D5.2 INITIAL REPORT ON ORCHESTRATION AND MONITORING SOLUTIONS AND CYBERSECURITY THREATS FOR 6G-NTN

Revision: v.1.0

Work package	WP 5
Task	5.2, 5.3
Due date	30/06/2024
Submission date	21/07/2024
Deliverable lead	TH-SIX
Version	1.0
Authors	Farid Benbadis (TH-SIX), Alice Piemonti (MAR), Vito Cianchini (MAR)
Reviewers	Massimo Neri (MAR), Madivanane Nadarassin (TASF)
Abstract	*This deliverable describes the proposed orchestration and management architectures for 6G networks over ad hoc constellations. The main focus will be drawn on the split of different functions and VNFs across the proposed architecture in WP2 with a particular attention on open interfaces and generic frameworks for integrating AI and ML algorithms. The document will also assess the cybersecurity threats and vulnerabilities induced by the proposed virtualization framework. A comprehensive threat analysis stemming from extending the 6G network to the satellite payloads will be conducted. Led by Task 5.2 with input from Task 5.3.*
Keywords	6G-NTN, Dynamic Orchestration, Autonomous Monitoring, Kubernetes, Security, Non-Terrestrial Networks, Machine Learning, API, Authentication, Scalability

Grant Agreement No.: 101096479
Call: HORIZON-JU-SNS-2022

Topic: HORIZON-JU-SNS-2022-STREAM-B-01-03
Type of action: HORIZON-JU-RIA

Document Revision History

Version	Date	Description of change	List of contributor(s)
V0.1	01/02/2024	Document creation	Farid Benbadis (TH-SIX)
V1.0	21/07/2024	Approved version for submission	Alessandro Vanelli-Coralli (UniBo)

DISCLAIMER

Co-funded by the European Union

6GSNS

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs, Education and Research EAER
State Secretariat for Education, Research and Innovation SERI

6G-NTN (*6G Non Terrestrial Network*) project has received funding from the **Smart Networks and Services Joint Undertaking (SNS JU)** under the European Union's **Horizon Europe research and innovation programme** under Grant Agreement No 101096479. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them. This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI).

COPYRIGHT NOTICE

Project co-funded by the European Commission in the Horizon Europe Programme		
Nature of the deliverable:	**R**	
Dissemination Level		
PU	*Public, fully open, e.g. web (Deliverables flagged as public will be automatically published in CORDIS project's page)*	✓
SEN	*Sensitive, limited under the conditions of the Grant Agreement*	
Classified R-UE/ EU-R	*EU RESTRICTED under the Commission Decision No2015/ 444*	
Classified C-UE/ EU-C	*EU CONFIDENTIAL under the Commission Decision No2015/ 444*	
Classified S-UE/ EU-S	*EU SECRET under the Commission Decision No2015/ 444*	

* R: Document, report (excluding the periodic and final reports)

DEM: Demonstrator, pilot, prototype, plan designs

DEC: Websites, patents filing, press & media actions, videos, etc.

DATA: Data sets, microdata, etc.

DMP: Data management plan

ETHICS: Deliverables related to ethics issues.

SECURITY: Deliverables related to security issues

OTHER: Software, technical diagram, algorithms, models, etc.

EXECUTIVE SUMMARY

D5.2 is a mid-project deliverable that presents a detailed exploration of the proposed orchestration and management architectures tailored for 6G networks operating across ad hoc constellations. The primary focus revolves around the strategic distribution of diverse functions and Virtual Network Functions (VNFs) within the framework outlined in Work Package 3 (WP3). Particular emphasis is placed on open interfaces and generic frameworks designed to facilitate the seamless integration of advanced Artificial Intelligence (AI) and Machine Learning (ML) algorithms. By dissecting the architecture's intricacies, the document aims to provide insights into the scalability, flexibility, and adaptability required for the dynamic 6G landscape.

In addition to architectural considerations, the deliverable delves into the critical domain of cybersecurity, evaluating potential threats and vulnerabilities inherent in the proposed virtualization framework. Task 5.2 leads a comprehensive threat analysis, extending its scope to encompass the integration of 6G networks with satellite payloads. This collaborative effort is enriched by inputs from Task 5.3, ensuring a multi-faceted understanding of the security landscape. The objective is to identify potential risks and challenges early in the development cycle, facilitating the formulation of robust countermeasures.

Through this deliverable, we strive to offer a holistic perspective on the foundational aspects, integration capabilities, and security implications essential for the successful deployment of 6G networks over ad hoc constellations. By combining technical insights from WP2 with a thorough cybersecurity assessment led by Tasks 5.2 and 5.3, this document contributes significantly to the ongoing progress of the project.

Co-funded by the European Union

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

3GPP	3rd Generation Partnership Project
5GC	5G Core network
AI	Artificial Intelligence
AMF	Access and Mobility Management Function
BS	Base Station
BVLoS	Beyond Visual Line-of-Sight
BWP	Bandwidth Part
C2	Command and Control
C2CSP	C2 Link communication service provider
CoW	Cell on Wheels
CP	Cyclic Prefix
C-SWaP	Cost Size Weight and Power
CT	Core Network and Terminals
E2E	End-to-end
EAB	External Advisory Board
ECC	Electronic Communication Committee
EASA	European Union Aviation Safety Agency
ECC	Electronic Communication Committee
FR	First Responder
GEO	Geostationary Earth Orbit
gNB	Next-generation Nobe-B
GNSS	Global Navigation Satellite Systems
GSO	Geostationary Orbit
HAP	High Altitude Platform
HD	High Definition
HV	Host Vehicle
IoT	Internet of Things

Co-funded by the European Union

ISL	Inter-Satellite Link
INL	Inter-Node Link
LEO	Low Earth Orbit
LIDAR	Light Detection and Ranging
LoS	Line of Sight
M2M	Machine-to-Machine
MC	Multi-Connectivity
MC data	Mission Critical data
MCPTT	Mission Critical Push-to-Talk
MC video	Mission Critical Video
MFCN	Mobile/Fixed Communications Networks
MEO	Medium Earth Orbit
MNO	Mobile Network Operator
NASA	National Aeronautics and Space Administration
NGSO	Non-geostationary orbit
NLoS	Non Line-of-Sight
NR	New Radio
NTN	Non-Terrestrial Network
OOBE	Out-of-Band Emission
PAPR	Peak-to-Average Power Ratio
PHEM	Pre-Hospital Emergency Medicine
PPDR	Public Protection and Disaster Relief
PTT	Push-To-Talk
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
Rel	Release
RF	Radio Frequency
RIC	Radio Intelligent Controller

SAR	Search and Rescue
SD	Standard Definition
SI	Study Item
SNO	Satellite Network Operator
SON	Self-Organizing Networks
TN	Terrestrial Network
TR	Technical Report
TS	Technical Specification
UAM	Urban Air Mobility
UAV	Uncrewed Aerial Vehicle
UAV-C	UAV controller
UC	Use Case
UE	User Equipment
USSP	U-Space Service Provider
vLEO	Very Low Earth Orbit
VLoS	Visual Line-of-Sight
VNF	Virtualized Network Function
VR	Virtual reality
VTOL	Vertical Take-Off and Landing
WI	Work Item
WPAN	Wireless Personal Area Network
WP	Work Package

Co-funded by the European Union

1 INTRODUCTION

In the rapidly evolving landscape of wireless communication technologies, the transition from 5G to 6G non-terrestrial networks represents a pivotal moment in the quest for enhanced connectivity, unprecedented data speeds, and transformative applications. As we stand on the cusp of this new era, characterized by the convergence of advanced technologies and the proliferation of interconnected devices, the need for a robust and adaptable infrastructure becomes increasingly apparent. This document serves as a comprehensive exploration of the key drivers, challenges, and opportunities inherent in the migration towards 6G non-terrestrial networks. By delineating the scope and objectives of the 6G-NTN project, elucidating the underlying principles of dynamic orchestration and autonomous monitoring, and delving into the intricacies of cybersecurity in a rapidly evolving threat landscape, this document seeks to provide a holistic understanding of the technological advancements shaping the future of wireless communication. Through a combination of theoretical analysis, practical insights, and forward-looking perspectives, this document aims to inform and inspire researchers, industry stakeholders, and policymakers alike, catalysing innovation and driving progress towards a more connected, intelligent, and secure digital future.

1.1 SCOPE AND OBJECTIVES

This deliverable is part of WP5, tasks 2 and 3. Its scope and objectives are multifaceted, reflecting the complexity and breadth of the research and development efforts within the 6G-NTN project. At its core, this document aims to provide a comprehensive exploration of the transition from 5G to 6G non-terrestrial networks, elucidating the key innovations and potentials to be explored within this domain. By delving into various aspects such as dynamic orchestration, autonomous monitoring, and cybersecurity, the document seeks to address the evolving challenges and opportunities associated with the next generation of wireless communication technologies. Moreover, it aims to define the structural framework and technical solutions required to realize the vision of 6G non-terrestrial networks, including new virtualized and cloud-native architectures, ML-based traffic prediction techniques, and proactive security mechanisms. Through this comprehensive analysis, the document aims to lay the groundwork for future research endeavours and industry initiatives aimed at advancing the state-of-the-art in wireless communication technologies and driving the development of more resilient, efficient, and secure network infrastructures.

1.2 RELATION TO OTHER WORK PACKAGES IN 6G-NTN

The relation of tasks 2 and 3 of WP5 to the rest of the 6G-NTN project is illustrated in **Error! Reference source not found.**.

