

A decorative graphic consisting of several parallel lines in shades of blue and grey, forming a stylized 'L' or corner shape.

Net-Zero self-adaptive activation of distributed self-resilient augmented services

D4.6 NATWORK Federated repository for B5G/6G networks (cross layer AI-based attacks Datasets).r2

Lead beneficiary	CERTH	Lead author	Antonios Lalas, Anastasios Drosou
Reviewers	Francesco Paolucci (CNIT), Mays Al-Naday (UESSEX)		
Type	R & DATA	Dissemination	PU
Document version	V1.0	Due date	30/06/2026



Co-funded by
the European Union

6G SNS

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



UK Research
and Innovation

Project information

Project title	Net-Zero self-adaptive activation of distributed self-resilient augmented services
Project acronym	NATWORK
Grant Agreement No	101139285
Type of action	HORIZON JU Research and Innovation Actions
Call	HORIZON-JU-SNS-2023
Topic	HORIZON-JU-SNS-2023-STREAM-B-01-04 Reliable Services and Smart Security
Start date	01/01/2024
Duration	36 months

Document information

Associated WP	WP4
Associated task(s)	T4.1 - AI-based attacks & Datasets formulation T4.5 - Federated repository for boosting AI-powered 6G security
Main Author(s)	Antonios Lalas, Anastasios Drosou (CERTH)
Author(s)	Alexandros Papadopoulos, Asterios Mpatziakas, Evangelos V. Kopsacheilis, Dimitrios Trakos, Nikolaos Vakakis, Virgilios Passas, Eleni Chamou, Maria-Anna Pistela, Sotirios Kopsinos, Georgios Tsismanidis, Evi Vogiatzi, Konstantinos Nikiforidis, Dimitrios Manolakis, Nikolaos Makris, Donatos Stavropoulos, Thanasis Korakis, Georgios Agrafiotis, Konstantinos Giapantzis, Anastasios Drosou (CERTH), Wissem Soussi, Gökcan Cantali, Gürkan Gür (ZHAW), Maxime Vinh Hoa La, Edgardo Montes de Oca (MONT), Roberto Gonzalez, Jaime Fuster (NEC), Ioannis Markopoulos, Angelos Lampropoulos (NOVA), Péter Vörös (ELTE), Shankha Gupta, Mays AL-Naday (UESSEX), Vincent Lefebvre (TSS), Nasim Nezhadsistani, Andy Aidoo (UZH), Joachim Schmidt, Eryk Schiller (HESSO), Jorge Pose Eiroa, Julio Suárez Gómez (GRAD), Layal Ismail, Rana Abu Bakar, Francesco Paolucci (CNIT)
Reviewers	Francesco Paolucci (CNIT), Mays Al-Naday (UESSEX)
Type	R & DATA – Document, Report & Data
Dissemination level	PU – Public
Due date	M30 (30/06/2026)
Submission date	30/06/2026

Document version history

Version	Date	Changes	Contributor (s)
r1.v0.1	29/01/2025	Initial table of contents	Anastasios Drosou (CERTH)
r1.v0.2	10/02/2025	Contributions for various sections	Asterios Mpatziakas, Evangelos V. Kopsacheilis (CERTH)
r1.v0.3	10/04/2025	Contributions for various sections	All authors
r1.v0.4	20/05/2025	Contributions for Annexes	Asterios Mpatziakas, Virgilios Passas, Eleni Chamou, Evi Vogiatzi, Konstantinos Nikiforidis (CERTH), Wissem Soussi, Gökcan Cantali, Gürkan Gür (ZHAW), Edgardo Montes de Oca (MONT), Roberto Gonzalez, Jaime Fuster (NEC), Péter Vörös, Mohammed Alshawki (ELTE), Shankha Gupta, Mays AL-Naday (UEssex), Vincent Lefebvre (TSS), Tom Goethals (IMEC)
r1.v0.5	11/06/2025	First version refinement for internal review	Asterios Mpatziakas, Evangelos V. Kopsacheilis (CERTH)
r1.v0.6	14/06/2025	Version ready for review	Antonios Lalas (CERTH)
r1.v0.7	20/06/2025	Review complete and feedback to co-authors	Ioannis Markopoulos, Angelos Lampropoulos (NOVA), Tom Goethals (IMEC)
r1.v0.8	24/06/2025	Refinement based on review comments	All authors
r1.v0.9	26/06/2025	Quality review complete	Joachim Schmidt, Leonardo Padial (HES-SO)
r1.v0.9.5	01/07/2025	Final review and refinements	Asterios Mpatziakas, Evangelos V. Kopsacheilis, Virgilios Passas (CERTH)
r1.v1.0	03/07/2025	Final version for submission	Antonios Lalas (CERTH)
r2.v0.1	05/02/2026	ToC	Alexandros Papadopoulos (CERTH)
r2.v0.2	10/02/2026	Adding the template for dataset description	Alexandros Papadopoulos, Eleni Chamou (CERTH)
r2.v0.3	15/03/2026	Adding the description of the NATWORK DMP	Alexandros Papadopoulos, Dimitrios Trakos (CERTH)
r2.v0.4	31/03/2026	Adding the description of the generated during NATWORK datasets	All authors
r2.v0.4.5	05/05/2026	Addressing format issues and commenting for missing information	Alexandros Papadopoulos (CERTH) and CERTH Team

Version	Date	Changes	Contributor (s)
r2.v0.5	13/05/2026	Addressing the comments and adding information in Section 5	All authors
r2.v0.6	15/05/2026	Adding section 7 on Exploitation	Evangelos V. Kopsacheilis (CERTH)
r2.v0.6.5	18/07/2026	Proof reading, adapting sections 1,2,3,4 and formatting	Alexandros Papadopoulos (CERTH) and CERTH Team
r2.v0.7	29/05/2026	Review complete and feedback to co-authors	Francesco Paolucci (CNIT), Mays Al-Naday (UESSEX)
r2.v0.8	24/06/2025	Refinement based on review comments	All authors
r2.v0.9	26/06/2026	Quality review complete	Leonardo Padial, Eryk Schiller (HES-SO)
r2.v0.9.5	29/06/2026	Final review and refinements	Alexandros Papadopoulos (CERTH) and CERTH Team
r2.v1.0	30/06/2026	Final version for submission	Antonios Lalas (CERTH)

Disclaimer

Co-funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or 6G-SNS. Neither the European Union nor the granting authority can be held responsible for them. The European Commission is not responsible for any use that may be made of the information it contains.

While the information contained in the documents is believed to be accurate, the authors(s) or any other participant in the NETWORK consortium make no warranty of any kind with regard to this material including, but not limited to the implied warranties of merchantability and fitness for a particular purpose.

Neither the NETWORK Consortium nor any of its members, their officers, employees, or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

Without derogating from the generality of the foregoing neither the NETWORK Consortium nor any of its members, their officers, employees, or agents shall be liable for any direct or indirect or consequential loss or damage caused by or arising from any information advice or inaccuracy or omission herein.

Copyright message

© NETWORK Consortium. This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation, or both. Reproduction is authorized provided the source is acknowledged.

Contents

List of acronyms and abbreviations	10
List of figures	12
List of tables	15
Executive summary	17
1 Introduction	18
1.1 Purpose and structure of the document	18
1.2 Intended Audience	19
1.3 Interrelations	19
1.4 Advancements with respect to D4.5	20
2 Existing Knowledge	22
2.1 Main Technologies: Data Management Platforms in Federated Repositories	22
2.1.1 Event Processing	22
2.1.2 Massive Scaling	23
2.1.3 Edge Data Repositories	24
2.2 Commercial Solutions	24
2.3 Telecommunication Data Repositories	28
2.4 Solutions in terms of EU projects	29
2.4.1 SHOW Data Management Platform	29
2.4.2 Gaia-X / International Data Spaces (IDS)	30
2.4.3 TEADAL	31
2.4.4 IoTFeds	31
2.4.5 FIESTA	31
2.4.6 5G-EPICENTRE	32
2.4.7 EPOS: The European Plate Observing System	33
2.4.8 DICE - Data Infrastructure Capacity for EOSC	34
3 Data in B5G/6G networks	35
3.1.1 Data Granularity and Temporal Resolution	36
3.1.2 Data Sensitivity and Privacy	37

3.1.3	Spatial and Contextual Metadata	37
3.1.4	Data Format and Interoperability	38
3.2	Volume of data & techniques	39
4	NATWORK Repository	40
4.1	Functional & Non-Functional Requirements	40
4.2	Abstract Architecture.....	44
4.3	Selected tools and technology.....	45
4.4	CKAN Platform	47
4.4.1	Registration	50
4.4.2	Organisation.....	51
4.4.3	Group and Organization.....	52
4.4.4	Dataset	54
4.4.5	Visualisation	56
4.5	NATWORK Ecosystem	56
5	Cross-Layer AI-based Attacks Datasets.....	59
5.1	Classification of attacks.....	59
5.2	AI-DoS attack Tool - GORGO	63
5.2.1	System Architecture.....	63
5.2.2	Experimental Setup and validation	67
5.3	Jamming Attack.....	68
5.3.1	System Architecture.....	69
5.3.2	Experimental Setup and validation	70
5.4	OT-ICS-attacks	71
5.4.1	System Architecture.....	72
5.4.2	Experimental Setup and Validation	73
5.5	Eavesdropping Attack	78
5.5.1	System Architecture.....	78
5.5.2	Experimental Setup and validation	78
5.6	DoSt Attack	79

5.6.1	System Architecture.....	80
5.6.2	Experimental Setup and validation.....	80
5.7	Mirai botnet attack	81
5.7.1	System Architecture.....	81
5.7.2	Experimental Setup and validation.....	83
5.8	5G Interface Flooding Attacks.....	84
5.8.1	System Architecture.....	84
5.8.2	Experimental Setup and Validation	85
5.9	Inter-Container Attacks in Kubernetes	85
5.9.1	System Architecture.....	86
5.9.2	Experimental Setup and Validation	87
6	Data Generation during NATWORK Project.....	88
6.1	Jamming Detection	88
6.2	Channel Estimation	92
6.3	Jamming Mitigation	95
6.4	Anomaly Detection on Kubernetes (GATAKU).....	97
6.5	Stateful CNF Live Migration Dataset.....	98
6.6	Channel Estimation for sub-THz bands.....	100
6.7	DetAction jamming Dataset.....	103
6.8	Jamming Detection	105
6.9	DDoS Detection	107
6.10	Edge-IIoT Adversarial Dataset.....	109
6.11	5G Network Traffic dataset.....	113
6.12	OT-ICS-attacks dataset.....	120
6.13	Denial of Sustainability Dataset	124
6.14	Cyber Threat Intelligence dataset.....	127
6.15	A Multimodal 5G Network & Container Metrics Dataset.....	129
6.16	MARLIN-5G: Multi-Agent Reinforcement Learning Intrusion Network dataset for 5G	
	136	

7	Exploitation Aspects and Potentials	145
7.1	Open science and open access to research data	145
7.2	Open science and Intellectual Property Rights (IPR) - Exceptions to open science ...	146
7.3	Market Demand and Economic Outlook for AI-Native Telemetry and Datasets	147
7.4	NATWORK dataset exploitation potential strategies and models.....	148
7.5	Considerations on open dataset exploitation aspects	149
7.5.1	Commercial Exploitation.....	149
7.5.2	AI Training on Open Datasets	150
7.6	Operational Blueprint for Consolidated Dataset Exploitation.....	150
7.6.1	Federated Institutional Stewardship Framework.....	150
7.6.2	Institutional Knowledge Transfer and B2B/B2G Consultancies.....	151
7.6.3	Policy Support and European Standardization Mechanisms.....	151
7.6.4	Dual-Licensing Legal Architecture.....	151
7.7	Legal and Licensing Issues.....	152
7.7.1	The regulatory framework for NATWORK datasets	152
8	Conclusions	153
9	References	155
	Annexes.....	160
A.1	Data generated per partner in the project consortium.....	160
A.2	Collection of information by each data related partner.....	163
A.3	NATWORK DMP documentation	176

List of acronyms and abbreviations

Abbreviation	Description
3GPP	Third Generation Partnership Project
AI	Artificial Intelligence
AMF	Access and Mobility Function
API	Application Programming Interface
ASN.1	Abstract Syntax Notation One
AV	Autonomous Vehicles
AWS	Amazon Web Services
CAIDA	Center for Applied Internet Data Analysis
CCAM	Cooperative Connected and Automated Mobility
CEP	Complex Event Processing
CNF	Cloud-Native Function
CKAN	Comprehensive Knowledge Archive Network
CPU	Central Processing Unit
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
CTI	Cyber Threat Intelligence
DDoS	Distributed Denial-of-Service
DICE	Data Infrastructure Capacity for EOSC
DMP	Data Management Platform
DOI	Digital Object Identifier
DoS	Denial of Service
DoSt	Denial of Sustainability
EOSC	European Open Science Cloud
EPOS	European Plate Observing System
ESP	Event Stream Processing
ETL	Extract, Transform, Load
eMBB	Enhanced Mobile Broadband
ETSI	European Telecommunications Standards Institute
EU	European Union
FCC	Federal Communications Commission
GDPR	General Data Protection Regulation
HDF5	Hierarchical Data Format version 5
HDFS	Hadoop Distributed File System
HTTP	HyperText Transfer Protocol
HTTP2 or HTTP/2	HyperText Transfer Protocol version 2
I/Q	In-phase/Quadrature-phase
ID	Identity Document
IDS	International Data Spaces
IoT	Internet of Things

Abbreviation	Description
IP	Internet Protocol
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LPWA	Low-Power Wide-Area
ML	Machine Learning
MIMO	Multiple Input Multiple Output
mMTC	Massive Machine-Type Communications
MQTT	Message Queuing Telemetry Transport
NMS	Network Management Systems
NR	New Radio
PCAP	Packet CAPture
PL	Physical Layer
PPDR	Public protection and disaster relief
QoE	Quality of Experience
QoS	Quality of Service
RAM	Random Access Memory
RIA	Research Innovation Action
RIS	Reconfigurable Intelligent Surface
RSSI	Received Signal Strength Indicator
SDN	Software-Defined Networking
SDR	Software Defined Radio
SHOW	Shared Automation Operating Models for Worldwide Adoption
SNR	Signal to Noise Ratio
SNS	Social Networking Services
SNS-JU	Smart Networks and Services Joint Undertaking
SQL	Structured Query Language
symbloTe	Symbiosis of smart objects across IoT environments
TCP	Transmission Control Protocol
TCS	Thematic Core Services
TTL	Time To Live
UE	User Equipment
UI	User Interface
UPF	User Plane Function
URLLC	Ultra-Reliable and Low Latency Communications
VNF	Virtualized Network Function
WP	Work Package
XML	eXtensible Markup Language
YANG	Yet Another Next Generation

List of figures

Figure 1: Basic Workflow of SHOW Data Management Platform	30
Figure 2: Abstract overview of NATWORK's repository architecture	45
Figure 3: Overview of mechanisms used during the real-time process.	46
Figure 4: The introduction page of NETWORK CKAN	51
Figure 5: The Organization menu of NETWORK CKAN	51
Figure 6: The Group menu of NETWORK CKAN.....	53
Figure 7: The Dataset menu of NETWORK CKAN	55
Figure 8: CERTH lab with experimental testbed.	69
Figure 9: Jammer user interface in GNU Radio.	71
Figure 10: System architecture for OT-ICS attacks	73
Figure 11: Reconnaissance phase of the attack, showing network scanning using Nmap to identify hosts, open ports, and vulnerabilities.	74
Figure 12: Initial access phase, demonstrating exploitation of the Log4Shell vulnerability using Metasploit to compromise a target system.	75
Figure 13: Execution phase, where the attacker establishes a remote shell and executes commands on the compromised machine.	75
Figure 14: Lateral movement phase, illustrating network scanning, identification of the SCADA server, and brute-force attack against ScadaBR.....	76
Figure 15: Command-and-control phase using MODBUS/TCP protocol to communicate with compromised devices while blending with legitimate industrial traffic	76
Figure 16: Impact phase, showing manipulation of machine states via MODBUS/TCP, leading to disruption of industrial processes.....	77
Figure 17: MMT-based anomaly detection architecture, including MMT-Probe, MMT-Security, and SACD dashboard components.	77
Figure 18: Functional lab architecture for eavesdropping and PKG.....	79
Figure 19: NCL Testbed Infrastructure.....	80
Figure 20 Dedicated IoT traffic test bed.	82
Figure 21 Client nodes	83
Figure 22: Server nodes.	84
Figure 23: An overview of the Kubernetes setup on the cloud for the GATAKU dataset.	87
Figure 24: Structure of Jasmin Dataset.....	89

Figure 25: Constellation of clear BPSK signal.....	91
Figure 26: Constellation of clear 16-QAM signal	92
Figure 27: Doppler Spectrum of Highway scenario (a) and Urban Scenario (b).....	93
Figure 28: Structure of Channel Estimation Dataset.	94
Figure 29: Structure of Jamming Mitigation dataset.....	95
Figure 30: Structure of files of Jamming Mitigation Dataset.....	96
Figure 31: The structure of the GATAKU dataset.	97
Figure 32: Top features of the GATAKU dataset based on value uniqueness.....	98
Figure 33: The structure of the CNF Stateful Live Migration dataset.....	99
Figure 34: SubTHzPKG_Dataset_Structure	101
Figure 35: Structure of DetAction dataset.....	103
Figure 36 Carrier Frequency Offset over the PRBs. White bands represent unjammed windows, while grey-shadowed bands represent jammed windows.....	106
Figure 37: Heatmap DDoS Detection results.	108
Figure 38: Structure of the Edge-IIoT Adversarial Dataset.....	110
Figure 39: Dataset Class Distribution.....	110
Figure 40: Payload Length Distribution across Classes.....	112
Figure 41: Feature Correlation Heatmap.....	112
Figure 42: Structure of 5G Network Traffic dataset	114
Figure 43: 5G testbed used to generate the flooding-attack dataset, showing the UE, gNB, 5G core functions, UPF, data network, and adversarial node, together with the monitored interfaces.	116
Figure 44: Representative benign packet-rate traces web-browsing scenarios	117
Figure 45: Representative benign gaming traffic traces.....	117
Figure 46: Representative benign streaming traffic traces	118
Figure 47: N2 SCTP flooding trace, showing sustained signaling-plane overload behavior during the attack interval	118
Figure 48:N3 ICMP flooding trace.....	119
Figure 49: N4 PFCP flooding trace	119
Figure 50: N6 TCP flooding trace captured after the UPF, showing sustained overload behavior on the external data interface.	120
Figure 51: Structure of the OT-ICS-attacks dataset.	121

Figure 52: Workflow of the Apache Tomcat compromise scenario in the OT/ICS environment.	121
Figure 53: Workflow of the Command-and-control activity scenario in the OT/ICS environment.	123
Figure 54: Structure of DoSt Dataset.	124
Figure 55: Average Benign and Attack RPS	125
Figure 56: Average CPU usage	126
Figure 57: Average Memory Usage	126
Figure 58. A subset of the STIX ontology, including all entities	128
Figure 59 Structure of Multimodal 5G Network & Container Metrics Dataset	130
Figure 60 Normal vs. Attack Traffic — Packet Rate & Byte Rate	134
Figure 61: 5G Container CPU — During Attack	135
Figure 62: Structure of MARLIN-5G dataset	137
Figure 63: Attack sub-category distribution across all RL rounds. Colour coding: red = high-intensity (flood/storm), orange = medium (burst), blue = low (probe), green = technique-based.	144
Figure 64: Confirmation e-mail for user registration	178
Figure 65: Dashboard Page (Member role user)	179
Figure 66: Dashboard Page (Editor role user)	180
Figure 67: Create Dataset initial form	181
Figure 68: Create Dataset subsequent form	182
Figure 69: Organizations Page	183
Figure 70: Organization Page	184
Figure 71: Dataset info through the Organization Page	185
Figure 72: Preview Dataset Page	186
Figure 73: Data API	187

List of tables

Table 1: Big Data Management Software Solutions	28
Table 2: Representative Attributes of Data Types in B5G/6G Networks	36
Table 3: Sensitivity Classification of B5G/6G Data Types in NATWORK Project.....	37
Table 4: Functional Requirements of NATWORK DMP.....	40
Table 5: Non-functional requirements	44
Table 6: Mapping CKAN features with Functional and Non-Functional Requirements of NATWORK DMP	47
Table 7: Classification of attacks.....	59
Table 8: HTTP2 extension per-agent action space versus baseline per-agent action space.....	65
Table 9: GTP-U extension per-agent action space versus baseline per-agent action space.....	66
Table 10: PFCP extension per-agent action space versus baseline per-agent action space	67
Table 11 :Comparison of protocol aware vs generic transport layer fuzzing mean per-episode reward over 100 episodes	68
Table 12. Main parameters of Jasmin Dataset	89
Table 13: Statistical metrics of the data per case and modulation	90
Table 14: Main parameters of Channel Estimation Dataset.....	94
Table 15: Signal power statistics for Highway and Urban NLOS scenarios.	95
Table 16. Main parameters of jamming mitigation dataset.....	96
Table 17: Distribution of samples in the GATAKU dataset	97
Table 18: The distribution of migration types in the dataset.....	100
Table 19: Main parameters of Channel Estimation Dataset.....	102
Table 20: Statistical Metrics.....	102
Table 21: Parameters of the .pt file.	104
Table 22: Summary of the collected datasets.	105
Table 23: Edge-IIoT Adversarial Dataset illustrating TCP header features and corresponding labels.	111
Table 24: Main parameters of the 5G Network Traffic dataset.	114
Table 25: Statistical metrics of the data per case and modulation	115
Table 26: Statistical metrics of the data per case and modulation	115

Table 27: Key characteristics of Apache Tomcat compromise scenario.	122
Table 28: Key characteristics of Command-and-control activity scenario.	123
Table 29: DoSt Dataset Feature Description.....	124
Table 30: Dataset statistics: number of reports per source, and report length in words and sentences.	127
Table 31: Topics covered by the reports in the dataset.	127
Table 32: Dataset statistics by STIX bundle. Final column shows percentage of bundles containing each object or relation at least once.	128
Table 33: Summary of the main parameters captured in Resource monitor CSV	131
Table 34: Summary of the main parameters captured in Network traffic PCAP	131
Table 35: Attack timeline and short description	133
Table 36: Dataset File Summary	133
Table 37: MARLIN-5G dataset parameters.....	137
Table 38: Label record fields.....	139
Table 39: Metadata features	139
Table 40: Statistical features.....	140
Table 41: GTP-U protocol features	140
Table 42: PFCP Protocol features.....	141
Table 43: HTTP/2 protocol features.....	141
Table 44: Data generated per partner in the project consortium, input from D1.2	160
Table 45: Template for gathering data handling information related to NATWORK's Data repository.....	163
Table 46: CERTH data handling information.....	164
Table 47: ZHAW data handling information	168
Table 48: NEC data handling information.....	169
Table 49: MONT data handling information	170
Table 50: ELTE data handling information.....	171
Table 51: TSS data handling information.....	173
Table 52: UESSEX data handling information	173

Executive summary

This deliverable presents the progress and main outcomes of Task 4.5, “*Federated repository for boosting AI-powered 6G security*”, up to Month 30. It updates and extends D4.5 by consolidating the NATWORK federated repository, the associated Data Management Platform (DMP), and the project-generated datasets related to AI-based security, cross-layer attacks, network monitoring, and B5G/6G experimentation.

The NATWORK repository supports the secure collection, storage, organisation, and sharing of data among consortium partners, while also preparing selected project outputs for wider visibility within the SNS-JU community. The platform is based on a Comprehensive Knowledge Archive Network (CKAN)-driven environment and provides a common interface for dataset publication, metadata management, controlled access, visualisation, and reuse. The deliverable also updates the description of the repository architecture, functional and non-functional requirements, selected technologies, and operational workflow.

A major advancement of this revision is the detailed description of the datasets generated and shared during the NATWORK project. These include datasets related to jamming detection, channel estimation, jamming mitigation, Kubernetes anomaly detection, Cloud-Native Function (CNF) live migration, sub-THz channel estimation, Distributed Denial-of-Service (DDoS) detection, OT/ICS attacks, denial-of-sustainability attacks, edge-cloud workload, Cyber Threat Intelligence (CTI), and multimodal 5G network and container metrics. The deliverable also extends the description of cross-layer AI-based attacks. Finally, D4.6 defines the exploitation perspective of the NATWORK repository and clarifies how the collected datasets, documentation, and platform services can support technical integration, AI model development, validation activities, and post-project reuse.

1 Introduction

This deliverable, *D4.6 NATWORK Federated repository for B5G/6G networks (cross-layer AI-based attacks Datasets).r2* describes the overall progress and results for task T4.5. The task transparently integrates NATWORK's autonomous, decentralised data sources by adding a data-abstraction layer and uniform interfaces. Query wrappers allow user queries to be tailored to the native language of each source, feeding into a federated repository that underpins NATWORK AI search. This AI-ready dataset outlives the project, powering B5G/6G network innovation across Europe, and also incorporates CTI information gathered in task T4.4. The deliverable is also associated with data generated by the consortium during the project's technical activities, particularly under T4.1 - AI-based attacks & Datasets formulation.

The demand for secure, energy-efficient, and intelligent network infrastructures has become more critical than ever, as technological advantage and associated cyber threats stimulate the development of NATWORK's advanced framework, *"Net-Zero self-adaptive activation of distributed self-resilient augmented services"*.

1.1 Purpose and structure of the document

The purpose of this document is to provide a comprehensive overview of the platform's architecture, data models, and the tools used for data collection, storage, and management. It also includes information on the implementation of the attacks in NATWORK, as well as a detailed description of the datasets generated during the project.

Following the Introduction, which sets the stage for the document's purpose, audience, and its interconnections within the project's framework, the structure continues as follows:

2. **Existing Knowledge:** Presents the main technologies, tools, and techniques currently employed for federated data collection and manipulation platforms in the telecommunication domain and in EU projects.
3. **Data on B5G/6G:** Describes the general attributes and structure of data in the telecommunication domain and the framework used by each NATWORK partner.
4. **NATWORK Repository:** Includes the functional and non-functional requirements of the NATWORK Data Management Platform (DMP). Moreover, it also describes the main architecture and the key platform tool.
5. **Cross-Layer AI-based Attacks Datasets:** Provides a classification of the employed attacks. Describes briefly each attack associated with specific datasets, either open or custom-generated by the different testbeds of the members of the consortium.

6. **Data Generation during NATWORK Project:** Provides a detailed description of all the datasets that have been generated during the NATWORK project and shared via the Data Management Platform.
7. **Exploitation Aspects and Potential:** Describes the exploitation plan up to the end of the project and after it within the SNS community.
8. **Conclusions:** Wraps up the document, reflecting on the project's strategic orientation and its final outputs.

1.2 Intended Audience

The NATWORK *D4.6 NETWORK Federated repository for B5G/6G networks (cross-layer AI-based attacks Datasets).r2* is designated for public use in the context of the project's datasets generation by the NATWORK consortium, comprising members, project partners, and affiliated stakeholders. This document primarily focuses on the project's data owners and serves as a reference tool throughout the project's lifespan. It also highlights the project's strategic blueprint and collective vision, ensuring that all collaborative efforts are harmonized and directed toward fulfilling the project's ambitions.

1.3 Interrelations

The NATWORK consortium integrates a multidisciplinary spectrum of competencies and resources from academia, industry, and research sectors, focusing on user-centric service development, robust economic and business models, cutting-edge cybersecurity, seamless interoperability, and comprehensive on-demand services. The project brings together 15 partners from 10 EU member states and associated countries (UK and CH), ensuring broad representation for addressing the security requirements of emerging 6G Smart Networks and Services in Europe and beyond.

NATWORK is categorized as a "Research Innovation Action - RIA" project and is methodically segmented into 7 WPs, further subdivided into tasks. With partners contributing to multiple activities across various WPs, the structure ensures clarity in responsibilities and optimizes communication amongst the consortium partners, boards, and committees. The interrelation framework within NATWORK facilitates smooth operations and collaborative innovation across the consortium, ensuring the interconnection of diverse expertise from the various entities (i.e., Research Institutes, Universities, SMEs, and large industries), thereby enabling scientific, technological, and security advancements in the realm of 6G. The *D4.6 NETWORK Federated repository for B5G/6G networks (cross-layer AI-based attacks Datasets).r2* addresses all activities of the NATWORK project related to data generation, collection, processing, storage, and sharing. The deliverable is directly related to *D1.2 Quality Assurance, Risk Assessment, Data Management Plan, Ethics & Regulatory issues.r1*, where the NATWORK Data Management Plan was presented.

The deliverable is also associated with the data generated by the consortium during the project's technical activities, and especially under T4.1 - AI-based attacks & Datasets formulation. The repository also includes the information collected from public CTI sources in T4.4 - CTI, Security, and explainability for AI. Moreover, it is relevant to WP3, WP4, WP5, and WP6, as it serves as the repository for data generated during the development activities of these WPs. Still, it is also relevant for the project's evaluation activities of WP6.

1.4 Advancements with respect to D4.5

This deliverable, D4.6, represents the second revision of D4.5 and reports on the progress achieved since the initial release of the NATWORK federated repository. While D4.5 mainly introduced the repository concept, the initial DMP implementation, and the planned data collection activities, D4.6 provides a mature and consolidated view of the platform and the datasets generated within the project.

The main advancements are the following:

- The NATWORK DMP description has been updated to reflect the current implementation status, including the CKAN-based repository, user roles, dataset publication workflow, metadata handling, visualization capabilities, and access-control mechanisms.
- The repository architecture and the mapping between CKAN features and NATWORK functional/non-functional requirements have been refined to better represent the operational use of the platform.
- The cross-layer AI-based attack section has been extended. In particular, in addition to the AI-DoS attack tool GORGO, D4.6 now includes setups for jamming, eavesdropping, Denial of Sustainability (DoSt), Mirai, interface flooding, and Kubernetes inter-container attacks. Their system architecture, experimental environment, and validation process are also presented.
- A new major section on data generation during the NATWORK project has been introduced. It provides detailed descriptions of the datasets shared by partners, covering wireless jamming, channel estimation, Kubernetes security, Cloud-Native Function (CNF) migration, 5G traffic, Distributed Denial-of-Service (DDoS), OT/ICS attacks, Cyber Threat Intelligence (CTI), and multimodal network/container monitoring.
- The deliverable now includes dataset-level analysis, figures of the datasets' structure and the included data values, statistical summaries, file structures, and parameter tables, making the repository content more transparent, reusable, and useful for AI-based security research.

- A dedicated exploitation plan has been added, describing how the NATWORK repository and datasets can be used during the remaining project activities and beyond the project lifetime.

Overall, D4.6 moves the deliverable from an initial repository definition toward a concrete project data asset, documenting both the operational platform and the datasets that support NATWORK's AI-powered B5G/6G security objectives.

2 Existing Knowledge

2.1 Main Technologies: Data Management Platforms in Federated Repositories

To fulfill the requirements of the NATWORK project, a DMP has been developed to meet the specific needs arising from the format and volume of the generated data. A DMP is a software system designed to collect, process, and manage large and complex datasets derived from diverse sources. These platforms are widely used across various sectors, such as industry, healthcare, environmental monitoring, and logistics, where real-time data processing is essential for decision-making, statistical analysis, and deriving actionable insights.

2.1.1 Event Processing

Event Processing is a common technique for monitoring, storing, analyzing, and processing data streams. The goal is to derive meaningful insights and enable appropriate responses. DMPs employ various event processing methods to collect and process data in real time, as outlined below.

2.1.1.1 Complex Event Processing (CEP)

Complex Event Processing (CEP) offers significant advantages for managing and analysing continuous data streams, particularly in the context of Beyond 5G/6G networks. These next-generation networks generate vast amounts of real-time data from diverse sources, including IoT devices, edge computing nodes, and network infrastructure. CEP provides a robust framework for **efficiently** handling this data, enabling faster decision-making and improved network performance [1].

In general, CEP is used in applications where the following are required:

- **Complex patterns:** The event detection may include complex correlation patterns of temporal or/and spatial nature.
- **Low latency:** The time between event occurrence and system response must be in the order of a few milliseconds or less.
- **High event rate:** The incoming data rate can be hundreds or thousands of events per second.

Below are the key advantages of using CEP for 5G/6G data.

- **Real-time data processing:** CEP excels at processing continuous data streams in real time, which is critical for 5G/6G networks where low latency and high-speed data transmission are essential. By analysing data, CEP enables immediate detection of network anomalies,

security threats, or performance bottlenecks, allowing for rapid response and mitigation [1], [2].

- **Pattern recognition and anomaly detection:** CEP is designed to identify complex patterns and correlations within data streams. In 5G/6G networks, this capability is invaluable for detecting unusual behaviours, such as cyberattacks, network congestion, or equipment failures. By recognizing these patterns early, CEP helps prevent disruptions and ensures network reliability [3].
- **Scalability for High-Volume Data:** 5G and 6G networks generate massive volumes of data from millions of connected devices. CEP systems are highly scalable, capable of handling large-scale data flows without compromising performance. This scalability ensures that network operators can manage the increasing data demands of next-generation networks effectively [4], [5].
- **Energy Efficiency and Resource Optimization:** CEP supports energy-efficient network operations by analysing data flows and optimizing resource allocation. For example, it can dynamically adjust network configurations, such as Multiple Input Multiple Output (MIMO) settings or Reconfigurable Intelligent Surfaces (RIS), to reduce energy consumption while maintaining performance. This is particularly important for achieving the net-zero energy goals of future networks [6].

2.1.1.2 Event Stream Processing (ESP)

While CEP specialises in recurrent correlations between low-level events, Event Stream Processing (ESP) refers to a range of computational methods that are employed to process the stream and apply analysis. ESP can handle data from spatially and temporally continuous streams of events, whereas CEP cannot.

ESP algorithms are considered faster for performing stateless processing. This is performed by storing the state of event patterns for short time periods, which allows for more efficient use of memory and processing resources. While ESP primarily focuses on handling general event streams and traditionally lacks the advanced capability to detect complex events, modern ESP solutions have begun to address this limitation at a basic level. They achieve this by leveraging incident time, i.e., the actual time an event occurs, rather than processing time, which refers to when the event data reaches the system for analysis. This shift enables more accurate and timelier event detection, even in scenarios involving complex patterns [7].

2.1.2 Massive Scaling

Data scalability refers to the ability of a system to adapt the usage of computational resources (CPU frequency, RAM, disk) according to the application's usage demands. The volume of data generated by wireless networks is growing exponentially due to the increasing number of connected devices, higher data transmission speeds, and the proliferation of bandwidth-

intensive applications like augmented reality, autonomous vehicles, and IoT. This massive data growth necessitates data scalability for efficient utilisation.

Two types of scaling are commonly used: vertical scaling, also known as "scaling up," and horizontal scaling, also known as "scaling out. Vertical scaling involves increasing the capacity of a single server or node by adding more resources, such as CPU, memory, or storage, to handle higher workloads. While this approach is straightforward and easier to manage, it has hardware limitations and can be costly. On the other hand, horizontal scaling focuses on distributing the workload across multiple servers or nodes, enabling systems to handle greater data volumes and traffic by adding more machines to the network. This distribution is a fundamental aspect of creating robust systems, often forming the basis for more advanced architectures such as federated or decentralized platforms, in which data and control are managed across independent entities. This method is highly scalable and fault-tolerant, making it ideal for modern applications like 5G/6G networks.

2.1.3 Edge Data Repositories

Edge computing is a distributed computing method that brings data processing and storage closer to the source of data generation, such as remote devices, sensors, and user equipment, rather than relying solely on centralized cloud servers. This is achieved by deploying computing resources (e.g., servers, storage, and analytics tools) at the edge of the network, such as base stations, routers, or local data centres. The stored data will eventually be synchronized with the cloud or processed by an edge service [8], [9]. Edge computing is a critical enabler of 5G networks, providing low-latency, scalable, and efficient data storage and processing capabilities. By bringing computation and storage closer to the data source, edge computing addresses the challenges of bandwidth congestion, data privacy, and real-time analytics in 5G networks. It also supports advanced applications like network slicing, federated learning, and autonomous systems, making it an indispensable practice for the future of 5G and beyond.

