert-defined loss function
Limitations of ML vs. benchmark	A two-stage hierarchical classifier consumes more switch resources than a monolithic classifier	A flow-level classifier consumes more switch resources than a packet-level classifier	Not provided	Slightly more complexity	Slightly more complexity
Advantages of ML vs benchmark	The classification performance in terms of F1-score, Precision and Recall of the hierarchical classifier is better than the monolithic classifier	The classification performance in terms of F1-score, Precision and Recall of the flow-level classifier is better than the packet-level classifier	Adapts to data with high variation and close-to-zero values Stability	It is able to characterize a complex or even unknown loss function during training. This knowledge can be transferred to other cases and allows explainability.	It is able to characterize a complex or even unknown loss function during training. This knowledge can be transferred to other cases and allows explainability.
Optimality Gap	3% more resources are used by the solution	from 12% to 18% more resources are used by the solution	Not provided	No optimal. It performs better than the "trainer" for standard loss functions.	No optimal. It performs better than the "trainer" for standard loss functions.
Tradeoff ML vs. Benchmark	7%, 21%, 27% better Precision, Recall, F1-score	From 2% to 39% better F1-score, Precision and Recall in four different use cases	Not provided	For complex or unknown loss functions, a small increase of complexity provides significant gains	An internal structure designed with the generic problem in mind (anticipatory decision making) improves considerably the performance and allows to learn how to behave in cases where the optimal solution is not known

Bib Key	NinaSK2023CCNC [21]	Zhu2021[78]	ayalagarcia2020 [17]	nakanoyasato2019 [79]	xiaozhang2019 [80]
ML Method	Supervised Learning (SL)	Reinforcement Learning (RL)	Reinforcement Learning (RL)	Reinforcement Learning (RL)	Reinforcement Learning (RL)
ML Problem	Regression	Learning resource management policy (pricing policy with DDPG, and offloading policy with MADDPG)	Control	Control	Control
Algorithm	Support Vector Regression and The Technique for Order of Preference (TOPSIS)	Deep Deterministic Policy Gradient (DDPG), and Multi-Agent DDPG (MADDPG)	Neural networks (DDPG)	Gradient Boosting Decision-Tree - Monte Carlo Value Iteration	Deep Policy Gradient
Resource Awareness	No	Yes	Yes	No	No
Model Description	Input: CPU, RAM, storage, and end-to-end latency. Output: Predicted end-to-end latency that will be experienced by users	Deep Reinforcement Learning Resource Management (DRLRM)	input: Encoded representation of the scheduler context output: scheduling policy	State: VNF presence at PoPs, allocated CPU and memory per VNF. Action: VNF scaling and migration	State: resource utilization across all servers, links, resource demands of VNs Action: server to host the VNF
Loss / Reward Function	R-squared	Reward of client/server	Minimize operational cost when CPU capacity is sufficient or meet performance target when there is computing deficit	Step1: throughput/latency, Step2: utility/cost	minimize OPEX, maximize throughput
Baseline Comparison	No forecasting	Not provided	1. CVrain-Rlegacy: Proposed CPU orchestrator and legacy scheduler 2. R-Optimal: knows the required CPU and scheduling policies that maximizes the reward 3. T-Optimal: similar to R-Optimal but maximizing throughput 4. Heuristic: lineal model between MCS and CPU load	Conventional RL (without performance profile)	greedy algorithms and a Bayesian learning method
Limitations of ML vs. benchmark	Limited number of edge nodes, possibility to obtain conflicting decisions made at edge and cloud levels	sample efficiency and stability	No optimality guarantees	Additional effort to profile SFCs in terms of performance	neural networks inherently reduce the explainability capacity compared to, e.g., a heuristic method, parameter tuning, training
Advantages of ML vs benchmark	Awareness of quality perceived by the end clients (vehicle), improved end-to-end latency	unstable training of DDPG	Supports non-linear contextual traffic and mobility patterns	Faster convergence and adaptability	adaptability to varying network environment, fast decision-making
Optimality Gap	Increase service reliability to 99.999%, decrease service latency for 105ms	Not provided	2% below optimal	Not provided	Not provided
Tradeoff ML vs. Benchmark	Increase service reliability to 99.999%, decrease service latency for 105ms	Not provided	Not provided	Not provided	Not provided

Bib Key	**quanghadjadj-aoul2019** [81]	**peihong2019** [82]	**zhengtian2019** [83]	**solozabalceberio2019** [84]	**foukasradunovic2021** [85]
ML Method	Reinforcement Learning (RL)	Reinforcement Learning (RL)	Reinforcement Learning (RL)	Reinforcement Learning (RL)	Supervised Learning (SL)
ML Problem	Control	Control	Control	Control	Prediction