Work Package 5, entitled "Enablers for E2E integration in 6G system architecture," plays a crucial role in the 6G-NTN project by ensuring the seamless integration of Non-Terrestrial Networks (NTN) into the broader 6G ecosystem. This WP is connected with all the other work packages, facilitating a cohesive and integrated approach towards the project's objectives. Here's how WP5 is related to other work packages:

Coordination and Oversight: WP1 is responsible for the overall project management, ensuring that all WPs, including WP5, adhere to the project's timeline and deliverables. WP5 is under the supervision of WP1 that ensures that its activities are well-coordinated and that any risks or issues are promptly addressed.

Defining Requirements: WP2 focuses on defining the project use cases and identifying requirements. WP5 relies on the requirements set forth by WP2 to ensure that the integration of NTN in the 6G network meets the specified performance targets and technical specifications. This synergy ensures that the developed solutions in WP5 are aligned with the overall project goals and user needs.

Architectural Alignment: WP3 is tasked with designing the 3D network architecture for 6G NTN, encompassing various flying network nodes. WP5 builds upon this architectural framework to integrate NTN into an end-to-end (E2E) 6G network. The collaboration involves assessing different architectural scenarios and their implications on system integration, thus ensuring that the solutions developed in WP5 are feasible and effective within the proposed architecture.

Technology Integration: WP4 focuses on designing and implementing a unified air-interface and data-driven algorithms for NTN. WP5 works closely with WP4 to ensure that the developed radio access technologies are seamlessly integrated into the overall 6G system. This includes collaborating on dynamic orchestration, autonomous monitoring, and cybersecurity aspects to ensure a robust and secure E2E network.

Dissemination and Exploitation: WP6 is dedicated to maximizing the impact of the project through dissemination, standardization, and community building activities. WP5 supports WP6 by providing the technical advancements and integration solutions needed for demonstrations and proof-of-concept validations. Additionally, WP5 contributes to the standardization efforts and policy recommendations by sharing insights and results from its integration work.

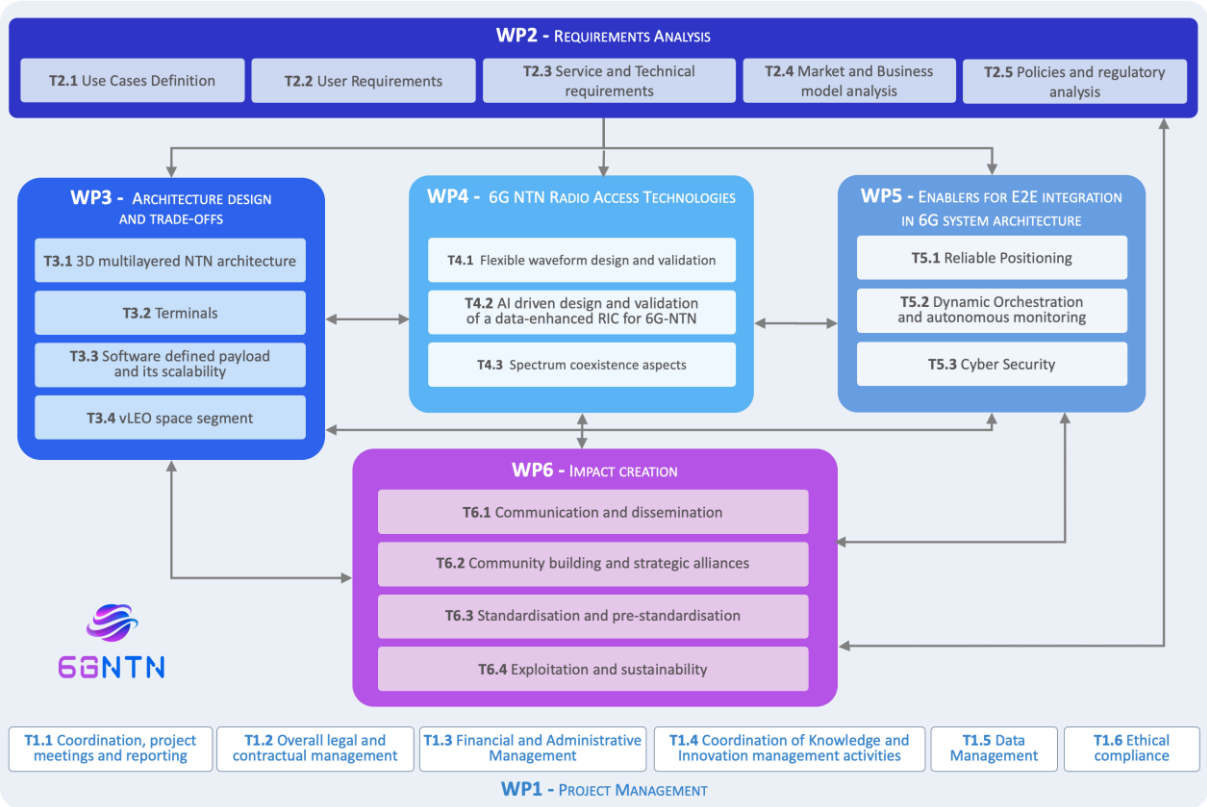

FIGURE 1: 6G-NTN WORK ORGANIZATION

1.3 STRUCTURE OF THE DOCUMENT

This document is structured to provide a comprehensive overview of the ongoing research and development efforts within the scope of the 6G-NTN project. The structure is designed to facilitate easy navigation and understanding of the key objectives, methodologies, and findings presented in each section.

1. **Introduction**: This section introduces the document by outlining its scope and objectives within the context of the 6G-NTN project. It also establishes the relationship between this work package and other components of the project. Additionally, the structure of the document is briefly outlined to guide the reader through the subsequent sections.

2. **From 5G to 6G Non-Terrestrial Networks**: This section provides insights into the objectives and innovation potentials of the 6G-NTN project, highlighting the transition from 5G to 6G networks and the unique challenges and opportunities it presents.

3. **Dynamic Orchestration and Autonomous Monitoring**: This section delves into the new virtualized and cloud-native architecture proposed within the project, focusing on core network integration, monitoring, configuration, management APIs, authentication, security, and ML techniques for traffic and resource prediction.

4. **Cybersecurity**: This section addresses the context and technical problems related to cybersecurity in the context of 6G NTN. It outlines a three-phase security mechanism and presents a technical solution for proactive detection before deployment.

5. **Conclusion**: This section summarizes the key findings and insights presented in the document, highlighting the significance of proactive security measures, dynamic orchestration, and autonomous monitoring in ensuring the success and resilience of 6G NTN.

Each section is structured to provide clear objectives, methodologies, and findings, supported by relevant data, analysis, and references. Together, these sections offer a comprehensive overview of the ongoing efforts within the WP5 in tasks 2 and 3 of the 6G-NTN project and their implications for the coming phases of the project.

Co-funded by the European Union

2 FROM 5G TO 6G NON-TERRESTRIAL NETWORKS

The evolution of 5G into beyond 5G (5G-Advanced) and 6G networks aims at responding to the increasing need of our society for ubiquitous and continuous connectivity services in all areas of our life: from education to finance, from politics to health, from entertainment to environment protection. It is generally understood that the terrestrial network alone cannot provide the flexibility, scalability, adaptability, and coverage required to meet the above requirements, and the integration of the NTN component is a key enabler.

In this section, we describe how the work done in tasks 5.2 and 5.3 of WP5 can help achieve some of the objectives of the project detailed in the Direction of Work.

The dynamic network orchestrator we are designing is a crucial component in achieving some key objectives and exploring the innovation potentials of the 6G-NTN project. This section details how the ICS will respond to these objectives and innovation potentials over the next 18 months.

- **Objective #6:** *AI-enhanced Radio Intelligent Controller (RIC)*

The system can integrate AI-enhanced RIC capabilities to analyse traffic patterns and predict variations at both large and small scales. This predictive capability will enable the our system to proactively allocate resources, adjust network configurations, and optimize performance across the 3D network infrastructure. The AI algorithms will be continuously refined to improve accuracy and responsiveness.

- **Objective #7:** *VNF orchestration*

The dynamic network orchestrator will leverage advanced VNF orchestration techniques to integrate TN and NTN components within the 6G Edge and Core architectures. By utilizing lightweight micro-service orchestrators, the system will dynamically deploy VNFs and edge services on physical nodes. We believe this approach will ensure low-latency, high-throughput, and reliability services across the network.

Regarding the innovation potentials we planned to explore within the project, we provide here a list of the ones we target.

- **Innovation potential #1:** *Performance enhancement*

The dynamic orchestrator will significantly enhance network performance compared to 5G-NTN by employing advanced AI algorithms and dynamic orchestration techniques. These capabilities will optimize latency, data rates, and the number of devices managed by the network. It will ensure that new services requiring high performance are achievable, providing seamless connectivity to any 6G-NTN terminal.

- **Innovation potential #2:** *Ubiquitous connectivity and Resiliency*

The system will play a crucial role in reinforcing network resilience and providing ubiquitous coverage. By dynamically adjusting network configurations and resource allocations in response to real-time traffic variations and network conditions, it will ensure global service continuity and meet diverse Quality of Service (QoS) requirements. This capability will be essential for maintaining network operations during outages and other disruptions, with the ability to reconfigure resources in short time (order of a minute).

- **Innovation potential #5:** *Solutions as-a-Service*

By leveraging virtualized capabilities and cloud computing techniques, our dynamic orchestrator will enable fast (near-instantaneous) deployment of network functions. This approach will support Network as a Service (NaaS) and Infrastructure as a Service (IaaS) models, allowing for flexible and scalable service offerings. The system main goal is to ensure that network resources are efficiently utilized and can be quickly reconfigured to meet changing demands.

- **Innovation potential #6:** *Space Edge Computing*

Because the system we are designing is supposed to include all the components of our core network, it will incorporate Space Edge Computing to enhance latency-sensitive services by bringing computation closer to data sources. This will be particularly beneficial in remote and hard-to-reach locations. The orchestrator will manage and orchestrate edge computing resources located at various non-terrestrial layers, including satellites, High Altitude Platforms (HAPs), and aerial base stations.