2.2 Commercial Solutions

In this section, popularly used commercial solutions for big data collection and management are outlined and compared.

Apache Hadoop [10] is a foundational open-source framework designed for the distributed storage and processing of big data across clusters of commodity hardware. It is the most common software solution for big data processing. It addresses key challenges of large-scale data by enabling scalable, fault-tolerant, and cost-effective solutions, primarily through a batch processing scheme. The Hadoop Distributed File System (HDFS) stores large volumes of data across multiple nodes, ensuring redundancy and high availability, and is commonly used for logs, sensor data, and historical records. In addition, MapReduce [11], its core batch processing

paradigm, processes vast datasets in parallel by splitting tasks into smaller chunks (map) and aggregating results (reduce).

While Hadoop excels at batch processing for big data, certain use cases require real-time or near-real-time data processing capabilities. Stream processing is the preferred approach for these scenarios, enabling continuous analysis of live data flows rather than delayed batch operations. To address this need for real-time processing, the Hadoop ecosystem incorporates specialized stream processing tools like Apache Spark, Apache Storm, Apache Samza, and Apache Flink. Unlike MapReduce's periodic job-based approach, these streaming frameworks process data continuously as it arrives, providing immediate insights and faster response times for time-sensitive applications.

Apache Spark [12] is a versatile cluster computing platform renowned for its high performance, scalability, and accessibility. It offers comprehensive libraries and developer-friendly interfaces in Python, Java, Scala, and SQL, making it adaptable to diverse programming environments.

The advantages of this platform are outlined below:

- **Performance Optimization:** Spark enhances the traditional MapReduce model by enabling interactive queries and real-time stream processing. In addition, it accelerates computations through in-memory processing, significantly reducing latency.
- **General Purpose Flexibility:** Spark is designed to support a broad spectrum of workloads (batch processing, streaming, machine learning, and graph analytics) within a unified framework. Also, it eliminates the need for multiple distributed systems, reducing complexity and maintenance. One more advantage is the ability to run on single machines at first, and scaling to large clusters later.
- **Cost Efficiency:** Spark can consolidate multiple data processing modes (e.g., ETL, analytics, ML) into one platform, thereby lowering infrastructure costs.

Apache Storm [13] is a distributed, open-source solution designed to process unbounded data streams with ultra-low latency and high reliability in real time. Unlike batch-processing systems (e.g., Hadoop MapReduce), Storm handles continuous, real-time data flows, making it ideal for time-sensitive applications. It is straightforward to use and compatible with any programming language.

Storm's use cases include online machine learning, real-time analytics, social media analysis, network traffic analysis, ETL and so on. Its key benefits include:

- **Ultra-low latency:** Processes data in milliseconds.
- **Scalability:** Horizontally scales to handle millions of events per second.
- **Fault-tolerance:** Automatically recovers from failures without data loss.

- **Language flexibility:** Supports Java, Python, and more.
- **Simple integration:** Works seamlessly with popular data sources.

Apache Samza [14] excels at real-time processing of network data within Kafka-centric architectures. It is ideal for stateful operations like tracking device session states in a data centre or detecting DDoS attack patterns by analysing traffic flows with low latency. Its benefits include:

- **Lightweight and simple**
- **Isolation**
- **Scalability**

Unlike batch-processing systems such as Apache Hadoop, Samza employs continuous data streaming methods to achieve sub-second response times [5]. The work in [15] presents an animal health monitoring platform for beehives, where data from multiple sources is offloaded to the cloud for analysis. The system implements a lambda architecture, incorporating LPWA and 3GPP protocols for connectivity. Data is transmitted in real time to enable immediate processing and generate actionable insights and analytics. Apache Kafka serves as the messaging layer, bridging the gap between data-producing nodes and the processing engines running Apache Samza algorithms.

Apache Flume [16] is a distributed, open-source framework designed for secure, large-scale log data collection, aggregation, and transfer. Specializing in streaming data flows, it features a simple yet scalable architecture with tunable reliability mechanisms for failover, recovery, and fault tolerance. Its extensible model makes it ideal for real-time applications.

In [17], Flume is part of a big data pipeline for autonomous vehicles (AVs), where it collaborates with Kafka, Zookeeper, HDFS, and Spark to ingest and route sensor data from AVs to HDFS for storage and analysis. Similarly, [18] proposes a cloud-based IoT data pipeline leveraging Flume to ensure uninterrupted Quality of Service (QoS) and cost-efficient resource allocation through optimised data handling.

Apache Flink [19] is an open-source software that combines both stream and batch processing. Flink is the go-to for sub-millisecond analysis of network event streams, such as identifying intrusions in firewall logs or correlating latency spikes across distributed nodes. Its stateful processing enables real-time anomaly detection and adaptive QoS routing in Software-Defined Networking (SDN). It is designed to run in diverse cluster environments, delivering in-memory processing speeds while scaling seamlessly to handle workloads of any size.

Unlike other high-throughput, low-latency solutions, Flink distinguishes itself with native support for event-time processing and robust state management. It integrates seamlessly with

distributed storage systems like Amazon Kinesis, Apache Kafka, and HDFS, enabling flexible data ingestion and persistence.

Flink's versatility extends to cloud, edge/fog computing [6] and IoT ecosystems [20] where it processes real-time data streams from sensors, devices, and other distributed sources with high efficiency and reliability.

Apache Kafka [21] is an open-source, distributed stream-processing platform primarily developed in Scala and Java, but with language-agnostic compatibility, supporting integration across diverse programming environments. It establishes a high-throughput, low-latency pipeline for efficiently managing real-time data streams. Its key components and features include:

- **Kafka Connect:** A framework for seamless integration with external systems (data import/export).
- **Kafka Streams:** A lightweight Java library for stream processing (e.g., transformations, aggregations).

Kafka employs a binary TCP protocol that:

- Batches messages to minimize network round-trip overhead.
- Maximizes throughput through:
 - Larger network packets.
 - Sequential disk operations.
 - Contiguous memory allocation. This design converts random message writes into linear, sequential writes, enhancing efficiency for producers and consumers.

Kafka's protocol and architecture make it ideal for real-time analytics, event sourcing, and log aggregation in scalable systems (e.g., financial transactions, IoT telemetry).

Another option available are cloud solutions such as:

Amazon Web Services (AWS) [22] is a leading cloud computing platform that provides on-demand access to scalable infrastructure (e.g., servers, storage, databases). Its use cases span web hosting, big data analytics, AI/ML, IoT, and enterprise IT migration, enabling users to deploy applications without requiring physical hardware. Some advantages include its scalability, cost efficiency and security provided.

Unlike non-cloud solutions, AWS eliminates the need for capital-intensive hardware, reduces maintenance overhead, and enables rapid innovation through serverless and AI services.

In Table 1, the software solutions presented here are summarised.

Table 1: Big Data Management Software Solutions

Name	Company	Supported Languages	Processing Type
Hadoop	Apache Foundation	Java, Perl, Python, Ruby, Php	Batch
Spark	Apache Foundation	Scala, Java, R, Python	Stream/Batch
Storm	Apache Foundation	Java, Perl, Python, Ruby	Stream
Samza	Apache Foundation	Java, Scala, Python, Go	Stream
Flume	Apache Foundation	Java	Batch
Flink	Apache Foundation	Scala, Java, Python	Stream/Batch
Kafka	Apache Foundation	Java, C/C++, Go, .NET, Python, Scala, Erlang, Groovy, Haskell, Kotlin, Lua, Node.js, Ocaml, Php, Ruby, Rust, Tcl, Swift and more	Stream
Amazon Web Services	Amazon	Python, JavaScript, Java, Go C#/.NET, Ruby, PHP, Rust	Stream

2.3 Telecommunication Data Repositories

Telecommunication researchers have access to a rich and mature ecosystem of public-interest data platforms: long-running Internet-backbone archives, constantly updated crowd-sourced performance tiles, specialist mmWave channel soundings and even Kubernetes-based edge-cloud logs. Together, these repositories cover every OSI layer and most regulatory or socio-economic perspectives while remaining either completely open or available under light academic agreements.

The main platforms highlighted include:

- **CRAWDAD** [23]: This platform serves as a community resource for archiving wireless data. It hosts numerous Wi-Fi, Bluetooth and mobility traces uploaded by academic researchers and conference organisers. Each dataset includes a plain-language abstract, a Digital Object Identifier (DOI) for easy citation, and user-friendly text or CSV files compatible with Excel or Python.
- **CAIDA repository** [24]: For an overview of the Internet backbone, CAIDA (Center for Applied Internet Data Analysis) provides anonymised packet headers and routing snapshots. These downloads are available as standard PCAP or JSON files and only require a simple click on the licence, making the archive accessible to new researchers.
- **Measurement Lab (M-Lab)** [25]: M-Lab offers real-world network speed figures, including raw test logs and query-ready BigQuery tables. This data is updated daily, allowing researchers to monitor network upgrades in real time.
- **Ookla** [26]: Known for its Speedtest service, Ookla publishes worldwide quarterly median download, upload and latency values. These values are provided in the form of Parquet

and Shapefile tiles ready to be mapped, offering a visual representation of network performance.

- **OpenCellID** [27]: This community project maintains a massive database of over 49 million cell tower identifiers, including GPS locations and antenna directions. All CSV data is free after creating an API key, and an interactive map on their website allows quick checks.
- **Federal Communications Commission (FCC) Broadband Data Collection Portal** [28]: In the US, the FCC provides shapefiles twice a year that show exactly where fixed and mobile broadband services are reported to be available.
- **NYU WIRELESS** [29]: For practical radio research, NYU WIRELESS offers open databases on channel propagation and a MATLAB/ns-3 simulator. Downloads are free after registration and include sample scripts for replaying measurements or integrating them directly into models.
- **COSMOS Testbed in New York** [30]: This testbed supports outdoor city testing by collecting its own measurements at 28 GHz and allowing researchers to download existing logs for further analysis.
- **EdgeNet Project** [31]: EdgeNet transforms ordinary university servers into a shared Kubernetes edge cloud. Its public site provides links to source code and actual node manifests, offering researchers a preconfigured sandbox to test core 5G functions or content delivery strategies without incurring commercial cloud costs.

2.4 Solutions in terms of EU projects

This section presents the status of available solutions or services regarding federated repositories in the telecommunications-related sector, stemming from EU-funded research, and other development initiatives of the European Union.

2.4.1 SHOW Data Management Platform

The Shared Automation Operating Models for Worldwide Adoption (SHOW) project [32] is a Horizon 2020 European initiative with a specific focus on autonomous public transport fleets. Since the complexity of collecting, manipulating and storing data related to Automated Vehicles (AVs), bears many similarities to the challenges faced by NATWORK's database, it is worth being referred to.

The main part of the SHOW infrastructure is the Data Management Platform (DMP). The DMP comprises all mechanisms dedicated to the collection, manipulation, and storage of the data within the project. Due to the variety of inputs from the partners involved in the Automated Mobility project, models and formats are also included in addition to these mechanisms. The abstract architecture is described in Figure 1.

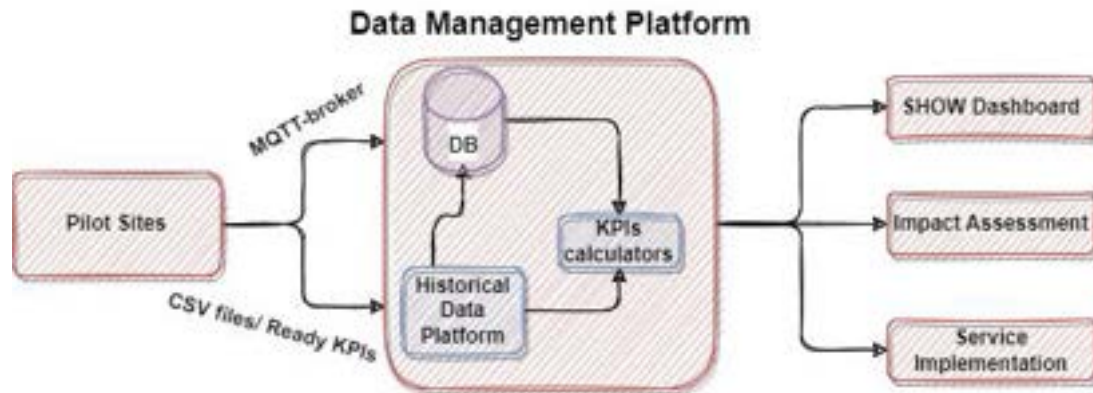


Figure 1: Basic Workflow of SHOW Data Management Platform

The pilot sites participating are connected to the DMP through three different alternatives:

1. **The site provides real-time data**, delivered directly to the database directly by a Message Queuing Telemetry Transport (MQTT) broker. The respective KPIs are calculated and then uploaded to the database and displayed on the SHOW Dashboard.
2. **The site provides CSV files with historical data**. The files are uploaded on a daily, weekly, or monthly basis. A technical team transfers the data to the database, following the same procedure as in the previous option. The delivery of historical data includes uploading past records, which can be utilized for creating and training new services.
3. **The site provides ready-to-use KPIs**. Key Performance Indicators (KPIs) are metrics derived from performance data that are calculated from historical data or directly from the system's sensors and infrastructure. Then these metrics are publicly available on the SHOW Dashboard, demonstrating the operational flow, after being uploaded to the historical database.

The data is afterwards used according to the needs of the service, for estimation, service implementation and assessment.

2.4.2 Gaia-X / International Data Spaces (IDS)

Gaia-X is a European initiative to create a federated, secure data infrastructure, where data is shared and users retain control over their data access and usage [33]. The initiative is based in Belgium and has the legal form of an international non-profit organization. It aims to develop a proposal for the next generation of data infrastructure for Europe and promote the digital sovereignty of European users of cloud-based services.

Gaia-X Hubs are established at national level to animate the Gaia-X communities locally. Data-sharing operates under transparent governance and user-controlled access via open interfaces

and standards—implemented notably in International Data Spaces (IDS) open-source frameworks.

Gaia-X's Policy Rules aim to identify clear controls to demonstrate the European values of Gaia-X, such as Openness, Transparency, Data Protection, Security and Portability. Every service offering has to be provided under the umbrella of / via the Gaia-X framework and must comply with these requirements.

2.4.3 TEADAL

The EU-funded TEADAL project [34] delivered key technologies that create stretched data lakes spanning the cloud-edge continuum and multi-cloud, enabling reliable, accurate and fast decision-making. These technologies ensure privacy, confidentiality and energy-efficient data management by using a shared approach to define, enforce and track these requirements while balancing energy consumption. The project aimed to establish trusted, verifiable and energy-efficient data flows across a mediator-less data lake federation, facilitating data sharing and enhancing decision-making capabilities across various industries.

2.4.4 IoTfeds

An open-source middleware enabling IoT platform providers to form federations and operate decentralized IoT data marketplaces using blockchain and Symbiosis of smart objects across IoT environments (symbIoTe) foundations. The IoTfeds project [35] proposes a solution that will enable small IoT companies to form partnerships to address common needs and collectively operate a controlled market for the exchange of IoT data. The IoTfeds solution introduces an Open-Source middleware that allows IoT platform providers to form IoT federations and operate a distributed IoT data marketplace, without the need for a centralized entity to operate the market. The IoTfeds project is based on the outcomes of the H2020 symbIoTe project and combines it with Blockchain technologies towards the implementation of a fully distributed federation management and marketplace framework. The project is co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation.

2.4.5 FIESTA

The FIESTA project [36] delivered a federated semantic IoT/cloud testbed, offering unified APIs, semantic interoperability, certification tools, secure AAI and data access mechanisms.

FIESTA opened new horizons in IoT experimentation at a global scale, based on the interconnection and interoperability of diverse IoT testbeds. The project produced a first-of-its-kind blueprint experimental infrastructure (tools, techniques, and best practices) that enabled testbed operators to interconnect their facilities in an interoperable way, while at the same time facilitating researchers in conducting integrated experiments that seamlessly transcended the

boundaries of multiple IoT platforms. FIESTA was validated and evaluated based on the interconnection of four testbeds (in Spain, the UK, France, and Korea), as well as through the execution of novel experiments in the areas of mobile crowd-sensing, IoT applications portability, and dynamic intelligent discovery of IoT resources.

To achieve global outreach and maximum impact, FIESTA integrated an additional testbed and experiments from Korea and collaborated with IoT experts from the USA. The participation of a Korean partner (based on its own funding) maximized FIESTA's value for EC money. Moreover, the project took advantage of open call processes to attract third parties that engaged in integrating their platforms within FIESTA or conducting value-added experiments. As part of its sustainability strategy, FIESTA established a global market confidence programme for IoT interoperability, which enabled innovative platform providers and solution integrators to ensure and certify the openness and interoperability of their developments.

2.4.6 5G-EPICENTRE

The 5G-EPICENTRE project [37] delivered a 5G testbed and microservices repository facilitating secure interoperability evaluation across Enhanced Mobile Broadband (eMBB), Massive Machine-Type Communications (mMTC), and Ultra-Reliable and Low Latency Communications (URLLC) services, particularly for Public protection and disaster relief (PPDR) verticals, leading towards an end-to-end experimentation platform for 5G public safety services. Following the best DevOps practices of the time (such as containerisation of micro-services), the EU-funded 5G-EPICENTRE project developed an open experimentation 5G platform focused on solutions serving PPDR needs. It aimed to facilitate the adoption of existing services to leverage 5G networks' advanced capabilities, as well as to provide a solid foundation for the proliferation of such services. The project provided concrete experiments and libraries, VNF chains, and NetApps. SMEs and developers were able to learn about the latest 5G applications and approaches for first responders and crisis management, and to build and experiment with their own solutions.

5G-EPICENTRE aimed to establish an environment that bridged the development and operations worlds for PPDR services. 5G was considered the mainstream broadband wireless technology of the time, with the potential to enhance the efficiency and effectiveness of high-demand daily operations such as PPDR. The ITU regarded LTE-Advanced systems and 5G as mission-critical PPDR technologies capable of addressing the needs of mission-critical intelligence by supporting mission-critical voice, data, and video services as an IMT radio interface.

The 5G-EPICENTRE project delivered an open end-to-end experimentation 5G platform focused on software solutions serving PPDR needs. The envisioned platform enabled SMEs and developers to acquire knowledge of the latest 5G applications and approaches for first responders and crisis management, as well as to build and experiment with their solutions. The 5G-EPICENTRE platform was based on an open service-oriented architecture, following the best

DevOps practices of the time (e.g., containerisation of micro-services), and could accommodate and provide open access to 5G network resources—thus acting as a 5G open-source repository for PPDR network applications.

To assess the platform, several use cases were envisioned and implemented as part of a PPDR vertical. The purpose of these use cases was to span the full range of the three ITU-defined service types (i.e., eMBB, mMTC, and URLLC) and to provide a framework for evaluating the platform’s secure interoperability capabilities beyond vendor-specific implementations.

2.4.7 EPOS: The European Plate Observing System

The EPOS Data Portal [38] is powered by a three-tier federated architecture with thematic core services, interoperability catalogues, centralized/distributed hubs, and rich UI—managed via DevOps methods [39]. The European Plate Observing System (EPOS) is a multidisciplinary, distributed research infrastructure that facilitates the integrated use of data, data products and facilities from the solid Earth science community in Europe by integrating diverse European Research Infrastructures under a common federated framework. EPOS is the sole research infrastructure representing solid Earth science in Europe.

EPOS initiatives have existed for more than a decade, initially as a project included in the European Strategic Forum on Research Infrastructures Roadmap in 2008, then implemented through two European Projects, namely EPOS-PP Preparatory Phase (2010–2014), and EPOS-IP Implementation Phase (2015–2019), and finally established as a European Research Infrastructure Consortium in 2018, a specific legal form that facilitates the establishment and operation of RIs with pan-European dimensions.

The EPOS Data Portal provides homogeneous access to heterogeneous resources made available by data providers in the EPOS community. EPOS adopted a specific methodology to comply with the FAIR principles which relies on a federated approach for providing access to Data, Data Products Services and Software, and faced ethical challenges related to data collection, data use as well as with respect to their promotion.

The main technical challenge in the EPOS case was to integrate more than 250 services, delivering more than 30 different types of data formats – spanning for instance from miniSeed waveforms for Seismology to tiff images for Satellite Data and pdf reports for Volcanology – covering more than 800 TB of data in total, described by more than 20 different types of metadata. Some metadata follows international or community-based standards, but some had to be designed and developed from scratch. This process was accelerated thanks to sharing of best practices among individual communities in EPOS.

The EPOS architecture is based on a three-component structure. The first component consists of the main data providers that collect solid Earth data through various National Research

Infrastructures, which maintain their integrity with respect to data ownership, storage and availability. Data collected by the NRIs are then combined at the middle layer (Thematic Core Services – TCS), where community-based standards are developed and data are made available to the entire European scientific community (e.g., ORFEUS7 in seismology). Services provided through these TCS form the basis for the third component where resources are integrated and made interoperable in a metadata structure. This component represents the EPOS Integrated Core services with the associated EPOS Data Portal.

2.4.8 DICE - Data Infrastructure Capacity for EOSC

The Data Infrastructure Capacity for EOSC (DICE) [40] consortium brings together a network of computing and data centres, research infrastructures, and data repositories with the purpose to enabling a European storage and data management infrastructure for EOSC, providing generic services and building blocks to store, find, access and process data in a consistent and persistent way. Specifically, DICE partners offered 14 state-of-the-art data management services together with more than 50 PB of storage capacity. The service and resource provisioning are accompanied by enhancements to the current service offering in order to fill the gaps still present in supporting the entire research data lifecycle; solutions are provided for increasing the quality of data and their re-usability, supporting long term preservation, managing sensitive data, and bridging the gap between data and computing resources.

All services provided via DICE are offered through the EOSC Portal and interoperable with EOSC Core via a lean interoperability layer to allow efficient resource provisioning from the very beginning of the project. The partners have closely monitored the evolution of the EOSC interoperability framework and guidelines to comply with a) the rules of participation for onboarding services into EOSC, and b) the interoperability guidelines for integration with the EOSC Core functions. The data services offered via DICE through EOSC are designed to be agnostic to the scientific domains to be multidisciplinary and to fulfil the needs of different communities. The consortium demonstrated the effectiveness of the service offering by integrating services with community platforms as part of the project and by engaging with new communities coming through EOSC.

3 Data in B5G/6G networks

Working on 5G/6G networks generates a diverse range of data types, each serving different purposes and originating from diverse sources. Managing and analysing this data effectively is key to unlocking the full potential of the NETWORK project's experimentation, research, and implementation. The following section presents an overview of the various types of data produced in or by B5G/6G networks.

The diverse types of data in B5G/6G networks come with a variety of attributes that determine their structure, usefulness, and processing requirements. Understanding these attributes is fundamental to the design and deployment of data-driven architectures for experimental setups, AI-driven network functions, and secure data exchange protocols within the NETWORK project.

Below is a description of the key data types that may be collected:

Network Signal and Characteristics Data: When conducting an experiment, the original's setting metadata may include timestamps, localization settings, signal frequency, received signal (I/Q) data as well as original source application's type or ID.

Network Performance Data: This includes metrics such as latency, throughput, packet loss and signal strength, and signal to noise ratio (SNR), which are critical for monitoring and optimizing network performance. These metrics help ensure high Quality of Service (QoS) and Quality of Experience (QoE) for users.

User Data: Data generated by end-users, such as location information, device usage patterns, and application usage statistics, is essential for personalising services and improving user experience. However, this data must be handled with strict adherence to privacy regulations like the GDPR.

Security and Threat Data: This includes information about cyberattacks, unauthorised access attempts, and anomalous behaviour detected in the network. Real-time analysis of this data is vital for maintaining network security and preventing breaches.

Energy Consumption Data: With the focus on energy efficiency in 5G/6G networks, data on power usage, energy-saving metrics, and carbon footprint is collected to optimize energy consumption and support net-zero goals.

AI/ML Training Data: Data used for training artificial intelligence (AI) and machine learning (ML) models, such as network configuration data, user behaviour patterns, and predictive maintenance logs, enables intelligent automation and optimization of network operations.

Table 2 provides a summary of the main data categories and their representative attributes.

Table 2: Representative Attributes of Data Types in B5G/6G Networks

Name	Features	Description
Network Signal Data	Timestamp, Localization Coordinates, I/Q Samples, Signal Frequency, Source Application ID, e.g., [41]	Captures low-level radio characteristics and experimental context for signal processing tasks.
Network Performance Data	Latency, Throughput, Packet Loss, Jitter, RSSI, SNR, e.g., [42]	Used for performance evaluation and QoS/QoE optimization.
User Data	Device ID, Location History, Session Duration, App Usage Statistics, e.g., [43]	Enables personalized services and behavioural modelling; must be anonymized in compliance with GDPR.
Security/Threat Data	Alert/Attack Type, Timestamp, Source IP, Destination IP, Source Port, Destination Port, Protocol used, Packet Size, e.g., [44]	Enables real-time threat detection and forensic analysis.
Energy Consumption Data	Power Usage (W), Energy Efficiency Index, Carbon Emission Estimates, Idle/Active States, e.g., [45]	Important for evaluating green networking strategies and supporting sustainability targets.
AI/ML Training Data	Labelled Logs, Feature Vectors, Model Targets, Annotations, e.g., [46]	Utilized in the development of intelligent control, prediction, and automation systems.

Based on the available literature, for all these data types there are four aspects that must be clearly defined to ensure a high-quality dataset: a) Data Granularity and Temporal Resolution, b) Data Sensitivity and Privacy Levels, c) Spatial and Contextual Metadata, and d) Format and Interoperability.

3.1.1 Data Granularity and Temporal Resolution

Different data types possess varying levels of granularity and temporal resolution, depending on their role within the network ecosystem. For instance, **network signal data**—such as in-phase and quadrature (I/Q) samples or signal strength indicators—often require high-frequency sampling at millisecond or even microsecond intervals to accurately capture rapid fluctuations in radio conditions and ensure precise signal analysis. In contrast, **energy consumption metrics** typically exhibit slower variability and are therefore collected at coarser time intervals, often aggregated over seconds or minutes. This reduces the volume of data generated, lowers processing overhead, and aligns with long-term performance monitoring or sustainability evaluations. Choosing the appropriate resolution is critical for balancing data fidelity with system efficiency.

3.1.2 Data Sensitivity and Privacy

Data attributes must be classified according to their sensitivity and privacy risk. User data, for instance, is considered highly sensitive and requires pseudonymisation or encryption prior to processing. On the other hand, network performance data may be non-sensitive but remains essential for internal network management. A proposed sensitivity model, aligned with GDPR and other relevant policies, is shown in Table 3.

Table 3: Sensitivity Classification of B5G/6G Data Types in NATWORK Project

Data Type	Sensitivity Level	Privacy Safeguards
User Data	High	Anonymization, Consent Management
Network Signal Data	Medium	Controlled Access, Aggregation
Performance Metrics	Low	Internal Use, Minimal Anonymization
Security Logs	High	Encryption, Access Logging
Energy Metrics	Low	Aggregated Reporting
AI/ML Datasets	Variable	Depending on Source Data Composition

3.1.3 Spatial and Contextual Metadata

Spatial and environmental information can play a critical role in the interpretation and application of network data, particularly in experimental and simulation contexts. This includes several key types of metadata:

Geolocation Data (latitude, longitude, altitude): Precise geographical coordinates provide essential context for analysing signal behaviour, mobility patterns, and location-based services.

Environment Descriptors (urban, rural, indoor, outdoor): Classifying the physical environment helps in understanding how propagation conditions, interference, and signal attenuation vary across different settings. Urban environments, for example, typically involve complex multipath effects, while rural areas may offer clearer line-of-sight but fewer infrastructure elements.

Node Role and Position (edge, core, relay, UE): Identifying the functional role and relative position of a node within the network architecture allows for a more accurate assessment of data flow, processing responsibilities, and performance expectations. For instance, edge nodes often handle latency-sensitive tasks, while core nodes are central to routing and data aggregation.

These contextual attributes when available can enable **scenario replication**, allowing other researchers to reproduce experimental conditions accurately; enhance **simulation fidelity**, ensuring that digital models reflect real-world dynamics; and support **performance benchmarking**, allowing fair comparisons across diverse deployment environments and network configurations.

3.1.4 Data Format and Interoperability

A fundamental requirement for developing NETWORK's federated repository for beyond B5G and 6G networks is the establishment of a common data model. In 5G and 6G networks, data models are crucial for structuring, organizing, and managing the enormous volumes of data generated by these advanced communication systems. These models provide a standardised framework for representing network entities, relationships, and interactions, enabling interoperability, efficient data processing, and seamless integration across diverse systems. Without a common data model, the integration process could face significant challenges due to potential ambiguities in the semantics of exchanged data. As a result, achieving seamless interoperability across systems is only feasible if all stakeholders adopt consistent definitions, structures, and interpretations of their data.

In general, a data model comprises of three key components:

1. **A collection of data structure types:** Defines how data is organized and represented.
2. **A set of operators or inferencing rules:** Specifies how data can be manipulated, queried, or extracted from the structures.
3. **A set of general integrity rules:** Ensures the consistency and coherence of database states or changes to those states.

Below are some key data models and their relevance in 5G/6G networks:

- **3GPP Data Models:** Standardized models for network functions, interfaces, and protocols, ensuring interoperability across 5G networks [47].
- **ETSI NFV Data Models:** Define virtualized network functions (VNFs) and network services, supporting dynamic deployment and orchestration [48].
- **YANG Data Models:** Used for network configuration and management in Software-Defined Networking (SDN) and Network Management Systems (NMS) [49].

Data must be structured using interoperable formats, particularly to support integration with AI pipelines and data-sharing platforms. Common formats include:

- **JSON/CSV/XML** for structured telemetry
- **PCAP** for raw packet traces
- **HDF5/Parquet** for large-scale ML datasets
- **5G NR-specific logs** (e.g., ASN.1-encoded messages)

3.2 Volume of data & techniques

Working with 5G/6G networks produces data in the scale of Big Data. The term refers to extremely large and diverse collections of structured, unstructured, and semi-structured data which continue to grow exponentially over time. These datasets are vast and complex in terms of volume, velocity, and variety that traditional data management systems cannot store, process, or analyse them.

Big data processing involves several stages to transform raw data into actionable insights. These stages are often referred to as the big data pipeline or data lifecycle. Here are the key stages of big data processing:

1. **Data collection**, which consists of gathering raw data from our sources, in our case the network and its component devices. Designated brokers for real-time data or APIs for historical data are used.
2. **Data cleaning**, also called pre-processing, may include handling missing values, removing duplicates and correcting errors.
3. **Data processing**, in this stage the data is processed and transformed into usable format, while performing aggregations, filtering and joining.
4. **Data storage** is the next process, storing data in scalable and accessible formats.

4 NETWORK Repository

The federated NETWORK repository serves as a key tool for collecting, storing, and sharing data among both the project consortium and the broader SNS community. It supports the provision of both real-time and historical data through a unified user interface.

4.1 Functional & Non-Functional Requirements

This section outlines the functional and non-functional requirements for the NETWORK Data Management Platform (DMP), categorised by priority and user roles:

- **"Must"** Critical requirements essential for core portal functionality. Non-compliance would severely impair usability or management.
- **"May"** Optional enhancements that improve success but are not mandatory.

Moreover, functional requirements are expressed as **user stories**, following agile development principles for clarity and standardisation. Three key roles are defined:

- **Reader:** Discovers and utilises portal data.
- **Editor:** Publishes or modifies datasets and content.
- **Administrator:** Manages technical configurations (themes, metadata schemas, etc.).

These roles reflect typical DMP user archetypes and their data access levels—not a permission system—ensuring security and alignment with SNS service implementation.

Functional Requirements are defined in Table 4, they specify the NETWORK DMP's core capabilities, derived from NETWORK's framework. The requirements specification served as prerequisites for architectural design and system implementation. This ensured traceability from design to deployment, role-based accountability and compliance with NETWORK's objectives.

Table 4: Functional Requirements of NETWORK DMP

Req_Id	Name	Description
FunReq_001	Reader Account Creation	As a reader , I must be able to create an account by using an email address, so I can be able to use the functionalities of the NETWORK DMP.
FunReq_002	Profile Management	As a reader , I must be able to generate, modify, or delete profile information to control my visibility towards other NETWORK DMP users.
FunReq_003	Catalogue Access & Search	As a reader , I must have the capability to access and search the data catalogue to find available data assets efficiently.

Req_Id	Name	Description
FunReq_004	Keyword Dataset Search	As a reader , I must be able to search for datasets by using one or more keywords.
FunReq_005	Read-Only Data Access	As a reader , I must have read-only access to data assets that I am authorized to view.
FunReq_006	Data Categorization	As a reader , I must be able to categorize data, to explore and find easier the necessary data.
FunReq_007	Machine-Readable Catalogue Export	As a reader , I must have the option to download a catalog in a format that is machine-readable, making automated use easy.
FunReq_008	Dataset Description Page Access	As a reader , I must be able to access a description page for each dataset, allowing me to retrieve information about data resources, metadata, and other documentation.
FunReq_009	Multi-Format Data Download	As a reader , I must be able to download data assets in various formats (e.g., CSV, Excel) to get the datasets in a more efficient manner.
FunReq_010	Data Visualization Tool Access	As a reader , I must have access to data visualization tools to explore and analyse data in a user-friendly way.
FunReq_011	Browser-Based Data Preview	As a reader , I may have the ability to preview data resources within their web browser, such as a short display of rows and columns or a map. This allows for a basic overview of the dataset.
FunReq_012	Dataset Review & Commenting	As a reader , I may be able to review (comments, etc.) individual data sets, to give valuable insights for other users of the NETWORK DMP.
FunReq_013	Catalogue Review & Commenting	As a reader , I may be able to review (comments, etc.) the data catalogue, to help other users of the NETWORK DMP.
FunReq_014	Visualization Configuration Saving	As a reader , I may be able to save visualized data with configuration settings, so that they can be accessed again later.
FunReq_015	Public Data Resource Proposal	As a reader , I may be able to propose data resources for the purpose of making them known to the public.
FunReq_016	Catalogue Entry Management	As an editor , I must be able of adding, modifying or deleting catalogue entries to make them accessible or inaccessible to NETWORK DMP users.

Req_Id	Name	Description
FunReq_017	Data Asset Maintenance	As an editor , I may be able to update or remove data assets, ensuring data accuracy and completeness.
FunReq_018	Catalogue Entry Privacy Control	As an editor , I must be able to classify catalogue entries as private, making sure they are not visible to the public.
FunReq_019	Group Access Control Definition	As an editor , I must be able to define data access controls for specific data assets and user groups.
FunReq_020	Data Resource Generation	As an editor , I must have the ability to generate new data resources so other NATWORK DMP users can have access to them.
FunReq_021	Proposed Resource Review	As an editor , I may be able to review user-proposed data resources, while being able to make decisions about their inclusion or exclusion.
FunReq_022	Proposed Resource Notification	As an editor , I may be able to answer to user-proposed data resources, enabling me to inform the user about their inclusion or exclusion.
FunReq_023	Data Asset Change Auditing	As an editor , I may be able to track and audit changes made to data assets for Alignment and accountability.
FunReq_024	Shared Visualization Generation	As an editor , I may be able to generate data visualizations that users have access to through the data catalogue or data resource pages.
FunReq_025	Proposed Resource Status Tracking	As an editor , I may be able to edit the status of user-proposed data resources, facilitating user notifications and maintaining records.
FunReq_026	Cross-Role Collaboration	As an editor , I may collaborate with other editors and readers by sharing data assets and insights within the platform.
FunReq_027	User Review Responses	As an editor , I may be able to respond to user reviews and so that other users may also be informed.
FunReq_028	User Comment Moderation	As an editor , I may be able to assess and authorize the display of user comments and feedback, aiming to preserve only constructive comments.
FunReq_029	Granular Access Delegation	As an editor , I may be able to determine the accessibility of my data resources, allowing me to grant access to specific registered users.