Algorithm	Deep Deterministic Policy Gradient based on the Actor-Multiple Critics paradigm	Double deep Q-network (DDQN)	Q-learning (e-greedy)	Neural combinatorial optimization	Decision Tree (quantile decision tree)
Resource Awareness	No	No	No	No	Yes
Model Description	State: resource requirements across all VNFs, Vlinks, Action: ranking of node-to-node and link-to-link pairs.	State: average available bandwidth, memory, CPU and #cores on links, nodes, VNFIs across each region. Action: region combination	State: SFC type to be deployed, Action: server to place the SFC	State: sequence of VNFs (SFC), Action: mapping of VNFs to servers	input: Set of features describing the state of the base station (number of scheduled UEs and their transport block sizes, number of layers)
Loss / Reward Function	acceptance rate	weighted cost	weighted cost	minimize energy consumption with constraint violation penalization	N/A
Baseline Comparison	DDPG	MGSAS (primarily introduced to accelerate embedding solvers by limiting the solution space), Eigen-decomposition (operations on adjacency matrices)	ILP, Bicriteria approximation algorithm, greedy	MILP, First-Fit heuristic	Benchmark for the prediction model: 1. linear regression 2. (non-linear) gradient boosting model
Limitations of ML vs. benchmark	computational complexity (MCN), incorporation of tailored heuristic	repetitive re-training, region selection (not PoP selection) is a strong simplification	optimality gap, memory consumption (conventional Q-learning)	benchmark performs better on small sized problems; constraint satisfaction is not guaranteed - but the probability of occurrence is reduced	Not provided
Advantages of ML vs benchmark	improved performance	improved performance	no need for a-priory knowledge of resource requirements	informative guidance to heuristic for improved performance on large scale networks	Ability to predict task execution times
Optimality Gap	Not provided	Not provided	32% worse than the optimal	within 10%	Not provided
Tradeoff ML vs. Benchmark	Not provided	Not provided	Not provided	Not provided	Not provided

Bib Key	**zhaoliang2019** [86]	**Bakri2021** [87]	**tripathipuligheddu2021** [88]	**mismarchoi2019** [89]	**xiongZilberman2019** [90]
ML Method	Reinforcement Learning (RL)	Reinforcement Learning (RL)	Reinforcement Learning (RL)	Reinforcement Learning (RL)	Supervised Learning (SL)
ML Problem	Prediction	Classification /Control	Prediction	Prediction	Classification
Algorithm	Dueling Double DQN	Deep Q-Learning / Regret Matching	SARSA	Deep neural networks / Deep Q-learning	Decision Tree, SVM, Naive Bayes, K-means
Resource Awareness	no	No	Yes	No	No
Model Description	input: List of allowed actions to be taken by all UEs output: Optimal sequence of actions to achieve QoS requirements of all UEs	Input: Slices	input: SNR, buffer state, and the status of aggregate traffic load already hosted on the available links output: policy to select the best available link and transmission parameters for packet transfer	Input: Initial downlink SINR and target SINR for the voLTE downlink closed loop power control and a set of handling actions in a network for the Fault management solution	input: 11 flow features (e.g., size of the packet, source and destination ports, etc); output: classification of the type of device (e.g., sensor, video, etc)
Loss / Reward Function	The reward function focus on guaranteeing the quality of service (QoS) requirement of UEs.	Not provided	The reward function focus on accomplish the packet loss and latency requirements, being as close as possible to the optimal value	VoLTE PC: the reward function ensures base station's radio link power is constantly tuned to meet the target SINR. Fault management: solves the impact of impairments on DL throughput as experienced by UEs	F1 score, where F1 = 2(precision*recall)/(precision+recall)
Baseline Comparison	Optimal policy	QL- DQL-RM	1. Modified version of the proposed solution where the reward is evaluated by averaging the reward over all the RL agents	1. Fixed power allocation 2. Maximum SINR	DT model with tree depth equal to 11, is implemented in a bmv2 software switch and compared against: 1) the same implementation with a smaller tree depth; 2) an hardware implementation in a NetFPGA with only 5 levels.
Limitations of ML vs. benchmark	Not provided	Not provided	No optimality guarantees	Not provided	Loss of accuracy due to a reduced three depth in both cases
Advantages of ML vs benchmark	near-optimal solution with a small number of iterations	Not provided	Effectively addresses the need for a solution that can swiftly adapt to the underlying channel network dynamics for context-aware radio resource allocation in heterogeneous vRANs	Effectively tunes downlink SINR and number of active faults through exploration and exploitation without the interaction of the UE	Better accuracy
Optimality Gap	Near-optimal results	Not provided	Not provided	Not provided	Loss of accuracy of: 1) 1-2% per each level of the tree; 2) about 9%.