- **Innovation potential #7:** *Fast adaptation to traffic variations*

Finally, the orchestrator will provide high dynamicity and re-configurability to absorb traffic variations at both large and small scales. By continuously monitoring traffic patterns and predicting future demands, it will dynamically adjust network configurations and resource allocations. This will ensure that the network can handle fluctuating traffic loads efficiently, maintaining optimal performance and user experience.

The Dynamic network orchestrator is poised to address several key innovation potentials of the 6G-NTN project, enhancing network performance, connectivity, and service flexibility. By integrating advanced AI, dynamic orchestration, and space edge computing capabilities, it will ensure that the 6G-NTN network is robust, resilient, and capable of meeting future demands.

3 DYNAMIC ORCHESTRATION AND AUTONOMOUS MONITORING

In the quest for pioneering advancements within the realm of 6G and Non-Terrestrial Networks (NTN), ability to have a dynamic orchestration and to monitor in an autonomous way the infrastructure emerges as a pivotal domain. This section goal is to clarify how 6G-NTN project seeks to revolutionise the landscape of network management and operational efficiency.

3.1 NEW VIRTUALIZED AND CLOUD NATIVE ARCHITECTURE

The essence of our 6G-NTN architecture lies in its inherent ability to dynamically orchestrate resources, intelligently allocating and reallocating computing, storage, and networking assets in real-time. Through the orchestration layer's prowess, the network can seamlessly adapt to changing demands, efficiently provisioning resources where and when they are needed most.

At the heart of this architecture lies the concept of autonomy. Leveraging cutting-edge technologies such as Artificial Intelligence (AI) and Machine Learning (ML), autonomous monitoring mechanisms continuously observe network performance, identify anomalies, and proactively take corrective actions. This proactive stance not only enhances network reliability and robustness but also minimizes downtime and service disruptions, ensuring a seamless and uninterrupted user experience.

Moreover, the platform we provide in the scope of the project fosters innovation by providing a ground for experimentation and exploration. By abstracting underlying hardware complexities and providing standardized interfaces, it empowers rapid prototyping of new services, and iterate towards novel solutions.

The platform architecture is designed to integrate a 5G/6G core network, orchestrated by an orchestrator, and monitored using a monitoring tool. Regarding the core network, our platform integrates open5GS, free5GC, and OpenAir Interface. The orchestrator that has been chosen for now is Kubernetes, and the elected monitoring tool is Prometheus. We believe this setup provides a flexible and scalable foundation for managing and monitoring the network infrastructure efficiently.

3.1.1 Review of Existing Technologies

In this subsection, we describe the different technologies we have identified to build our 6G-NTN test and integration platform and justify our choices.

3.1.1.1 Kubernetes and Container Orchestration

Kubernetes is the de facto standard for container orchestration in cloud environments. It automates the deployment, scaling, and operations of application containers across clusters of hosts, providing highly resilient and scalable solutions. While Kubernetes offers robust orchestration capabilities, it needs adaptation for the dynamic and variable conditions of 6G-NTNs. Enhancements are required to handle high latency, mobility, and intermittent connectivity. Kubernetes, in its current form, does not inherently account for the high-latency, intermittent connectivity, and varying resource availability that 6G-NTNs face. There is a need for customized scheduling algorithms, enhanced fault tolerance mechanisms, and more robust network management features tailored for 6G-NTNs.

3.1.1.2 Virtualization software

OpenStack is an open-source cloud computing platform that provides IaaS solutions. It allows for the deployment of virtual machines and other instances that handle different tasks for managing a cloud environment. OpenStack's modular architecture can be leveraged for managing virtualized network functions (VNFs) in 6G-NTNs. However, similar to Kubernetes, it requires customization for the unique challenges presented by 6G-NTNs. The modularity of OpenStack is advantageous, but its components need to be adapted to handle the dynamic nature of 6G-NTNs. This includes the development of distributed control planes, enhanced security features, and improved interoperability with edge computing frameworks.

VMware offers a suite of cloud computing and virtualization software and services. VMware's virtualization technology provides a robust and flexible infrastructure for managing virtualized environments. VMware's solutions can support the deployment and management of VNFs in 6G-NTNs. Its advanced management tools and integration capabilities are beneficial for handling complex virtualized environments. While VMware provides a comprehensive set of tools for virtualization, its proprietary nature can be a limitation compared to the open-source flexibility of OpenStack. Additionally, similar to OpenStack, VMware needs customization to handle the specific requirements of 6G-NTNs, such as enhanced mobility management and real-time resource allocation.

OpenStack vs. VMWare

OpenStack, being open-source, offers more flexibility for customization and integration with other open-source tools. This can be crucial for adapting to the unique needs of 6G-NTNs. On the other hand, VMware, while offering robust features, may have limitations in terms of customization due to its proprietary nature. We provide here a comparison between these two virtualization solutions.

- VMware provides advanced integration tools that can seamlessly work with existing enterprise systems.
- OpenStack, although highly integrable, might require more effort to achieve the same level of seamless integration.
- OpenStack generally has lower upfront costs due to its open-source nature. However, the total cost of ownership can vary depending on the implementation complexity and required support.
- VMware, being a commercial product, involves licensing costs but often comes with comprehensive support and professional services.
- OpenStack has a large community of developers and users contributing to its ecosystem, which can be advantageous for troubleshooting and continuous improvement.
- VMware offers dedicated professional support, which can be beneficial for enterprise environments but may come at an additional cost.

3.1.1.3 Edge Computing

Edge computing brings computation and data storage closer to the location where it is needed, improving response times and saving bandwidth. Deploying edge computing nodes in 6G-NTNs can significantly reduce latency and improve service reliability. Edge nodes can manage VNFs locally, reducing dependency on central cloud resources. Existing edge computing solutions need to be enhanced to handle the unique challenges of 6G-NTNs, such as mobility management, synchronization across distributed nodes, and dynamic resource allocation based on changing network conditions.

3.1.1.4 Network Slicing

Network slicing allows for the creation of multiple virtual networks on a shared physical infrastructure. Each slice can be tailored to meet the specific needs of different applications or

services. Network slicing is vital for 6G-NTNs, where different applications might have varying requirements for latency, bandwidth, and reliability. It enables efficient resource utilization and service differentiation. To implement network slicing in 6G-NTNs effectively, there is a need for advanced slicing orchestration frameworks that can dynamically adjust slices based on real-time network conditions, manage inter-slice interference, and ensure end-to-end quality of service.

Note that edge computing and network slicing, while mandatory in a 6G network are not considered in the current scenario for dynamic network orchestration, which focuses on an intelligent deployment and orchestration of VNFs.

3.1.2 6G-NTN dynamic VNFs orchestrator solution

In this section, we describe the solution we suggest for our 6G-NTN dynamic orchestration. This solution is based on a traditional 6G core network to which we add a VNF orchestrator based on machine learning techniques to make configuration and redeployment decisions for various network functions. This orchestrator collects real-time information from the core network, obtains traffic forecasts and hardware resource needs from a component using artificial intelligence techniques to predict future network conditions, and, based on this information, makes adaptive decisions for the core network that it deems necessary and applies them.

This solution is depicted in **Error! Reference source not found.**, and we describe each of the components in the remainder of this section.

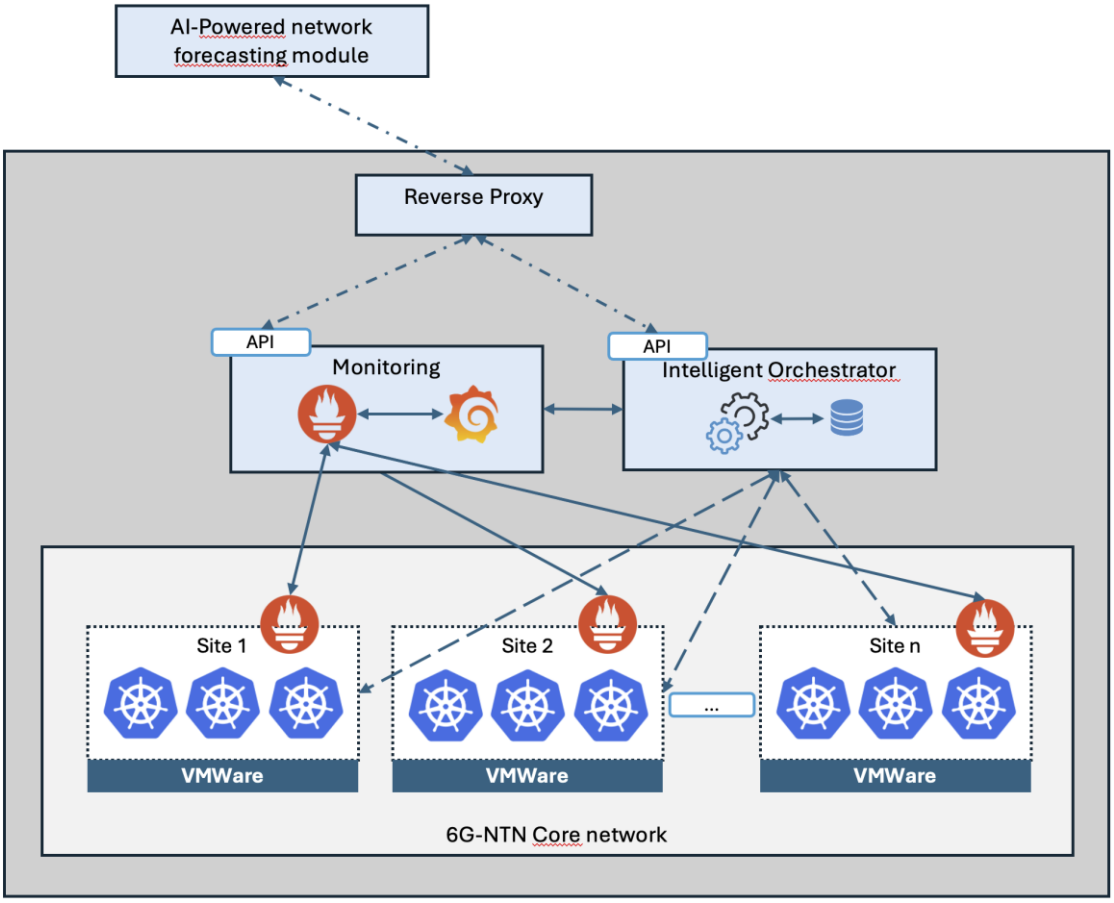

FIGURE 2: HIGH-LEVEL VIEW OF THE SOLUTION ARCHITECTURE DESCRIBING THE DIFFERENT COMPONENTS AND THE INTERACTIONS BETWEEN THEM.