Req_Id	Name	Description
FunReq_030	Role & Permission Control	As an administrator , I must have full control over user access permissions and roles within the NATWORK DMP, to amplify overall security.
FunReq_031	User Account Management	As an administrator , I must be able to create and manage user accounts, including creating, modifying, and deleting them.
FunReq_032	Metadata Category Management	As an administrator , I must be able to generate, modify, and archive metadata categories, ameliorating the categorization of data for users.
FunReq_033	Consumer Presentation Visibility Control	As an administrator , I must be able to manage the visibility of consumer-generated data presentations for other consumers, ensuring the privacy and security of the DMP.
FunReq_034	System Performance Monitoring	As an administrator , I must be able to monitor and manage system performance and data storage capacity to ensure optimal performance.
FunReq_035	Usage & Compliance Reporting	As an administrator , I must be able to generate and export usage and compliance reports for auditing purposes.
FunReq_036	Basic UI Customization	As an administrator , I may be able to customize global settings for the data catalogue, including aspects like colour theme, branding/logo, and titling, with the aim of advancing the user experience.
FunReq_037	Advanced Layout & CSS Customization	As an administrator , I may be able to customize further global settings for the data catalogue, such as custom Cascading Style Sheets (CSS), homepage display elements and layout, displayed catalogue elements, and more, all geared towards enhancing the user experience.
FunReq_038	Default Sorting Preferences	As an administrator , I may be able to set default sorting preferences for data catalogues, aiming to enhance the user experience.
FunReq_039	Automated Backup & Disaster Recovery	As an administrator , I may set up automated backups and disaster recovery procedures to protect data integrity.
FunReq_040	User Feedback Analysis	As an administrator , I may be able to monitor and analyse user feedback to provide deeper understanding of the DMP.

Non-functional requirements define the system's operational boundaries and quality attributes. Failure to meet these specifications could render the platform ineffective. Key requirements are presented in Table 5.

Table 5: Non-functional requirements

Req_Id	Name	Description
NonFunReq_01	Open Data Compliance	Full public accessibility of open data with no restrictions.
NonFunReq_02	High Reliability	Continuous operation with minimal downtime across all scenarios.
NonFunReq_03	Maintainability	Automated recovery to predefined operational states after failures or maintenance.
NonFunReq_04	Service Availability	Guaranteed uptime for user access during specified operating periods.
NonFunReq_05	Extensible Architecture	Modular design supporting seamless integration of new features.
NonFunReq_06	System Robustness	Stable performance under variable loads and adverse conditions.
NonFunReq_07	Intuitive User Experience	Simplified interface for non-technical users.
NonFunReq_08	Comprehensive Security	Multi-layered protection against cyber threats and unauthorized access.
NonFunReq_09	Cross-Device Usability	Consistent, device-agnostic interface for all user types.
NonFunReq_10	Public Accessibility	Official city-domain hosting for standardized public access.

4.2 Abstract Architecture

The architectural depiction of the NATWORK DMP is presented in Figure 2. Each partner is required to establish a connection with the central NATWORK server, choosing between two options, according to the user system's abilities and requirements.

The first option can be through real-time connectivity (Figure 3), which can be achieved through the provisioning of data via a dedicated broker in JSON formats, which will transmit the data from the designated source (sensor, SDR, internal system etc) to the central NATWORK database. Although this requires the alignment of the respective partner's side with the requirements of

the broker, such as compliance with message format standards, and determination of an appropriate data transmission frequency, it would facilitate the interaction between partners and database, enhancing performance.

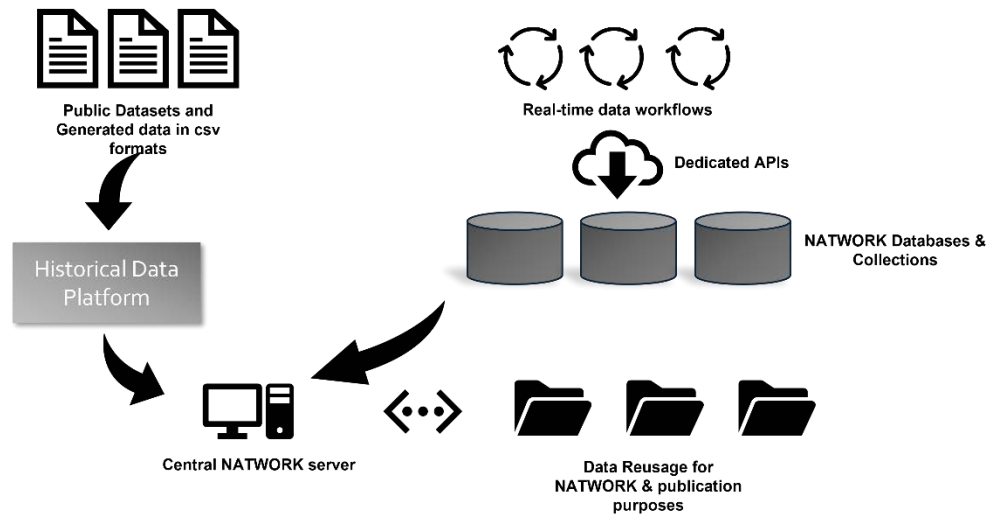


Figure 2: Abstract overview of NATWORK's repository architecture

To be able to evaluate the data provisioning process and its quality, the real-time data is monitored with a middleware visualization tool. This tool facilitates the assessment of the broker's stability and enhances the pre-analysis inspection process, making it more efficient and easier to manage. Real-time data is then be archived into the NATWORK's databases and collections with the assistance of designated APIs and subsequently forwarded to the central NATWORK server, where they can be accessed in case of future reference.

The second option available is for the data to be stored locally during their production and subsequently uploaded to the Historical Data Platform as Datasets in CSV format. Within the structure of the Historical Data Platform, operational partners provide historical data from past operational periods. They also share data message formats or simulation data, if available. This collaborative initiative aims to enable the sharing of valuable information while ensuring its security and centralising it within a single repository.

4.3 Selected tools and technology

Figure 3 presents an overview of the components dedicated to real-time data provision. Data generated by NATWORK entities are formatted as predefined JSON messages and transmitted through an MQTT broker/API to the central DMP database. The datasets are then converted into

CSV files and periodically uploaded to the historical data provision component, which also serves as the DMP user interface.

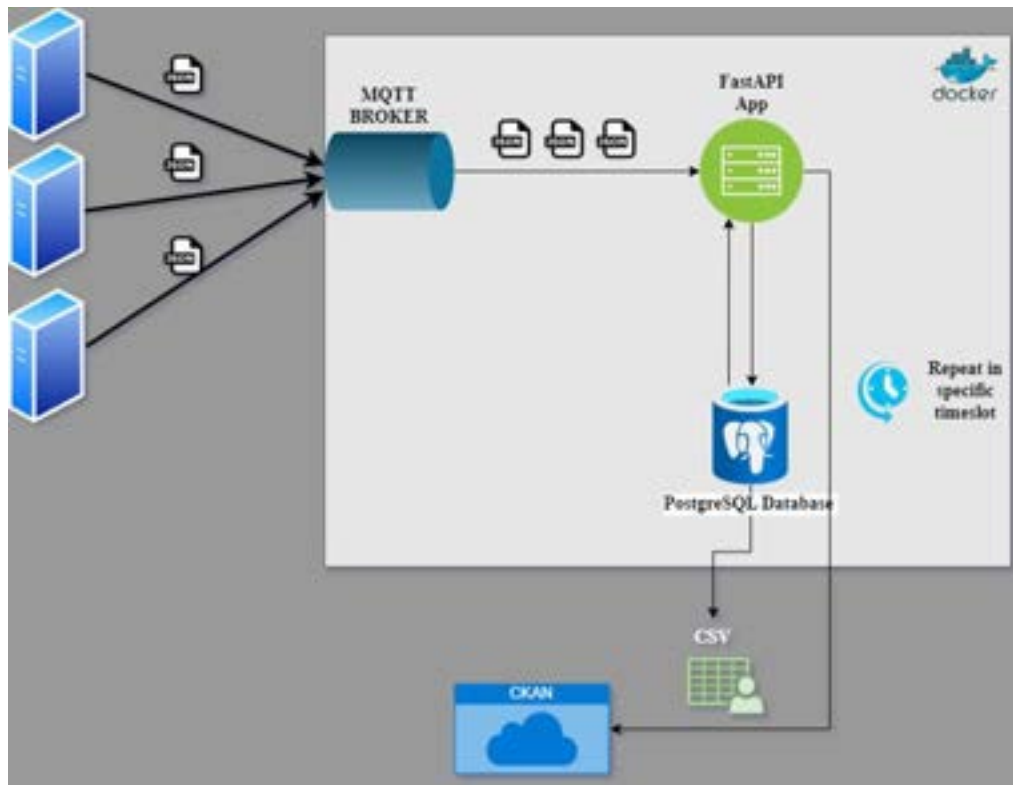


Figure 3: Overview of mechanisms used during the real-time process.

Specifically, the tools used for the NETWORK DMP are:

- **Docker** [50]: Docker is a platform that allows the development, shipping and running of applications in lightweight, portable containers. These containers package an application along with its dependencies, libraries, and configuration files, ensuring that it runs consistently across different environments. Docker simplifies the process of deploying applications by isolating them from the underlying infrastructure, making it easier to manage and scale. The dockerised-based approach allows also the capability to extend the system in multiple physical and virtual machines achieving the decentralization and the more effective management of both the tools and the captured data.
- **MQTT (Message Queuing Telemetry Transport)** [51]: MQTT is a lightweight, versatile protocol that enables efficient, reliable, and scalable communication in IoT, machine-to-machine communication and other distributed systems. It operates on a publish-subscribe model, allowing devices to communicate by publishing messages to specific topics and subscribing to receive messages on those topics. Thanks to its minimal resource requirements, MQTT is perfect for environments with limited bandwidth and power,

facilitating seamless, real-time communication between devices and applications. In terms of NATWORK, Mosquito is utilized as the MQTT component [52].

- **Fast API App:** FastAPI is a modern, high-performance web framework for building APIs (Application Programming Interfaces) with Python. It is designed to be fast, easy to use, and highly efficient, making it a popular choice for developers. It can act as a middleware layer that processes, validates, and stores data received from the broker into the database, while also providing APIs for querying and retrieving data.

The main database components are:

- **PostgreSQL [53]:** PostgreSQL is a powerful, open-source relational database known for its reliability, scalability, and advanced features. It supports SQL (Structured Query Language) for querying and managing data, making it ideal for structured data storage. PostgreSQL is highly extensible, allowing users to define custom data types and functions. Additionally, PostgreSQL supports JSON and NoSQL-like features, enabling it to handle semi-structured data.

4.4 CKAN Platform

The main tool used for historical data provision is CKAN (Comprehensive Knowledge Archive Network) [54]. CKAN is an open-source data platform that supports integration with various tools for data collection, processing, visualization, and cybersecurity. It was selected primarily due to its alignment with both the functional and non-functional requirements defined in Section 4.1. The mapping between these requirements and CKAN’s features is presented in Table 6. Each requirement is classified as: “**included**” if it is supported by the standard CKAN version, “**added**” if CERTH has developed relevant extensions, “**rejected**” if its implementation was not needed for the NATWORK project and the consortium’s final needs.

Table 6: Mapping CKAN features with Functional and Non-Functional Requirements of NATWORK DMP

Req_Id	Mapping with CKAN features	Justification for rejection
FunReq_001	Included	N/A
FunReq_002	Included	N/A
FunReq_003	Included	N/A
FunReq_004	Included	N/A
FunReq_005	Included	N/A
FunReq_006	Included	N/A
FunReq_007	Included	N/A
FunReq_008	Included	N/A
FunReq_009	Included	N/A
FunReq_010	Included	N/A

Req_Id	Mapping with CKAN features	Justification for rejection
FunReq_011	Included	N/A
FunReq_012	Included	N/A
FunReq_013	Added	N/A
FunReq_014	Rejected	This requirement was not included in the final implementation because saving visualization configurations is considered an advanced user-personalization feature. The final implementation prioritizes core data catalogue, access, download, and visualization capabilities.
FunReq_015	Rejected	This requirement was rejected because the proposal of public data resources introduces an additional review and governance workflow. This was considered outside the core scope of the final DMP implementation.
FunReq_016	Included	N/A
FunReq_017	Included	N/A
FunReq_018	Included	N/A
FunReq_019	Included	N/A
FunReq_020	Included	N/A
FunReq_021	Rejected	This requirement was not included because reviewing user-proposed resources depends on the availability of the resource proposal workflow, which was rejected. Therefore, this functionality was also excluded.
FunReq_022	Rejected	This requirement was rejected because it is directly linked to the user-proposed resource review process. Since this workflow was not implemented, notification responses about inclusion or exclusion were not required.
FunReq_023	Added	N/A
FunReq_024	Included	N/A
FunReq_025	Rejected	This requirement was not included because status tracking for proposed resources depends on a

Req_Id	Mapping with CKAN features	Justification for rejection
		dedicated proposal-management workflow. As this workflow was excluded, status tracking was also considered unnecessary for the final implementation.
FunReq_026	Added	N/A
FunReq_027	Rejected	This requirement was rejected because responding to user reviews introduces an additional interaction and moderation layer. The final implementation focuses on catalogue and data management rather than extended discussion functionalities.
FunReq_028	Rejected	This requirement was not included because comment moderation requires additional administrative processes and moderation rules. It was considered outside the minimum functional scope of the final DMP implementation.
FunReq_029	Added	N/A
FunReq_030	Included	N/A
FunReq_031	Included	N/A
FunReq_032	Included	N/A
FunReq_033	Included	N/A
FunReq_034	Included	N/A
FunReq_035	Included	N/A
FunReq_036	Included	N/A
FunReq_037	Rejected	This requirement was rejected because advanced layout and CSS customization goes beyond the basic configuration needs of the platform. The final implementation supports standard customization options, while deeper interface modifications were considered non-essential.
FunReq_038	Rejected	This requirement was not included because default sorting preferences are a usability enhancement rather than a core functional need. The

Req_Id	Mapping with CKAN features	Justification for rejection
		final implementation relies on the standard catalogue browsing and sorting mechanisms already available in the platform.
FunReq_039	Added	N/A
FunReq_040	Added	N/A
NonFunReq_01	Added	N/A
NonFunReq_02	Added	N/A
NonFunReq_03	Added	N/A
NonFunReq_04	Added	N/A
NonFunReq_05	Included	N/A
NonFunReq_06	Included	N/A
NonFunReq_07	Included	N/A
NonFunReq_08	Added	N/A
NonFunReq_09	Added	N/A
NonFunReq_10	Included	N/A

The following presents the main functionalities of the CKAN platform used in the NATWORK DMP, together with their development status. The platform has been publicly available since the first year of the project at: <https://network-data-portal.iti.gr/>.

4.4.1 Registration

To be able to use and have access to the platform, users must create an account or log in to their current one. To do that, the submission of specific information is required, including:

- **Username:** The registrant is required to select a username including solely alphanumeric characters, hyphens (-), and underscores (_).
- **Full Name:** This name will be publicly displayed on the user's profile.
- **E-Mail Address:** This information is to remain confidential and will not be disclosed to other platform users.
- **Password:** The registrant must create the same secured password in both designated fields.

After a successful login, the reader will subsequently have access to the platform. The respective screen is illustrated in Figure 4. It is worth noting that the option to retain this pertinent information for registered readers is available but not mandatory.



Figure 4: The introduction page of NATWORK CKAN

4.4.2 Organisation

Organisations serve as the primary users that have access to, as well as the ability to create and modify datasets within CKAN. Within this system, each dataset is associated with a single organization, and each organization is granted the authority to regulate access to its datasets. The respective page is shown in Figure 5.



Figure 5: The Organization menu of NETWORK CKAN

Datasets are classified as either public or private within the system. Public datasets are openly accessible to all registered CKAN users. On the other hand, private datasets are restricted to authenticated users who are also designated members of the associated organization. Notably, private datasets are excluded from search results unless the user (or an API-authenticated request) has explicit authorization to view them.

After joining an organisation, the user is assigned one of three distinct roles by an organisation administrator: member, editor, or admin.

A **member** has the ability to:

- **Access the organisation's private datasets.**

An **editor** enjoys the privileges of a member and, in addition, possesses the authority to:

- **Upload new datasets** to the organization.
- **Modify or delete** any of the organization's datasets.
- **Alter the visibility status** of datasets, switching them between public and private.
- **Versioning:** Track changes with CKAN's revision history or integrate Git for version control.

An organisation **administrator** is granted the full range of privileges held by an editor, and, in addition, has the power to:

- **Include users** within the organization, with the option to **assign their role** as a member, editor, or admin.
- **Modify the roles** of any user within the organization, including other administrators.
- **Remove members, editors, or fellow administrators** from the organization.
- **Modify the organization's attributes**, such as its title, description, or image.
- **Delete** the organization itself.

When a user initiates the creation of a new organization, they automatically assume the role of administrator for that organization. Specific tools for the protection of the platform from the massive creation of fake accounts will also be integrated.

4.4.3 Group and Organization

CKAN Groups serve as a valuable tool for the creation and management of dataset collections. These collections can group datasets by project, team, theme, or simply help users organize and discover their published content. The page of NATWORK CKAN groups is illustrated in Figure 6.

A key distinction between Organisations and Groups is primarily rooted in their designated authorization. Organisation editors/admins can create new datasets and modify existing ones within their Organisations. Group editors/admins, however, are restricted to managing dataset-group associations - they can add or remove datasets from the group but cannot create new datasets or edit community content. Furthermore, only Organisations can contain private datasets, with access limited to their designated members.

Within CKAN, Organizations serve as the primary governance structure for dataset management. They determine which users can create, modify, and publish datasets in the system. Every dataset

must belong to exactly one Organization. Users with editor or administrator privileges within an Organization gain significant capabilities: they can initiate new datasets within their organizational domain and maintain editorial control over all datasets owned by that Organization.

Groups fulfil a distinct but complementary function as organizational tools for dataset discovery and curation. Unlike Organizations, Groups cannot create or modify content but instead serve as thematic containers that logically organize existing datasets. Group administrators can:

1. Assemble relevant datasets from across the platform
2. Remove datasets from their collections
3. Create thematic groupings (by subject, project, or data type)

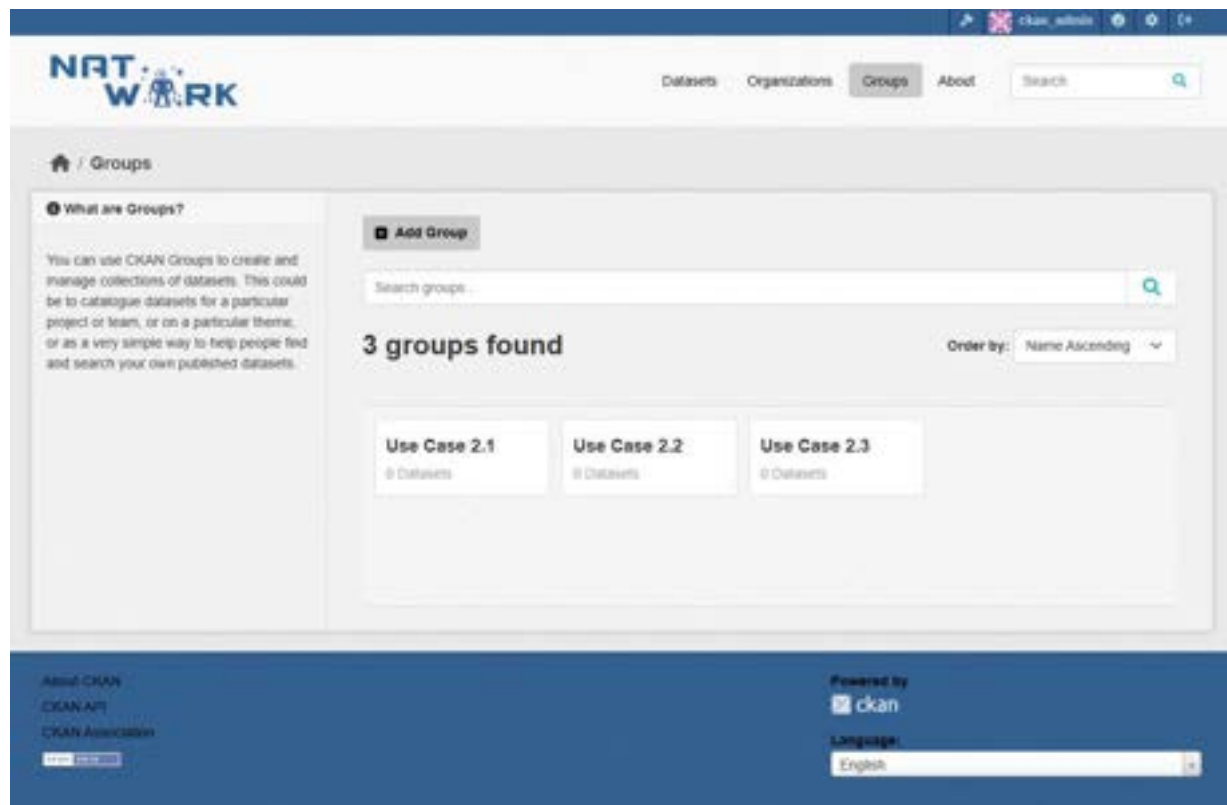


Figure 6: The Group menu of NATWORK CKAN

Several critical distinctions govern these entities:

4. **Creation Privileges:** Only Organisations can create new datasets.
5. **Modification Rights:** Organisational editors can alter dataset contents, while Group administrators can only manage collection membership.
6. **Visibility Management:** Organisations can host private datasets with restricted access, while Groups inherently work with publicly available content.

7. **User Roles:** Groups typically serve as data consumers and analysts rather than publishers

This dual structure enables both rigorous access control (through Organisations) and flexible content organisation (through Groups), making CKAN adaptable to various institutional needs while maintaining clear data governance boundaries. The system ensures publishers maintain control over their datasets while allowing the community to organize information in ways that enhance discoverability and usability.

4.4.4 Dataset

While adding a new dataset, the information provided must include the following attributes:

- **Title:** The dataset should have a distinctive title across the platform.
- **Description:** A comprehensive description of the dataset should be included, covering information such as the data's source and any additional information that users will require when utilizing the dataset.
- **Tags:** Tags serve as powerful metadata tools for organizing, classifying, and retrieving information efficiently. Users have the option to add tags that facilitate data discovery and establish associations with related datasets. Tags should be separated by hitting the <return> key. Any incorrectly entered tags can be deleted using the provided delete button prior to saving the dataset.
- **License:** It is crucial to specify the licensing information of each dataset so that users are informed about the permitted legal use of the data. The platform supports a predefined set of licenses, which can be selected through a dropdown menu when a dataset is uploaded. Each partner may select the most appropriate license for its own dataset, according to its internal policies and the intended use of the data. If a required license is not included in the supported list, the partner must contact the site administrator for further assistance.
- **Organisation:** If the user is a member of an organization, they can select the appropriate organisation from a dropdown menu to claim ownership of the dataset. It is crucial to ensure that the default selection is the correct organisation before proceeding. The manner that a dataset is displayed within NATWORK CKAN platform is shown in Figure 7.



Figure 7: The Dataset menu of NATWORK CKAN

After this initial input, CKAN proceeds to display the **"Additional Data"** screen, which includes the following elements:

- **Visibility:** Users must specify the visibility of the dataset. A "Public" dataset is accessible to any user of the site, while a "Private" dataset can only be viewed by members of the organization that owns the dataset. Private datasets will not appear in searches conducted by other users.
- **Author:** Provide the name of the individual or organization responsible for producing the data.
- **Author email:** Enter an email address for the author, which users can use to send queries related to the dataset.
- **Maintainer / Maintainer email:** If necessary, provide details for a second person responsible for maintaining the dataset, including their name and email address.
- **Custom Fields:** Users can include additional fields for the dataset by specifying the field name and its corresponding value in this section.

For existing datasets, users have the ability to explore them by either previewing or downloading the dataset.

4.4.5 Visualisation

CKAN's resource views enable dynamic visualisation of dataset contents through tables, charts, maps, and other formats. Key features include:

- **Multiple Views per Resource:** A single dataset can support simultaneous visualisations (e.g., a tabular grid and bar chart of the same data).
- **Editorial Customisation:** Dataset editors' control which views to display, their order, and configuration (e.g., adjusting chart parameters).
- **Embeddable Visualisations:** Views can be embedded externally (e.g., on websites or dashboards) for broader sharing.

CKAN offers diverse visualisation capabilities through a plugin-based architecture. These specialised plugins, when activated by system administrators, enable flexible data presentation options. Key aspects include:

1. **Plugin Framework:** Visualisation types are implemented as modular plugins that can be selectively enabled for each CKAN instance.
2. **Administrative Control:** Administrators can:
 - Set **default view preferences**
 - **Match visualisation** types to data formats
 - Configure **display parameters**
3. **Automatic Format Detection:** The system intelligently pairs data types with appropriate visualisations:
 - **Tabular** data with **Grid/table views** (when stored in Datastore)
 - **Geospatial data**

These resource views and plugins enhance CKAN's capability to effectively visualise and present data in various formats, catering to the specific requirements of users and datasets.

4.5 NETWORK Ecosystem

A first description of the NETWORK ecosystem regarding the data generated and shared was given in D1.2 – “*Quality Assurance, Risk Assessment, Data Management Plan, Ethics & Regulatory issues.r1*”. Moreover, the first description of the NETWORK DMP was given in D4.5 – “*NATWORK Federated repository for B5G/6G networks (cross layer AI-based attacks Datasets).r1*”. The exact input by each partner is included in Annex A.1, while the more detailed data-handling information collected for the NETWORK repository is included in Annex A.2.

Up to this period, the NETWORK ecosystem has progressed from an initial mapping of partner data practices to an operational data-sharing environment supported by the NETWORK DMP. The ecosystem now includes the consortium partners acting as data providers, the CKAN-based

repository acting as the common data catalogue and access point, and the project WPs acting as data consumers and producers for AI-based security, validation, integration, and exploitation activities.

The cumulative conclusions at this stage are the following:

- **Data Sources:** The repository ecosystem includes mainly datasets generated during NETWORK activities. These datasets cover several NETWORK technical activities, including jamming detection, channel estimation, jamming mitigation, Kubernetes anomaly detection, CNF live migration, 5G traffic analysis, DDoS detection, OT/ICS attacks, denial-of-sustainability attacks, CTI, and multimodal 5G network/container monitoring.
- **Repository Operation:** The NETWORK DMP provides the common interface for dataset registration, metadata description, controlled access, dataset preview, visualisation, and download. The platform supports the organisation of datasets through CKAN entities such as organisations, groups, datasets, and resources.
- **Documentation & Quality:** Dataset usability is supported through structured descriptions, metadata fields, file-structure information, statistical summaries, and parameter tables. This improves traceability and enables partners to understand the origin, structure, and intended use of each dataset.
- **Data Formats and Interoperability:** The shared datasets use heterogeneous formats, including CSV files, PCAP traces, structured logs, ML-ready files, JSON/STIX-based CTI information, and other domain-specific structures. The repository therefore acts as a harmonised catalogue layer that preserves the native format of each dataset while improving discoverability and reuse.
- **Research Outcomes:** The collected datasets support NETWORK's AI-based security research, cross-layer attack analysis, model training, validation, and technical integration activities. Where allowed by access and confidentiality constraints, selected datasets may also support wider dissemination through publications or future sharing within the SNS-JU community.
- **Cost and Storage:** Data storage remains mainly supported by partners' own facilities and by the NETWORK DMP infrastructure. This approach limits additional cost while allowing the repository to provide a common catalogue and access mechanism.
- **Data Security:** No datasets contain personal data. Datasets with potential sensitivity are handled through controlled access, partner-side restrictions, or institutional repositories when needed.

- **Ethics and Personal Data:** No personal data processing is reported in the shared datasets, and no ethics or legal issues have been identified by the consortium partners for the data currently described in the repository.

5 Cross-Layer AI-based Attacks Datasets

This section provides a classification of the attacks employed, as elaborated under T4.1. Moreover, it describes briefly each attack associated with datasets, generated by the different testbeds of the members of the consortium (as presented in Annex A.2).

5.1 Classification of attacks

As part of the T4.1, the NATWORK project continues to develop the attack datasets. NATWORK partners are working on various attacks targeting HyperText Transfer Protocol version 2 (HTTP/2), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Access and Mobility Function (AMF), among others. The partners have classified their attacks to facilitate data collection activities by uploading the generated datasets to the NATWORK DMP. A glimpse of the structure can be seen in Table 7.

Table 7: Classification of attacks

Attack Name	Reported Subsection	Type of attack	Target Protocol	Target
Denial-of-Service (DoS) attacks and port scans		Denial of Service	ICMP/UDP/TCP/HTTP	Server deployed in the 5GTN MEC environment
AI-DoS attack	5.2	Denial of Service	HTTP2/PFCP /GTP-U	AMF and UPF of CERTH's testbed
SCTP Session Flooding		Denial of Service	SCTP	AMF of CERTH's testbed
HTTP2 Ping Flooding attack		Denial of Service	HTTP2	SMF of CERTH's testbed
HTTP2 Slow Get Flooding attack		Denial of Service	HTTP2	SMF of CERTH's testbed
TCP SYN Flooding		Denial of Service	TCP	AMF or SMF of CERTH's testbed
UDP Flooding		Denial of Service	UDP	UPF of CERTH's testbed
SSH brute force attack		SSH Brute Force	SSH	CERTH testbed
OT/ICS attacks (Log4Shell, Brute Force attack)	5.4	SSH brute force	SSH	MONT's testbed
		Log4Shell	TCP / HTTP	
Inter-container DoS attacks	5.9Inter-Container Attacks in Kubernetes	Denial of Service	HTTP	ZHAW testbed

Attack Name	Reported Subsection	Type of attack	Target Protocol	Target
Inter-container Port Scan attacks	5.9	Scanning	TCP	ZHAW testbed
Inter-container ZAP Scan attacks	5.9	Scanning	HTTP	ZHAW testbed
DoSt Attack	5.6	App HTTP	HTTP	UEssex Testbed
Mirai botnet attack	5.7	UDP generic	UDP	HES-SO's testbed
		TCP SYN	TCP	
		App HTTP	HTTP	
Jamming Attack	5.3	Jamming	IEEE 802.11.p	CERTH Attack testbed
Eavesdropping Attack	5.5	Passive Eavesdropping	PL	GRAD Lab5G testbed
5G Interface Flooding Attacks	5.8	Denial of Service	SCTP / GTP-UDP / GTP-TCP / GTP-ICMP / PFCP / TCP	N2, N3, N4 and N6 interfaces of UZH's emulated hybrid 5G testbed

The different attacks and associated datasets are the following:

DoS attacks and port scans: 5G-NIDD [55] is a fully labelled dataset built on a functional 5G test network that can be used by those who develop and test AI/ML solutions. 5G-NIDD contains data extracted from a 5G testbed. The testbed is attached to 5G Test Network in University of Oulu, Finland. The data are extracted from two base stations, each having an attacker node, several benign 5G users. The attacker nodes attack the server deployed in 5GTN MEC environment. The attack scenarios include DoS attacks and port scans. Under DoS attacks, the dataset contains ICMP Flood, UDP Flood, SYN Flood, HTTP Flood, and Slowrate DoS. Under port scans, the dataset contains SYN Scan, TCP Connect Scan, and UDP Scan. The dataset files are available in different formats. These files belong to a series of post-processing steps from network capture (pcapng) to encoded data (csv) ready to feed ML algorithms.

AI-DoS attack: This use case aims to formulate a variety of AI-based Attacks dedicated to B5G/6G infrastructure with enhanced capabilities. Indicatively, an AI-powered DoS attack will use Reinforcement Learning algorithms to identify and exploit weaknesses in the network that can be used to flood it with traffic or cause it to crash. Also, a protocol fuzzer is utilised to identify vulnerabilities in the network protocols and test them for potential exploitation. Finally, it has

the ability to apply techniques that will bypass IDS detection making it more effective in attacks. An overview of the tool that will be used is shown in section 5.2.

SCTP Session Flooding: A DoS attack targeting the Stream Control Transmission Protocol (SCTP) is a cyberattack strategy that disrupts an existing SCTP session or connection, hindering communication between two services. There are several techniques attackers might employ to achieve this disruption [56]. In this use case, the attack targets the SCTP protocol by attempting to set up numerous SCTP sessions.

HTTP2 Ping Flooding attack: The HTTP/2 Ping Flooding attack exploits the "PING" frame feature of the HTTP/2 protocol, which is designed to measure the minimum round-trip time between the client and the server [57]. Attackers execute this type of attack by dispatching an excessive number of PING frames to the target server in quick succession. The primary goal of this attack is to deplete the server's resources. By inundating the server with these frames, the attacker forces it to allocate significant processing capacity and bandwidth to handle and respond to each PING request. This excessive demand on the server's resources can lead to a slowdown or even a complete halt in serving legitimate requests, effectively disrupting the service for genuine users.

HTTP2 Slow Get Flooding attack: The HTTP/2 Slow Get Flooding attack is a DoS variant tailored to exploit the HTTP/2 protocol. In this attack, the malicious user initiates numerous connections to the target server and deliberately keeps these connections active for an extended period by gradually sending incomplete requests [57]. The strategy behind this slow-paced submission of requests is to occupy server resources indefinitely, preventing the server from closing the connection due to inactivity.

TCP SYN Flooding: The TCP Flooding Attack is designed to compromise network integrity by flooding the target with an excessive number of TCP packets. This attack aims to exhaust the resources of a server, causing DoS by sending a large number of TCP SYN packets, a technique known as SYN flooding [58].

UDP Flooding: The UDP Flooding attack leverages the connectionless nature of the User Datagram Protocol (UDP) to overwhelm a system with an excessive number of session requests. This can cause service degradation or complete unavailability, particularly impacting 5G network components such as User Plane Function (UPF) that depend on the UDP protocol for operations [59].

SSH brute force attack: SSH Brute force is an attack method where an attacker methodically attempts to gain unauthorized access to a server by cycling through numerous combinations of usernames and passwords. This process continues until the attacker successfully discovers the correct credentials, enabling them to log in to the server [60].

OT/ICS attacks (Log4Shell, Brute Force attack): OT/ICS attacks provide a simulation of an operational disruption in an OT/ICS network. The setup involves configuring a network with a vulnerable web application, OpenPLC, SCADA, and Tomcat services. The scenario includes two key vulnerabilities: CVE-2021-44228 (Log4Shell) and CVE-2009-3548 (weak credentials in Apache Tomcat). These vulnerabilities allow attackers to gain remote code execution through Log4j exploits [61] and WAR file uploads, respectively, affecting industrial control systems using OpenPLC and ScadaBR.

Data Exfiltration: Using side-channel attacks, a malicious actor might attempt to fetch confidential data from a container or VM on the cloud infrastructure. In case of success, the privacy of the data is compromised. For instance, some old Python versions (3.8 or 3.9.15) had a vulnerability in the TLS implementation that raised a possibility of data exfiltration, as explained in CVE-2023-40217. Mitigation: A proactive MTD action can make it difficult for an attacker to perform data exfiltration and a reaction MTD action can cut the exfiltration process[62].

Malware Infection: Some vulnerabilities of common software components, due to using outdated versions, may enable attackers to perform remote code execution, which will allow them to get a control of the encapsulating CNFs/VNFs with malware infection. Once a container/VM is infected, a stateless migration can remove the threat. Mitigation: A stateless MTD action can remove the malware by re-instantiating a clean container from scratch[63].