Tradeoff ML vs. Benchmark	Not provided	Not provided	Not provided	Not provided	Not provided

Bib Key	**gijon21_longterm** [91]	**gutterman19** [92]	**yangcao2020** [93]	**liuyu2020** [94]	**camelomennes2020** [95]
ML Method	Supervised Learning (SL)	Supervised Learning (SL)	Supervised Learning (SL)	Unsupervised Learning (UL); Reinforcement Learning (RL)	Semi-Supervised Learning (SSL); Supervised Learning (SL)
ML Problem	Forecasting	Forecasting	Classification	Clustering; Decision Making	Recognition; Forecasting
Algorithm	DT, SARIMA, AHW, RF, ANN, ANN–LSTM, SVR.	Hybrid (X_LSTM: ARIMA+LSTM)	Convolutional Neural Network (CNN)	K-Means; DQN	CNN and Context Tree Weighting (CTW)
Resource Awareness	No	No	No	No	No
Model Description	Input: Past Traffic volume Output: Predicted Traffic Volume	Input: Past Traffic volume Output: Predicted Traffic Volume	Master-CNN Input: IQ samples of RF traces Output: Number of colliding STAs Slave-CNN Input: IQ samples of RF traces Output: ID of the colliding STAs	K-Means Input: User Priority Output: Which devices must perform local task computation. DRL Actions: transmission power of the device States: channel gain, task queue, remaining computation capacity of each device.	Technology Recognition Input: RF traces of different radio technologies and idle noise Output: technology presence in a given spectrum voxel Spectrum Usage Pattern Predictor Input: The transmission pattern of an incumbent Output: Forecast the pattern of future incumbent transmission
Loss / Reward Function	MAE	REVA: combining the amount of average Physical Resource Blocks with individual channel bearer conditions	Cross-entropy	K-Means: Silhouette Coefficient and the sum of squared error (SSE) for selecting the number of clusters. DRL: weighted sum of energy consumption and task execution latency of computing the tasks locally or at the edge	Technology Recognition Binary Cross-Entropy Spectrum Usage Pattern Predictor N/A
Baseline Comparison	Among themselves	simple LSTMs	IEEE802.11 DCF implemented in ns-2	Heuristics	Not provided
Limitations of ML vs. benchmark	longer to converge/train	Not provided	Trade-off between the performance gain and the inference accuracy.	No optimality guaranteed	Not provided
Advantages of ML vs benchmark	Better MAE/MAPE	degree of prediction accuracy with a MAPE	The colliding transmissions can be rescheduled, improving the overall throughput (performance gain w.r.t. standard IEEE802.11 DCF)	adaptability to varying network environment; independent decision-making	Fast Fourier Transform (FFT) representation instead of raw IQ samples reduces the amount of samples to be processed in the TR. The CTW has low time and space complexities with theoretical performance guarantees; the algorithm does not require offline training
Optimality Gap	Not provided	Not provided	Not provided	Not provided	Not provided
Tradeoff ML vs. Benchmark	Not provided	Not provided	Not provided	the DRL optimizes the system cost but does not outperform baselines.	The execution time of the two-step approach can be easily implemented in a RIC.

Bib Key	**yange2020** [96]	**wangmao2021** [97]	**jiayang2021** [98]	**nakashimakamiya2020** [99]	**xucheng2018** [100]
ML Method	Reinforcement Learning (RL)	Reinforcement Learning (RL)	Reinforcement Learning (RL)	Reinforcement Learning (RL)	Unsupervised Learning (UL)
ML Problem	Control	Control	Control	Decision Making	Clustering
Algorithm	A3C and GCN	Double Deep Q-network	A3C	Deep Q-Learning; Graph Convolutional Neural Network	Non-parametric Bayesian Model
Resource Awareness	No	No	No	No	No
Model Description	State: max CPU and BW on each node, residual CPU and BW on each node, virtual node CPU and BW requirements, remaining virtual nodes to be placed Action: physical node to embed current virtual node	State: initial, occupied, reserved resources of each DC, similar state of each link, features of current SFC. Action: (a,b) pair where a is the DC for actual deployment and b is the DC for the standby SFC instance	State: VNF type, number of VNFs remaining in the chain, length of SFC chain, VNF computation load, remaining length of the chain, remaining time to deadline. Action: defer rate, i.e., the probability that a VNF scheduling will be deferred for the next scheduling event.	State: adjacency matrix and current channel allocation Actions: new channel allocation	Input: Timeseries of spectrum sensing data of different Secondary Users (SUs) Output: Number of spectrum states
Loss / Reward Function	Shaped reward that combines acceptance ratio, revenue, cost, load balancing, and eligibility traces	{1, if SFC is placed, -1 otherwise}	based on whether the SFC execution occurred prior to its deadline	The reward function is the average throughput of the lower 40% APs	Not provided
Baseline Comparison	MCTS (MCVNE), relaxed MILPs (R-Vine, D-Vine), NodeRank, GRC	Random greedy, best-fit greedy, near optimal sorting greedy, deep q network	Earliest Finish First, Earliest Start First, DQN	1. Random Allocation 2. DQN + CNN 3. Potential game-based	1. Energy Detection 2. Gaussian Mixture Model - Expectation Maximization 3. Gaussian Mixture Model - Bayesian Information Criterion 4. Mean Shift
Limitations of ML vs. benchmark	The proposed algorithm requires lots of computational resources for its parallel implementation.	offline training, neural networks reduce explainability	A3C typically trains multiple agent workers to improve stability, which requires many computational resources.	Not provided	The spectrum sensing performance degrades when more Primary Users (PUs) are present for all proposed methods
Advantages of ML vs benchmark	Improved performance and adaptability w.r.t. varying VN request types	fast decision making	faster convergence compared to the DQN, improved acceptance rate	Using GCN instead of CNN the learning performance improves.	The proposed method is more robust since it takes advantages of the spatial-temporal characteristics of the timeseries data.