3.1.2.1 Core Network Integration

We aim at deploying a 6G core network based on Kubernetes, thus, the emphasis lies on achieving service flexibility and agility rather than solely focusing on raw throughput performance. The deployment procedure of 6G Network Functions (NFs) is pivotal in this regard, with options ranging from Physical NFs (PNFs), Virtual NFs (VNFs), to the emerging trend of Containerized NFs (CNFs). While VNFs initially gained traction due to their virtualization benefits in terms of efficiency, scalability, and cost, CNFs are now becoming increasingly popular among operators. CNFs offer advantages such as enhanced scalability, operational efficiency, energy savings, and suitability for resource-constrained edge applications.

Containers, as defined by Docker, serve as standardized units of software that encapsulate code and its dependencies, enabling applications to run consistently across various computing environments. Container images, being lightweight and self-sufficient, contain everything required to execute an application seamlessly. Container deployments, spanning multiple hosts, are managed by an orchestrator that automates container lifecycle processes such as creation, deletion, and modification without disrupting services. This aligns closely with the lifecycle management principles of Network Functions Virtualization (NFV). In our deployment, Kubernetes serves as the chosen container orchestrator due to its widespread adoption across industries for managing high-demand services with intricate configurations.

Our deployment strategy revolves around leveraging Helm charts, which are collections of files describing related Kubernetes resources. By utilizing Helm charts, we streamline the deployment process of both the 6G Core Network Functions (NFs) and the monitoring system within the Kubernetes cluster. With a few commands, the entire framework can be deployed, simplifying the setup process significantly. Once the deployments are initiated, establishing connectivity among containers and external services becomes essential.

3.1.2.2 Monitoring

Monitoring plays a crucial role in our project focused on orchestrating a 6G core network using Kubernetes. It involves analysing behavioural data from infrastructure, network events, and user interactions to gather relevant metrics and manage unexpected events effectively. To achieve this, we rely heavily on Prometheus, an open-source monitoring and alerting toolkit that seamlessly integrates with Kubernetes for automatic deployment and service discovery. Prometheus not only collects data from various sources but also provides interfaces to third-party applications like web UI or Grafana. For infrastructure monitoring, we use Kube-Prometheus, an end-to-end Kubernetes cluster monitoring stack that comes pre-configured with default dashboards and alerting rules. This allows us to monitor resource usage metrics such as CPU, memory, and networking at different levels within the Kubernetes cluster.

To visualise the gathered metrics, we leverage Grafana, an open-source analytics and visualization tool included in the Kube-Prometheus stack.

3.1.2.3 Configuration and management API

In today's dynamic telecommunications landscape, the ability to quickly adapt and reconfigure core network infrastructure in response to changing requirements is paramount. To achieve this agility, a custom software solution with a RESTful API can serve as a powerful tool for managing and orchestrating core network resources. This section outlines the design and functionality of a custom REST API tailored for dynamic core network management.

Our custom REST API is designed to provide a flexible and extensible interface for receiving instructions and configuration parameters from external sources. Built on RESTful principles,

the API adheres to standard HTTP methods for interacting with resources. It follows a resource-oriented architecture, with endpoints representing different entities and operations within the core network infrastructure. Description of the endpoints is provided in Section 3.4.

The documentation for our RESTful API, including detailed descriptions of endpoints, request/response formats, and authentication mechanisms, will be provided in the form of a Swagger document. This Swagger document will serve as a comprehensive reference for developers and users, facilitating the implementation and utilization of the API within their applications.

3.1.2.4 Integration and Interoperability

The architecture is designed with a focus on integration and interoperability, enabling seamless communication and collaboration between different components and systems within the platform. By adhering to industry-standard APIs and protocols, the platform ensures compatibility with a wide range of third-party tools and services, fostering a vibrant ecosystem of solutions and innovations.

The custom REST API is designed for seamless integration with existing orchestration platforms, network management systems, and external applications. It supports standard protocols like OpenAPI (Swagger) for documenting the API specification, enabling interoperability and ease of integration with third-party tools and frameworks.

3.1.2.5 Authentication and Security

In order to ensure secure access to the API and manage user authentication and authorization, we will leverage Keycloak, an open-source identity and access management solution. Keycloak provides centralized authentication and authorization capabilities, allowing us to authenticate users via various mechanisms such as password-based login, social logins, X.509 certificates, and more. Additionally, Keycloak enables us to define granular authorization policies to control access to API resources based on user roles and permissions.

One of the key features of Keycloak is its ability to generate access tokens based on OAuth 2.0 and OpenID Connect standards, such as JSON Web Tokens (JWTs). These tokens will be issued to authenticated users and can be used by clients to access protected API resources. By integrating Keycloak with our API, we can ensure that only authorized users with valid tokens are able to access sensitive data and perform authorized actions.

Furthermore, Keycloak offers extensive user management capabilities, including user registration, user profile management, password reset, and more. It provides a comprehensive administration console for managing users, roles, groups, and client applications. Additionally, Keycloak supports user federation, allowing us to integrate with external identity providers such as LDAP, Active Directory, and social identity providers.

With Keycloak's robust authentication and authorization features, we can enforce secure access control policies, protect sensitive data, and provide a seamless user experience for accessing our API. The documentation for integrating Keycloak with our API will be provided in Swagger format, making it easy for developers to understand and implement the necessary authentication and authorization mechanisms.

3.2 6G-NTN INTELLIGENT ORCHESTRATOR

The dynamic intelligent orchestrator we design in the context of 6G-NTNs must be able to dynamically manage and deploy Virtual Network Functions (VNFs) based on real-time and predictive data. This orchestration involves leveraging traffic load predictions, current core

network status, and intelligent decision-making algorithms to ensure optimal performance, resource utilization, and service reliability.

3.2.1 Orchestrators key functions

The key functions of the orchestrator are as follows. Note that some of the key sub-functions will not be implemented but we believe it is worth mentioning them and design the system with these functionalities in mind.

Traffic Load Prediction Integration

- Prediction data reception: The orchestrator receives data from an external module, namely the AI-Powered network forecasting module, that predicts traffic loads based on historical data, user behaviour, and other relevant metrics. This predictive module uses machine learning algorithms to forecast traffic patterns and peak usage times.
- Response: The orchestrator uses these predictions to proactively allocate resources and deploy VNFs to areas expecting high traffic, thereby preventing congestion and ensuring seamless service delivery.

Real-Time Network Monitoring

- Core network status: A network and resources monitoring module continuously monitors the core network status, collecting data on current load, latency, throughput, and resource availability.
- Anomaly detection: The orchestrator can implement anomaly detection algorithms to identify and respond to unexpected changes or failures in the network in real-time and adapt network reconfiguration to mitigate these anomalies.
- Resource adjustment: The orchestrator dynamically adjusts the deployment and scaling of VNFs based on real-time network conditions, ensuring optimal performance and resource utilization.

Intelligent VNF Deployment

- Adaptive scaling: The system automatically scales VNFs up or down based on current and predicted traffic loads to maintain service quality and efficiency.
- Optimal placement: It utilizes custom algorithms to determine the best locations for deploying VNFs, considering factors such as latency, resource availability, and predicted demand.
- Resource optimization: The orchestrator ensures efficient use of computational and network resources by balancing the load across available infrastructure depending on the current and predicted traffic and available resources.

3.2.2 Implementation process

In the following, we give details about how the orchestrator is designed.

Integration with Traffic Prediction Module

The orchestrator ingests data from the AI-Powered network forecasting module, which provides forecasts at regular intervals (e.g., every 5 minutes, hourly) or on-demand. Using this data, the orchestrator performs predictive analysis to anticipate network demand and potential bottlenecks.

Monitoring Core Network Status

As part of our system, we implement tools to continuously monitor core network metrics such as CPU usage, memory consumption, bandwidth utilization, and latency to providing a real-time dashboard for network operators to visualize the current status and predicted trends and use this information, in conjunction with the traffic prediction to optimise network utilisation.

Decision-Making Algorithms

As for the AI-Powered network forecasting module, we can develop and traine machine learning models to predict the optimal deployment and scaling of VNFs based on historical data and real-time metrics. However, we can also use heuristic algorithms in the meanwhile and for quick decision-making in situations where machine learning models might be too slow or complex.

Dynamic Resource Allocation

Our orchestrator can proactively scale the resources in anticipation of predicted traffic surges, ensuring that sufficient capacity is available before demand peaks but can also make real-time adjustments to resource allocation in response to sudden changes in traffic or network conditions. One main feature is to implement load balancing techniques to distribute traffic evenly across the network, preventing any single node from becoming a bottleneck.

VNF Placement Optimization

The VNFs will be geographically distributed in locations that minimize latency and maximize service reliability, considering the geographical distribution of users and network infrastructure, as well as geographical resources demand and traffic prediction. Of course, we will ensure that VNFs are deployed on nodes with sufficient resources, avoiding overloading nodes. This VNFs deployment should be done taking into account the possible interferences between the VNFs and ensuring that they have no impact on each other's performances.