Mirai botnet attack: Although Mirai first appeared several years ago, it remains a relevant threat for ISPs because its variants continue to exploit vulnerable IoT devices and generate large-scale DDoS traffic. Mirai-based malware typically compromises devices with weak credentials or exposed services, enrolls them into a botnet, and uses them to overwhelm network infrastructure, servers, or service providers. Moreover, the Mirai codebase has evolved into several variants with different scanning, infection, and attack strategies. To capture this evolving threat landscape, we created datasets that reflect different types of Mirai-based attacks [64].

Jamming attack: This jamming dataset contains time-domain signal representations (I/Q samples) for the IEEE 802.11p protocol, covering all supported modulation schemes: BPSK, QPSK, 16-QAM, and 64-QAM. It includes two scenarios: clean signal and jamming attack. The data was generated using an SDR setup with varying distances between transmitter, receiver, and jammer, and across different SNR levels [65].

Eavesdropping attack: The PKG dataset “PKG_SubTHz_OFDM_TDD_dataset” added on CKAN NETWORK Repository contains physical-layer channel measurements passively recorded by Eve (attacker) during the exchange between two legitimate nodes (Alice and Bob). The data is collected at varying distances between Eve and the main nodes, ranging from within two to six wavelengths up to spatially decorrelated positions, and across different SNR levels, allowing

analysis of the conditional min-entropy leakage and key reconstruction feasibility from Eve's perspective.

DoSt Attack: A DoSt attack aims to disrupt services by overwhelming the target's resources with oscillating requests. This action leads to resource strain or failure in Cloud-Native Functions (CNFs). This type of attack initiates a surge in traffic, causing scaling of CNFs, which can degrade performance and impact service availability. For example, in 6G environments, such attacks may trigger unintended CNF scaling operations, leading to excessive resource usage and potentially crashing critical system functions. Mitigation: Proactively monitoring 6G services and leveraging Anomaly Detection systems for early detection. At the same time, orchestrators can apply scaling controls or distribute the services using Cyber Threat Intelligence (CTI) to maintain service stability [66].

5G Interface Flooding Attacks: These attacks target key 5G network interfaces by generating high volumes of malicious traffic that overload control-plane or user-plane communication. The dataset was collected in an emulated hybrid 5G testbed and includes both benign traffic and flooding attacks against the N2, N3, N4, and N6 interfaces. Each interface is associated with a different part of the 5G architecture, so attacks against them can disrupt signalling, data forwarding, session control, or external network connectivity.

5.2 AI-DoS attack Tool - GORGO

As network infrastructures evolve toward 5G and 6G paradigms, their increasing complexity and interconnectivity necessitate more advanced cybersecurity assessment methodologies. Traditional penetration testing tools often fail to simulate the dynamic and adaptive nature of real-world adversaries. In the following section GORGO, an AI-powered DoS attack tool is introduced. GORGO is designed to autonomously identify and exploit vulnerabilities in next-generation networks, providing a robust evaluation framework for system resilience.

5.2.1 System Architecture

GORGO employs a Reinforcement Learning framework, leveraging Deep Q-Learning agents capable of simulating and orchestrating DoS attacks. These agents do not rely on pre-configured datasets or manual tuning; instead, they learn and evolve through ongoing interaction with their environment. This allows GORGO to model highly adaptive adversarial behaviours, aligning with the dynamic nature of real-world cyber threats. Its design enables the tool to respond to network conditions in real time, altering its methods of attack as the scenario evolves.

The technical foundation of GORGO supports an extensive range of features that contribute to its effectiveness as a penetration testing instrument. The system is not limited to a single protocol, as it can launch attacks targeting transport layer protocols such as TCP, UDP, or SCTP

as well as application layer protocols such as PFCP, GTP-U, HTTP2 depending on the target and scenario. Its intelligence enables it to determine the most disruptive strategy based on the specific service or network component under attack. For instance, when targeting critical elements such as AMF or UPF, GORGO adjusts its tactics accordingly.

The tool can conduct protocol-level fuzzing: it autonomously generates and manipulates network packets to expose vulnerabilities that may not be detectable with standard testing tools. Moreover, the system facilitates collaborative learning among multiple agents, orchestrating synchronized attacks that are more difficult to detect and mitigate. By continually analysing the impact of its actions on QoS metrics, such as latency and throughput, GORGO refines its strategy to enhance the overall effectiveness of the attack.

GORGO is a cooperative MARL framework designed to autonomously discover DoS vulnerability patterns in networked systems. The baseline architecture uses nine independent Deep Q-Network (DQN) agents with a shared QoS-based reward signal. Agents are partitioned into two groups:

- Transport agents (Agents 1–5): select protocol and port, session-handling strategy, thread concurrency, requests per thread, and request rate.
- Payload agents (Agents 6–9): control payload content through fuzzed byte sequences, seed values, encoding patterns, and structural variants.

This baseline is effective for socket-level stress generation, but 5Gcore security evaluation requires application-layer support: high-impact faults in HTTP/2, GTP-U, and PFCP arise from parser logic, state-machine transitions, and procedure semantics that are not exposed by transport-only mutations.

Extending the baseline to application protocols introduces three main challenges:

1. Protocol-specific fuzzing semantics: each protocol has different valid message families (e.g., HTTP methods, GTP-U message types, PFCP procedures), so naive shared payload actions become semantically noisy.
2. Statefulness and sequencing constraints: some actions are only valid after prior exchanges (e.g., PFCP session operations after association setup), requiring state-aware control.
3. Payload-action incompatibilities: many joint actions are mutually inconsistent (e.g., nobody HTTP methods with large payload actions), which can collapse learning if executed as-is.

These issues are addressed while preserving the original cooperative MARL structure through two mechanisms. First, conditional action masking uses shared state to remove invalid actions

before sampling, depending on the target protocol. For example, if the target protocol is HTTP2 then all non-HTTP2 actions are removed from the action space of each agent. Second, post-selection conflict resolution repairs semantically incompatible joint actions prior to transmission and applies a small coordination penalty to the yielding agent, enabling stable learning without discarding exploratory behaviour. This is implemented by defining a dominant agent per-protocol, whose action selection is prioritized. The remaining payload agents have to adjust their selections to be compatible with the dominant agent's selection.

5.2.1.1 HTTP/2 Extension

HTTP/2 is the primary transport for all 5G SBA interfaces. Unlike raw TCP fuzzing, effective HTTP/2 DoS requires semantic understanding of methods, headers, frame types, and stream multiplexing.

Dominant agent: For the HTTP extension, Agent 8 is considered the dominant agent because it selects the HTTP method to be used. The other related agents then take this selection into account before making their own decisions. This ensures that their actions remain aligned with the method selected by Agent 8 and that the overall HTTP configuration is consistent.

Conflict resolution: If Agent 8 selects GET or HEAD, Agent 6's body-size choice is corrected to NONE (RFC 7231 prohibits request bodies on these methods). If the corrected body size is NONE, Agent 7 receives δ_{conflict} since a payload seed is irrelevant without a body.

Attack vectors enabled: Stream exhaustion (heavy multiplexing with sufficient requests per thread to open a full concurrent batch per connection), body flooding (large bodies saturating server I/O buffers), HPACK decompression overload (many/large header blocks), and parser stress via invalid methods or malformed header blocks.

Table 8: HTTP2 extension per-agent action space versus baseline per-agent action space

Agent	Baseline Action Space	HTTP2 Action Space
Agent 1	Port selection	Port Selection
Agent 2	Session action (keep open / close)	HTTP2 Multiplexing level - Streams dispatched concurrently within each batch on a connection
Agent 3	Number of threads	Number of concurrent persistent HTTP2 connections
Agent 4	Number of requests per thread	Number of streams per HTTP2 connection
Agent 5	Time between requests	Time between stream batches
Agent 6	Number of characters per payload	Request body size
Agent 7	Payload seed	Payload seed
Agent 8	Unicode category	HTTP2 method

Agent	Baseline Action Space	HTTP2 Action Space
Agent 9	Range of Unicode characters	HTTP2 header formation strategy

5.2.1.2 GTP-U Extension

The N3 interface carries all subscriber data traffic. Disrupting the UPF’s GTP-U processing pipeline can degrade or sever service for all active sessions simultaneously.

Dominant agent: Agent 6 (message type) is the dominant agent for the GTP-U extension. Its selection — g_pdu, echo_request, error_indication, or end_marker— is written into the dominant_agent_action slot before agents 7–9 choose their inner-payload and header-encoding actions.

Conflict resolution: Control-plane message types (echo_request, error_indication, end_marker) carry no user-plane payload; Agent 7’s inner-payload choice is corrected to none. An End Marker is meaningful only once per tunnel teardown sequence; Agent 4’s requests-per-thread is corrected to 1 when this message type is active.

Attack vectors enabled: TEID exhaustion via sequential Agent8 forcing session table entries at the UPF, large inner IP payloads that stress forwarding and may trigger fragmentation depending on MTU, reassembly/parser stress via malformed_ip encapsulation, and GTP-U header field corruption.

Table 9: GTP-U extension per-agent action space versus baseline per-agent action space

Agent	Baseline Action Space	GTP-U Action Space
Agent 1	Port selection	Port Selection
Agent 2	Session action (keep open / close)	Session action (keep open / close)
Agent 3	Number of threads	Number of threads
Agent 4	Number of requests per thread	GTP-U PDUs sent sequentially by each thread
Agent 5	Time between requests	Time between successive PDUs within a thread
Agent 6	Number of characters per payload	GTP-U opcode
Agent 7	Payload seed	Encapsulated payload strategy
Agent 8	Unicode category	TEID strategy
Agent 9	Range of Unicode characters	Header encoding

5.2.1.3 FCP Extension

The SMF/UPF control channel is a single point of failure for session management. PFCP flooding or session-table exhaustion disables the UPF’s ability to process any newsubscriber sessions.

Dominant agent: Agent 6 (PFCP procedure) is the dominant agent. Its procedure selection (association_setup, heartbeat, session_establishment, etc.) is written into the dominant_agent_action slot before Agents 7–9 choose their IE-population and TLV-encoding actions. Agent 2’s action space is additionally extended with pfcpl_association_strategy actions, controlling association reuse and orphan-association patterns across rounds.

Conflict resolution: large_ruleset applies only to session-level messages; corrected to minimal for node-level procedures. Sequential or collision F-SEID strategies are corrected to random for node-level procedures. The orphan association leaves the association half-open, so the active procedure is corrected to heartbeat.

Attack vectors enabled: Session-table exhaustion via sequential F-SEID embedded consistently in the PFCP header and F-SEID IE with session_establishment; Heartbeat event-loop exhaustion; session-deletion volumetric stress; association-level churn (release_cycle); rule engine stress (large_ruleset); and TLV parser attacks (zero_padded, length_overflow).

Table 10: PFCP extension per-agent action space versus baseline per-agent action space

Agent	Baseline Action Space	PFCP Action Space
Agent 1	Port selection	Port Selection
Agent 2	Session action (keep open / close)	Association strategy
Agent 3	Number of threads	PFCP messages sent sequentially by each thread.
Agent 4	Number of requests per thread	GTP-U PDUs sent sequentially by each thread
Agent 5	Time between requests	Time between successive messages within a thread
Agent 6	Number of characters per payload	PFCP procedure type
Agent 7	Payload seed	Information Element (IE) strategy
Agent 8	Unicode category	F-SEID strategy
Agent 9	Range of Unicode characters	Type-Length-Value (TLV) encoding

5.2.2 Experimental Setup and validation

Experiments run on a single Ubuntu 24.04 host with kernel gtp5g v0.9.5 loaded for GTP-U offload. The 5G core is free5gc-compose v4.2.0 1, deployed via Docker Compose (NRF, AMF, SMF, UDM, AUSF, UDR, PCF, CHF, NEF, UPF, WebUI containers) on a br-free5gc bridge in subnet 10.100.200.0/24. UERANSIM runs as an additional container on the same bridge, acting as both gNB and UE simulator.

¹ <https://github.com/free5gc/free5gc-compose>

Each protocol extension is validated by a two-phase procedure. Phase 1 trains the nine-agent team for 100 episodes (six rounds each) with the protocol-aware action blocks active and always writes the resulting checkpoints; these are the checkpoints later used for dataset capture. Phase 2 repeats the same hyper-parameters but agents emit unstructured TCP/UDP fuzz as in the Gorgo baseline: it is a performance baseline only — no checkpoint is saved. Comparing the two phases isolates the contribution of protocol awareness from the contribution of the nine-agent control loop itself. The protocol-aware advantage tracks the strictness of the application-layer parser at the target. HTTP/2 gains the most (+36.5%): the SBI rejects malformed framing outright, so unstructured TCP fuzz wastes most bursts on syntactically invalid segments while protocol-correct HEADERS/DATA patterns are accepted and processed. GTP-U gains only +5.2% because the UPF parses raw UDP at the GTP layer but silently drops most malformed packets either way — the marginal value of protocol-correct headers over random bytes is small, and dominant agent inspections show that the team most often converged on g_pdu with small_ip payload at sequential TEIDs (i.e., TEID exhaustion). PFCP is the outlier: protocol-aware PFCP scored –62.9% below generic UDP (R = 10.18 vs. 27.43), with the generic side showing extreme variance (max = 188.07, min<0). Generic UDP fuzzing on port 8805 instead occasionally hits a TLV-parser pathology (the 188 outlier was a single episode where the UPF stalled for tens of seconds) and those jackpots dominate the cumulative mean. Protocol-aware PFCP still generates the structured traffic the dataset needs (real session_establishment, heartbeat, large_ruleset bursts) rather than random bytes that happen to crash one parser.

Table 11 :Comparison of protocol aware vs generic transport layer fuzzing mean per-episode reward over 100 episodes

Protocol	Phase 1 mean reward	Phase 2 mean reward	Δ_{abs}	$\Delta\%$
HTTP2	30.40	22.28	+8.12	+36.5%
GTP-U	19.48	18.52	+0.96	+5.2%
PFCP	10.18	27.43	-17.25	-62.9%

5.3 Jamming Attack

As wireless communication systems evolve toward 5G and 6G paradigms, their increased reliance on the physical layer makes them vulnerable to jamming attacks, which can severely disrupt connectivity and degrade Quality-of-Service. Simulated testing approaches often struggle to reproduce realistic and adaptive interference scenarios. To address this, this section presents an SDR-based jamming framework as a representative approach for generating attack scenarios, together with the associated datasets. The developed testbed allows the selection across multiple jamming strategies under realistic operating conditions, supporting the evaluation of system robustness.

5.3.1 System Architecture

The proposed framework is built upon an SDR-based architecture, enabling flexible and programmable control over both legitimate transmissions and adversarial interference. The system is designed to support vehicular and wireless communication protocols, while its modular structure allows extension to different technologies and frequency bands. The overall signal processing pipeline, including transmission, reception, and jamming functionalities, is implemented using the GNU Radio framework.

The testbed consists of three main components: a legitimate transmitter–receiver pair and a dedicated jammer node (Figure 8). The transmitter generates and transmits signal packets through a GNU Radio-based processing chain, while the receiver continuously captures raw baseband in-phase and quadrature (I/Q) samples. The jammer node is implemented using an SDR platform, allowing the adjustment of transmission parameters according to the selected attack scenario. In the examined setup, the jammer emits white noise signals to interfere with the legitimate transmission.

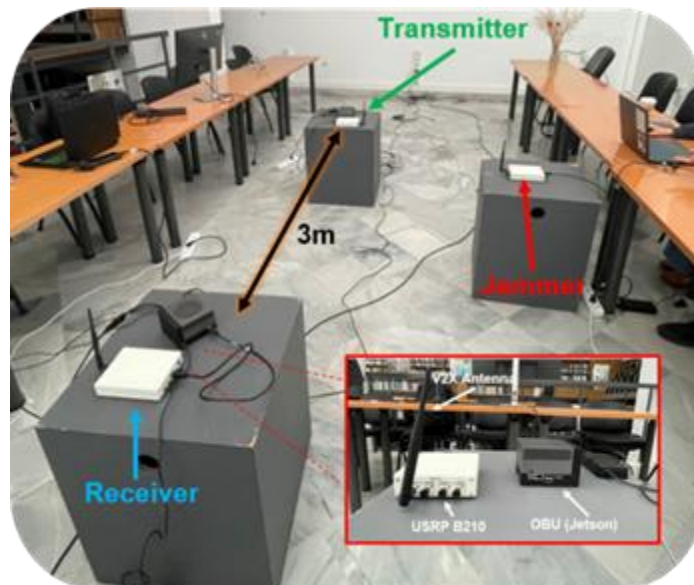


Figure 8: CERN lab with experimental testbed.

The framework supports multiple types of jamming attacks, including constant, periodic, and reactive jamming. In constant jamming, the adversary continuously transmits interference signals to overwhelm the channel. In periodic jamming, interference is introduced at predefined intervals, targeting specific transmission patterns. Reactive jamming represents a more advanced strategy, where the jammer monitors the channel and transmits only when legitimate activity is detected, increasing both efficiency and stealth.

The SDR-based implementation provides fine-grained control over key parameters such as transmission power, frequency offset, timing, and bandwidth. This flexibility enables the generation of diverse and realistic attack scenarios that closely resemble real-world adversarial conditions. The jamming framework has been designed in alignment with the objectives of Use Case 2.2 of the NATWORK project, which focuses on enabling AI-based jamming detection and mitigation in multi-path routing environments.

5.3.2 Experimental Setup and validation

The experimental evaluation of the proposed framework was conducted using the CERTH SDR-based testbed, providing a controlled and reproducible environment for assessing the impact of jamming attacks on wireless communications. The testbed operates on IEEE 802.11p vehicular communication at 5.9 GHz and enables the configuration of diverse transmission and interference scenarios.

The experiments were designed to evaluate the effect of jamming across multiple modulation schemes, including BPSK, QPSK, 16-QAM, and 64-QAM, allowing the analysis of system performance under different levels of robustness to noise. The jammer was configured to transmit white noise signals, while parameters such as transmission power and timing were varied to emulate different interference conditions.

A graphical user interface (GUI) is presented in Figure 9 and is provided to enable real-time configuration of the jamming framework. The interface includes control elements for adjusting key parameters such as transmission gain, which controls the signal power of the transmitter, operating frequency, which defines the channel of operation, jamming interval, which determines the timing between interference transmissions, and noise amplitude, which regulates the intensity of the injected interference. Additionally, the user can select the modulation scheme (encoding), allowing experimentation across different signal constellations. These controls enable flexible and efficient configuration of the attack scenario during operation.

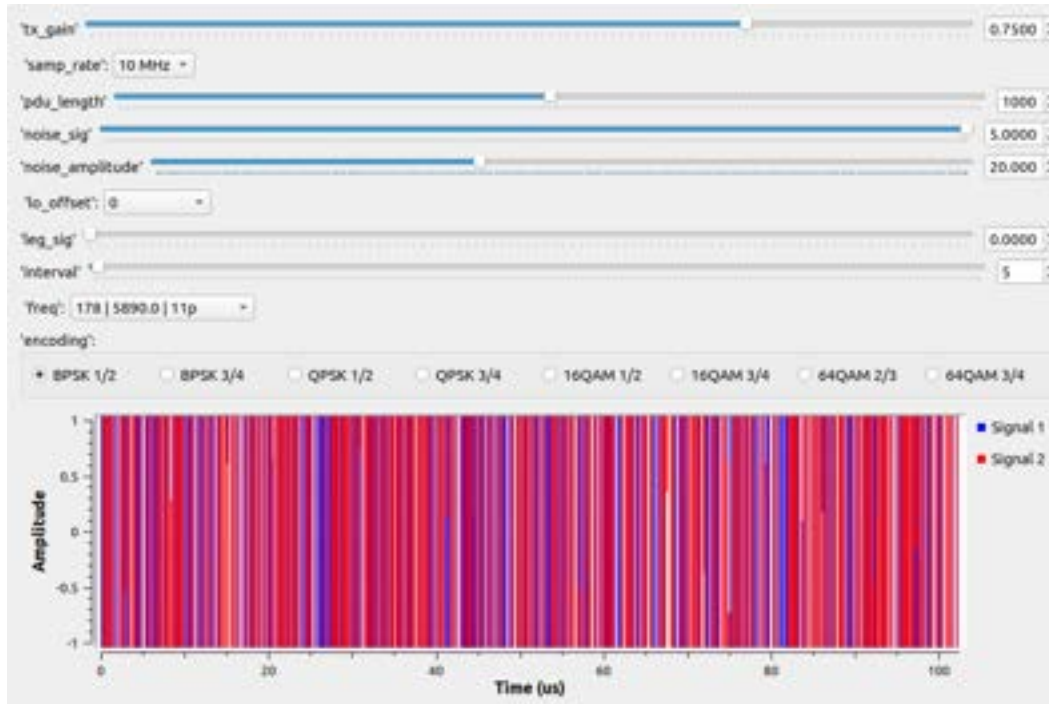


Figure 9: Jammer user interface in GNU Radio.

During operation, the receiver continuously captures raw baseband I/Q samples, which are forwarded via a UDP-based streaming mechanism to a processing module. This module applies to an AI-based model to classify the received signal in real time as either clear or jammed, enabling immediate assessment of channel conditions without requiring offline processing.

The evaluation focuses on the impact of jamming on signal integrity and communication reliability. The results indicate that even simple white noise interference can significantly degrade system performance, with more pronounced effects observed in higher-order modulation schemes.

5.4 OT-ICS-attacks

The considered attack scenario targets a modern industrial control system (ICS) environment where IT and OT infrastructures are tightly interconnected through SCADA and PLC components. In this context, the architecture is designed to reflect a realistic deployment in which supervisory systems such as ScadaBR interact with programmable logic controllers like OpenPLC over industrial protocols, notably MODBUS/TCP, while also exposing web-based services over HTTP/HTTPS. This convergence of enterprise and operational networks, combined with the use of legacy industrial protocols that lack built-in security, creates a broad attack surface that can be exploited through well-known vulnerabilities such as Log4Shell and weak authentication mechanisms.

5.4.1 System Architecture

The system architecture, as demonstrated in Figure 10, models a realistic industrial control system (ICS) environment that integrates both IT and OT layers, reflecting modern industrial deployments where operational processes are interconnected with enterprise networks. At the core of the architecture is a SCADA-based supervision layer that bridges the gap between external access systems and the industrial control infrastructure. This design enables centralized monitoring and control of physical processes while also exposing typical attack surfaces found in converged IT/OT environments.

The SCADA component, implemented using ScadaBR, acts as the central monitoring and control platform. It collects real-time data from industrial devices and provides operators with a graphical interface to supervise and manage processes. This SCADA system communicates with the control layer, which is implemented using OpenPLC, an open-source programmable logic controller responsible for executing control logic and interacting directly with industrial processes such as factory automation, robotics, and embedded systems. Together, these components emulate the behavior of real-world industrial environments where SCADA systems supervise PLC-controlled operations.

At the lower level, the field or process layer represents the physical or simulated industrial assets controlled by the PLC. These include manufacturing systems, robotic devices, and embedded components, forming the operational backbone of the system. On the upper side, an IT layer provides external connectivity through internet-facing services and remote management interfaces. While this enables operational flexibility, it also introduces vulnerabilities that can be exploited by attackers, especially when security mechanisms such as proper segmentation or patch management are insufficient.

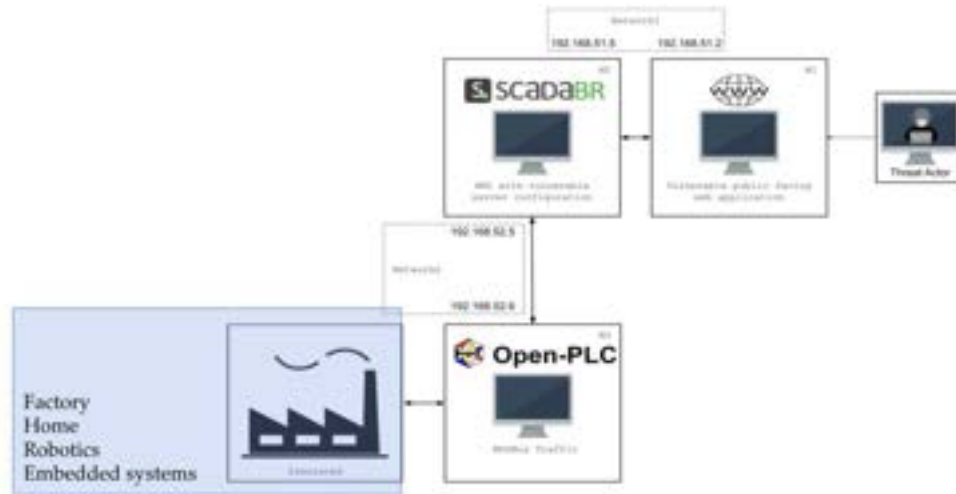


Figure 10: System architecture for OT-ICS attacks

Communication within the system relies on a combination of industrial and standard network protocols. MODBUS/TCP plays a central role in enabling communication between the SCADA system and the PLC, facilitating both telemetry collection and command execution. In addition, HTTP/HTTPS is used for web-based interfaces and APIs, while other protocols such as OPC, DNP3, RDP, and Telnet may be present for integration and remote access purposes. These heterogeneous protocols create a rich but complex communication environment, which is essential for realistic experimentation but also increases the attack surface.

To support monitoring and security analysis, the architecture integrates MMT-based components, including MMT-Probe for traffic capture and feature extraction, and MMT-Security for anomaly detection. These components analyze network behavior and feed results into a SADC dashboard that provides visualization of alerts and system status. This monitoring infrastructure enables the observation of both normal and malicious activities, supporting the evaluation of detection mechanisms.

Overall, the architecture is deployed as a controlled testbed environment, likely leveraging virtualization or containerization technologies. It is specifically designed to support the execution of cyberattacks alongside real-time monitoring, enabling the generation of datasets and the evaluation of detection techniques. By replicating realistic IT/OT convergence and common industrial protocols, the system provides a suitable platform for studying advanced cyber-physical threats and resilience strategies.

5.4.2 Experimental Setup and Validation

The experimental setup is designed to reproduce a realistic multi-stage cyberattack against an OT/ICS environment while enabling detailed monitoring and validation of detection capabilities. The testbed integrates industrial components such as ScadaBR (SCADA system) and OpenPLC

(controller), interconnected within a network that simulates a factory-like environment. This setup allows the attacker to interact with both IT-facing services and OT components, reflecting real-world deployment conditions. Security monitoring is performed using MMT-Probe and MMT-Security, which capture and analyze network traffic and generate alerts through the SACD dashboard.

```
carillag@mtlinage:~/Documents/marcl-$ sudo nmap -A 10.208.89.15
Starting Nmap 7.95 ( https://nmap.org ) at 2024-07-22 12:18 CEST
Nmap scan report for 10.208.89.15
Host is up (0.0026s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
22/tcp    open  ssh          OpenSSH 8.9p1 Ubuntu 3ubuntu0.18 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_  256 8b:86:7d:61:6f:ae:c0:71:ce:9a:5f:cd:89:15:12:8a (ECDSA)
|_  256 de:80:b7:ab:4b:9c:6d:f3:58:53:08:eb:7d:66:6a:8a (ED25519)
8080/tcp  open  nagios-nrca Nagios NSCA
|_ http-title: Site doesn't have a title (application/json).
Device type: general purpose|router
Running: Linux 5.X, MikroTik RouterOS 7.X
OS CPE: cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3
OS details: Linux 5.0 - 5.14, MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)
Network Distance: 2 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT ADDRESS
1 2.65 ms 10.208.89.15

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.31 seconds
```

Figure 11: Reconnaissance phase of the attack, showing network scanning using Nmap to identify hosts, open ports, and vulnerabilities.

The experimental execution follows a step-by-step attack scenario. In the first phase (Figure 11), reconnaissance is conducted by scanning the target network using tools such as Nmap with aggressive scanning options. This step enables the attacker to identify active hosts, open ports, and potential vulnerabilities within the infrastructure. The results of this phase reveal exposed services that can be exploited in subsequent stages.

In the second phase (Figure 12), initial access is achieved by exploiting a known vulnerability, specifically Log4Shell, using the Metasploit framework. This allows the attacker to compromise an internet-facing component and gain a foothold in the system. Once access is obtained, the third phase (Figure 13) involves execution, where the attacker opens a remote shell and runs commands on the compromised machine, effectively establishing control over the system.

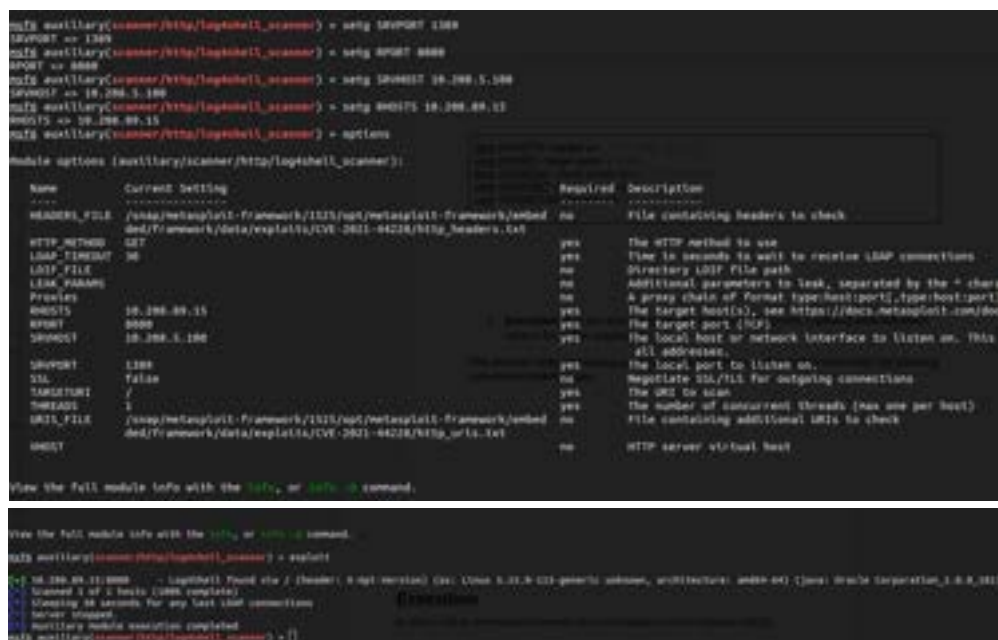


Figure 12: Initial access phase, demonstrating exploitation of the Log4Shell vulnerability using Metasploit to compromise a target system.

The fourth phase (Figure 14) focuses on lateral movement within the network. The attacker scans the internal environment to discover additional assets, identifies the SCADA server (ScadaBR running on a Tomcat server), and performs a brute-force attack to gain access. This step demonstrates how weak authentication and insufficient segmentation between IT and OT networks can allow the attacker to move deeper into critical infrastructure components.

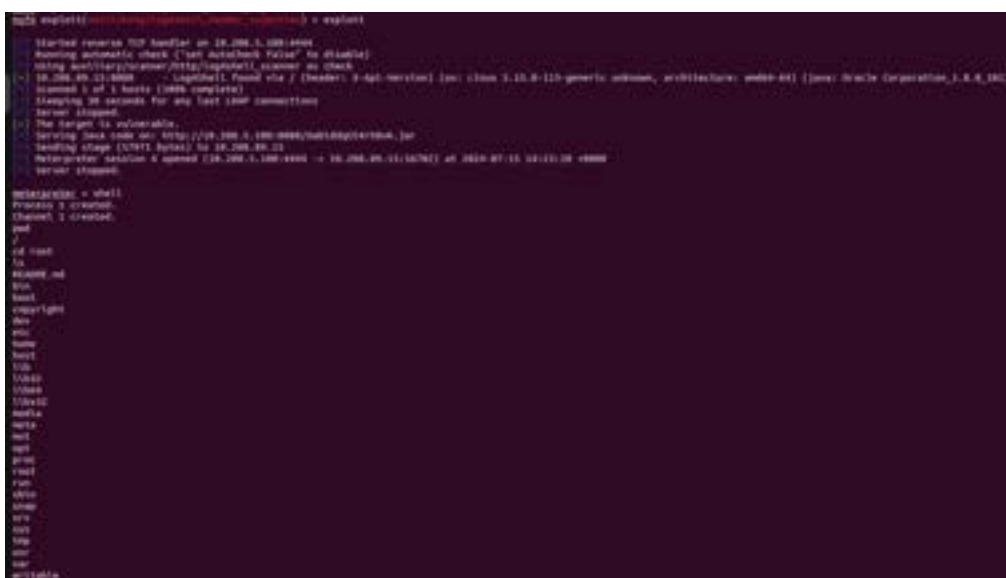


Figure 13: Execution phase, where the attacker establishes a remote shell and executes commands on the compromised machine.

In the fifth phase (Figure 15), the attacker establishes command and control (C2) by leveraging the MODBUS/TCP protocol, typically operating on port 502. By using tools such as modpoll, the attacker communicates with compromised devices while blending malicious traffic with legitimate industrial communication. This technique makes detection more difficult, as the traffic appears consistent with normal ICS operations.



Figure 14: Lateral movement phase, illustrating network scanning, identification of the SCADA server, and brute-force attack against ScadaBR.



Figure 15: Command-and-control phase using MODBUS/TCP protocol to communicate with compromised devices while blending with legitimate industrial traffic

Finally, in the impact phase (Figure 16), the attacker disrupts the operational process by manipulating machine states through MODBUS/TCP commands. This directly affects the physical processes controlled by the ICS, demonstrating the cyber-physical consequences of the attack. In a broader ransomware scenario, this stage can also include encrypting critical systems such as HMIs and databases, leading to production downtime and operational failure.

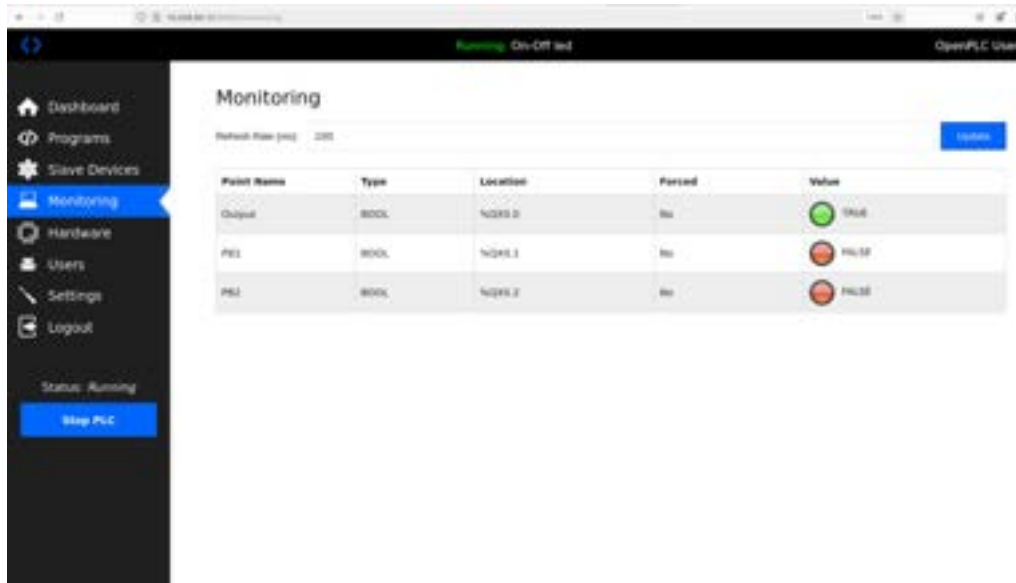


Figure 16: Impact phase, showing manipulation of machine states via MODBUS/TCP, leading to disruption of industrial processes.

Validation of the experiment is performed through the MMT-based monitoring infrastructure as depicted in Figure 17. MMT-Probe captures network traffic across all phases of the attack, while MMT-Security analyzes behavioral patterns and detects anomalies such as unusual scanning activity, exploitation attempts, brute-force login behavior, and abnormal MODBUS commands. The SACD (Security Analytics and Cyber Detection) dashboard provides visual evidence of these detections, including alert timelines and detailed logs of suspicious events. This validation approach ensures that each stage of the attack is both reproducible and observable, supporting the evaluation of detection accuracy and system resilience.