Optimality Gap	Not provided	slightly worse than a near-optimal method	in terms of reliability (which is the scope of the paper), at least 5% better	No optimal results are available	Not provided
Tradeoff ML vs. Benchmark	Not provided	Not provided	Not provided	The proposed method achieves better reward.	Not provided

Bib Key	**manousis2021** [101]	**perinoyang2020** [102]	**iyerli2018** [103]	**navarrohuet2021arxiv, navarrorossi2020** [104], [105]	**kattadigeraman2021** [106]
ML Method	Unsupervised Learning (UL)	Supervised Learning (SL)	Likely supervised (not clearly indicated)	Unsupervised Learning (UL)	Supervised Learning (SL)
ML Problem	Prediction	Classification	Classification and regression	Classification	Classification
Algorithm	Gaussian Processes	XGBOOST	Multi-task learning with ensembles	10 anomaly detection algorithms (based on distance, density, clustering, subspace) for batching and stream and feature scoring algorithms.	XGBOOST
Resource Awareness	No	No	No	No	No
Model Description	Not provided	Input: features built on alarms from cell site and additional information (e.g., weather, location, power supply type) Output: failure permanent or temporal.	Input: features built on bearer records, signaling records, TCP flow level statistics, network elements records. PCA is then used for grouping of datasets across different cells. Output: cell drop rate classification or throughput prediction.	Input: router KPIs Output: tickets for anomalies	Input: features derived from packet level traces or TCP flow level information and video session level. Output: is the flow a 360 video streaming or regular streaming
Loss / Reward Function	Not provided	Not provided	weighted sum of the component of a given base station and the component of the group of base stations (defined by the PCA)	Not provided	Not provided
Baseline Comparison	Not provided	1. Original policy 2. LSTM 3. Experience-based threshold 4. probability -based threshold	1. per base station modelling 2. different spatial grouping methods for cells 3. grouping only	The paper compares the 10 AD algorithms and the 10 FS algorithms.	1. Heuristics based on threshold on input fields 2. Different ML models as CNN, Multi-layer Perceptron, KNN, naive bayes
Limitations of ML vs. benchmark	Not provided	The ML approach is more expensive than heuristics	It requires a more complex design than the benchmarks	Not provided	The ML approach is more expensive than heuristics based on thresholds
Advantages of ML vs benchmark	Not provided	The approach is simpler and less computational expensive than LSTM while providing the same benefits.	Superior performance with limited data available. Best delay/performance tradeoffs	Not provided	XGBOOST performs the best among different other models tested.
Optimality Gap	Not provided	Not provided	Not provided	Not provided	Not provided
Tradeoff ML vs. Benchmark	Not provided	Not provided	Not provided	Not provided	Not provided

Bib Key	**manglahalepovic2020** [107]	**subramanya2021centralized** [108]	**rahman2018auto** [109]	**huang2021scalable** [110]	**zhang2020tiki** [111]
ML Method	Supervised Learning (SL)	Supervised Learning, Federated Learning	Supervised Learning (SL)	Reinforcement Learning; Federated Learning	Joint Deep Learning and Statistical Modelling
ML Problem	Classification	Forecasting	Classification	Control	Classification
Algorithm	SVM, k-NN, XGBoost, RF and MLP	FNN; LSTM; CNN-LSTM	DT; RF; MLP and Bayesian Networks	Deep Q-Learning	DNN, CNN, C-LSTM, DSE
Resource Awareness	No	No	No	No	No
Model Description	Input: features derived from TLS flow level information and video session level. Output: target QoE metric (i.e., video quality, rebufferig ratio, combined QoE)	input: avg traffic load per second or number of VNF in a given time. Output: expected avg traffic load in next prediction window or the number of VNFs in the next prediction window	Input: traffic load and traffic load change between time intervals. Output: instances in the next time interval.	State: information of all network resources and configurations (remaining and demanded capacity of nodes and links) Actions: VNF deployment and perception and allocation	Input: flow features Output: one-to-all classification: binary (attack vs no attack) and one-to-one classification (which attack type of 14 available)
Loss / Reward Function	Not provided	Huber and MSE	Not Provided, probably cross-entropy.	Minimize the total weighted cost of deploying an SFC. The cost is composed of the communication, setup and operation of SFC.	cross-entropy
Baseline Comparison	1. the different ML methods 2. packet level traces (still based on ML techniques)	Naïve approach: the expected avg traffic load in the next prediction window is the same as the previous one.	Moving Average	Neural Combinatorial Optimization and Branch and Bound	DNN, CNN and C-LSTM
Limitations of ML vs. benchmark	The approach based on packet level is 5-7% superior. But it is also based on ML.	CNN-LSTM model outperforms purely LSTM since the use of the CNN provides extra information and learn internal representation of the time-series data	No optimality guaranteed	The utilization margin of the ML method compared to the baseline reduces as the SFC length and number increases.	The new method requires implementing a voting scheme, re-train the NNs and implement a new DSE neural network
Advantages of ML vs benchmark	The approach based on flow level information is amenable to actual usage, while packet level traces one do not scale.	All the ML algorithms outperform the baseline	All the ML algorithms outperform the baseline	The ML method achieves maximal utilization ratio. The	The new approach fully prevents five mainstream black-box adversarial attacks from compromising deep learning-based NIDS
Optimality Gap	Not provided	Not provided	Not provided	Regarding network cost, the BandB method perform the best, but the ML method closely follows. The deviation is not big	No optimal results are available
Tradeoff ML vs. Benchmark	Not provided	The FL approach does not perform better than the centralized. The model average does not provide better results than not averaging the weights but to using them for the next node.	Random Forest provides higher precision highest ROC area and lowest false positives. The pattern of data and feature set favors decision tree algorithms. More features do not imply better accuracy due to repetitive patterns in the input data.	The ML method exhibits better convergence performance, higher average reward, and smaller average resource consumption than the baseline policies over a variety of network scenarios	Not provided

Bib Key	**zhu2021network** [113]	**yan2021acc** [114]	**wang2022Hive** [115]	**he2021towards** [116]	**rossi2019horizontal** [117]