Feedback Loop

Finally, the system will continuously monitor the performance of deployed VNFs and the overall network, feeding performance data back into the machine learning models to improve future predictions and decision-making. That will allow to regularly update and refine the algorithms and models used for orchestration based on new data and insights

3.3 ML TECHNIQUES FOR TRAFFIC AND RESOURCE PREDICTION

3.3.1 Architecture

To address the dynamic demands of microservices orchestration, a novel AI-powered network forecasting platform is proposed. This platform leverages advanced AI algorithms to predict the utilization and demand for virtualized network resources, enabling proactive resource allocation and optimization.

Below is an illustrative diagram outlining the key components and interactions of the **AI-powered network forecasting platform.**

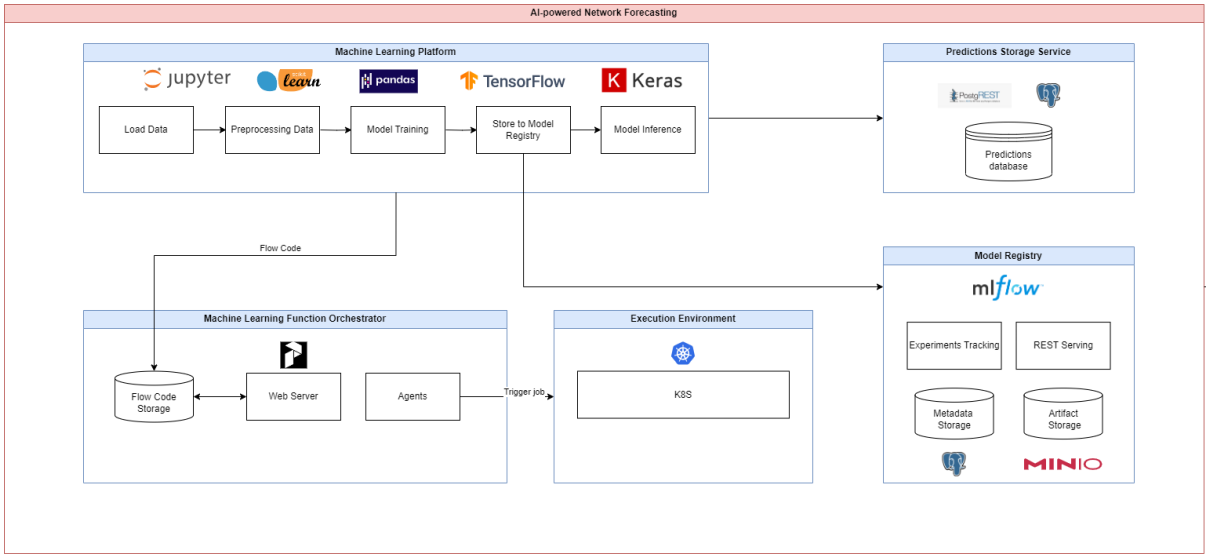

FIGURE 3: ARCHITECTURE OF AI-POWERED NETWORK FORECASTING

The **Machine Learning Platform** serves as the backbone for executing standard machine learning procedures across all utilized models, facilitating the automatic training and retraining of the models. The execution of Machine Learning (ML) and Deep Learning (DL) experiments is triggered by orchestrated workflows that leverage cloud-native infrastructure.

Initially, the platform retrieves monitored data, primarily in the form of time-series, from the monitoring component using Prometheus APIs. Subsequently, the data undergoes pre-processing, encompassing tasks such as cleaning, interpolation to fill missing values, sub-sampling to align with forecasting requisites, and normalization as needed. Once prepared, the model undergoes training through the standard split of data into training and test sets. The resulting model weights are then stored in a Model Registry, which assumes a fundamental role in experiment tracking. Leveraging a PostGres database for metadata storage and MINIO for artifact storage, the Model Registry ensures record-keeping and accessibility of trained model parameters. Once the model is trained and saved, it's prepared to predict future points across the specified time-series.

Notably, the frequency and scope of inference, including the number of predicted points and their temporal lag, are adjusted to suit the requirements and needs of the orchestrator. Furthermore, these variables consider the model performance, encompassing both prediction accuracy and energy consumption.

For implementing machine learning and deep learning functionalities, the platform utilizes popular libraries and frameworks such as scikit-learn, TensorFlow, and pandas. The code is developed in **Python** using Jupyter notebooks.

Moving forward, the predictions generated by the Machine Learning Platform are stored in the Prediction Storage Service, which can be accessed through PostgREST APIs.

Furthermore, the AI-powered network forecasting component leverages Prefect as its **Machine Learning Function Orchestrator (MLFO)**. Prefect is a workflow orchestration tool designed for building and managing data pipelines. Within this component, Prefect is used to convert Python code into Flow Code, enabling the execution of machine learning tasks as data pipelines that can run concurrently when possible. Acting as the MLFO, Prefect is responsible for initiating the creation of **Kubernetes (K8S)** pods for executing the machine learning workflow and scheduling the training and inference of ML models.

3.3.2 Forecasting strategies

In 6G-NTN, network functions are virtualized and treated as Kubernetes pods. Therefore, **CPU usage** and **memory usage** for each pod that implements a network function can be used as target metrics to predict the overall resource usage of the network. Forecasting these metrics can assist the orchestrator in better allocating the appropriate amount of logical resources on the shared physical resources. To satisfy this objective, various types of forecasting methods are being explored and evaluated by the AI-powered network forecasting component. The aim is to identify the optimal approach for assisting the orchestrator effectively, with a focus on model accuracy and energy efficiency.

Initially, **different forecasting frequencies** are being examined, considering the balance between prediction reliability and the quality of information provided. For instance, when handling missing values in a dataset, down sampling to a higher frequency (such as 1-minute frequency) yields highly detailed information. However, a higher frequency tends to maintain more missing values, resulting in a lower quality of training data and less accurate predictions. Conversely, utilizing a lower frequency (such as 1-hour frequency) reduces the number of missing values, enhancing prediction reliability. However, it also entails less granular information in each prediction.

Furthermore, the development encompasses both **univariate and multivariate forecasting** techniques. Univariate forecasting involves predicting a target metric, such as the CPU usage of a Kubernetes (K8S) pod, using only historical values of that metric. On the contrary, multivariate forecasting entails predicting the target metric by leveraging past values of the target metric along with historical and current values of additional features. This assumes a certain level of dependency between the target metric and the auxiliary features used. Multivariate forecasting typically yields superior performance due to its utilization of a broader dataset for prediction. However, it requires larger and slower machine learning models, necessitating careful configuration and usage to avoid resource inefficiencies and overlong training times.

Finally, the AI-powered network forecasting component implements both **single-step and multi-step forecasting** methods. Single-step forecasting involves predicting only the next element in the time series, providing a short-term forecast. On the other hand, multi-step forecasting predicts multiple values over a specified period, offering a broader temporal outlook. Notably, when employing multi-step forecasting, short-term predictions typically exhibit higher accuracy, whereas the accuracy of long-term forecasts may be inadequate. Consequently, multi-step forecasting may not provide additional useful information compared to single-step forecasting, yet it typically requires more resources for implementation, training, and execution.

3.3.3 Implementation

The dataset utilized in this initial phase is the 5G Core network's AMF dataset[1], selected for its similarity to the data that will be collected by Prometheus from the platform in the 6G-NTN context. This dataset comprises non-regular simultaneous registration requests that are sent

[1] M. Mekki, N. Toumi and A. Ksentini, "Microservices Configurations and the Impact on the Performance in Cloud Native Environments," 2022 IEEE 47th Conference on Local Computer Networks (LCN), Edmonton, AB, Canada, 2022, pp. 239-244, doi: 10.1109/LCN53696.2022.9843385.

to an instance of the AMF in a period of approximately 17 days, containing circa 27,000 rows and 15 columns, inclusive of timestamps.

The **CPU usage** has been selected as the target feature for forecasting. However, it is worth noting that this work can be extended to include other metrics, such as RAM usage, CPU limit, and RAM limit. Various data analyses have been conducted, and machine learning models have been trained and evaluated using different forecasting strategies to predict the maximum CPU usage value that may be reached within a specific future time frame.

While only the most promising models are detailed in this document, additional models (Decision Trees, Random Forests, Convolutional Neural Networks) have been trained to establish performance baselines. The noteworthy models for time-series forecasting include:

- **ARIMA** (Auto Regressive Integrated Moving Average): is a powerful statistical model for time series forecasting. It utilizes linear relationships with lagged observations and past errors to make accurate predictions. Unlike traditional machine learning algorithms, ARIMA does not require a separate training phase or the splitting of datasets into training and test sets. This makes it a convenient and efficient tool for forecasting without the need for extensive pre-processing or intensive model training.
- **LSTM** (Long Short-Term Memory) is a specialized type of Recurrent Neural Network (RNN) designed to overcome the limitations of traditional RNNs in capturing long-range dependencies in sequential data. LSTM are well-suited for time series forecasting, due to their capability to effectively capture both short-term and long-term dependencies.
- **Prophet** is an open-source forecasting tool that leverages an additive regression model with four main components: a trend change detector, yearly and weekly seasonal components, and a holiday list component. Due to its structure, Prophet performs better when time series exhibit strong seasonality. It excels in detecting trend changes and handling missing data, and it can also manage outliers effectively. Prophet is designed to be scalable and handle large amounts of data.

As a first step, univariate, single step forecasting of CPU usage is performed using both ARIMA model and a LSTM model. Data undergoes common pre-processing techniques, including data cleaning, interpolation, and normalization. As stated in Section 3, a down sampling frequency of 1 hour has been chosen to balance the trade-off between prediction information and reliability. The data is then split into training and test sets for subsequent evaluation. Following these steps, a univariate single-step ARIMA model and a univariate single-step LSTM model are trained and evaluated.