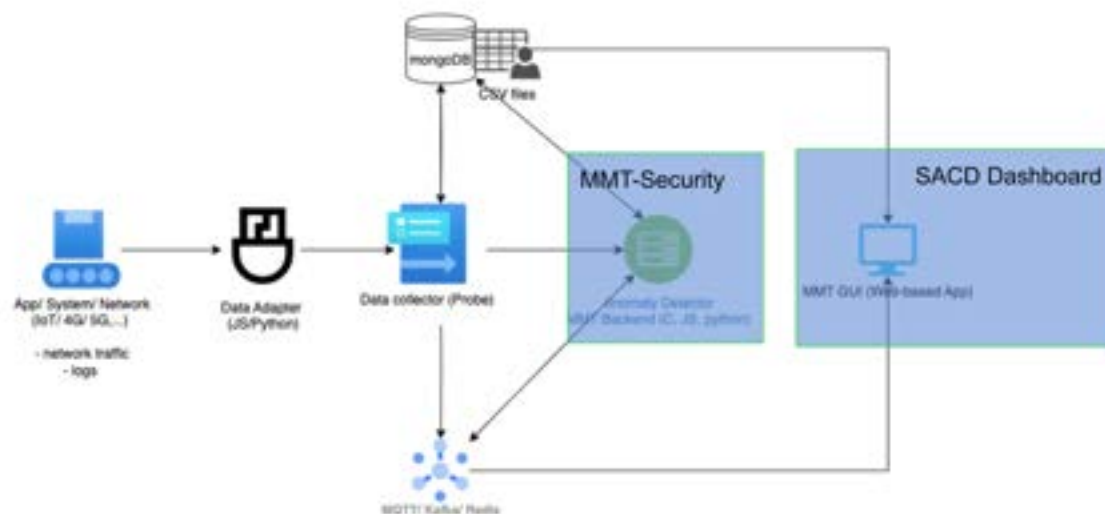


Figure 17: MMT-based anomaly detection architecture, including MMT-Probe, MMT-Security, and SACD dashboard components.

5.5 Eavesdropping Attack

Physical-Layer Key Generation (PKG) has emerged as a promising encryption-free mechanism for securing wireless links by exploiting the inherent randomness of the radio channel. However, the assumption that spatial decorrelation alone guarantees security against passive adversaries has been increasingly challenged. In the following section, an indoor PKG eavesdropping assessment framework is introduced as an indicative example of passive attack generation and related dataset collection. The framework is designed to evaluate the resilience of PKG schemes against a co-located passive eavesdropper (Eve), providing a controlled evaluation environment for physical-layer security mechanisms.

5.5.1 System Architecture

The proposed framework is built upon a simulation-based architecture using QuaDRiGa in MATLAB, enabling flexible and reproducible generation of indoor sub-THz channel data for all three nodes involved: Alice (Base Station), Bob (UE), and Eve (passive eavesdropper). The system emulates a TDD half-duplex PKG exchange at 95 GHz under the 3GPP_38.901_Indoor channel model.

The framework consists of three main components: a legitimate Alice-Bob channel pair and an independent Eve channel node. Alice and Bob share the same physical channel separated by the TDD delay, emulating the reciprocity impairment of a real half-duplex system. Eve's channel is generated independently, capturing her passive observation of the wireless environment without any signal injection or interaction with the legitimate link. The complex CFR is estimated at the OFDM receiver after propagating a reference signal through the QuaDRiGa channel, providing Eve with the raw channel observations she would exploit to infer key-relevant information. A complete description of the dataset structure, propagation variants, and OFDM numerology is provided in Section 6 on this deliverable.

5.5.2 Experimental Setup and validation

The experimental evaluation of the proposed framework was conducted using the QuaDRiGa-based simulation environment, providing a controlled and reproducible setting for assessing Eve's passive eavesdropping capability within a PKG exchange. The simulation operates at 95 GHz under the 3GPP_38.901_Indoor channel model and covers three propagation variants of increasing complexity: LOS, NLOS, and NLOS with AWGN at six controlled SNR levels ranging from 5 to 30 dB.

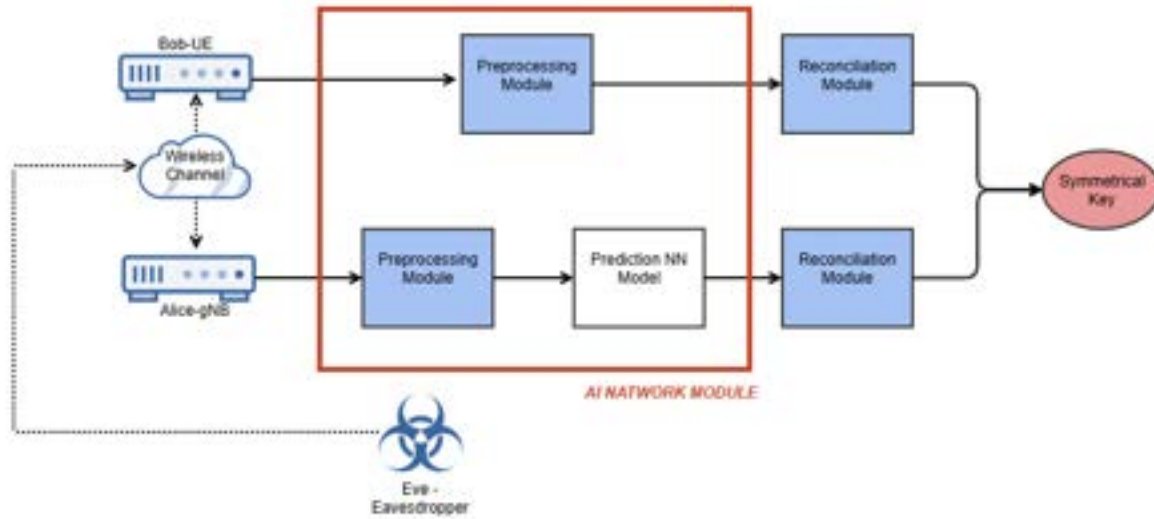


Figure 18: Functional lab architecture for eavesdropping and PKG

The evaluation is structured around a stage-by-stage comparison between Eve's reconstructed key material and the legitimate Alice-Bob link across the full PKG pipeline. At the channel probing stage, the CFR correlation between Eve's independent channel and the legitimate link is measured to quantify raw leakage. After quantization, the Key Disagreement Rate (KDR) between Eve's bit sequence and Alice-Bob's is computed to assess how much key-relevant information Eve retains. The reconciliation stage is then evaluated in terms of information leakage introduced by the error correction process, and finally, the effectiveness of SHA-256 privacy amplification is assessed by determining whether Eve can reconstruct the final key under each propagation scenario.

5.6 DoSt Attack

The DoSt [68] attack is an application-layer Denial of Service (DoS) variant that targets resource sustainability in cloud-native systems. It operates primarily over HTTP over TCP/IP and affects Kubernetes-based Cloud-Native Functions (CNFs), including proxy components (acting as API gateways) and backend service pods. The attack is characterized by oscillatory and rapidly fluctuating request patterns, where periods of high-intensity traffic are interleaved with low or idle phases. This behavior exploits the auto-scaling mechanisms and elasticity of cloud-native environments, forcing continuous scale-in and scale-out operations. As a result, instead of causing immediate service disruption like traditional DoS attacks, DoSt induces persistent CPU and memory oscillations, increases control-plane overhead, degrades Quality of Service (QoS), and leads to inefficient and unsustainable resource utilization, particularly in edge-cloud deployments.

Such behaviour represents a realistic threat to cloud-native 5G/6G infrastructures where dynamic scaling is essential. Unlike conventional DoS attacks, its oscillatory nature requires pattern-based and temporal detection approaches to distinguish between benign and attack patterns, making it well-suited for the development and evaluation of AI-driven anomaly detection and resilience mechanisms across application and system layers.

5.6.1 System Architecture

The DoSt datasets are generated within the Network Convergence Laboratory (NCL) at the University of Essex, which provides a multi-tier edge–cloud infrastructure integrating core cloud clusters, cloudlets, and edge nodes interconnected via a programmable SDN network. The environment supports deployment of Kubernetes-based cloud-native services across distributed resources, enabling realistic edge–cloud processing scenarios. Containerized traffic generators are deployed as user-emulation clusters to simulate both benign and adversarial user behaviour, producing realistic HTTP request patterns toward CNF-based services. These services include a proxy component (acting as an API gateway) and backend service pods running within the Kubernetes environment. Figure 19 demonstrates a high-level view of the system architecture.

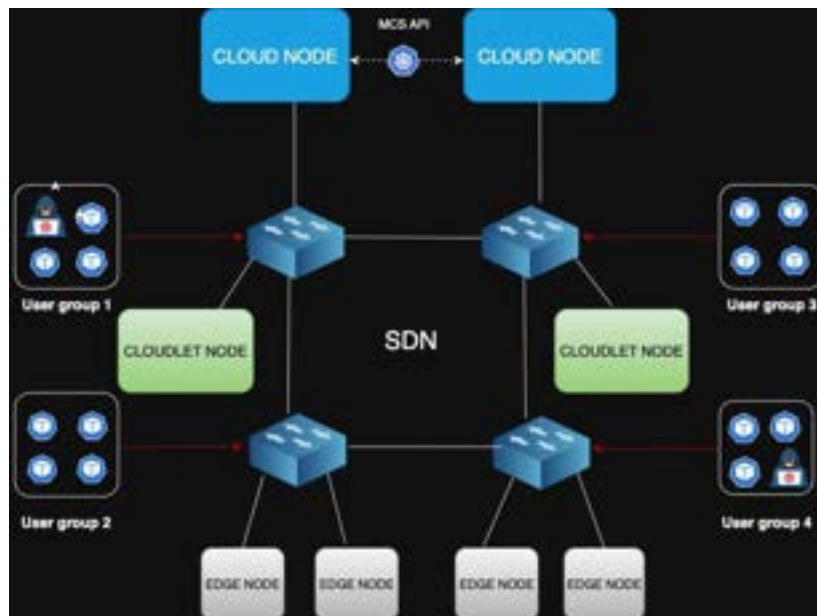


Figure 19: NCL Testbed Infrastructure

5.6.2 Experimental Setup and validation

The dataset is generated using a Kubernetes-based deployment of cloud-native services within the NCL edge–cloud infrastructure. The experimental setup consists of a proxy component (acting as an API gateway) and backend service pods, which are subjected to dynamically varying workloads generated by containerized user-emulation clusters.

Benign traffic is continuously generated to emulate realistic user behavior, while attack traffic is introduced in oscillatory patterns with alternating low and high-intensity phases, replicating DoSt conditions. These workloads are scaled dynamically across multiple pods to create varying system states, including idle, benign-dominant, and attack scenarios. The dataset spans approximately three weeks and represents roughly 100–200 concurrent benign users, alongside interleaved attack patterns.

Metrics from these components are continuously collected using Prometheus [69] Time Series Database (TSDB), capturing CPU, memory, network, and request-rate behavior at per-minute granularity for dataset generation.

5.7 Mirai botnet attack

The Mirai botnet is one of the most well-known malware families used to perform large-scale Distributed Denial of Service (DDoS) attacks. It primarily targets Internet of Things (IoT) devices by exploiting weak authentication mechanisms and insecure default configurations. Once compromised, these devices become part of a distributed network of bots capable of generating significant volumes of malicious traffic under the control of a command-and-control infrastructure. Mirai supports multiple attack vectors, including UDP flooding, TCP SYN flooding, and application-layer HTTP attacks, which are representative of common DDoS strategies observed in operational networks.

For this reason, Mirai traffic constitutes a representative and realistic workload for the evaluation of DDoS detection mechanisms. The attack profiles considered in this work, namely UDP generic flooding, TCP SYN flooding, and HTTP application-layer attacks, provide complementary perspectives for analysing the behaviour of both network-layer and application-layer DDoS attacks in wireless communication environments.

Mirai botnet attack	UDP generic	UDP
	TCP SYN	TCP
	App HTTP	HTTP

5.7.1 System Architecture

For DDoS attack generation, a set of Raspberry Pi devices hosts a Mirai botnet connected through smartphone tethering. Each Raspberry Pi hosts a set of Docker containers acting as traffic sources or receivers. The test bed supports representative legitimate traffic models: MQTT via Mosquitto, UDP with Python, HTTP with nginx and curl, and video streaming with mediamtx and ffplay. It also includes isolated Mirai virtual machines. The embedded Raspberry Pi 5 devices host QEMU/KVM virtual machines to contain malicious code and control it remotely.

Legitimate services are deployed as lightweight containers connected directly to the test interface exposed by the tethered smartphone. Malicious VMs are managed by a dedicated Malware Management and Control (MMC) framework consisting of mmc-vm-daemon, mmc-host-daemon, and mmc-cli. Control commands for the malware are exchanged via AF_UNIX sockets on the Raspberry Pi host side and virtual serial ports on the guest side, enabling fine-grained orchestration while maintaining secure separation of the malicious code. Traffic of interest for DDoS detection is captured at the N3 interface between the gNB and the UPF. The captured traces are processed by CICFlowMeter, stored in Parquet format, and consumed by a dedicated container running Python-based XGBoost models. Training and inference are currently performed offline.

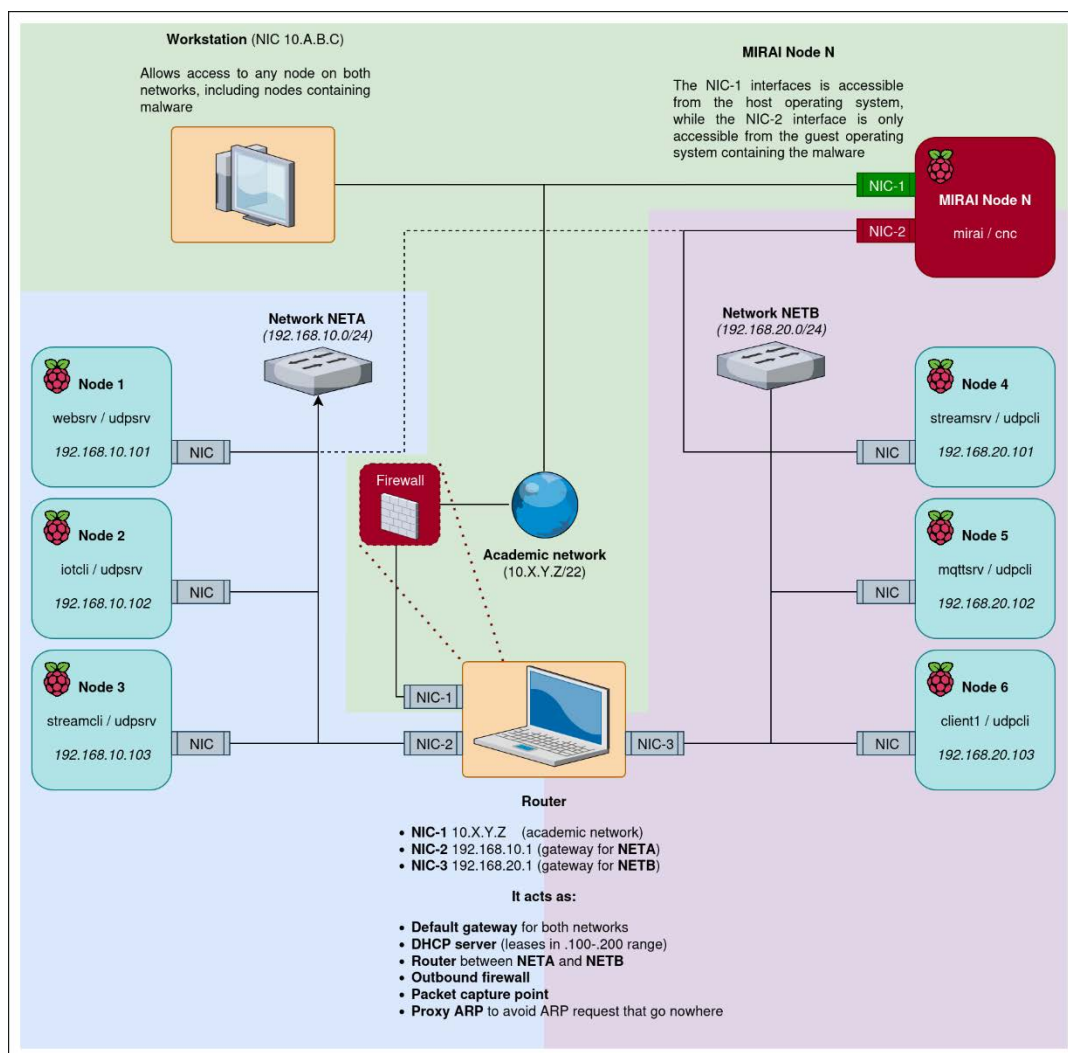


Figure 20 Dedicated IoT traffic test bed.

Figure 20 illustrates the overall architecture of the dedicated IoT traffic test bed, highlighting the interaction between the legitimate service containers, the Mirai-controlled virtual machines, the

MMC orchestration framework, and the traffic analysis pipeline. The figure also shows the placement of the traffic capture point at the N3 interface, and the subsequent processing chain used for flow extraction and offline DDoS detection.

5.7.2 Experimental Setup and validation

The testbed employed for data generation is based on a distributed architecture comprising, on the one hand, servers responsible for producing various types of data streams and, on the other hand, servers tasked with consuming these streams. This configuration enables the reproduction of a controlled network environment while maintaining conditions that are representative of an operational setting.

A Mirai botnet has also been integrated into the network in the form of a virtual machine hosted on a Raspberry Pi. This instance has been specifically modified to prevent any self-propagation capability, thereby ensuring that its use remains strictly experimental and fully controlled. This arrangement enables the injection of malicious traffic in a distributed manner, particularly using a mobile telephone as a tethering device. Consequently, the DDoS attack traffic can be transmitted over the air interface, allowing its effects to be investigated within an environment that incorporates wireless communications.

The testbed is therefore not limited to the assessment of DDoS attack detection mechanisms. It also serves as an experimental platform for the study and detection of radio jamming phenomena. In parallel, legitimate traffic may be generated by means of a mobile telephone connected to the base station and used as a modem. This approach makes it possible to reproduce mixed scenarios in which legitimate traffic coexists with malicious traffic, thereby providing a relevant framework for the analysis, characterisation, and detection of DDoS attacks in realistic communication environments.



Figure 21 Client nodes



Figure 22: Server nodes.

Figure 21 illustrates the client-side nodes of the distributed testbed, including the devices responsible for generating legitimate and malicious traffic over the wireless interface. Figure 22 presents the server-side infrastructure used for traffic reception, service hosting, data collection, and offline analysis of DDoS attacks and radio jamming phenomena.

5.8 5G Interface Flooding Attacks

As 5G systems rely on multiple control-plane and user-plane interfaces, their security evaluation requires datasets that preserve protocol and interface-specific characteristics. The UZH 5G Network Traffic dataset provides such a basis by capturing benign and flooding traffic in an emulated hybrid 5G testbed. The dataset is designed to support the analysis of normal and malicious behaviour across different 5G interfaces and to enable the development and evaluation of interface-aware intrusion detection mechanisms.

5.8.1 System Architecture

The UZH testbed is implemented as an emulated hybrid 5G environment built with Mininet and virtual machines. The architecture includes the main components of a 5G system, including the UE, gNB, 5G core functions, UPF, data network, and an adversarial node. Traffic is monitored at multiple observation points in order to capture the behavior of both control-plane and user-plane interfaces.

The dataset includes captures from three main monitoring points. Traffic on the N3 interface is recorded at r0 inside the GTP tunnel, traffic on the N6 interface is recorded at r1 after the UPF, and N4 control-plane traffic is captured separately on the Control VM. This monitoring design

makes it possible to preserve the protocol context of each scenario and to analyze how flooding attacks affect different parts of the 5G architecture.

The dataset is organized into two main categories: benign and flooding. The benign subset includes realistic service traffic such as web browsing, streaming, gaming, and ping activity. The flooding subset includes attack traffic targeting the N2, N3, N4, and N6 interfaces. The corresponding protocols include SCTP flooding on N2, GTP/UDP, GTP/TCP and GTP/ICMP flooding on N3, PFCP flooding on N4, and TCP flooding on N6.

5.8.2 Experimental Setup and Validation

Traffic traces were collected in PCAP format using packet capture tools compatible with Wireshark and tcpdump. The dataset contains fourteen PCAP files in total, including eight benign captures and six flooding captures. The link layer is Linux cooked-mode capture v1, and the packet traces were truncated to a snap length of 96 bytes. This truncation removes application payload while preserving the protocol-level information needed for traffic characterization and intrusion detection experiments. MAC address pseudonymization was applied to support anonymization.

The benign traffic was generated to approximate realistic user and service behavior. Web browsing traffic was generated against a local web server, gaming traffic was produced using a TCP socket-based client/server implementation with jitter and traffic variations, streaming traffic was generated through concurrent video and audio streams, and ping traffic was generated using standard ICMP echo requests.

The flooding scenarios were designed to stress different 5G interfaces and protocols. The N2 SCTP flooding scenario represents a signaling-plane overload case. The N3 flooding scenarios include GTP/UDP, GTP/TCP, and GTP/ICMP flooding inside the GTP tunnel. The N4 PFCP flooding scenario targets the control-plane interface, while the N6 TCP flooding scenario represents an overload attack after the UPF on the external data interface.

The collected traces show clear differences between benign and attack traffic. Benign traffic follows application-driven patterns with intermittent or service-oriented behavior, while the flooding traces show persistent abnormal load patterns. Therefore, the dataset is suitable for evaluating 5G intrusion detection approaches, especially those that consider protocol context, capture point, and interface-specific traffic characteristics.

5.9 Inter-Container Attacks in Kubernetes

In real-life deployments of applications, network access from external sources is almost always restricted to certain extent by security tools, such as firewalls or load balancers. However, due to the inevitable communication across different components of deployed applications, internal

network traffic rules are much looser. For this reason, a silent infiltration on a container or VM could lead to a stealthy compromise of additional components via internal traffic, which are usually harder to detect and prevent.

The same applies to container and container orchestration platforms. Since 6G systems will heavily utilize Containerized Network Functions (CNFs), securing containers is of utmost importance for future wireless networks. Given the popularity of Kubernetes as a container orchestration platform, malicious users may attempt to infiltrate into a Kubernetes cluster and conduct attacks on benign pods or nodes, disrupting the communication inside the cluster itself. Once a container is compromised, the infection could spread to other pods or containers via internal traffic, which is assumed to be safe. To prevent such malicious network actions, sophisticated anomaly or intrusion detection systems should be developed, focusing on inter-container network traffic.

To simulate such a possible scenario in a realistic manner, we deploy a Kubernetes cluster and spawn infected pods along with benign ones and generate malicious inter-container network data, forming the anomaly detection dataset for Kubernetes, namely GATAKU.

5.9.1 System Architecture

The corresponding testbed for generating the inter-container attacks is implemented using virtual resources deployed on an OpenStack-based cloud environment. A microservice application, with its components shipped in containers, is deployed in a Kubernetes cluster that is managed by k3s. As the Container Network Interface (CNI) of the Kubernetes cluster, eBPF-based Cilium is utilized, and the data is collected using an OpenTelemetry backend. Regular application network flows comprise the benign data while malicious data is crafted using manually triggered attacks.

An overview of the cloud setup used in the data collection process is provided in Figure 23, where the leveraged tools are displayed. The Kubernetes cluster includes five nodes in total, one for the control-plane (which also undertakes load like a worker) and four worker nodes. Note that the control-plane also undertakes load like a worker so the microservice components are deployed across all these five nodes.

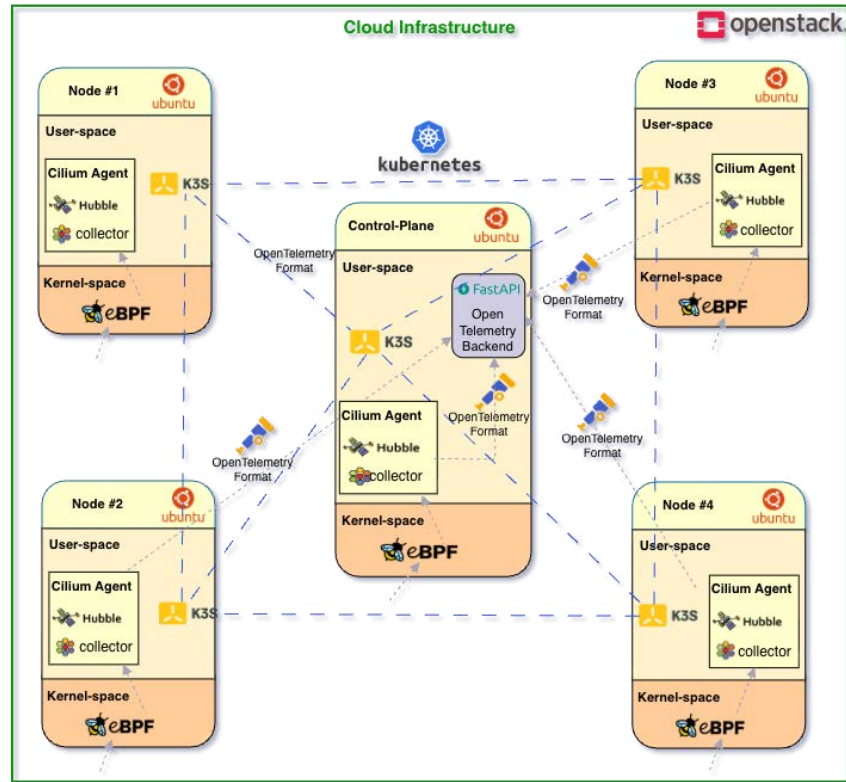


Figure 23: An overview of the Kubernetes setup on the cloud for the GATAKU dataset.

5.9.2 Experimental Setup and Validation

The deployed microservice application consists of four components: a front-end, a back-end, a database, and an external service. Each component uses multiple replicas of containers for higher availability and fault-tolerance, except the database unit which is stateful and needs to store the entire data. Also, to simulate multi-tenancy, the concept of namespaces in Kubernetes is leveraged, where multiple instances of the application is deployed across isolated network slices.

For the malicious data, three types of attacks are initiated on the microservice applications:

- **Denial of Service (DoS):** A malicious pod in each namespace targets a benign pod in that namespace and sends excessive network requests using the *hey* application
- **Port Scan:** A malicious pod in a random namespace scans the entire subnet using *nmap*
- **ZAP Scan:** A malicious pod in each namespace targets a benign front-end pod in that namespace and uses the *OWASP ZAP* tool to scan the vulnerabilities on that pod

For the benign data, each service utilizes curl commands to periodically send requests to the services they are supposed to interact with. For instance, front-end containers send requests to back-end containers, which send requests to the external service and database containers. Note that interaction across containers is limited to those in the same namespace.

6 Data Generation during NATWORK Project

The NETWORK DMP provides the central data-management environment for the project, supporting the registration, organization, and controlled sharing of datasets among partners. At this stage, the platform includes 29 registered users and 12 registered organizations, reflecting the active participation of the consortium in the repository ecosystem. In total, 19 datasets have been uploaded, comprising 41 individual resources with an overall storage volume of 5.2086 GB. This state shows that the repository has moved beyond its initial setup phase and is now operating as a functional data catalogue, enabling partners to document, access, and reuse project datasets in a structured and traceable manner. Following, the description of the provided datasets is presented.

NATWORK created fifteen (15) datasets. Some of these datasets are related to the implemented attacks described in Section 5, while others cover additional data that are not directly associated with the attacks. The datasets are listed below:

1. Jamming Detection	9. DDoS Detection
2. Channel Estimation	10. Edge-IIoT Adversarial Dataset
3. Jamming Mitigation	11. 5G Network Traffic dataset
4. Anomaly Detection on Kubernetes (GATAKU)	12. OT-ICS-attacks dataset
5. Stateful CNF Live Migration Dataset	13. Denial of Sustainability Dataset
6. Channel Estimation for sub-THz bands	14. Cyber Threat Intelligence dataset
7. DetAction jamming Dataset	15. A Multimodal 5G Network & Container Metrics Dataset
8. Jamming Detection	

A detailed description of the datasets follow.

6.1 Jamming Detection

This section describes the JASMIN dataset, which has been generated to support the development and evaluation of AI-based jamming detection mechanisms for B5G/6G wireless systems. The dataset is directly linked to the JASMIN methodology. Figure 24 illustrates the structure of the JASMIN dataset. The dataset is organized according to channel condition (clear unjammed condition and jamming scenario). It includes data for 4 modulation schemes, namely, BPSK, QPSK, QAM-16 and QAM-64. Each modulation is represented under both normal channel conditions and intentional interference (jamming) conditions. This structure enables the training and evaluation of models that must distinguish between natural channel degradation and malicious interference. The dataset consists of complex baseband I/Q samples, captured using an SDR-based experimental setup operating under IEEE 802.11p/V2X communication conditions.

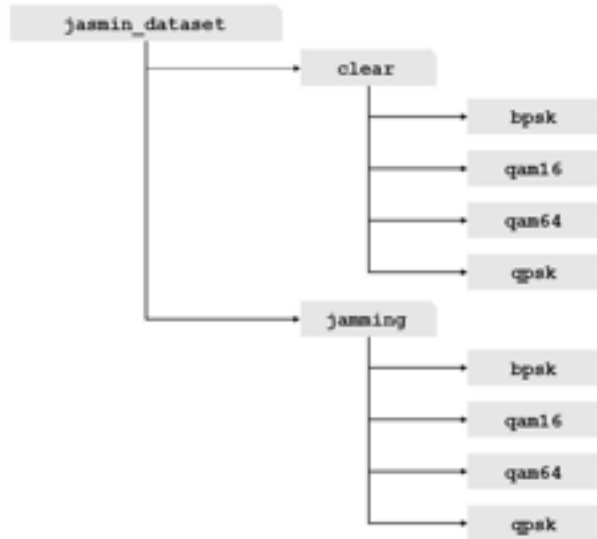


Figure 24: Structure of Jasmin Dataset

Table 12. Main parameters of Jasmin Dataset

Parameter	Size
Total Size	407 MB
Information format	I, Q samples
Scenarios	BPSK, QAM16, QAM64, QPSK clear and jammed
Total entries	12,900,000
BPSK clear entries	1,300,000
QAM16 clear entries	1,550,000
QAM64 clear entries	660,000
QPSK clear entries	2,180,000
BPSK jamming entries	1,100,000
QAM16 jamming entries	2,100,000
QAM64 jamming entries	2,900,000
QPSK jamming entries	1,100,000

Table 12 summarizes the main structural characteristics of the JASMIN dataset, which is included in the NATWORK repository. The dataset has a total size of 407 MB and contains 12.9 million complex baseband I/Q samples, organized across four modulation schemes (BPSK, QPSK, 16-QAM, 64-QAM) and two channel conditions (clear and jammed). The clear-signal subset represents legitimate transmissions under varying SNR levels, while the jammed subset captures signals distorted by intentional interference. The distribution of entries across modulations and

scenarios ensures sufficient data volume for both training on normal channel behavior and evaluating AI-based jamming detection performance under diverse signal conditions.

Table 13: Statistical metrics of the data per case and modulation

Case/Modulation	Mean distance from optimal	Standard deviation of distance from optimal
BPSK clear entries	0.143366	0.111220
QAM16 clear entries	0.286852	0.231116
QAM64 clear entries	0.169254	0.259270
QPSK clear entries	0.684550	0.125160
BPSK jamming entries	0.771155	0.553643
QAM16 jamming entries	0.355272	0.494433
QAM64 jamming entries	0.220967	0.502679
QPSK jamming entries	0.730981	0.438733

Table 13 provides statistical metrics describing the distortion of constellation points for each modulation and scenario, expressed through the mean distance from the ideal constellation and its standard deviation. Clear-signal cases exhibit lower mean distances and more stable variance, reflecting predictable noise levels under normal channel conditions. In contrast, jammed cases show increased mean distances and variability, indicating stronger and more irregular dispersion caused by interference. These statistical differences highlight the separability between simple channel noise and malicious jamming, forming the basis for anomaly-based detection approaches such as the Outlier Detector used in JASMIN.

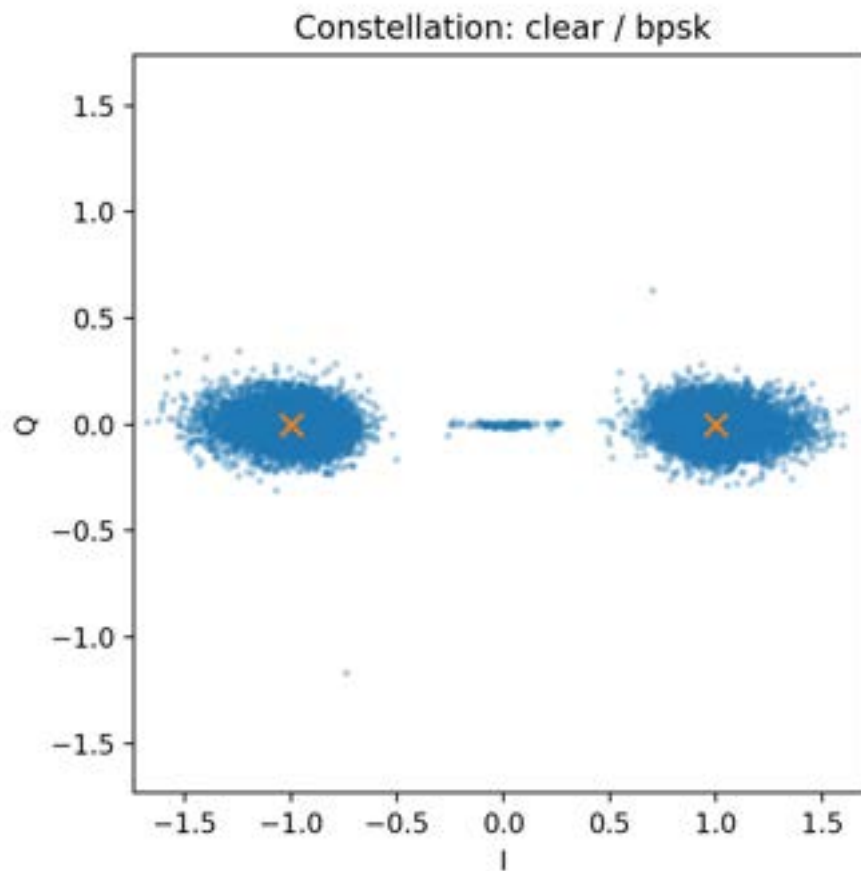


Figure 25: Constellation of clear BPSK signal

Figure 25 shows the constellation of a clear (unjammed) BPSK signal included in the Jasmin dataset. Two dominant clusters appear along the in-phase (I) axis, corresponding to the two BPSK symbols, while the quadrature (Q) component remains centered around zero. The orange marks indicate the ideal symbol locations, and the spread of the blue points around them reflects normal channel noise. The tight grouping of samples demonstrates the robustness of BPSK to noise and confirms stable reception under the unjammed conditions.