ML Method	Deep Reinforcement Learning	Multi-agent Deep RL	Split graph transformation (actually not ML)	Reinforcement Learning (RL)	Reinforcement Learning (RL)
ML Problem	Decision Making	Decision Making	multi-split ML problem	Control	Decision Making
Algorithm	Actor-Critic algorithm	Deep Q-learning, four-layer NN	distributed min-cost graph algorithm (actually not ML)	GNN + A3C	Modified Q-Learning
Resource Awareness	No	No	No	No	No
Model Description	states: work topology and the node features, actions: action representation indicates which link to select to add capacity and how much capacity to add	states: collectible statistics, actions: ECN setting (i.e., high marking threshold, low marking threshold, and marking probability)	Input: nodes and ML layers, Output: mapping of layers to be executed in which node, Actions: deploy the layers in the nodes	Input: VNF's status (input and output traffic rate, latency, memory and CPU utilization), NS chain. Output: scaling decisions (scale up, down or not) Model: A3C Actions: Scaling in/out or do nothing	State: number of containers, CPU utilization, CPU share Action: Vertical (increase or decrease the CPU share) or horizontal scaling (increase or decrease the number of replicas)
Loss / Reward Function	the goal is to minimize the cost of the network, the ultimate reward is the cost of a network plan	trade-off between high link utilization and low queue buildup for each switch	Not provided	reward based on minimizing the overall system cost (packet loss and VNF instance cost)	minimize the weighted sum of the performance, adaptation and resource cost
Baseline Comparison	ILP vanilla and ILP + heuristic	static ECN settings in the switches	non-splitable ML models	Other ML approaches including a NN and a decision tree	Deep Q-Learning, Model-based RL
Limitations of ML vs. benchmark	NeuroPlan requires training, optimality is not guaranteed	ACC requires offline training, and requires specific DRL agents in the switches	the ML splitting is not done through ML itself, but using graph theory	The baseline was compared against the GNN. The method for control in the proposed method and the baseline is the same (an RL algorithm)	This set of experiments has shown the importance of providing system knowledge to improve the learning task. This is exploited by the Model-based RL, which shows better performance above the others
Advantages of ML vs benchmark	NeuroPlan overcomes the scalability issues existing in ILPs	Dynamic ECN settings that improve application performance	lower latency, lower energy consumption	The GNN is able to capture the interdependencies between VNFs (in the form of graphs) while the other methods not.	Not provided
Optimality Gap	ILP fails to scale to large topologies. NeuroPlan outperforms ILP + heuristic on large topologies and avoids human efforts to tune the heuristics	Not provided	Not provided	Not provided	Not provided
Tradeoff ML vs. Benchmark	the relax factor provides a convenient and tunable knob for the trade-off between optimality and tractability	Not provided	Not provided	From the figures of the results, the proposed method reduces around 30% the overall cost compared to baselines	Increasing the action space size (considering more actions) difficult the learning convergence of the agent.

Bib Key	**khaleq2021intelligent** [118]	**zalokostas2022experimental** [119]	**oshea2018modulationclass** [120]	**Jentzsch2022quantizedmodclass** [34]	**rosa2022bacalhaunet** [121]
ML Method	Reinforcement Learning	Supervised Learning (SL)	Deep Learning	Deep Learning	Deep Learning
ML Problem	Control	Prediction	Classification and regression	Classification and regression	Classification and regression
Algorithm	Several RL algorithms including SARSA, DQN, PPO and Actor Critic	ARIMA, XGBoost, LSTM	CNN	CNN	CNN
Resource Awareness	No	No	No	Yes	Yes
Model Description	State: min/max/current num of replicas, current resource utilization (e.g., CPU utilization) and current response time. Actions: Scaling in/out or do nothing	Input: incoming requests to a web server Output: number of requests to the web server in a future time interval	Input: In-phase and Quadrature (IQ) samples of a modulated RF signal, Output: Modulation, Model: VGG10 and ResNet. Running platform: GPU/CPU	Input: In-phase and Quadrature (IQ) samples of a modulated RF signal, Output: Modulation, Model: VGG10 CNN with equal quantization value on all layers., running platform: FPGA	Input: In-phase and Quadrature (IQ) samples of a modulated RF signal, Output: Modulation, Model: CNN using Depth-wise Separable Convolutions with quantization + pruning, running platform: FPGA
Loss / Reward Function	Reward if response time is below a target, otherwise penalize	MAE	Categorical Cross-entropy	Categorical Cross-entropy	Categorical Cross-entropy
Baseline Comparison	Generic auto-scaler module (HPA) from Kubernetes	Kubernetes off-the-shelf horizontal pod auto-scaler based on a custom specific metric	XGBoost with features extracted from higher order moments	Same model but using 32-bits floating point, or quantized and compiled to run on GPU	Same model but using 32-bits floating point
Limitations of ML vs. benchmark	Not provided	ARIMA uses raw data, while for the other models, data must be prepared accordingly	DNN is more computational expensive and requires accelerated hardware.	Models quantized to run on FPGA are more energy efficient than the ones on GPU.	Models quantized and pruned are more energy-efficient. In addition, Depth-wise Separable Convolutions help to reduce the size of the network with lower impact on accuracy compared to traditional CNN
Advantages of ML vs benchmark	The RL auto-scalers can autonomously identify the autoscaling values or thresholds. The default HPA auto-scaler cannot determine such thresholds autonomously, since they are determined by expert knowledge.	As the scaling is based on forecasting, the scaling decisions happen earlier, achieving better CPU utilization	No need of expert knowledge and it achieves higher accuracy	Not provided	Not provided
Optimality Gap	The response time was improved 20% regarding the baseline	Not provided	No optimal results are available	No optimal results are available	No optimal results are available
Tradeoff ML vs. Benchmark	Not provided	The HPA is reactive but simpler.	DNN provides higher accuracy but computational are more expensive.	Contrary to expectations, the model quantized at 8 bits outperform the one with 32 bits floating point representation in accuracy. This may be an indication that the baseline is overparametrized or the representation is robust to support quantization.	Minimal impact in accuracy while reducing up to 63x the model size.