3.3.4 Results

For the evaluation of ARIMA and LSTM models, the following evaluation metrics are used:

- **MAPE** (Mean Absolute Percentage Error): it measures the average absolute percentage difference between actual and predicted values. It is expressed as a percentage.
- **MAE** (Mean Absolute Error): it measures the average absolute difference between actual and predicted values. It provides indication of error magnitude.
- **RMSE** (Root Mean Square Error): it measures square root of average of squared differences between actual and predicted values. It penalizes larger errors more heavily and it is expressed in the same units as the target variable.

These metrics have been chosen to represent the deviation of the predicted values with respect to the actual values.

The results of both models are shown in the images below. The orange line depicts the actual CPU usage extracted from the dataset, while the green line represents the predicted values. The evaluation metrics are displayed above each image.

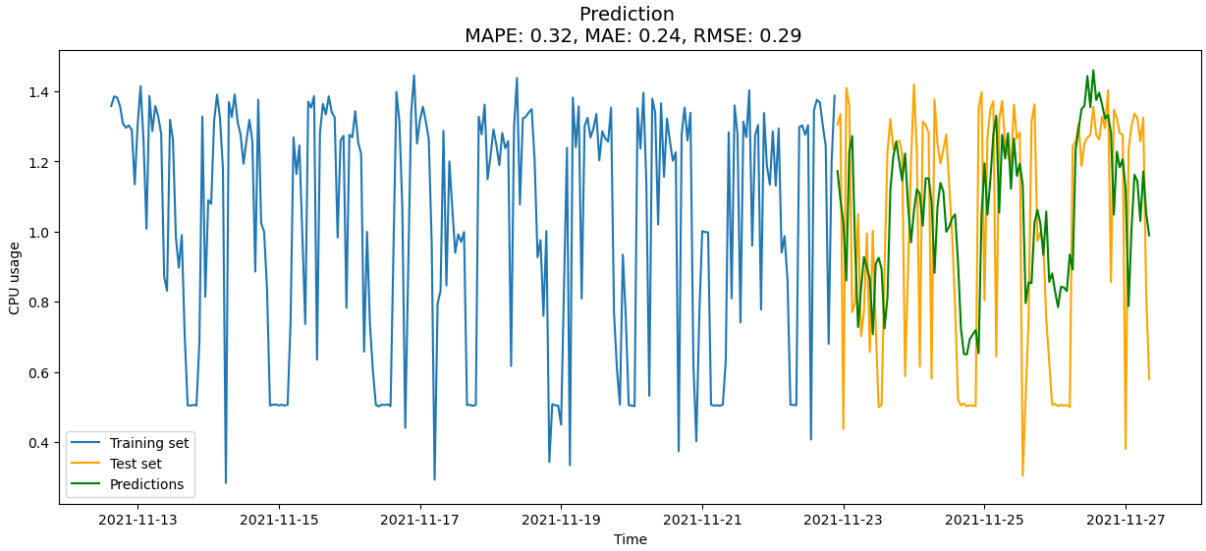

FIGURE 4: RESULTS OF ARIMA MODEL

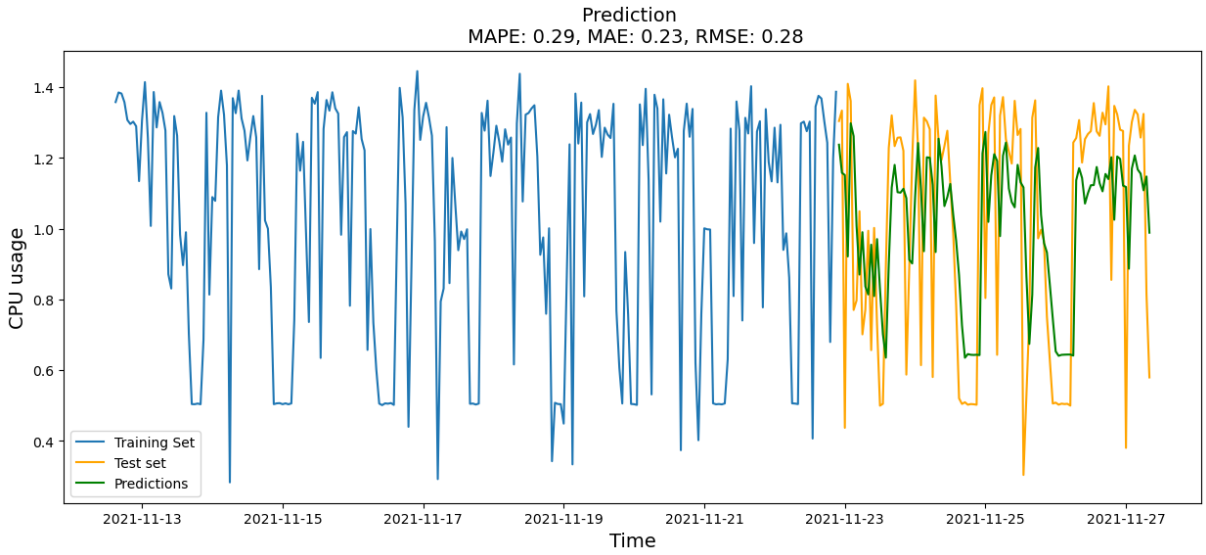

FIGURE 5: RESULTS OF LSTM MODEL

The results demonstrate that LSTM achieves better performance among the models tested, leading to reduced errors in terms of both absolute and percentage values, as it can effectively capture both short-term and long-term dependencies within the time series.

The next phase of development for this component involves implementing univariate multi-step forecasting to assess whether model performance deteriorates with longer prediction horizons. Additionally, multivariate forecasting will be introduced. The expected outcome is a slight reduction in prediction errors due to the utilization of multiple features for predicting the target. Furthermore, performance metrics such as training and inference time will be analyzed to consider energy consumption aspects.

3.4 MONITORING, CONFIGURATION, MANAGEMENT AND ORCHESTRATION APIS

The Platform Management API provides a RESTful interface for interacting with the Kubernetes-based platform orchestration layer. This document outlines the available endpoints, request/response formats, and usage guidelines for integrating with the platform management system.

3.4.1 Base URL

All API endpoints are relative to the base URL of the platform management system:

```
https://platform.6g-ntn.eu/api/v1
```

3.4.2 Authentication

Clients can obtain an access token by authenticating with the platform's authentication service. The access token should be included in the request headers as follows:

Authorization: Bearer <access_token>

3.4.3 Endpoints

3.4.3.1 Get Platform Status

GET /status

Returns the current status of the platform, including information about running services, resource utilization, and overall health.

Request

```
GET /api/v1/status
```

Response

```
{
  "status": "ok",
  "services": [
    {
      "name": "core-network",
      "status": "running",
      "version": "v1.0.0"
    },
    {
      "name": "monitoring",
      "status": "running",
      "version": "v2.1.0"
    }
  ],
  "resource_utilization": {
    "cpu_usage": "50%",
    "memory_usage": "75%"
  }
}
```

3.4.3.2 Scale Service

POST /services/{service_name}/scale

Scales the specified service to the desired number of replicas.

Request

```
1. POST /api/v1/services/core-network/scale
```

With the following JSON

```
1. {
2.   "replicas": 3
3. }
```

Response

```
1. {
2.   "status": "success",
3.   "message": "Service scaled successfully"
4. }
```

3.4.3.3 Deploy New Service

POST /services/deploy

Deploys a new service to the platform.

Request

```
1. POST /api/v1/services/deploy
```

With the following JSON

```
1. {
2.   "name": "new-service",
3.   "image": "registry.example.com/new-service:latest",
4.   "replicas": 1
5. }
```

Response

```
1. {
2.   "status": "success",
3.   "message": "Service deployed successfully"
4. }
```

3.4.3.4 Error handling

In case of errors, the API will return an appropriate HTTP status code along with a JSON error response containing details about the error.

3.5 ACTIVITY PLAN FOR SECOND PERIOD OF 6G-NTN

The first project period has been used to design the whole system that has been described in this document. In this section, we give insight on our plans regarding the tasks to be performed during the second period of the project that will lead us to provide this intelligent AI-Powered network orchestration system for 6G-NTN.

The activities plan is divided in the following 4 phases:

Advanced Algorithm Development

The objective of this phase is to enhance the decision-making capabilities of the orchestrator. We will continue developing and training advanced machine learning models for predictive analysis, we will implement heuristic algorithms for quick decision-making, and we will integrate anomaly detection capabilities for real-time network monitoring.

Testing and Optimization

The goal of this phase is to test the intelligent orchestration system and use the tests results to optimize and correct the system.

During this phase, we will conduct extensive testing mainly in simulated and emulated environments, and if time and conditions allow, in real-world situations. The output of these tests and the feedback should help optimize algorithms. As mentioned before, we will implement a feedback loop for continuous improvement of models and algorithms.

Deployment and Evaluation

If time and resources allow, we will deploy the intelligent orchestrator in a live 6G-NTN environment and evaluate its performances. This could be done in a 6G-NTN pilot setup, where we will monitor performances and gather feedback from users and stakeholders.

Report on tests and dissemination

The tests, deployment and evaluation will be finalised by the preparation of a comprehensive evaluation report and a plan for further features and improvements. This report will be at least provided in D5.3 which is the final report on orchestration solutions and cyber security framework for 6G-NTN.