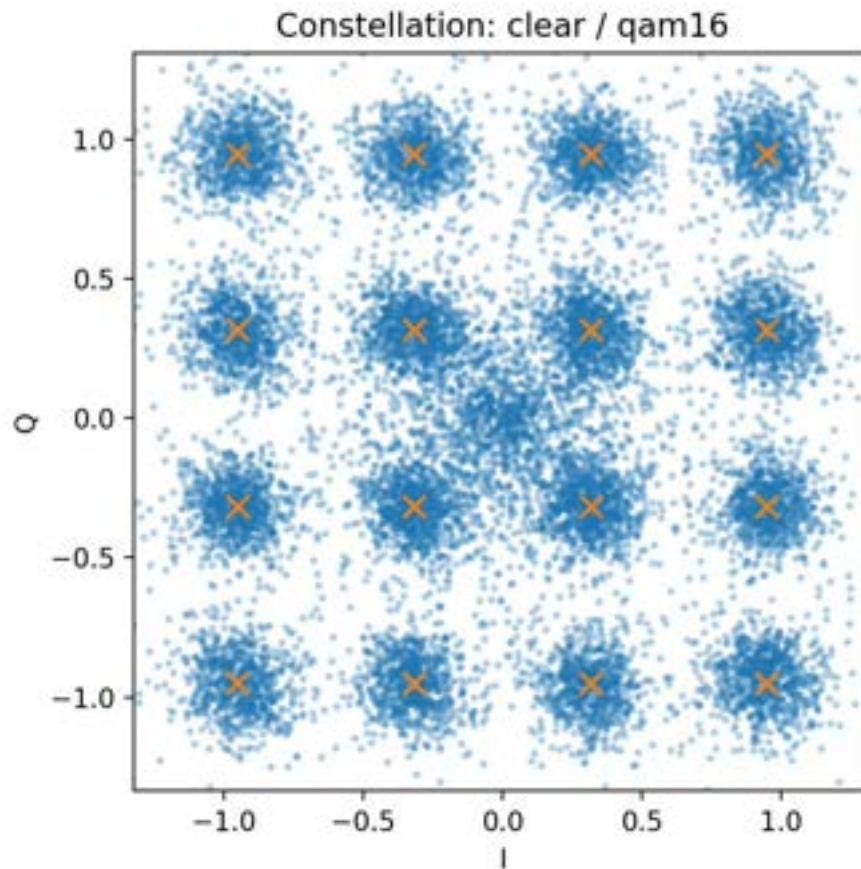


Figure 26: Constellation of clear 16-QAM signal

Figure 26 presents the constellation of a clear 16-QAM signal. Sixteen distinct clusters are visible in a square grid pattern, representing the modulation's symbol mapping. The blue clouds surrounding ideal reception indicate noise-induced dispersion. The clusters are closer to each other, illustrating the higher spectral efficiency and increased sensitivity of 16-QAM to channel noise, even in the absence of jamming. The concentration of blue samples that appears around the center of the I/Q plane is mainly caused by transient effects during symbol transitions, filtering, synchronization imperfections, and residual channel noise that exists in practical SDR-based systems.

6.2 Channel Estimation

The dataset contains simulated SISO V2X wireless channels for the IEEE 802.11p protocol at 5.9 GHz, generated using the QuaDRiGa channel simulator. It was created to support the development and evaluation of AI-based channel estimation models. Two propagation scenarios are considered. In both cases, the transmitter is modeled as a static base station, while the receiver is mobile.

The first scenario represents a highway non-line-of-sight (NLOS) environment based on the 3GPP 37.885 channel model. In this case, the receiver travels at speeds reaching 100 km/h, which leads to a maximum Doppler frequency of approximately $f_d \approx 1$ kHz, as presented in Figure 27 (a). The second scenario is based on the 3GPP 37.885 Urban NLOS model. Here, the vehicle speed is limited to 50 km/h, producing a Doppler spread of roughly $f_d \approx 500$ Hz Figure 28 (b). In both scenarios, the generated channel realizations include path-loss, shadow fading, and small-scale fading effects. While the Doppler spectra highlight the different levels of time selectivity between the two environments, the overall channel complexity is additionally governed by the multipath structure and delay spread. No channel normalization is applied to the stored channel coefficients, thereby preserving large-scale fading effects in the dataset.

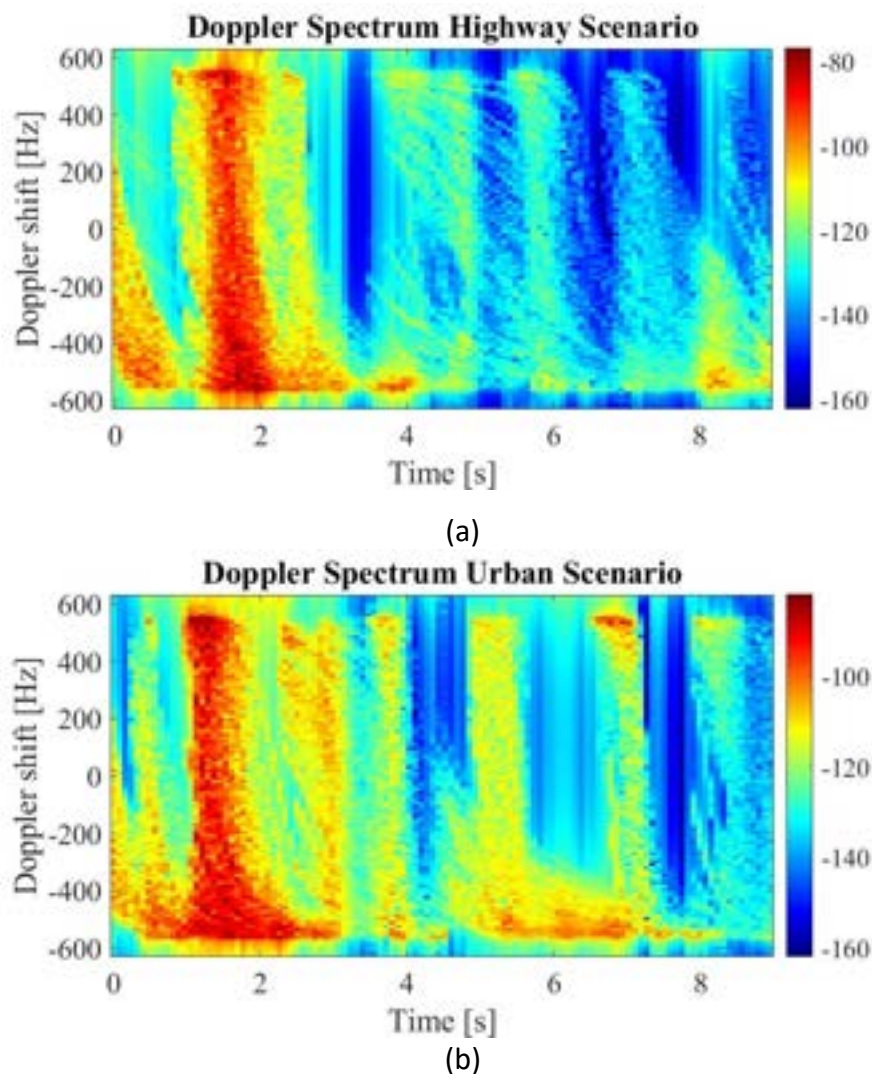


Figure 27: Doppler Spectrum of Highway scenario (a) and Urban Scenario (b).

Each of the two propagation scenarios consists of 6 different data files (.mat files), four of which are used for training and validation, and the other two for testing (Figure 28). Each data file

provides a complex channel frequency response (CFR) for every OFDM symbol and subcarrier. Specifically, for each .mat file there is an H_{full} complex matrix of size $[N_s, N_{fft}]$, where N_s is the length of the CFR sequence over time (number of OFDM symbols) and N_{fft} is the total number of subcarriers per symbol (64 in this protocol).

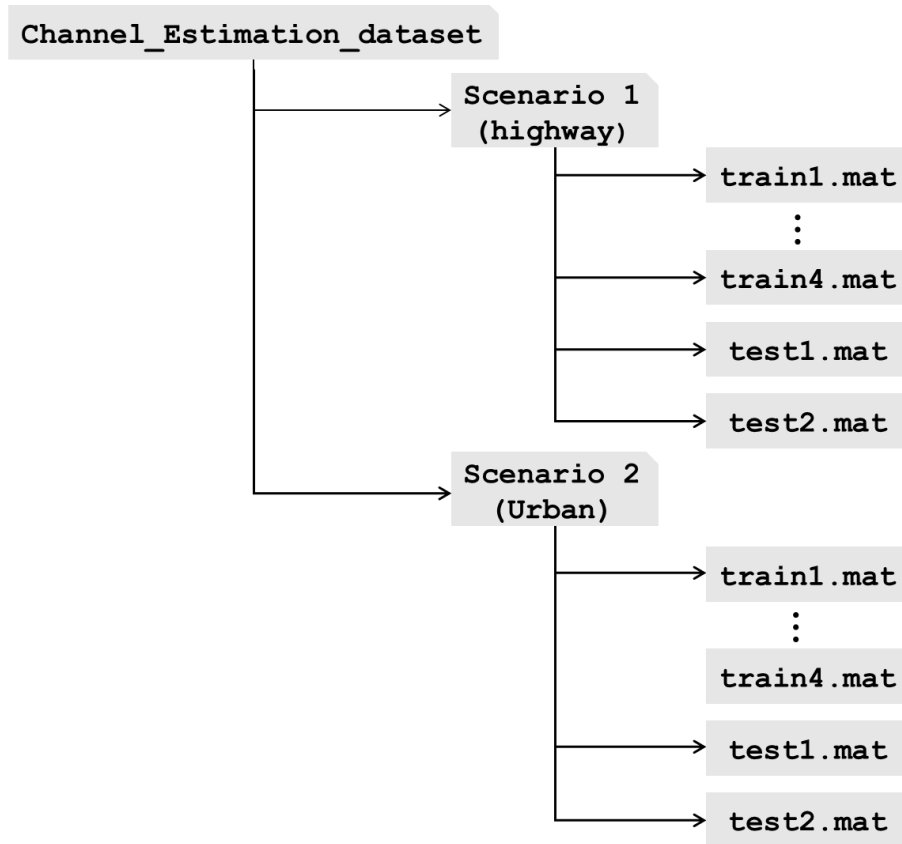


Figure 28: Structure of Channel Estimation Dataset.

Table 14: Main parameters of Channel Estimation Dataset.

Attribute	Description
Total Size	2.17GB
Data Type	Complex single-precision floating point (I, Q samples)
Structure	2D matrix (time snapshots × subcarriers)
Scenarios	3GPP_37.885_Highway_NLOS, 3GPP_37.885_Urban_NLOS
3GPP_37.885_Highway_NLOS	
Total files	6
Total Snapshots	2700000
3GPP_37.885_Urban_NLOS	
Total files	6
Total Snapshots	2,700,000

Table 14 summarizes the main characteristics of the Channel Estimation dataset. It includes the total dataset size, data type (complex single-precision I/Q samples), and structure (2D matrices of time snapshots and subcarriers). Additionally, it provides details for the two propagation scenarios, the 3GPP_37.885_Highway_NLOS and 3GPP_37.885_Urban_NLOS, each consisting of six data files and 2,700,000 snapshots. Table 15 presents the statistical properties of the dataset for both propagation scenarios.

Table 15: Signal power statistics for Highway and Urban NLOS scenarios.

Metric	Value
3GPP_37.885_Highway_NLOS	
Mean Power	2.2547e-08
STD Power	3.7634e-08
3GPP_37.885_Urban_NLOS	
Mean Power	1.6281e-09
STD Power	6.3985e-09

6.3 Jamming Mitigation

The dataset contains the received signal samples of a MIMO receiver operating under the IEEE 802.11p protocol for V2X communications. It was generated to support the development and evaluation of end-to-end jamming detection, identification, and mitigation techniques. The dataset covers multiple communication scenarios, including a baseline case with only a legitimate transmitter, as well as scenarios where the legitimate transmission is affected by one, two, three, or four jammers (Figure 29). For each scenario, multiple frame samples are generated under different angular configurations of the transmitters and jammers to ensure sufficient variability in the dataset. Each scenario is provided as a separate .npz file.

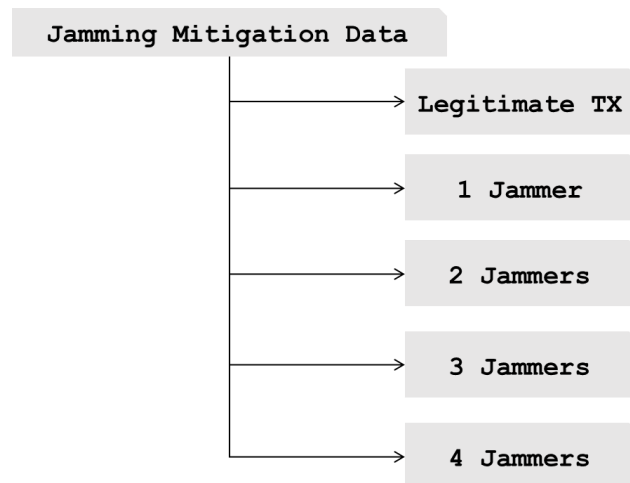


Figure 29: Structure of Jamming Mitigation dataset.

In addition to the jammer configurations described earlier, the dataset specifies the angular placement of the transmitters and jammers. The admissible angular grid is defined as $A = (-80^\circ, -79^\circ, \dots, 79^\circ, 80^\circ)$. A minimum angular separation of 5° is enforced between any two jammers, along with a 10° guard interval between each jammer and the legitimate transmitter. For the two-jammer scenario, all valid unordered angle pairs that satisfy these constraints are enumerated exhaustively, yielding 9,453 distinct configurations. For the three- and four-jammer cases, the additional jammer directions of arrival (DoAs) are randomly sampled from the same angular grid under the same separation rules. Table 16 provides an overview of the jamming mitigation dataset, including its total size, data format, and the distribution of entries across different scenarios.

Table 16. Main parameters of jamming mitigation dataset.

Parameter	Size
Total Size	36.7GB
Information format	I, Q samples
Scenarios	Only legitimate, 1 Jammer, 2 Jammers, 3 Jammers
Total entries	403,275
Legitimate entries	160
1 Jammer entries	5964
2 Jammer entries	132,482
3 Jammer entries	132,476
4 Jammer entries	132,193

Each .npz file contains raw complex baseband array data for direction-of-arrival (DoA) estimation, including the desired signal DoA, the DoAs of multiple jammers, and multiple time-domain frames (Figure 30). Each frame is represented by complex IQ samples arranged in a matrix of size (6×1024) , corresponding to the number of antennas and time samples, respectively.

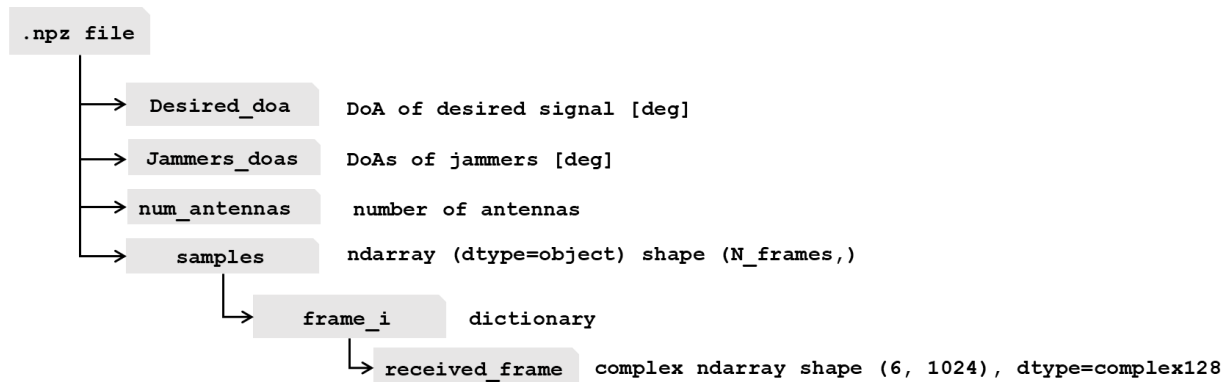


Figure 30: Structure of files of Jamming Mitigation Dataset.

6.4 Anomaly Detection on Kubernetes (GATAKU)

This dataset, namely GATAKU, is constructed via the simulation of a Kubernetes cluster, where malicious pods generate attack data across containers and a deployed microservice application components create benign network flows within the cluster. Three types of attacks are available in the GATAKU dataset: DoS, Port Scan, and ZAP Scan attacks. The details regarding how the malicious and benign flows are generated, are explained in Section 5.9.

The structure of the dataset is given in Figure 31 with a breakdown of the three different attack types comprising the anomalous network data. For benign flows, there is no further sub-categorization as they are all initiated by the same application.

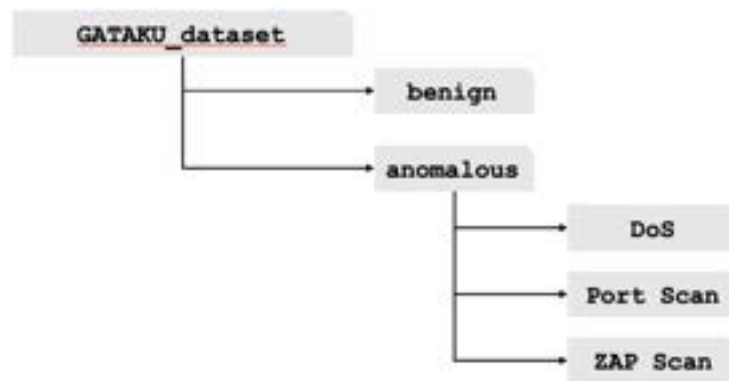


Figure 31: The structure of the GATAKU dataset.

The distribution of the class in the dataset is provided in Table 17 where the number of samples for each attack type plus the number of benign samples is provided. In total, there are 1,384,537 anomalous samples and 3,584,005 benign samples.

Table 17: Distribution of samples in the GATAKU dataset

Class	Number of Samples
Benign	3,594,005
DoS	1,228,749
Port Scan	155,267
ZAP Scan	521
Total entries	4,978,542

To provide an idea on the features and their relevance in the dataset, Figure 32 displays the most diverse features based on number of unique values they store. As one can observe, the most unique features are the source port and the destination port, which are followed by the 4th octet of the source and destination IP addresses, respectively.

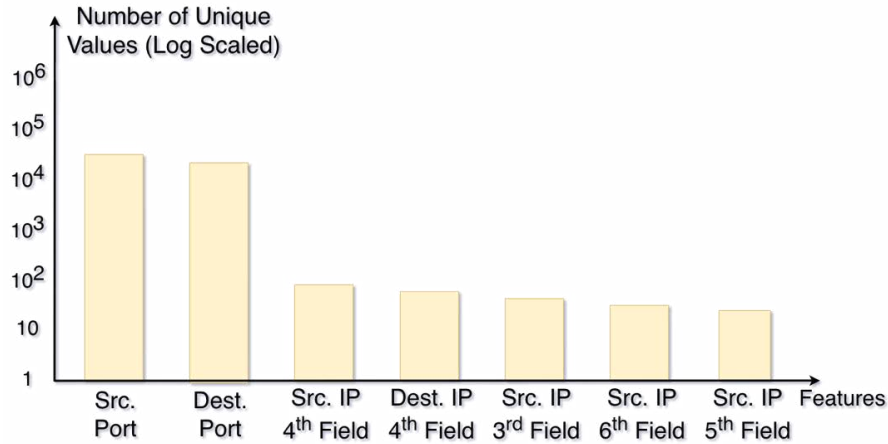


Figure 32: Top features of the GATAKU dataset based on value uniqueness.

6.5 Stateful CNF Live Migration Dataset

This dataset contains information regarding the stateful live migration of Containerized Network Functions (CNFs) in a cloud environment. In the context of this dataset, a container is migrated from one VM to another VM, both of which are hosted in an OpenStack-based cloud environment. As the container runtime, runc is used during the experiment phase and CRIU is utilized as the tool to save and restore state of the containers. The dataset comprises of four migration types, which are explained in detail below:

- **Cold:** The container is stopped, then moved to the target node along with its entire state and is started to run on the destination node with its prior state.
- **Pre-copy:** An initial subset of the container state is dumped and moved to the new place while the container is running on the existing node. Then, the container is stopped and moved to the target node along with the remaining state. Afterwards, the container is started to run on the destination node.
- **Post-copy:** The container is stopped and moved to the target node along with a minimal execution state, which is sufficient to run the container on the target node immediately. While the container is running on the new place, most of the prior state are copied from the old node simultaneously.
- **Hybrid:** A combination of *pre-copy* and *post-copy* techniques, where most of the state is transferred to the new node while the container is running on the existing node, like in *pre-copy*. Then, the container is paused and moved to the target node in a *post-copy* manner where the rest of the state is transferred after the container is started to run on the new node.

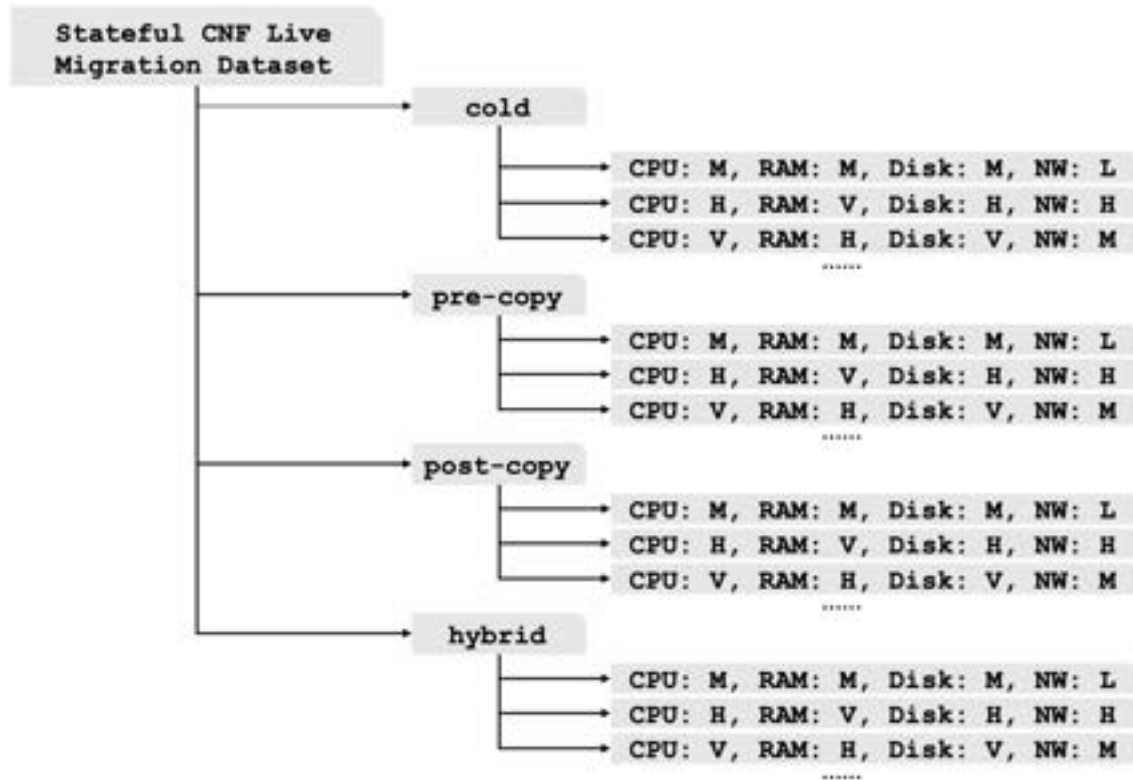


Figure 33: The structure of the CNF Stateful Live Migration dataset.

An overview of the dataset structure is provided in Figure 33, where all four migration types are presented. In addition to the migration type, the data is also divided into different subcategories based on the four resources: CPU, RAM, Disk, and Network (NW). The configuration indicates the resource consumption of the migrated CNF, with the following enumeration:

- 0: No consumption
- L: Low consumption
- M: Medium consumption
- H: High consumption
- V: Very high consumption

In the dataset, a container may have 0, L, M, or H assigned for the NW usage, but for other resources such as CPU, RAM, and Disk, it can only have M, H, or V assigned. For technical reasons, not all 108 (4*3*3*3) combinations are available in the measurements.

Other features available in the dataset are mostly numerical, indicating the time or size (bytes) related to the migration process. The following is a list of major features:

- dump_size

- dump_time
- dump_transfer_time
- restore_time
- dump_speed
- approx_downtime
- locust_downtime (measured by the *locust* tool)

In addition, there is also a column named `best_method`, which indicates the migration method/type that incurs the least downtime for a CNF with the corresponding set of resource configuration. Since this metric is aggregated across resource configuration, all rows that have the same unique value for the tuple (*CPU*, *RAM*, *Disk*, *NW*), have also the same value in the `best_method` field.

Table 18: The distribution of migration types in the dataset

Migration Method/Type	Number of Samples/Migrations
Cold	1546
Pre-copy	1510
Post-copy	1436
Hybrid	1398
Total	5890

The distribution of migration methods in the dataset is given in Table 18 Table 18 where it is shown that a total of 5890 migrations are available in the dataset. Although each migration type has a similar number of samples, the number of CNF migrations with cold and pre-copy methods are slightly higher than those with post-copy and hybrid methods. This is because the post-copy method, and consequently the hybrid method since it involves a phase with post-copy, has a higher risk of state corruption during the migration process, making it less reliable than cold or pre-copy techniques. This risk is also reflected in the experiments, as more migrations involving post-copy have failed, necessitating their removal from the dataset.

6.6 Channel Estimation for sub-THz bands

This dataset collection contains generated TDD channel data for three indoor scenarios, all with Alice (BS), Bob (UE) and Eve (as a passive eavesdropper), at a center frequency of 95GHz. It was created to support the development and evaluation of AI-based channel reciprocity compensation within a Physical Key Generation (PKG) pipeline. All the data was generated using QuaDRiGa in MATLAB under the 3GPP_38.901_Indoor channel model, considering both LOS and NLOS propagation conditions.

For each simulation case, the complex CFR (Channel Frequency Response) is estimated at the OFDM receiver after propagating a reference signal through the QuaDRiGa channel. Alice and Bob share the same physical channel separated by the time TDD delay, emulating the reciprocity impairment of a half-duplex TDD system. Eve's channel is generated independently. The dataset is structured as three variants of increasing complexity: 3GPP_38.901_Indoor_LOS, 3GPP_38.901_Indoor_NLOS and 3GPP_38.901_Indoor_NLOS_AWGN.

Is important to mention that each uploaded dataset is a portion (due to space limitations) of the real dataset used in the training of the channel estimation algorithm. The first variant corresponds to a Line-of-Sight (LOS) indoor environment. In this scenario, Alice and Bob have direct visual contact, producing a channel dominated by a single strong ray. This dataset serves as a baseline for evaluating reciprocity under ideal propagation conditions, where the channel structure is simple and highly stable across snapshots. The second variant models a Non-Line-of-Sight (NLOS) indoor environment, where obstacles between Alice and Bob introduce multipath propagation, making the reciprocity compensation task more challenging and the resulting PKG keys more secure against passive eavesdropping by Eve. The third variant extends the NLOS scenario by adding Additive White Gaussian Noise (AWGN) at six controlled SNR levels: 5, 10, 15, 20, 25 and 30 dB. This dataset enables evaluation of the AI model robustness under degraded channel estimation quality. All three options share the same OFDM numerology: 128 subcarriers, subcarrier spacing of 60 kHz (total bandwidth 7.68 MHz), cyclic prefix of 12 samples, and a TDD delay.

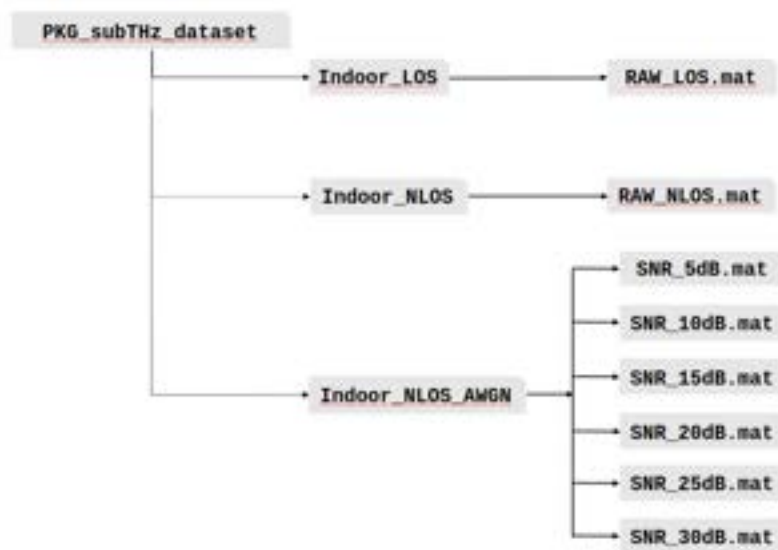


Figure 34: SubTHzPKG_Dataset_Structure

Figure 34 shows the structure of the Channel Estimation for the sub-THz PKG dataset. The dataset is organized in three variants of increasing complexity. The first two variants,

3GPP_38.901_Indoor_LOS and 3GPP_38.901_Indoor_NLOS, each consist of a single .mat file containing 1,000 channel realizations with 116 snapshots per case, for a total of 116,000 snapshots per variant. The third variant, 3GPP_38.901_Indoor_NLOS_AWGN, extends the NLOS scenario with additive white Gaussian noise at six SNR levels (5, 10, 15, 20, 25, and 30 dB). All files share the same internal structure, described in the following tables:

Table 19: Main parameters of Channel Estimation Dataset.

Attribute	Description
Total Size	1.63GB
Data Type	Complex single-precision floating point (I, Q samples)
Structure	2D matrix (subcarriers × snapshot)
Scenarios	3GPP_38.901_Indoor_LOS, 3GPP_38.901_Indoor_NLOS, 3GPP_38.901_Indoor_NLOS_AWGN
3GPP_38.901_Indoor_LOS	
Total files	1
Total Snapshots	116000
3GPP_38.901_Indoor_NLOS	
Total files	1
Total Snapshots	116000
3GPP_38.901_Indoor_NLOS_AWGN	
Total files	6
Total Snapshots	139200

Table 19 summarizes the structure of the Channel Estimation Dataset. The LOS and NLOS scenarios contain the same number of measurements, which allows a fair comparison between both propagation conditions. The AWGN option extends the NLOS case by adding different SNR levels, making it a useful tool to evaluate the robustness of estimation process under noisy conditions.

Table 20: Statistical Metrics.

Metric	LOS	NLOS
Mean Power Alice	8.351e-10	5.117e-11
STD Power Alice	1.038e-09	2.301e-10
Mean Power Bob	8.350e-10	5.119e-11
STD Power Bob	1.037e-09	2.302e-10
Mean Power Eve	8.385e-10	5.184e-11
STD Power Eve	9.686e-10	2.126e-10

Additionally, Table 20 shows the statistical power metrics for Alice and Bob in both scenarios. The LOS case presents higher values than the NLOS case, as expected due to the presence of a direct propagation path. In contrast, the NLOS scenario exhibits some attenuation, resulting in lower values.

6.7 DetAction jamming Dataset

This uploaded dataset is a portion (due to space limitations) of the original dataset used to train the CNN that serves as the jamming detection phase algorithm in the DetAction component. Each sample of the dataset was originally obtained from IQ samples of a real signal, captured using a USRP. Those IQ samples were postprocessed to clean and resample them. After that, an STFT with a fixed NFFT was used to obtain the spectrum of the whole bandwidth, and it was fragmented into PRBs, each one represented with 8192 frequency samples (each one of real and imaginary parts). Those fragments are finally normalized in mean and variance before being used to train the CNN.

This process is done for 5G signals (both from our lab using Amarisoft in the n78 band and the closest 5G radio station) and also to the jamming ones, differing only in the “cleaning” step, as the jammer was not affecting the whole recorded bandwidth and some portions of it were discarded as they would introduce noise to the classifier if they were labelled as “jamming”.

The jamming class fed to the CNN is represented by pure jamming recorded in the lab and also by legit 5G signals contaminated using jamming (this contamination is reproduced by adding a jamming signal to a legit one via software, not actually jamming the legit 5G communication due to legal limitations).

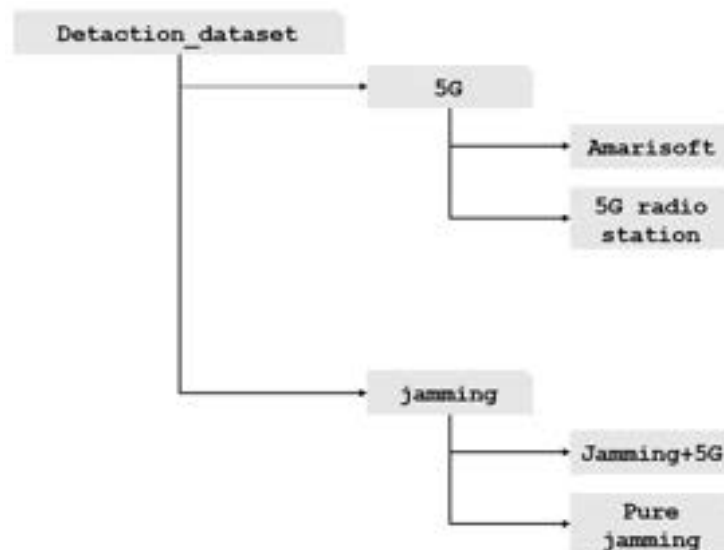


Figure 35: Structure of DetAction dataset.

Figure 35 shows the structure of the DetAction dataset. For the uploaded version, a subsample of 10,000 frequency fragments was randomly chosen (5,000 of the 5G class and the other 5,000 of the jamming class) and saved in .pt format (with a size of 625MB), having a payload described in the next table:

Table 21: Parameters of the .pt file.

Parameter	Description
Total size	625 MB
Data type	I/Q samples
Structure	10000 frequency fragments: 5000 labelled as 5G and 5000 labelled as jamming
Scenarios	Legitimate 5G signals with jamming signals
Total files	1
.pt file parameters	
created_utc	Date of creation
classes	Dict{0: "5G", 1: "jamming"}
signals	frequency fragments
labels	Each corresponding to "classes"
source	Parameters defining the FFT fragments saved

The "source" parameter itself is another Dict containing a set of parameters used to define the dataset:

- "step": jump between consecutive frequency fragments extracted from the whole bandwidth, it ranges from $0 < \text{step} < 1.0$, where 1.0 makes consecutive fragments independent, but for example 0.7 makes it so that a 30% of the spectrum is shared between two consecutive fragments. It is set to 0.7
- "fragment_size": 8192 samples per fragment
- "nfft": number of points of the whole bandwidth STFT
- "hop": similar to "step," but for the time domain, it defines the shared time samples between STFT
- "sample_rate": final sample rate used (IQ signals that were not recorded using it are resampled)
- "scs_hz": OFMD subcarrier spacing in 5G (15 kHz)
- "norm_fragments": flag to normalize the mean and variance of the fragments in training, set to True

6.8 Jamming Detection

The dataset presented in this section contains metrics collected from the gNodeB (srsRAN²) during experiments in which the legitimate link between the UE and the gNB was deliberately jammed. The experimental conditions vary across datasets according to the parameters described below.

Traffic type: Each experiment was conducted under one of two traffic profiles: TCP Live Video Streaming or TCP Continuous File Download. TCP was selected in both cases to maximize the number of exchanges over the air interface, thereby providing a richer set of observable metrics.

Jammer output power: The jammer was operated at three power levels, expressed as a fraction of its maximum output: 1.0 (100%), 0.75 (75%), and 0.5 (50%).

Faraday cage: All jamming experiments were conducted with the UE placed inside a small Faraday cage. In addition, the cage was used independently to simulate a channel outage condition: confining the UE within the cage while the gNB remained outside attenuated the received signal without any active jamming. This allowed a comparison between the effects of deliberate interference and those of a severely degraded channel.

Bandwidth and PRB configuration: Unless otherwise stated, the experiments were conducted on band n78 with a 20 MHz channel bandwidth, corresponding to 54 Physical Resource Blocks (PRBs). In some datasets, only 24 PRBs were observed; this is explicitly indicated where applicable.

Table 22 summarises all collected datasets, specifying for each the traffic type, jammer gain, environmental conditions, and bandwidth configuration.

Table 22: Summary of the collected datasets.