Bib Key	**fu2019** [122]	**koo2019deep** [123]	**DalgkitsisGarrido2022** [124]	**SantosLynn2021** [125]	**khan2020real** [126]
ML Method	Reinforcement learning	Reinforcement learning	Reinforcement learning	Reinforcement learning	Supervised Learning (SL)
ML Problem	decision making	decision making	decision making	decision making	Prediction
Algorithm	DRL	RL (Policy Gradient)	RL	RL	Multi-Layer Perceptrons; Support Vector Regressor; Decision Trees; Random Forest
Resource Awareness	yes	yes	Yes (energy consumption)	Yes (energy consumption)	No
Model Description	Input: the current state of the substrate network Output: the allocated server for a given VNF	The resource allocation problem is formulated as an MDP. They model a network service as a set of compute and transport resources without additional structure.	Multiple RL agents instantiated in each domain that perform VNF orchestration. Distributed decision engine with auction mechanism to decide the intra-domain offloading of VNFs	The input is the available resources (set of servers, etc.) and the output is a SFC placement composed of several VNFs.	Input: number of transmitting/receiving stations, RSSI, MCS, Data rate, Interarrival time, Channel Bandwidth Output: predicted throughput per station
Loss / Reward Function	reward for successful embedding of the SFC	reward on successful embedding	Maximal sum of rewards in each domain, considering energy consumption and latency	Reward on fulfilling the SFC requirements, successful placements, and reduced consume of energy	Mean Absolute Error, Mean Squared Error, R-squared
Baseline Comparison	Not provided	Not provided	Not provided	Not provided	Actual throughput values from the synthetic and the real dataset
Limitations of ML vs. benchmark	embeds one VNF at the time and rolls back if there is a failure	models a network slice as one workload	Only considers VNF deployments on servers (not edge or IoT Devices)	Limitations in the resource modeling (not consider link bandwidth, memory or storage). The energy consumption model is very simple	Different models must be defined depending on the number of stations and number of features.
Advantages of ML vs benchmark	Not provided	Not provided	Considers multi-domain SFC deployments using an Auction mechanism to allow inter-domain VNF migration	The RL agents learn to place VNFs in those nodes with more available resources and minimizing the energy consumption	ML learns and adapts to the varying environment and network conditions.
Optimality Gap	Not provided	Not provided	reduce average service latency by 103.4% and energy consumption by 17.1%compared to a centralized RL solution	Not provided	Not provided
Tradeoff ML vs. Benchmark	Not provided	Not provided	reduce average service latency by 103.4% and energy consumption by 17.1%compared to a centralized RL solution	Compare the acceptance rate of SFCs with PPO, A2C and greedy	Highly accurate prediction of transmission throughput in real-time. However, the results showed that the models were overestimating the throughput.