4 CYBER SECURITY

Our security work builds upon the foundation laid out in literature work regarding securing Kubernetes deployments[2], which explores the pervasive use of containers in contemporary deployment landscapes. We acknowledge the shift towards cloud-native microservice architectures, and the challenges associated with deploying and managing containers, particularly in complex environments orchestrated by platforms like Kubernetes. The outlined security concerns, including MITM attacks and configuration vulnerabilities, underscore the importance of robust security measures in container deployments. We recognize the need for methodologies to evaluate and mitigate these risks effectively. Therefore, our approach will draw inspiration from the proposed methodology in the summarized paragraph, focusing on extracting and analysing topological graphs, assessing risk factors, and identifying potential attack paths. By leveraging this established framework, we aim to contribute to the advancement of secure container deployments, emphasizing the importance of comprehensive security assessments and risk mitigation strategies. Through experimental evaluation and practical implementation, we seek to address the inherent security challenges in containerized environments and promote best practices for securing Kubernetes deployments.

4.1 CONTEXT AND TECHNICAL PROBLEM

Cloud computing has revolutionized the deployment of complex applications, particularly with the adoption of microservices architecture. Microservices offer agility, scalability, and ease of deployment, making them the de facto standard for modern software development. However, alongside these benefits come significant security challenges, especially when deploying microservices at scale on cloud infrastructure.

At the heart of managing and orchestrating microservices in cloud environments lies Kubernetes, an open-source container orchestration engine. Kubernetes provides a robust framework for automating the deployment, scaling, and management of containerized applications across clusters of servers. Its flexibility and scalability make it a popular choice for modern cloud-native applications. However, this flexibility introduces complexities and security risks that must be addressed.

One of the primary security risks induced by the virtualization inherent in cloud computing is the increased attack surface. With multiple containers running on shared hardware, there's a heightened risk of lateral movement between containers if one is compromised. Additionally, the dynamic nature of containerized environments can make it challenging to maintain consistent security configurations across all components.

Compared to traditional monolithic deployments, securing microservices architectures requires a paradigm shift in security mindset. In traditional deployments, security measures often revolved around perimeter defenses and securing the underlying infrastructure. However, in

[2] A. Blaise and F. Rebecchi, "Stay at the Helm: secure Kubernetes deployments via graph generation and attack reconstruction," 2022 IEEE 15th International Conference on Cloud Computing (CLOUD), Barcelona, Spain, 2022, pp. 59-69, doi: 10.1109/CLOUD55607.2022.00022. keywords: {Industries;Cloud computing;Microservice architectures;Feature extraction;Market research;Explosions;Security;Microservices;cloud computing;containerisation;orchestration;Kubernetes;Helm Charts}

microservices environments, the focus shifts towards securing each individual component and communication between them.

Furthermore, security configurations in Kubernetes environments can be complex and difficult to implement correctly. Kubernetes offers a wide range of security features, such as network policies, pod security policies, and role-based access control (RBAC). While these features provide granular control over security, configuring them requires a deep understanding of Kubernetes architecture and security best practices.

One of the challenges faced by organizations is the lack of visibility into the security posture of their Kubernetes deployments. Many teams struggle to ascertain what security measures are actually implemented and whether they align with industry best practices and compliance requirements. This lack of visibility can leave organizations vulnerable to security breaches and compliance violations.

In summary, while cloud-native architectures and Kubernetes offer unparalleled flexibility and scalability, they also introduce significant security challenges. Addressing these challenges requires a proactive approach to security, a thorough understanding of Kubernetes security features, and robust implementation of security best practices. Additionally, improving visibility into security configurations and compliance status is crucial for effectively managing security risks in cloud computing infrastructures.

4.2 KUBERNETES APPLICATION DEPLOYMENT

Kubernetes has emerged as the leading platform for deploying and managing containerized applications in modern cloud environments. Its robust features and automation capabilities make it the go-to choice for organizations looking to deploy applications at scale. Kubernetes abstracts away the underlying infrastructure complexities, providing a unified platform for orchestrating containerized workloads across clusters of servers. One of the key advantages of Kubernetes is its declarative approach to application deployment. Instead of manually configuring each component, developers define the desired state of their applications using YAML or JSON manifests. These manifests specify details such as container images, resource requirements, networking policies, and service dependencies. Kubernetes then ensures that the actual state of the system matches the desired state, automatically provisioning and scaling resources as needed.

Kubernetes offers a wide range of deployment options to suit different use cases and workload requirements. The most common deployment strategy is the Deployment resource, which manages the lifecycle of replicated Pods. Deployments enable rolling updates, rollback capabilities, and self-healing mechanisms, ensuring high availability and resilience for applications. In addition to Deployments, Kubernetes supports other deployment strategies such as StatefulSets for stateful applications, DaemonSets for running a single Pod on each node, and Jobs for batch processing workloads. Each deployment strategy provides specific capabilities tailored to different types of applications and use cases. Moreover, Kubernetes provides a rich ecosystem of tools and integrations to streamline the application deployment process. Helm, for example, is a package manager for Kubernetes that simplifies the installation and management of applications through reusable packages called charts. Operators extend Kubernetes functionality by automating operational tasks such as application lifecycle management, scaling, and monitoring. However, deploying applications in Kubernetes is not without its challenges. Managing complex configurations, ensuring security best practices, and optimizing resource utilization require careful planning and expertise. Additionally, monitoring and troubleshooting distributed applications in Kubernetes environments can be challenging due to their dynamic nature and scale.

4.3 3 PHASES SECURITY MECHANISM

Securing Kubernetes deployments is paramount to ensuring the integrity, confidentiality, and availability of containerized applications running in cloud environments. This task is typically approached in three phases: proactive analysis of Kubernetes configuration files, reactive detection during runtime, and mitigation by fixing the detected threats and anomalies. In this discussion, we will focus on the first phase – proactive analysis of Kubernetes configuration files – acknowledging that the subsequent phases will be addressed in later stages.

4.3.1 PROACTIVE Detection before deployment

In the proactive analysis phase, the goal is to identify security risks and vulnerabilities in Kubernetes configuration files before deploying applications. Kubernetes configuration files, typically written in YAML or JSON format, define the desired state of the cluster, including pods, services, deployments, and other resources. These files contain critical information about the deployment, such as container images, resource limits, network policies, and access controls. One of the key challenges in securing Kubernetes deployments is the complexity of these configuration files. As deployments grow in size and complexity, maintaining consistent security configurations becomes increasingly challenging. Misconfigurations or oversight in these files can expose the cluster to various security risks, including unauthorized access, privilege escalation, and data breaches.

To address these challenges, proactive analysis tools are used to perform automated checks and assessments of Kubernetes configuration files. These tools scan the configuration files for common security misconfigurations, adherence to best practices, and compliance with security policies. They identify potential risks such as overly permissive access controls, exposed sensitive information, insecure network configurations, and deprecated API versions. Furthermore, proactive analysis tools provide recommendations and remediation steps to address the identified security issues. This empowers administrators and developers to implement security best practices and harden the Kubernetes deployment before exposing it to production environments. By identifying and fixing security vulnerabilities early in the development lifecycle, organizations can minimize the risk of security incidents and ensure a secure foundation for their Kubernetes deployments.

In contrast to existing approaches, the methodology proposed in the previously cited paper surpasses basic compliance checks by generating comprehensive, interconnected graphs of microservice deployments. These graphs enhance the understanding of deployment configurations and highlight significant risks, including identifying the most critical attack paths. Unlike alternative solutions lacking prioritization, this methodology aids decision-making by prioritizing actions. Moreover, it integrates the Kubernetes threat matrix, effectively modelling deployment security and pinpointing potential vulnerabilities through accurate mapping to real-world attacks.

4.3.2 Real-time Reaction and Mitigation

In a dynamic Kubernetes environment where containers are constantly being deployed, scaled, and updated, traditional security approaches that rely solely on static configuration checks are insufficient. To address the evolving threat landscape, organizations must implement real-time monitoring and response mechanisms that can detect and mitigate security incidents in a timely manner. One approach to real-time security monitoring is the use of Kubernetes-native tools such as Prometheus and Grafana. These tools provide visibility into the health and performance of Kubernetes clusters, allowing administrators to monitor resource usage, network traffic, and application metrics in real time. By setting up alerts and thresholds, administrators can proactively detect abnormal behaviour indicative of security threats, such as unauthorized access attempts, excessive resource consumption, or unusual network activity.

In addition to monitoring for known security threats, we can leverage anomaly detection techniques to identify and mitigate previously unseen or emerging security risks. Machine learning algorithms can analyse patterns in application behaviour and network traffic to detect deviations from normal activity, flagging potential security incidents for further investigation. By continuously learning and adapting to new threats, anomaly detection systems can help organizations stay ahead of evolving security threats and respond quickly to potential security incidents.

4.3.3 Threat and anomaly Mitigation

Following real-time anomaly detection, the next critical step is effective mitigation of detected threats and anomalies. In addition to monitoring for known security threats, we can leverage anomaly detection techniques to identify and mitigate previously unseen or emerging security risks. Machine learning algorithms can analyse patterns in application behaviour and network traffic to detect deviations from normal activity, flagging potential security incidents for further investigation. By continuously learning and adapting to new threats, anomaly detection systems can help organizations stay ahead of evolving security threats and respond quickly to potential security incidents. Implementing automated response mechanisms, such as dynamic policy enforcement or container isolation, can further enhance the effectiveness of anomaly mitigation efforts, ensuring that security incidents are addressed promptly and comprehensively. Through a combination of real-time monitoring and proactive anomaly detection, organizations can strengthen the security posture of their Kubernetes deployments and minimize the risk of potential security breaches or disruptions.

4.4 TECHNICAL SOLUTION

To address the proactive configuration analysis phase, we have developed a Python-based tool that performs analysis of Kubernetes (k8s) configuration files. This tool is accessible via a web interface, allowing users to easily upload their k8s configuration files for analysis. Upon uploading the configuration file, the web service generates and returns an operation ID, which users can use to later access the analysis results when they are finalized.