Filename	Traffic Type	Jammer Gain	Environment	Bandwidth /PRBs	Notes
20260202_162101_001010000000002_gnb.csv	TCP Live Video Streaming	1.0 (100%)	Normal	10 MHz	—
20260202_165846_001010000000002_gnb.csv	TCP Live Video Streaming	0.5 (50%)	Normal	10 MHz	—
20260202_172622_001010000000002_gnb.csv	TCP Live Video Streaming	0.75 (75%)	Normal	10 MHz	—

² <https://www.srsran.com/>

Filename	Traffic Type	Jammer Gain	Environment	Bandwidth /PRBs	Notes
20260122_100934_001010000000002_gnb.csv	TCP Live Video Streaming	1.0 (100%)	Normal	24 PRBs	No signal during jamming
20260211_090027_999700000143915_gnb.csv	TCP Live Video Streaming	1.0 (100%)	Cage	10 MHz	—
20260211_121737_999700000143915_gnb.csv	TCP Live Video Streaming	1.0 (100%)	Cage	10 MHz	—
20260211_143309_999700000143915_gnb.csv	TCP Continuous File Download	1.0 (100%)	Cage	10 MHz	—
20260211_144530_999700000143914_gnb.csv	TCP Continuous File Download	0.8 (80%)	Cage	10 MHz	—

Packets on the PUSCH channel are transmitted between the UE and the base station. SINR (Signal-to-Interference-plus-Noise Ratio) in dB and carrier frequency offset (cfo [Hz]) are shown over time (seconds). The drop in signal quality caused by a broadband jammer is indicated by the highlighted bands.

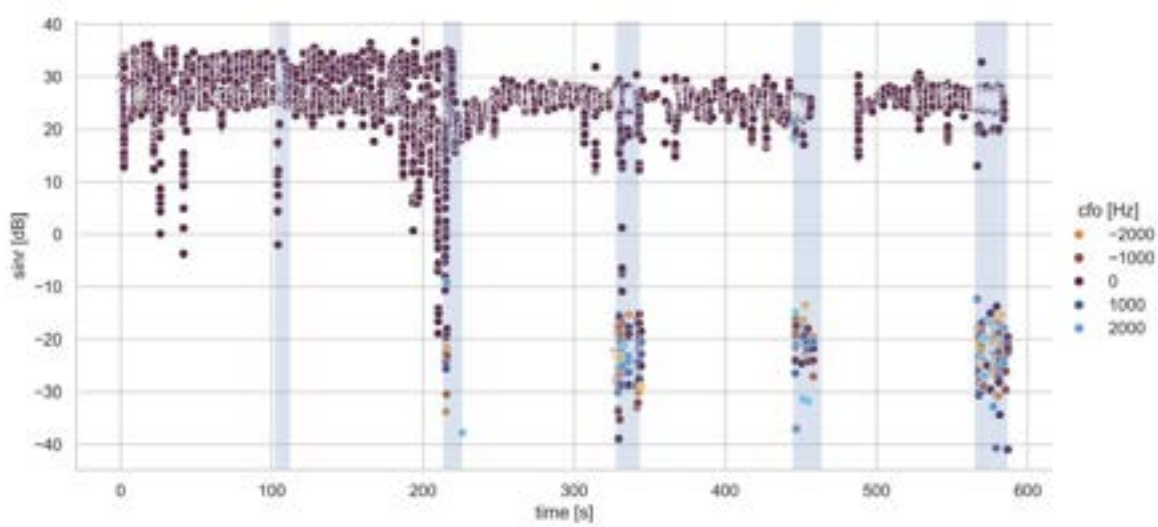


Figure 36 Carrier Frequency Offset over the PRBs. White bands represent unjammed windows, while grey-shadowed bands represent jammed windows.

6.9 DDoS Detection

The dataset `network_hes-so_ddos_dataset_0000_09122025.zip` constitutes an initial dataset for the project aimed at the early detection of Distributed Denial-of-Service (DDoS) attacks, in particular TCP SYN attacks generated by a Mirai botnet. It should be regarded as an initial working dataset, designed to initiate methodological developments, validate the first processing pipelines and provide an initial basis for experimentation. The dataset does not, at this stage, constitute a final reference corpus. Its main value lies in its role as a structured starting point for the construction of a more comprehensive collection of datasets, which is already in preparation, with datasets announced as being better balanced and more representative of conditions observed in the field.

In terms of content, the dataset covers seven capture scenarios, each lasting 600 seconds, collected between 8 and 10 September 2025. It includes scenarios of isolated or combined legitimate traffic, including HTTP, MQTT, RTSP and UDP, as well as standalone attack scenarios (`mirai-syn`) and a mixed scenario combining normal and malicious traffic (`http+mqtt+rtsp+udp+mirai-syn`). The observed disk volume is approximately 39 GB. The asset contains 7 main PCAP captures, 234 split PCAP files, 234 CSV files and 234 Parquet files.

The data has already reached an intermediate stage of processing. Raw captures are available as PCAP files, and a transformation into network flows has been performed with `CICFlowMeter4`, producing CSV and Parquet exports based on the source IP, source port, destination IP and destination port quadruplet. This structure is particularly useful for modelling work, as it allows for a rapid transition to classification or detection approaches based on aggregated features. However, the labelling remains incomplete from the perspective of fine-grained supervised learning: even though the malicious scenarios are known, the exports still contain a generic label such as `'NeedManualLabel'`, meaning that the dataset is not yet fully ready for rigorous multi-class supervised analysis without an additional labelling step.

One of the major limitations of this initial dataset is the significant imbalance between benign traffic and attack traffic. The exclusively benign scenarios account for a total of approximately 1,033 streams, comprising 101 HTTP streams, 149 MQTT streams, 219 RTSP streams, 221 HTTP+MQTT streams and 343 HTTP+MQTT+RTSP streams. Conversely, the `mirai-syn` scenario alone accounts for approximately 18,196,079 flows, whilst the mixed `http+mqtt+rtsp+udp+mirai-syn` scenario contains approximately 16,517,427. This contrast clearly shows that this initial corpus is useful for demonstrating the feasibility of initial detection, but that it remains insufficient to produce robust learning that is representative of real-world environments. It is precisely for this reason that other datasets are currently being developed, with the explicit aim of achieving more balanced distributions and scenarios that more closely resemble operational conditions.

This observation is reinforced by initial inference tests conducted on this dataset using a model trained on a synthetic dataset provided by the CIC in Canada. The results show that binary detection—that is, distinguishing between attack traffic and non-malicious traffic—is relatively satisfactory. However, multi-class classification does not work correctly: traffic flows are misclassified and confusion between classes remains significant. This observation highlights a classic limitation of transfer between synthetic data and real or quasi-real data: a model capable of detecting an anomaly in general terms does not necessarily succeed in correctly recognising the precise nature of traffic flows when confronted with traffic acquired in a real-world test bed. The heatmap associated with this experiment visually illustrates these classification errors and should be used to interpret the results (Figure 37).

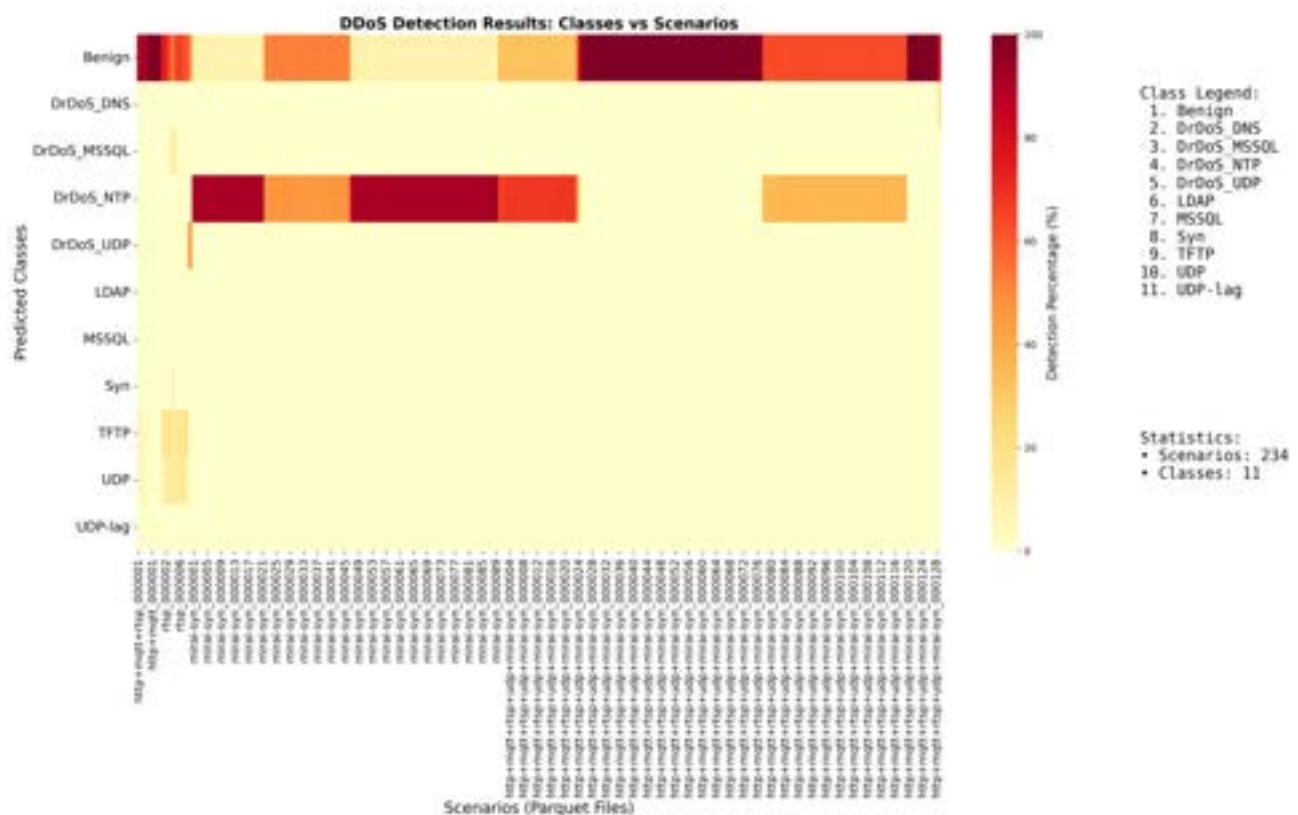


Figure 37: Heatmap DDoS Detection results.

In summary, this dataset fully fulfils its role as the project's first dataset: it enables us to structure the work, validate processing workflows, test initial detection hypotheses and objectively assess the limitations of approaches trained on synthetic data. Its maturity can be described as intermediate: the dataset is available, documented and technically usable, but it still has significant limitations in terms of balance, representativeness and fine-grained labelling. The scientific and operational value of this initial corpus therefore lies as much in what it already

enables as in the limitations it reveals, which directly justify the creation of new datasets that are more realistic, better balanced and more closely aligned with real-world conditions.

6.10 Edge-IIoT Adversarial Dataset

The Edge-IIoT Adversarial Dataset is a specialized extension of the Edge-IIoTset dataset [67] designed to evaluate the robustness of machine learning-based Intrusion Detection Systems (IDS) against adversarial manipulation in Industrial Internet of Things (IIoT) environments. As IIoT systems increasingly rely on edge computing for low-latency processing, they become vulnerable to adversarial examples—maliciously perturbed inputs crafted to deceive AI models. To address this challenge, the dataset integrates authentic IIoT traffic with adversarial samples generated using the Elastic-Net Adaptive Attack (EAD). This combination provides a rigorous benchmark for evaluating resilient anomaly detection mechanisms under realistic and adversarial conditions.

The dataset is generated within a multi-layer IIoT testbed comprising sensing, edge processing, networking, and control components.

1. **Physical Layer:** Raspberry Pi 4 devices and sensors (DHT11/22, ultrasonic sensors, pH meters) generate real-world telemetry.
2. **Infrastructure Layer:** Managed using Software-Defined Networking (SDN) via ONOS and fog computing via ThingsBoard.
3. **Communication Protocols:** MQTT, Modbus TCP/IP, CoAP, AMQP, and HTTP/TCP are supported.
4. **Data Capture:** Network packets are captured at multiple interfaces and transformed into a structured TCP header-based representation.

The resulting feature space is intentionally compact to enable high-speed inference in resource-constrained edge environments.

C. Traffic Composition

The dataset is organized into three balanced classes, as illustrated in Figure 38. At a high level, it consists of two main categories: legitimate and adversarial traffic. The adversarial category is further divided into two subclasses based on the magnitude of perturbations.

Specifically:

- **Legitimate Traffic:** Normal IIoT communication patterns, including telemetry and control signals.
- **Low-Magnitude Adversarial ($\epsilon = 0.25$):** Subtle perturbations designed to evade sensitive detection models.
- **High-Magnitude Adversarial ($\epsilon = 0.50$):** Stronger perturbations targeting robustness boundaries of IDS models.

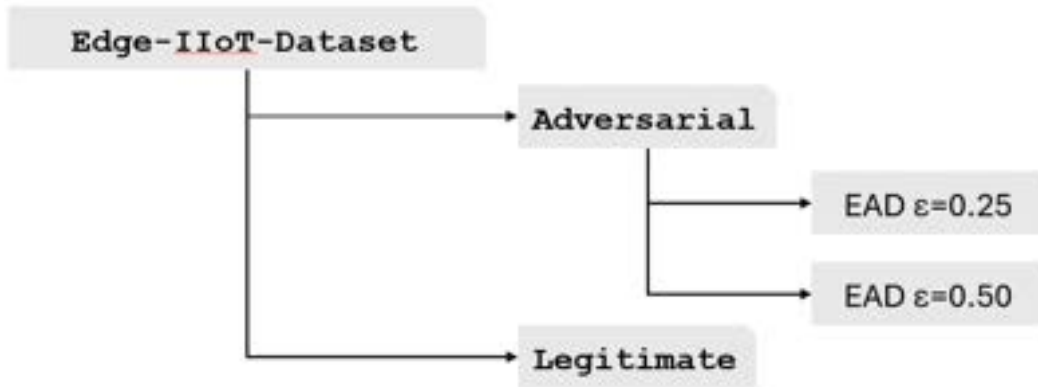


Figure 38: Structure of the Edge-IIoT Adversarial Dataset.

As shown in Figure 39, the three classes are balanced. A balanced distribution is adopted to prevent training bias and ensure fair evaluation across classes.

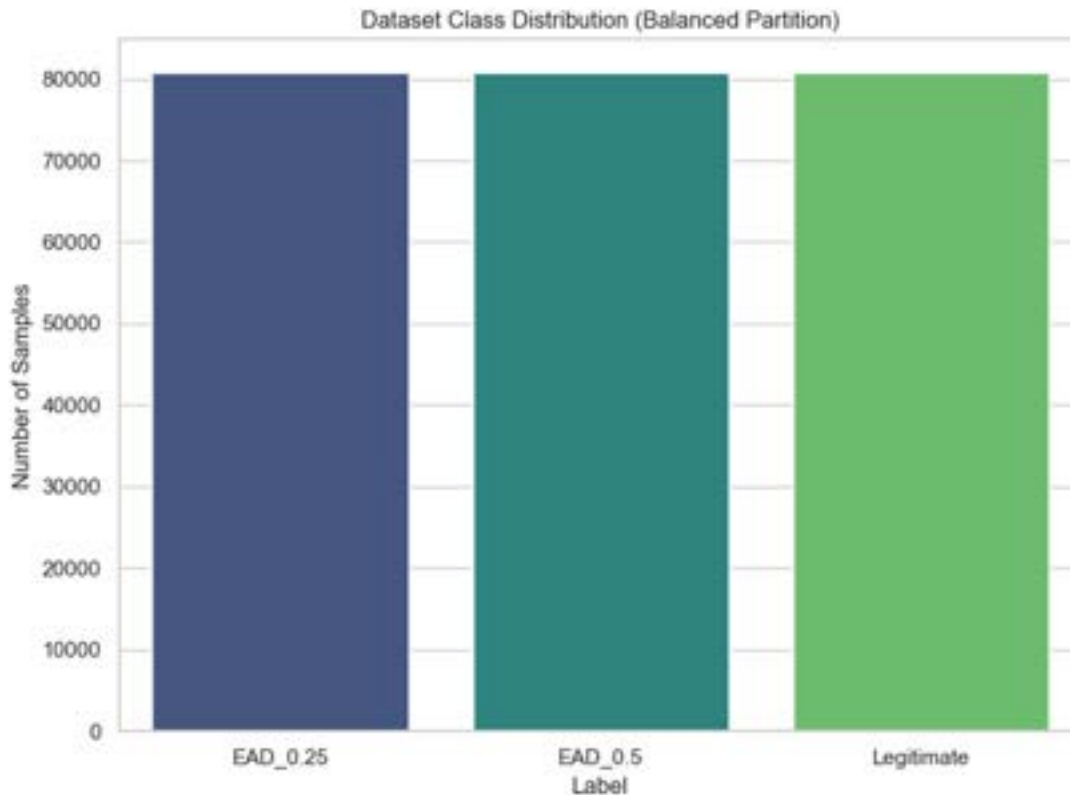


Figure 39: Dataset Class Distribution.

The dataset includes a diverse set of attacks covering multiple threat vectors:

- **DoS/DDoS Attacks:** High-volume traffic floods (TCP, UDP, ICMP) targeting resource exhaustion.
- **Reconnaissance Attacks:** Port scanning and system fingerprinting.

- **Man-in-the-Middle (MITM):** ARP spoofing and DNS poisoning.
- **Injection Attacks:** SQL injection (SQLi) and Cross-Site Scripting (XSS).
- **Malware/Backdoor Activity:** Unauthorized access and persistent compromise patterns.

These attacks reflect realistic threats targeting IIoT infrastructures across multiple layers.

The dataset is provided in CSV format with five key TCP header features described in Table 23.

Table 23: Edge-IIoT Adversarial Dataset illustrating TCP header features and corresponding labels.

Packet index	Acknowledgment number	Sequence number	Payload length	TCP control flags	label
1	3456721	9823412	20	18	Legitimate
2	3456800	9823500	0	2	EAD_0.25
3	3457000	9823700	21	16	EAD_0.5

Adversarial samples are generated using EAD, optimizing both $L1$ and $L2$ norms. Importantly, perturbations are constrained to preserve TCP protocol semantics, ensuring that modified packets remain valid and transmittable across the network stack.

These features capture:

- **Connection behavior (TCP control flags)**
- **Session dynamics (Sequence number, Acknowledgment number)**
- **Traffic characteristics (Payload length)**

This lightweight representation is well-suited for edge-based IDS deployment.

Statistical analysis provides insights into the underlying properties of the dataset. Figure 40 shows the payload length distribution across classes. Adversarial samples introduce subtle shifts while maintaining overlap with legitimate traffic, highlighting their stealthy nature. Figure 44 presents the feature correlation heatmap. A strong negative correlation between sequence number and payload length indicates structured traffic behaviour, while a moderate correlation between sequence numbers and flags reflects temporal dependencies in TCP state transitions.

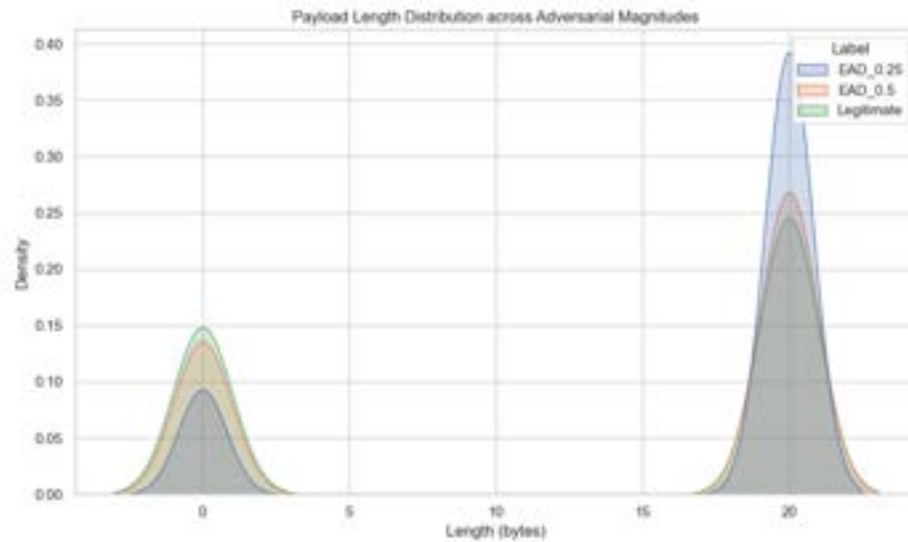


Figure 40: Payload Length Distribution across Classes.

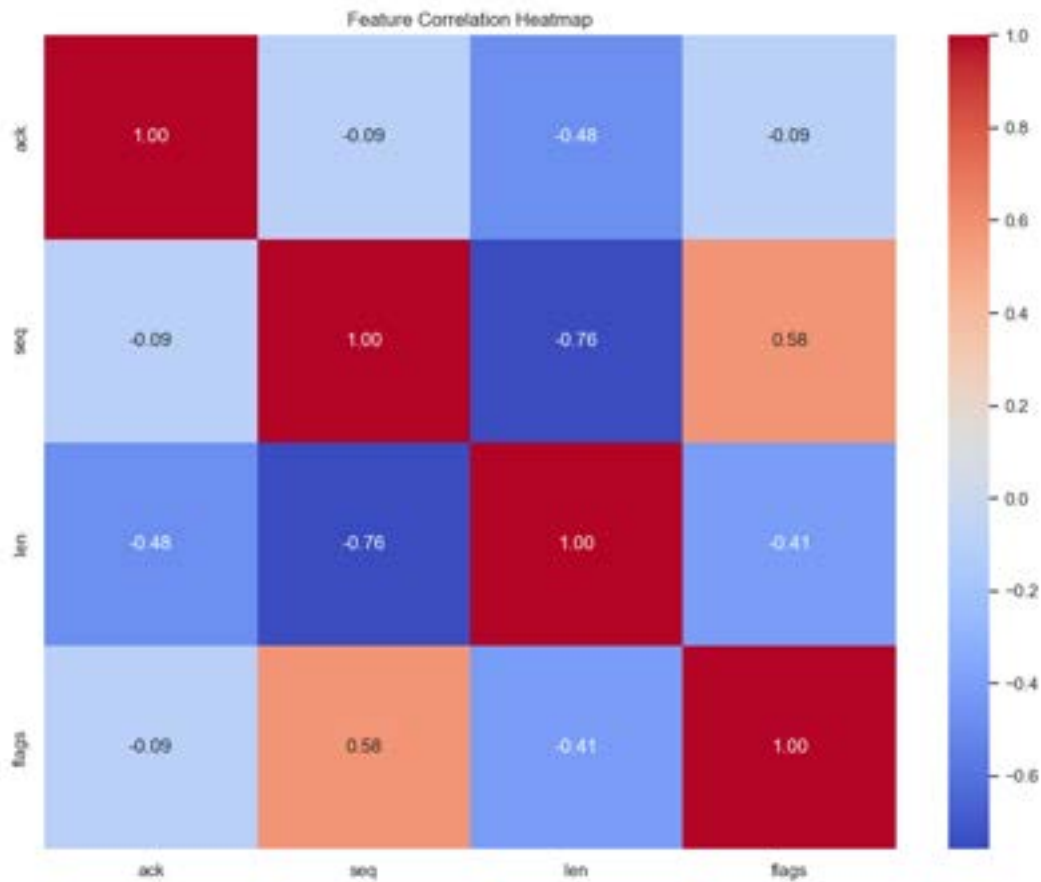


Figure 41: Feature Correlation Heatmap.

The Edge-IIoT Adversarial Dataset addresses a critical gap in IIoT security by enabling:

1. Evaluation of IDS robustness against adversarial attacks

2. Development of adversarial training strategies
3. Deployment of lightweight detection models for edge environments

Unlike traditional datasets, this dataset explicitly incorporates adversarial perturbations, making it highly relevant for next-generation AI-driven cybersecurity systems.

6.11 5G Network Traffic dataset

This dataset contains network traffic captured in an emulated hybrid 5G testbed environment built with Mininet and virtual machines. It includes both benign traffic and flooding attack traffic, allowing the analysis of normal and malicious behaviour under multiple 5G interface conditions. The captures were collected at multiple observation points in the emulated system. Traffic on the N3 interface was recorded at r0 inside the GTP tunnel, traffic on the N6 interface was recorded at r1 after the UPF, and N4 control traffic was captured separately on the Control VM. All packet traces are stored in PCAP format, truncated to a 96-byte snap length, and anonymized via MAC address pseudonymization. These design choices make the dataset practical for reproducible experimentation on attack detection, traffic characterization, and comparative analysis of benign and flooding scenarios. The dataset is divided into two main folders: benign and flooding. Each PCAP file corresponds to one traffic scenario and one capture point. The benign subset contains behaviourally modelled application traffic representing web browsing, streaming, gaming, and ping activity. The flooding subset contains attack traffic targeting different interfaces of the 5G architecture, namely N2, N3, N4, and N6. The overall dataset structure is illustrated in Figure 42.

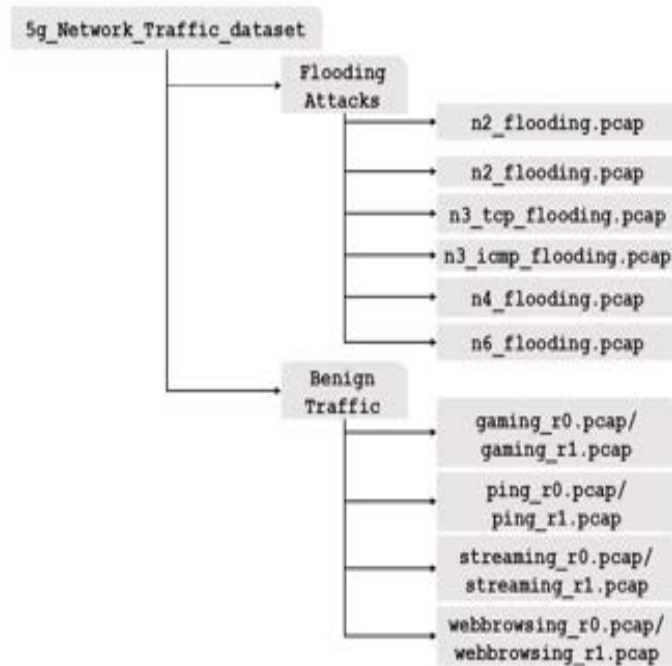


Figure 42: Structure of 5G Network Traffic dataset

Table 24 summarizes the main structural and technical characteristics of the dataset. In total, the collection includes fourteen PCAP files, split into eight benign captures and six flooding captures. The traces were collected at multiple points of the emulated 5G environment in order to preserve the protocol context of each scenario and to enable interface-specific analysis.

Table 24: Main parameters of the 5G Network Traffic dataset.

Parameter	Value
Dataset type	Network traffic dataset captured in an emulated hybrid 5G testbed
File format	PCAP (Wireshark/tcpdump compatible)
Total number of files	14
Benign files	8
Flooding files	6
Capture points	r0 (N3, inside GTP tunnel), r1 (N6, after UPF), Control VM (N4)
Link layer	Linux cooked-mode capture v1 (linux-sll)
Snap length	96 bytes
Anonymization	MAC address pseudonymization (seed = 42)
Traffic categories	Benign and flooding attack traffic

The benign subset was generated to approximate realistic service and user behaviour in the emulated environment. The traffic generation process was behaviourally modelled rather than

purely synthetic, so that the resulting traces reflect more plausible protocol dynamics and timing variations (cf. Table 25).

- **Web browsing:** generated with Puppeteer and Chromium against a local NGINX server, including randomized reading times and tab open/close actions.
- **Mobile gaming:** generated with a custom TCP socket-based client/server implementation with Gaussian-randomized jitter and spikes to emulate interactive gaming conditions.
- **Streaming:** generated through concurrent real video and audio streams executed in parallel threads.
- **Ping:** generated using standard ICMP echo traffic.

Table 25: Statistical metrics of the data per case and modulation

File	Scenario	Packets	Scenario Duration	Size
gaming_r0.pcap	GTP/TCP Sockets	1,793,742	14,400 s	200 MB
gaming_r1.pcap	TCP Sockets	1,790,424	14,400 s	165 MB
ping_r0.pcap	GTP/ICMP	42,916	14,400 s	4.806 MB
ping_r1.pcap	ICMP	38,236	14,400 s	4.282 MB
streaming_r0.pcap	GTP/UDP	14,583,521	14,400 s	1,407 MB
streaming_r0.pcap	UDP	10,547,352	14,400 s	1,099 MB
webbrowsing_r0.pcap	GTP/HTTP	4,620	14,400 s	0.517 MB
webbrowsing_r1.pcap	HTTP	790	14,400 s	0.071 MB

The flooding subset captures malicious traffic targeting several key interfaces of the 5G architecture. It includes SCTP, GTP/UDP, GTP/TCP, GTP/ICMP, PFCP, and TCP flooding scenarios. Most flooding scenarios were recorded for 600 seconds. However, some captures were shortened because very large files exceeded practical resource limits during cleaning and dataset preparation. Specifically, the N3 TCP flooding and N6 flooding traces were reduced to 180 seconds, while the N3 ICMP flooding trace was reduced to 360 seconds (

Table 26).

Table 26: Statistical metrics of the data per case and modulation

File	Scenario	Packets	Scenario Duration	Size
n2_flooding.pcap	SCTP Flooding	40'795'292	600 s	4'079 MB
n3_udp_flooding.pcap	GTP/UDP Flooding	8'763'825	600 s	981 MB
n3_tcp_flooding.pcap	GTP/TCP Flooding	8'293'347	180 s	928 MB
n3_icmp_flooding.pcap	GTP/ICMP Flooding	23'576'899	360 s	2'640 MB

File	Scenario	Packets	Scenario Duration	Size
n4_flooding.pcap	PFCP Flooding	1'761'924	600 s	151 MB
n6_flooding.pcap	TCP Flooding	4'983'055	180 s	358 MB

The captures use the PCAP format and are compatible with standard packet analysis tools such as Wireshark and tcpdump. The link layer is Linux cooked-mode capture v1 (linux-sll). Packet truncation to 96 bytes was intentionally applied to remove application payload while preserving the information needed for protocol-level traffic analysis. The dataset contains different encapsulation contexts depending on the interface: GTP-U tunnelled traffic on N3, raw traffic on N6, SCTP on N2, and PFCP on N4. Because the environment is synthetic and emulated, no IP remapping was required during anonymization. several technical observations are noteworthy. Gaming traffic shows retransmissions due to emulated jitter and spikes. SCTP packets are visible only at r0, between the 5GCN and the gNB, and not at r1. Streaming traffic shows IPv4 fragmentation caused by MTU limits, and some fragments appear outside the GTP tunnel. In addition, the N4 flooding attack reportedly succeeded after only two PFCP packets, suggesting that the control-plane behaviour deserves closer attention.

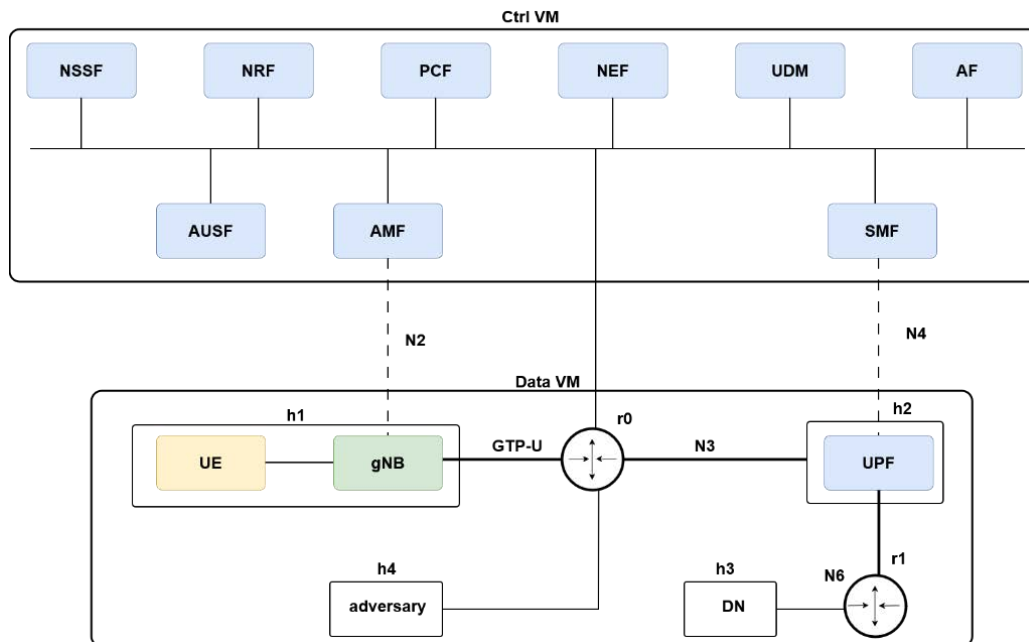


Figure 43: 5G testbed used to generate the flooding-attack dataset, showing the UE, gNB, 5G core functions, UPF, data network, and adversarial node, together with the monitored interfaces.

Figure 43 shows the emulated hybrid 5G testbed used for data generation. It identifies the main network components and the capture points, including the N3 side (r0), the N6 side (r1), and the control-plane monitoring point. This interface-aware setup explains the different traffic characteristics observed in the dataset. Figure 44 illustrates the representative benign packet-rate trace for the web-browsing scenario. This traffic was generated through browser-based

activity and represents normal user interaction with a web service. The packet-rate pattern is sparse and burst-oriented, reflecting short periods of page loading followed by idle intervals that correspond to reading time or inactivity between browsing actions.

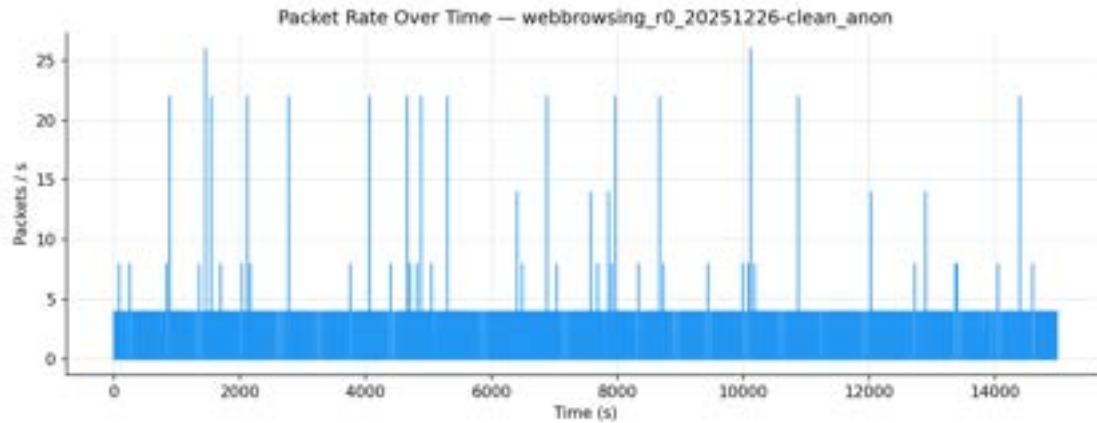


Figure 44: Representative benign packet-rate traces web-browsing scenarios

Figure 45 shows the representative benign packet-rate trace for the gaming scenario. Gaming traffic is more persistent than ping and web-browsing traffic and contains short-term fluctuations caused by interactive application behavior. In the dataset, this traffic was generated using a TCP socket-based client/server implementation with randomized jitter and spikes to emulate the behavior of real-time gaming communication.