Bib Key	**minovski2021throughput** [127]	**teixeira2023wi** [128]	**busseGrawitz2019** [129]	**Xieli2022** [130]	**Zhengzang2022** [131]
ML Method	Supervised Learning (SL)	Not provided	Supervised Learning (SL)	Supervised Learning (SL)	Supervised Learning (SL)
ML Problem	Prediction	Not provided	Classification	Classification	Classification
Algorithm	MLPs; Support Vector Regressor; XGBoost; Random Forest	Symbolic Regression + Unscented Kalman Filter	Random Forest	Random Forest	Random Forest
Resource Awareness	No	Yes	Yes	Yes	Yes
Model Description	Input: RSSI, RSRP, RSRQ, SINR, Band, Num of carriers, RSPath loss, cell load, CQI, CRI, etc. Output: available throughput per UE in UL and DL	Input: mean throughput, RSSI, speed, location, transmission data rate Output: predicted throughput	Input: more than 80 flow features (e.g., size of the packet, inter-arrival time, etc); Output: classification as malware or benign flow	Input: packet-level features (data extracted from ethernet, ip, tcp/udp headers). Output: classification of packets	Input: packet-level features (data extracted from ethernet, ip, tcp/udp headers). Output: classification of packets
Loss / Reward Function	Mean Squared Error, R-squared	Root-Mean-Squared Error	F1 score	Classification accuracy	Classification accuracy
Baseline Comparison	Not provided	Multiple Linear Regression (MLR); Support Vector Regression (SVR); Decision Tree (DT); Random Forest (RF); and Shallow Neural Network (SNN)	1) the same model in a floating-point-operation system and 2) an offline system (running on a server) that operates over the full flow	Offline model that does not perform any "knowledge distillation"; packet-level classifier	The in-switch classifier is compared with the same classifier running on a server
Limitations of ML vs. benchmark	No optimality guarantees	The best performing model must be chosen manually, while for the other ML models, Bayesian optimization can be used for hyperparameter tunning.	Loss of accuracy due to compression techniques for memory optimization; no floating point	Loss of accuracy due to the distillation process	The in-switch classifier has almost the same accuracy than the benchmark
Advantages of ML vs benchmark	Two of the models were deployed on real devices to perform throughput prediction	Given their simplicity, the proposed model is especially suited for embedded systems, such as those that, due to CPU and memory limitations, are unable to leverage more advanced machine learning algorithms.	The classification is performed at line-rate (directly at the switch)	The distilled model consumes less resources than the benchmark	The classification is performed at line-rate (directly at the switch)
Optimality Gap	Not provided	Not provided	2% below optimal (in accuracy)	1% less accuracy in some use cases with respect to the performance benchmark	Almost no gap
Tradeoff ML vs. Benchmark	The predictions are made during the idle period and not during the connected period, also the patterns seen during these two periods are very different which hinders the prediction accuracy.	Throughput measurement via active probes might introduce unwanted congestion. The proposed model passively observes the variables to produce an outcome	The in-switch classifier performs slightly worse than the offline classifier, but it classifies at line-rate	Up to 60% memory saving with respect to the resource benchmark	There is a trade-off memory vs performance

Bib Key	**begagramaglia2019** [132]	**begagramaglia2020** [133]	**Zhangpatras2018** [134]	**Zhangfiore2019** [135]
ML Method	Supervised Learning (SL)	Supervised Learning (SL)	Supervised Learning (SL)	Supervised Learning (SL)
ML Problem	Forecasting / Prediction	Forecasting / Prediction	Forecasting / Prediction	Forecasting / Prediction
Algorithm	Deep Neural Network	Deep Neural Network	Deep Neural Network	LSTM
Resource Awareness	No	No	No	No
Model Description	Input: Previous traffic measurements (But it allows other inputs, e.g., signal quality, occupied resource blocks, computational load of VNF) Output: forecast of the capacity required to accommodate the future demands for a specific network slice	Input: traffic generated by each slice during the preceding N re-orchestration opportunities. Data structured as a 3D matrix, where base stations represent different "pixels" and different slices represent different "color channels". Output: Allocated resources for each slice. Two-timescales: long-term resources, short-term readaptation.	Input: Traffic volume Output: Traffic Volume	Input: traffic measurements per 5 minutes over many antennas for different services Output: Forecast of those same time series
Loss / Reward Function	Tailored loss function for capacity forecasting (asymmetric cost between overprovisioning and under provisioning, because the latter = SLA violation)	The loss functions are tailored to capacity forecasting with different hyperparameters for each of the cases (e.g., one to avoid under provisioning, other restricting overprovisioning, etc.)	Least Square error (L2 Loss function)	MSE
Baseline Comparison	- Same ML architecture without tailored loss function (just Mean Absolute Error) - Naïve (Replicate last week value) - Infocom17 (first DL approach for mobile traffic prediction) - MobiHoc18 (SoA Network demand prediction) - Previous cases with overprovisioning	- wangtang2017: Custom-built DNN, traditional demand predictor, agnostic of all resource management costs - begagramaglia2019: Custom-built DNN, it takes anticipatory decisions on capacity allocation that aim exclusively at minimizing the trade-off of overprovisioning and non-serviced demands	Machine Learning, ARIMA HW-ExpS	Other NN configurations: MLP, CNN, LSTM
Limitations of ML vs. benchmark	Loss function (single) parameter needs to be tuned (also advantage)	- More hyperparameter configuration	Not provided	None (benchmarks are simpler ML models)
Advantages of ML vs benchmark	- Tailored Loss function allows to obtain much better performance because it adapts to the problem (of capacity allocation)	- Considers and optimizes instantiation costs, re-configuration costs, and two different hierarchical time-scales	More accuracy than with other methodologies	Much higher accuracy