The analysis performed by the tool includes checks for common security best practices, adherence to Kubernetes security policies, and potential misconfigurations that could lead to security breaches or vulnerabilities. This proactive analysis helps identify and address security issues before they can be exploited by malicious actors.

As of now, our efforts have been focused on developing and implementing this proactive configuration analysis tool. If resources and efforts allow in the future, we plan to work on the next phases of the project, which include reactive detection during runtime and mitigation by fixing the detected threats and anomalies. However, our current focus remains on the proactive analysis phase, which serves as a critical component in enhancing the security of Kubernetes deployments.

In the following, we provide the JSON code of the Swagger documentation of this webtool.

```
{
    "openapi": "3.1.0",
    "info": {
        "title": "FastAPI",
        "version": "0.1.0"
    },
    "paths": {
        "/upload/": {
            "post": {
                "summary": "Upload Files",
                "operationId": "upload_files_upload__post",
```

```
                "requestBody": {
                    "content": {
                        "multipart/form-data": {
                            "schema": {
                               "$ref": "#/components/schemas/Body_upload_files_upload__post"
                            }
                        }
                    },
                    "required": true
                },
                "responses": {
                    "200": {
                        "description": "Successful Response",
                        "content": {
                            "application/json": {
                                "schema": {}
                            }
                        }
                    },
                    "422": {
                        "description": "Validation Error",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "$ref": "#/components/schemas/HTTPValidationError"
                                }
                            }
                        }
                    }
                }
            }
        },
        "/status/{analysis_id}": {
            "get": {
                "summary": "Check Analysis Status",
                "operationId": "check_analysis_status_status__analysis_id__get",
                "parameters": [{
                    "required": true,
                    "schema": {
                        "type": "string",
                        "title": "Analysis Id"
                    },
                    "name": "analysis_id",
                    "in": "path"
                }],
                "responses": {
                    "200": {
                        "description": "Successful Response",
                        "content": {
                            "application/json": {
                                "schema": {}
                            }
                        }
                    },
                    "422": {
                        "description": "Validation Error",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "$ref": "#/components/schemas/HTTPValidationError"
                                }
                            }
                        }
                    }
                }
            }
        },
        "/results/{analysis_id}": {
            "get": {
```

```
                "summary": "View Analysis Results",
                "operationId": "view_analysis_results_results__analysis_id__get",
                "parameters": [{
                    "required": true,
                    "schema": {
                        "type": "string",
                        "title": "Analysis Id"
                    },
                    "name": "analysis_id",
                    "in": "path"
                }],
                "responses": {
                    "200": {
                        "description": "Successful Response",
                        "content": {
                            "text/html": {
                                "schema": {
                                    "type": "string"
                                }
                            }
                        }
                    },
                    "422": {
                        "description": "Validation Error",
                        "content": {
                            "application/json": {
                                "schema": {
                                    "$ref": "#/components/schemas/HTTPValidationError"
                                }
                            }
                        }
                    }
                }
            }
        },
        "/": {
            "get": {
                "summary": "Get Upload Form",
                "operationId": "get_upload_form__get",
                "responses": {
                    "200": {
                        "description": "Successful Response",
                        "content": {
                            "text/html": {
                                "schema": {
                                    "type": "string"
                                }
                            }
                        }
                    }
                }
            }
        }
    },
    "components": {
        "schemas": {
            "Body_upload_files_upload__post": {
                "properties": {
                    "files": {
                        "items": {
                            "type": "string",
                            "format": "binary"
                        },
                        "type": "array",
                        "title": "Files"
                    }
                },
                "type": "object",
                "required": ["files"],
```

```
                "title": "Body_upload_files_upload__post"
            },
            "HTTPValidationError": {
                "properties": {
                    "detail": {
                        "items": {
                            "$ref": "#/components/schemas/ValidationError"
                        },
                        "type": "array",
                        "title": "Detail"
                    }
                },
                "type": "object",
                "title": "HTTPValidationError"
            },
            "ValidationError": {
                "properties": {
                    "loc": {
                        "items": {
                            "anyOf": [{
                                "type": "string"
                            }, {
                                "type": "integer"
                            }]
                        },
                        "type": "array",
                        "title": "Location"
                    },
                    "msg": {
                        "type": "string",
                        "title": "Message"
                    },
                    "type": {
                        "type": "string",
                        "title": "Error Type"
                    }
                },
                "type": "object",
                "required": ["loc", "msg", "type"],
                "title": "ValidationError"
            }
        }
    }
}
```


4.5 SECURING 6G NON-TERRESTRIAL NETWORKS: CHALLENGES AND SOLUTIONS

The discussions and methodologies outlined previously have primarily focused on addressing security challenges within the context of 5G terrestrial networks. However, as we transition towards the era of 6G non-terrestrial networks (NTN), it becomes imperative to adapt and tailor these approaches to suit the unique characteristics and requirements of the next-generation network paradigm. While many principles of security remain consistent across different network types, the introduction of novel technologies, such as satellite communication and airborne platforms, introduces new challenges and considerations that must be accounted for. Therefore, this work serves as a foundational framework upon which further research and adaptation can be built to effectively secure 6G NTN deployments.

Risk Assessment and Threat Modelling: We need to conduct a comprehensive risk assessment and threat modelling exercise to identify potential security threats and vulnerabilities specific to non-terrestrial networks. We need to consider factors such as the increased exposure to physical and environmental risks, the unique attack surface presented

by satellite and airborne platforms, and the potential impact of communication disruptions or interference.

Security Architecture Design: We also need to develop a tailored security architecture that takes into account the distributed and dynamic nature of non-terrestrial networks by implementing security controls at multiple layers of the network stack, including the physical layer (e.g., satellite links), the network layer (e.g., routing protocols), and the application layer (e.g., communication protocols). Finally, we need to consider leveraging encryption, authentication, and access control mechanisms to protect data confidentiality, integrity, and availability.

Authentication and Access Control: One important aspect is to implement robust authentication and access control mechanisms to verify the identities of users, devices, and applications accessing the network and to consider the use of digital certificates, biometric authentication, or multi-factor authentication to enhance security. We also need to include the establishment of granular access policies based on user roles, privileges, and trust levels to limit unauthorized access to critical network resources.

Secure Communication Protocols: The deployment of secure communication protocols that are resilient to the challenges of non-terrestrial environments, such as latency, packet loss, and signal degradation is important. Thus, we need to use protocols with built-in security features, such as Transport Layer Security (TLS) for encrypted communication and Datagram Transport Layer Security (DTLS) for secure communications over unreliable networks.

Monitoring and Incident Response: We will implement continuous monitoring and real-time threat detection capabilities to detect and respond to security incidents in non-terrestrial networks. For this, we must consider the use of intrusion detection systems (IDS), security information and event management solutions, and anomaly detection algorithms to monitor network traffic, detect suspicious behaviour, and alert security teams to potential threats. The development of incident response plans and procedures, as described in the previous section, to effectively mitigate security breaches and minimize their impact on network operations is mandatory.

Collaboration and Standardization: We will foster collaboration and information sharing among stakeholders in the non-terrestrial communications ecosystem, including satellite operators, network providers, equipment manufacturers, and regulatory bodies and participate in industry forums, working groups, and standards organizations to develop best practices, guidelines, and security standards tailored to the unique requirements of non-terrestrial networks. Our final goal is to align security efforts with existing standards and frameworks, such as those developed by the International Telecommunication Union (ITU), IETF, and 3GPP.

5 CONCLUSION

This document has provided a comprehensive overview of the ongoing research and development efforts within the scope of the 6G-NTN project. From exploring the transition from 5G to 6G non-terrestrial networks to addressing the dynamic orchestration and autonomous monitoring challenges, each section has contributed valuable insights into the future of wireless communication technologies.

The objectives and innovation potentials of the 6G-NTN project have been discussed, highlighting the need for novel approaches to network architecture, monitoring, and management in the context of emerging 6G technologies. The proposed new virtualized and cloud-native architecture offers promising solutions to the complexities of core network integration, monitoring, and ML-based traffic and resource prediction.

Furthermore, this deliverable has shed light on the critical importance of cybersecurity in the 6G NTN ecosystem. By addressing the context and technical challenges associated with cybersecurity, and proposing a proactive security mechanism supported by a technical solution for detection before deployment, this work aims to enhance the resilience and security of future wireless communication networks.

As the project progresses, further research and development efforts will focus on refining the proposed methodologies, implementing advanced ML techniques, and addressing the unique challenges of securing 6G non-terrestrial networks. By leveraging collaborative efforts and interdisciplinary approaches, the 6G-NTN project aims to advance the state-of-the-art in wireless communication technologies and pave the way for a more connected and resilient future.

A novel 6G edge and core architecture will be developed to natively integrate TN and NTN components. This approach leverages the capabilities of regenerative payloads to host Virtual Network Functions (VNF) and edge services, using lightweight micro-service orchestrators. This integration is crucial for ensuring seamless operation and efficient resource management across the diverse components of the network.

The 6G-NTN will also incorporate advanced machine learning techniques for traffic and resource prediction. By leveraging these techniques, the network can intelligently anticipate and respond to varying traffic demands, optimizing the deployment and utilisation of network resources. This ML-based approach enhances the overall performance and responsiveness of the network, making it more adaptive to changing conditions.

Finally, the importance of cybersecurity is underscored by the proactive security mechanisms proposed in this project. Implementing technical solutions for threat detection before deployment, the 6G-NTN aims to enhance the resilience and security of the network. This proactive stance is essential in protecting the integrity and functionality of the network in the face of evolving cyber threats.

In summary, the findings presented in this document underscore the importance of proactive measures, dynamic orchestration, and autonomous monitoring in shaping the future of 6G non-terrestrial networks. With continued collaboration and innovation, the 6G-NTN project seeks to unlock the full potential of wireless communication technologies and drive towards a more sustainable and interconnected future.