Optimality Gap	Not provided	Not provided	Near optimal results	Mean absolute error w.r.t. real future traffic is only 13KBps
Tradeoff ML vs. Benchmark	Not provided	Solution cuts management costs down to 35-41% vs. an optimal static provisioning of resources. Solution reduces SLA violations due to insufficient available resources by 80% (0.7-1.21% of the re-orchestration opportunities versus at least 5.80%	Not provided	Not provided

Bib Key	**Trinhgiupponi2018** [136]	**camelo2022TrafficClassSpect** [57]	**oshea2016TraffClassSpec** [137]	**camelo2019TechClassSpect** [138]
ML Method	Supervised Learning (SL)	Supervised Learning (SL)	Supervised Learning (SL)	Semi-Supervised Learning (SSL)
ML Problem	Forecasting / Prediction	Classification	Classification	Classification
Algorithm	LSTM	Deep Neural Network	Deep Neural Network	Deep Neural Network
Resource Awareness	No	No	No	No
Model Description	Input: aggregated cell traffic measurement over several TTI Output: Multiple timesteps forecast of such traffic per cell	Input: In-phase and Quadrature (IQ) samples of a modulated RF signal, Output: Type of traffic at different layers (L1 up to L7), Model: a CNN and a RNN, running platform: GPU	Input: In-phase and Quadrature (IQ) samples of a modulated RF signal, Output: Type of traffic at different layers (L1 up to L7), Model: a RNN, running platform: GPU	Input: In-phase and Quadrature (IQ) samples of different radio technologies, Output: Radio Technologies, running platform: GPU
Loss / Reward Function	Normalized Root Mean Square Error (NRMSE)	Categorical Cross-entropy	Categorical Cross-entropy	Categorical Cross-entropy
Baseline Comparison	- Ground truth - ARIMA - Deep Feedforward Neural Network (FFNN)	CNN classifying byte-based packets	Not provided	CNN trained using SL
Limitations of ML vs. benchmark	Not provided	ML models are mainly based on DNN, so they act as black boxes and are computational expensive.	ML models are mainly based on DNN, so they act as black boxes and are computational expensive.	The SSL approach depends on finding models that can accurately extract features (unsupervised step) that later on will be used for the classification part.
Advantages of ML vs benchmark	Much higher accuracy	They can classify traffic using raw L1 packets (IQ samples), which can be modulated and encrypted.	They can classify traffic using raw L1 packets (IQ samples), which can be modulated and encrypted.	SSL allows using large amount of unlabeled data to reduce the need of labeled data and get high accuracy
Optimality Gap	Below 0.05 of NRMSE	>90% accuracy in the hardest classification task	>84% overall	Almost zero since SSL provides solutions with accuracy closed to the SL approach
Tradeoff ML vs. Benchmark	Error increases as the predictive step increases	The more higher layer protocol and granularity, the hardest to classify correctly.	The performance of RNN is good using short sequences of IQ samples. However, RNN consumed more resources that a CNN.	Two steps training but It reduces the size of the labeled data set to achieve high accuracy.

Bib Key	**camelo2020TraffClassSpec** [139]	**Dalgkitsis2021TransactionsITS** [140]	**Ma2020TransactionsWC** [141]	**Grasso2022TransactionsNSM** [142]
ML Method	Supervised Learning (SL)	Supervised Learning (SL)	N/A	Reinforcement learning
ML Problem	Classification	Decision making	Prediction	Decision making
Algorithm	Deep Neural Network	Deep Learning	Two-timescale Lyapunov optimization	Deep Reinforcement Learning, Neural network
Resource Awareness	No	No	No	No
Model Description	Input: In-phase and Quadrature (IQ) samples of a modulated RF signal, Output: Type of traffic (TCP vs UDP) and traffic pattern, Model: a CNN, running platform: GPU	Input: Resource consumption at the edge, Output: Average service delay in mobility scenarios with or without AI-assisted orchestration	Input: Long term migration cost, Output: Average user perceived latency, Average queue backlog	Input: Initial/Final exploration ratio, Number of hidden layers, minibatch size, discount factor, replay memory size; Output: Average processing delay and delay jitter (task execution at UAV level)
Loss / Reward Function	Categorical Cross-entropy	Not provided	Not provided	Mean Squared Error
Baseline Comparison	CNN classifying byte-based packets	Mobility + orchestration scenarios compared against Mobility only and Without mobility and orchestration	Always Migration Algorithm (AM), No Migration (NM), Lazy Migration (LM), Predictive Lazy Migraton (PLM)	Probabilistic Computation Offloading (PCO): each MEC server making independently online offloading decisions, and heuristics: Local Drone Only (LDO), and Uniform Selection (US)
Limitations of ML vs. benchmark	ML models are mainly based on DNN, so they act as black boxes and are computational expensive. It assumes L1 packets can be separated per user stream, which in reality is very hard at spectral level (user identification)	Simulation study, decreasing orchestration time not taken into account	Increased complexity	Management of energy of UAVs not considered resource-awareness not included
Advantages of ML vs benchmark	They can classify traffic using images representing the spectrum (IQ, FTT, short time FFT), which can be modulated and encrypted.	Measuring impact of service orchestration of challenging mobile services	Service quality improvements based on real-life datasets	Significant improvements in task execution time on the MEC level (UAV)
Optimality Gap	>96% overall	Average rejection rate of critical services reduced from 20.2% to 3.9%	Latency reduction ratio of 30.4%, reduction of average queue backlog 19.3%	Proposed solution meets requirements of various 6G applications
Tradeoff ML vs. Benchmark	Not provided	Average rejection rate of critical services reduced from 20.2% to 3.9%	Latency reduction ratio of 30.4%, reduction of average queue backlog 19.3%	Proposed solution meets requirements of various 6G applications