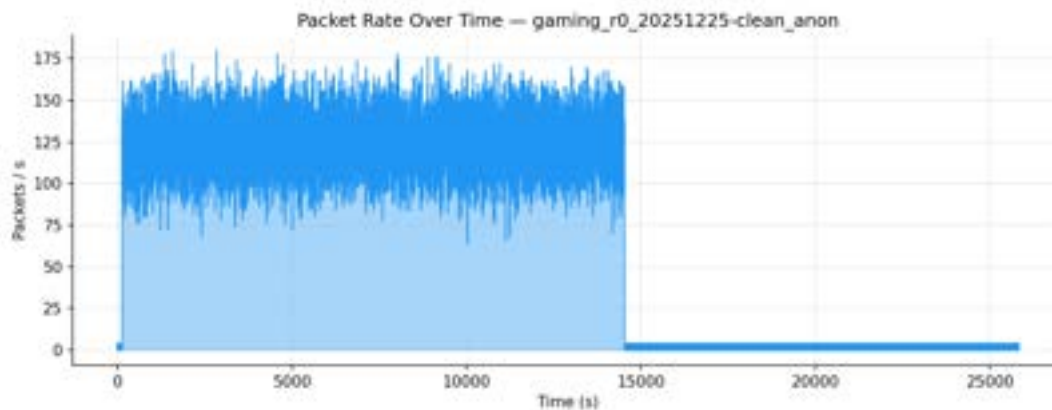


Figure 45: Representative benign gaming traffic traces

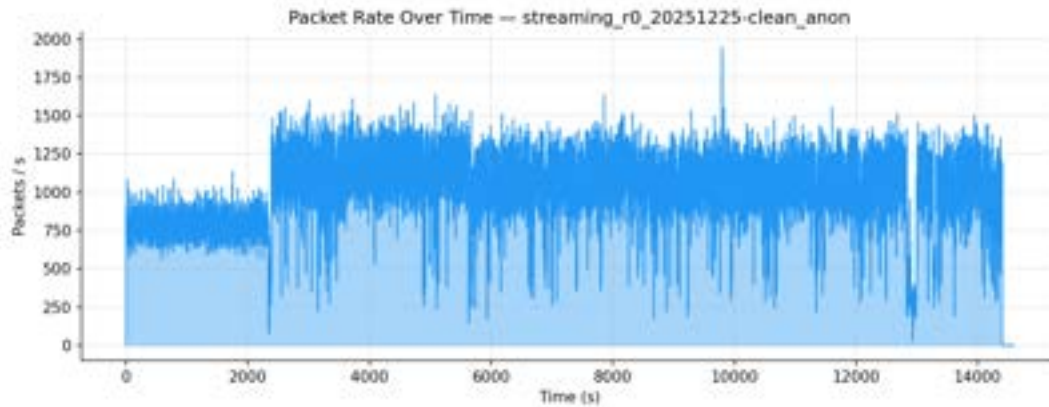


Figure 46: Representative benign streaming traffic traces

Figure 46 presents the representative benign packet-rate trace for the streaming scenario. Streaming traffic produces the highest traffic volume among the benign scenarios in the dataset. Unlike ping and web browsing, which are sparse and intermittent, streaming generates continuous service-oriented traffic due to parallel video and audio transmission. This figure is particularly important because it shows that high-volume traffic is not necessarily malicious. Although streaming may produce a large number of packets, its behaviour remains regular and consistent with normal application use. This makes the streaming trace a key benign reference for comparison with flooding attacks. Keeping this figure helps demonstrate that the dataset contains realistic high-load benign traffic and can therefore support more meaningful evaluation of attack detection methods.

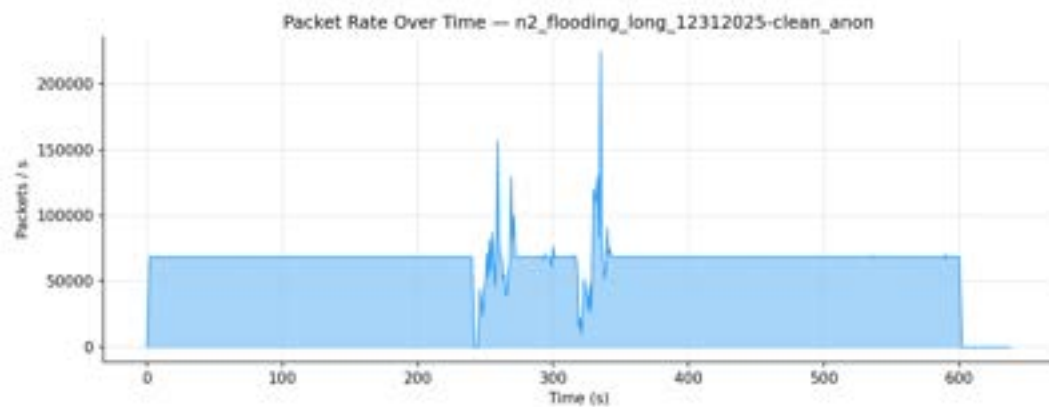


Figure 47: N2 SCTP flooding trace, showing sustained signaling-plane overload behavior during the attack interval

Figure 47 illustrates the N2 SCTP flooding scenario. This attack targets the signaling side of the 5G architecture and produces a sustained overload pattern during the attack interval. Since N2 is associated with signaling communication between the radio access side and the core network, this flooding scenario is different from data-plane attacks observed on N3 or N6.

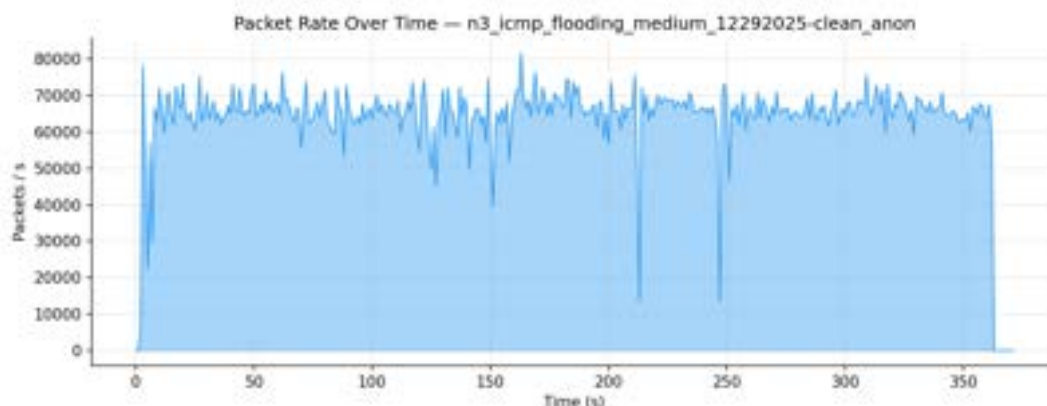


Figure 48: N3 ICMP flooding trace.

Figure 48 presents the N3 flooding scenarios for ICMP traffic inside the GTP-U tunnel. The N3 interface is one of the most important observation points in the dataset because it captures user-plane traffic between the gNB side and the UPF side before decapsulation. Therefore, traffic observed at this point preserves the GTP tunnelling context of the 5G user plane. The three N3 flooding traces show persistently elevated packet-rate behaviour compared with benign traffic.

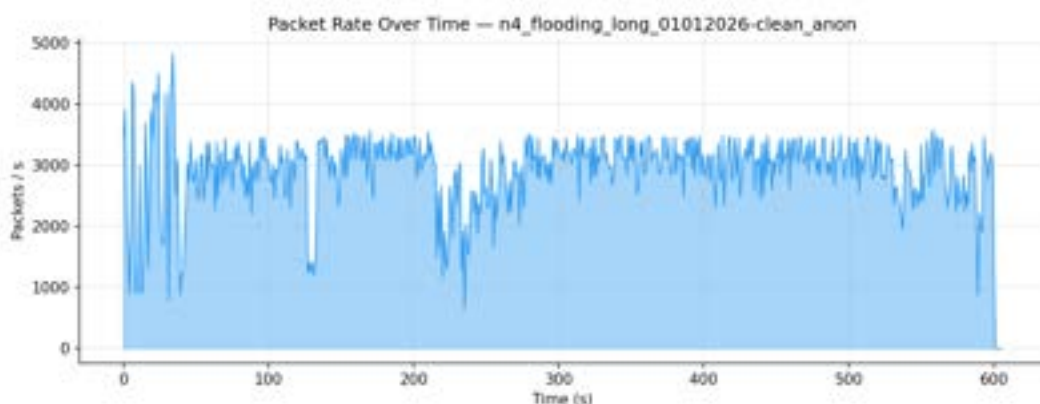


Figure 49: N4 PFCP flooding trace

Figure 49 illustrates the N4 PFCP flooding scenario. N4 is a control-plane interface between the SMF and UPF, and PFCP is responsible for control functions related to user-plane session management. Although the packet rate in this trace may be lower than some of the N2 or N3 flooding cases, the operational importance of this attack is high because it targets a sensitive control-plane protocol.

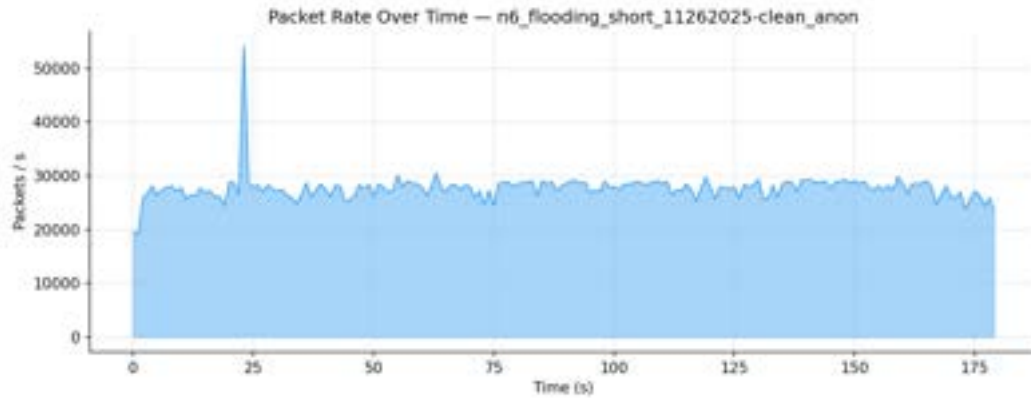


Figure 50: N6 TCP flooding trace captured after the UPF, showing sustained overload behavior on the external data interface.

Figure 50 presents the N6 TCP flooding scenario captured after the UPF. Unlike N3 traffic, which is observed inside the GTP-U tunnel, N6 traffic is captured on the external data-network side after decapsulation. This means that the N6 trace represents attack behaviour as it appears beyond the internal 5G tunnel and toward the data network. This figure is important because it shows that the dataset includes attack visibility at the boundary between the 5G system and the external data network. The sustained overload pattern in the N6 TCP flooding trace is clearly different from benign application traffic and confirms that the dataset covers malicious behaviour outside the tunnelled N3 context.

6.12 OT-ICS-attacks dataset

This section describes the OT-ICS-attacks dataset, which has been generated to support the development and evaluation of AI-based cybersecurity mechanisms for Operational Technology and Industrial Control System environments. The dataset is based on a simulated OT/ICS network that includes a vulnerable web application, OpenPLC, SCADA, and Apache Tomcat services, and it focuses on realistic attack progression in an industrial setting.

The dataset is organized around a shared OT/ICS environment and two main attack scenarios. The first scenario is related to CVE-2021-44228 (Log4Shell), which can enable remote code execution through a vulnerable Log4j-based service. The second scenario is related to CVE-2009-3548, reflecting weak credentials in Apache Tomcat and enabling compromise through malicious WAR file upload. This structure enables the study of attack execution paths, attacker behaviour, and the impact of exploitation steps in a controlled industrial simulation environment. Figure 51 illustrates the structure of the OT-ICS-attacks dataset.

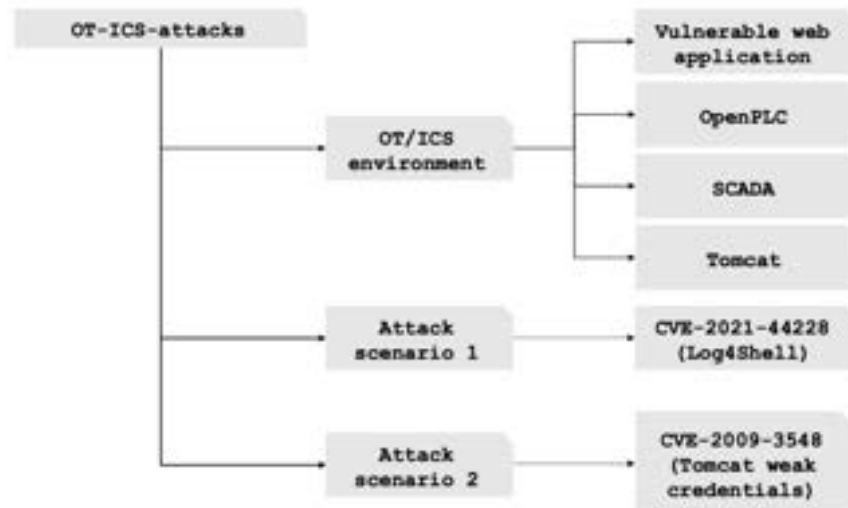


Figure 51: Structure of the OT-ICS-attacks dataset.

From a research perspective, the dataset can support several tasks, including OT/ICS intrusion detection, attack-stage recognition, traffic and event correlation, and resilience-oriented evaluation of industrial services under cyberattack conditions. Its modular structure and explicit vulnerability scenarios make it suitable for benchmarking AI-based detection and analysis methods in industrial cybersecurity research.

- **Apache Tomcat compromise**

This scenario models an intrusion path in which the attacker targets the Apache Tomcat service deployed within the OT/ICS testbed. In this scenario, the attack leverages weak credentials associated with Tomcat, allowing the adversary to gain access to the management interface and deploy a malicious Web Application Archive (WAR) file (Figure 52). Once deployed, the malicious application enables remote code execution on the target host, thereby establishing an initial foothold in the environment.

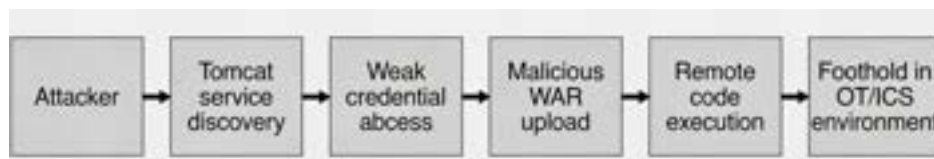


Figure 52: Workflow of the Apache Tomcat compromise scenario in the OT/ICS environment.

The relevance of this scenario lies in the fact that Tomcat coexists with other industrially relevant components, including a vulnerable web application, OpenPLC, and SCADA/ScadaBR services. As a result, this scenario supports the analysis of how a service-level compromise may affect interconnected OT/ICS components and how an attacker can pivot from an exposed application service toward more sensitive industrial assets. This is particularly important for understanding

the attack surface in converged IT/OT environments where web-based management interfaces are often deployed alongside industrial control systems.

From a data generation perspective, it is useful for capturing the network and system activity associated with credential abuse, malicious application deployment, remote code execution, and subsequent post-compromise behaviour. This makes the scenario suitable for research on intrusion detection, attack-stage identification, and resilience evaluation in OT/ICS cybersecurity environments. The dataset enables researchers to study both the technical indicators of compromise (e.g., HTTP traffic patterns, file writes, process creation) and the temporal progression of attacker activities following successful exploitation. Table 27 summaries the key characteristics of Apache Tomcat compromise scenario.

Table 27: Key characteristics of Apache Tomcat compromise scenario.

Characteristic	Description
Scenario name	Apache Tomcat compromise
Target service	Apache Tomcat application server
Primary vulnerability	CVE-2009-3548 (weak credentials, WAR deployment)
Attack vector	Credential abuse and malicious application deployment
Exploitation method	Authenticated access to Tomcat manager, WAR file upload
Initial impact	Remote code execution on Tomcat host
OT/ICS relevance	Service compromise in converged IT/OT environment; potential pivot point to industrial assets
Data types captured	HTTP traffic, authentication logs, application deployment events, process execution, file system activity
Research applications	Intrusion detection, attack-stage recognition, lateral movement analysis, resilience evaluation
Network context	Same environment as vulnerable web application, OpenPLC, and SCADA/ScadaBR services

- **Command-and-control activity**

This scenario represents the command-and-control phase of the attack lifecycle within the OT-ICS-attacks dataset (Figure 53). In this stage, the attacker has already established an initial foothold in the target environment and maintains communication with compromised assets in order to issue instructions, coordinate malicious actions, and preserve operational control over the intrusion. Within an OT/ICS context, this phase is particularly important because it reflects the persistence of an adversary inside an environment where industrial services and control-related components coexist.

The relevance of this scenario lies in its ability to model sustained attacker presence after the initial exploitation stages. Unlike earlier phases that focus on discovery or compromise, the

command-and-control stage reflects the operational continuity of the attack, where compromised hosts may receive follow-up commands, exchange status information, or support additional malicious objectives. In a simulated OT/ICS environment containing services such as a vulnerable web application, OpenPLC, SCADA/ScadaBR, and Apache Tomcat, this phase enables the study of how attacker control channels may persist and interact with industrially relevant infrastructure.

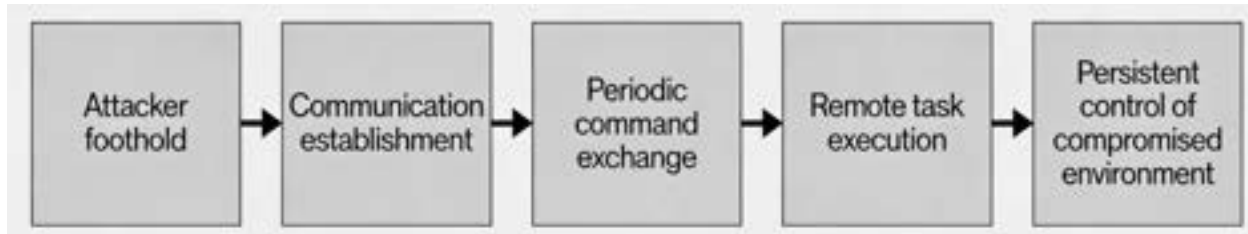


Figure 53: Workflow of the Command-and-control activity scenario in the OT/ICS environment.

From a data generation perspective, this scenario is useful for capturing communication patterns and system activities associated with post-compromise control. These may include periodic network exchanges, remote instruction flows, process execution traces, and other indicators of persistent adversarial activity. As a result, this scenario is highly relevant for research on intrusion detection, attack-stage recognition, behavioral profiling, and resilience evaluation in OT/ICS cybersecurity environments. It is particularly suitable for analysing temporal attack evolution and for developing AI-based methods capable of identifying persistent malicious behavior beyond the initial compromise stage. Table 28 summaries the key characteristics of Command-and-control activity scenario.

Table 28: Key characteristics of Command-and-control activity scenario.

Characteristic	Description
Scenario name	Command-and-control activity
Attack phase	Post-compromise / persistence stage
Primary objective	Maintain attacker presence and enable remote control
Operational role	Exchange commands and status information with compromised assets
Main behavior	Periodic communication, remote tasking, persistent malicious interaction
Initial prerequisite	Successful foothold established in previous attack stages
OT/ICS relevance	Sustained attacker control in an environment containing industrial services and control components
Data types captured	Network communication flows, periodic traffic patterns, process execution traces, post-compromise events
Research applications	Intrusion detection, behavioral profiling, attack-stage recognition, resilience evaluation

Characteristic	Description
Dataset role	Final-stage activity supporting the analysis of persistent adversarial behavior

6.13 Denial of Sustainability Dataset

To address DoSt attacks, NATWORK generated a CSV dataset capturing per-minute metrics, including CPU, memory, network RX/TX, and HTTP request rates, from both proxy and service pods. Each data point is labeled as either benign or an attack, reflecting the systems' operational state. Figure 54 shows the structure of this dataset.

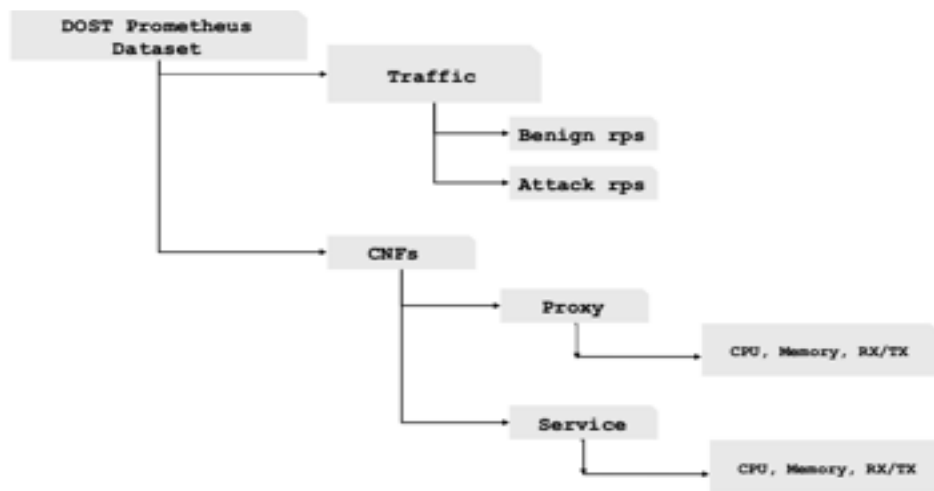


Figure 54: Structure of DoSt Dataset.

The dataset is provided with the following key features, collected per minute and labeled as either benign or attack:

Table 29: DoSt Dataset Feature Description.

Feature	Description	Units/Notes
timestamp	Time of the measurement (used as index)	per minute
proxy_cpu	CPU usage of the proxy (gateway to services)	cores per minute
proxy_memory	Memory usage of the proxy	MB per minute
proxy_rx/ proxy_tx	Network traffic received/transmitted by the proxy	Bytes per second, averaged per minute

Feature	Description	Units/Notes
service_cpu	Aggregate CPU usage of service pods	cores per minute
service_memory	Aggregate memory usage of service pods	MB per minute
service_rx / service_tx	Network traffic received/transmitted by service pods	Bytes per second, averaged per minute
attacker_rps	Requests per second generated by attack users	Averaged per minute
benign_rps	Requests per second generated by benign users	Averaged per minute
label	Class Label	Benign or Attack

When benign and attack users are active in the same time window, the label is marked as attack if at least one attack request is present.

Clear differences between attack and benign patterns can be observed: benign users exhibit realistic, varying workloads, while attack traffic generates sharp bursts and oscillations. A few visualizations are briefly discussed below, based on the overall dataset, to provide an overview of the system behavior under different conditions.

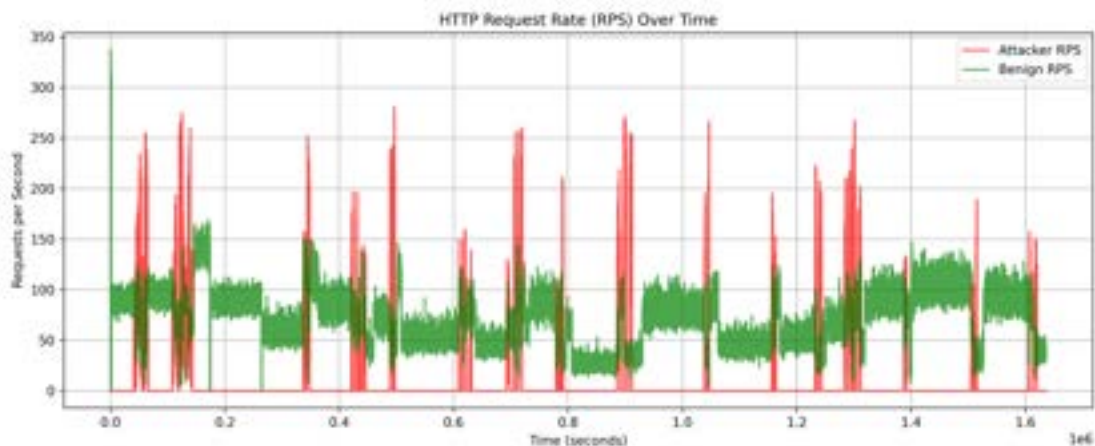


Figure 55: Average Benign and Attack RPS

Figure 55 shows attacker and benign request rates per second (RPS) over time. The benign traffic exhibits relatively smooth and continuous variations, representing realistic user behavior, whereas the attack traffic appears as intermittent high-intensity bursts with sharp spikes, clearly highlighting the oscillatory nature of DoSt patterns.

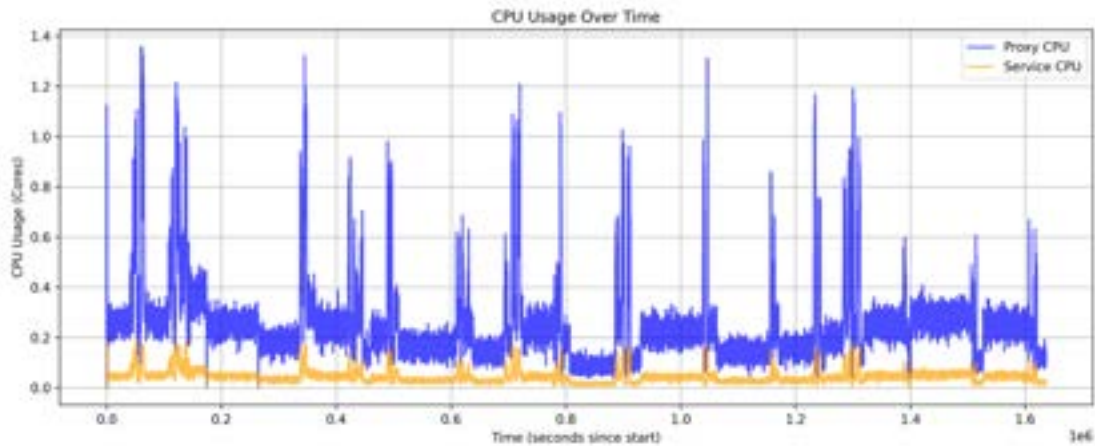


Figure 56: Average CPU usage

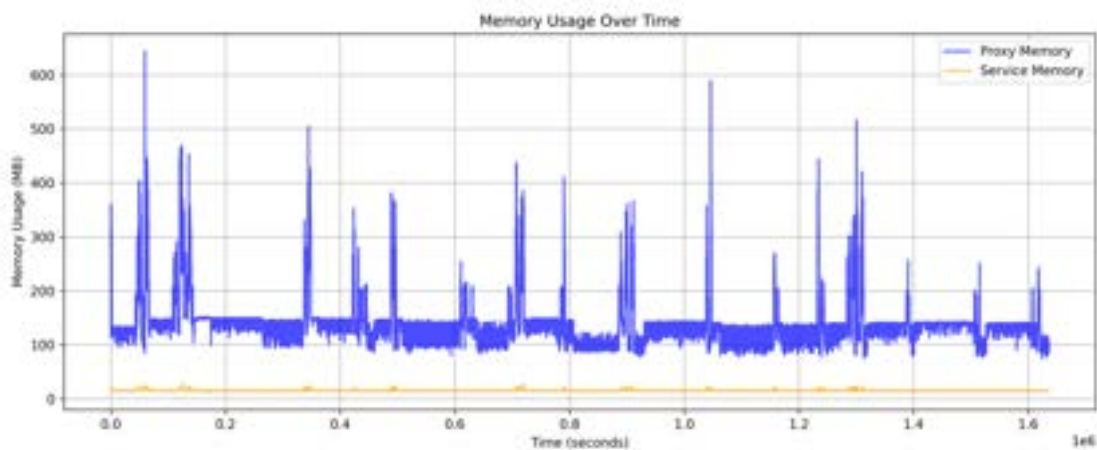


Figure 57: Average Memory Usage

Figure 56 and Figure 57 illustrate the average CPU and memory utilisation of the proxy and service components over time. Under benign conditions, both CPU and memory usage remain relatively stable. However, during attack phases, noticeable spikes in CPU utilisation are observed, reflecting increased processing demand due to elevated request rates. In contrast, memory usage exhibits moderate fluctuations rather than sharp spikes, indicating a more gradual impact on resource consumption. Together, these results highlight how attack conditions primarily stress computational resources while introducing variability in memory behaviour across the system.

This dataset provides high-resolution, per-minute telemetry of application-level traffic and resource utilization under realistic and adversarial conditions. Unlike traditional DoS datasets, it includes oscillatory DoSt attack patterns alongside benign traffic, allowing ML models to learn nuanced differences and patterns between normal and malicious behavior. Its detailed labeling, combined with proxy and service-level metrics, makes it highly suitable for training and

evaluating anomaly detection, intrusion detection, and workload prediction models in edge–cloud environments.

The dataset is available in the NETWORK Data Management Plan (DMP). Further data collection is ongoing, with tests being conducted on additional services and CNFs/slices to enable more realistic processing scenarios.

6.14 Cyber Threat Intelligence dataset

The Cyber Threat Intelligence (CTI) dataset comprises 204 CTI reports, each manually annotated with a corresponding STIX bundle. The reports are sourced from 62 well-known public entities, including Palo Alto Networks, Trend Micro, and Fortinet. On average, each source contributes 3.3 reports (cf. Table 30). Importantly, approximately 79% of the sources are referenced by the MITRE ATT&CK³ framework³ as external citations, confirming the representativeness and relevance of the selected materials within the global CTI landscape.

Table 30: Dataset statistics: number of reports per source, and report length in words and sentences.

Metric	Min	Avg	95p	Max
Reports per source	1	3.3	9	11
Words per report	504	2133.6	4015.8	6446
Sentences per report	11	86.3	172.5	358

The thematic focus of the dataset is summarized in Table 31. Roughly 75% of reports centre around malware, with a significant portion also covering associated threat actors (30%). An additional 15% describe threat actors alone or in combination with vulnerabilities. A minority of reports (10%) address broader topics such as cyber campaigns or threat infrastructure. This distribution ensures coverage across multiple intelligence use cases.

Table 31: Topics covered by the reports in the dataset.

Topic	Quota	Group
Malware	30%	
Malware + Threat Actor	30%	
Malware + Threat Actor + Vulnerability	8%	
Malware + Vulnerability	7%	75%
Threat Actor	11%	
Threat Actor + Vulnerability	4%	15%
Others (e.g., campaigns, infrastructure)	10%	10%

³ <https://attack.mitre.org>

The dataset achieves wide coverage of the MITRE ATT&CK Matrix for Enterprise, encompassing nearly 90% of its attack pattern classes. Furthermore, it includes all of the top 10 most prevalent ATT&CK techniques used by adversaries in 2022, with each technique appearing in multiple reports. Overall, the dataset mentions 188 unique malware variants and 91 distinct threat actors, ensuring robust diversity in threat representation.

Each of the 204 reports is associated with a STIX bundle, resulting in a total of 36.1k structured entities and 13.6k semantic relations. The ontology derived from this data is visualized in Figure 58 covering 9 STIX entity types and 5 types of relations and providing a structured foundation for knowledge graph-based CTI processing.

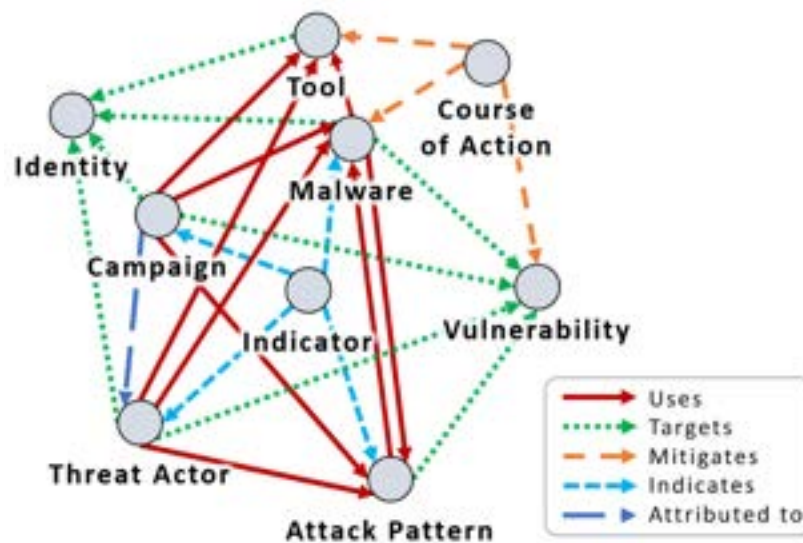


Figure 58. A subset of the STIX ontology, including all entities

Table 32 provides detailed statistics on the STIX bundles. On average, each bundle contains 177 STIX objects and 67 relations, with wide variance reflecting the richness of different reports. We also report the distribution of key entity and relation types across the dataset. Notably, Malware appears in 75% of bundles, Threat Actor in 54%, and Attack Pattern in 99%, reflecting their prominence in CTI narratives.

Table 32: Dataset statistics by STIX bundle. Final column shows percentage of bundles containing each object or relation at least once.

Metric / Type	Min	Avg	95p	Max	Quota
STIX Objects	13	177.1	525.8	1255	–
STIX Relations	5	67.0	180.3	429	–
Malware	0	0.9	2.0	5	75%
Threat Actor	0	0.6	1.0	2	54%
Attack Pattern	0	21.8	40.0	63	99%

Metric / Type	Min	Avg	95p	Max	Quota
Identity	1	1.7	2.0	5	100%
Indicator	1	41.9	163.1	395	100%
Campaign	0	0.6	1.0	4	55%
Vulnerability	0	0.5	2.0	11	21%
Tool	0	0.1	1.0	10	6%
Course of Action	0	0.0	0.0	1	2%
uses (relation)	1	23.6	48.8	64	100%
indicates (relation)	1	41.9	163.1	395	100%
targets (relation)	0	1.2	3.8	12	77%
attributed-to (relation)	0	0.3	1.0	2	26%
mitigates (relation)	0	0.0	0.0	2	2%

These statistics emphasize the richness and representativeness of the dataset, making it suitable for evaluating the performance of automated systems in structured CTI extraction tasks. While Table 32 is not used directly in our performance evaluation, it provides important context on the volume and diversity of information present in the annotated STIX bundles.

6.15 A Multimodal 5G Network & Container Metrics Dataset

This dataset was generated using a modular, containerized 5G Testbed that integrates SDN, a 5G core network, traffic orchestration, monitoring, and attack simulation components. The infrastructure is fully deployed using Docker containers, enabling reproducibility, isolation, and monitoring across all network layers. The testbed emulates a realistic 5G environment, including radio access (UE/eNB), core network functions, SDN control, external and internal adversarial nodes. Data collection groups system-level metrics, network traffic, and flow-level statistics, enabling comprehensive analysis for intrusion detection and network behavior. An overview of the dataset structure is illustrated in Figure 59.

The dataset is derived from interactions among the following subsystems:

- **SDN Layer:** Implements programmable network control (Floodlight controller, Mininet topology).
- **5G Core Network:** Based on Free5GC components, including AMF, SMF, UPF, HSS, PCRF, and MongoDB.
- **Monitoring Stack:** Prometheus, Grafana, Node Exporter, and custom logging services.
- **Traffic Management Layer:** Handles traffic generation and orchestration.
- **IDS Layer:** Performs flow monitoring and anomaly detection.
- **Attack Infrastructure:** Simulated adversarial nodes using Kali Linux containers.
- **User Interface:** Dashboard frontend/backend for visualization and control.

All containers operate continuously throughout the data generation period, ensuring consistent and synchronized data collection across all system layers. To emulate realistic user behavior within the 5G environment, controlled traffic generation scenarios were deployed between multiple User Equipment (UE) pairs. Specifically, two distinct communication profiles were implemented: (i) a constant bitrate flow of 1 Mbps between one UE pair, representing steady-state application traffic such as video streaming or VoIP, and (ii) a stochastic traffic flow between a second UE pair, modeled using a Poisson distribution to capture bursty and irregular transmission patterns typical of event-driven applications. This probabilistic model introduces variability in inter-packet arrival times, thereby increasing the realism of network load dynamics.

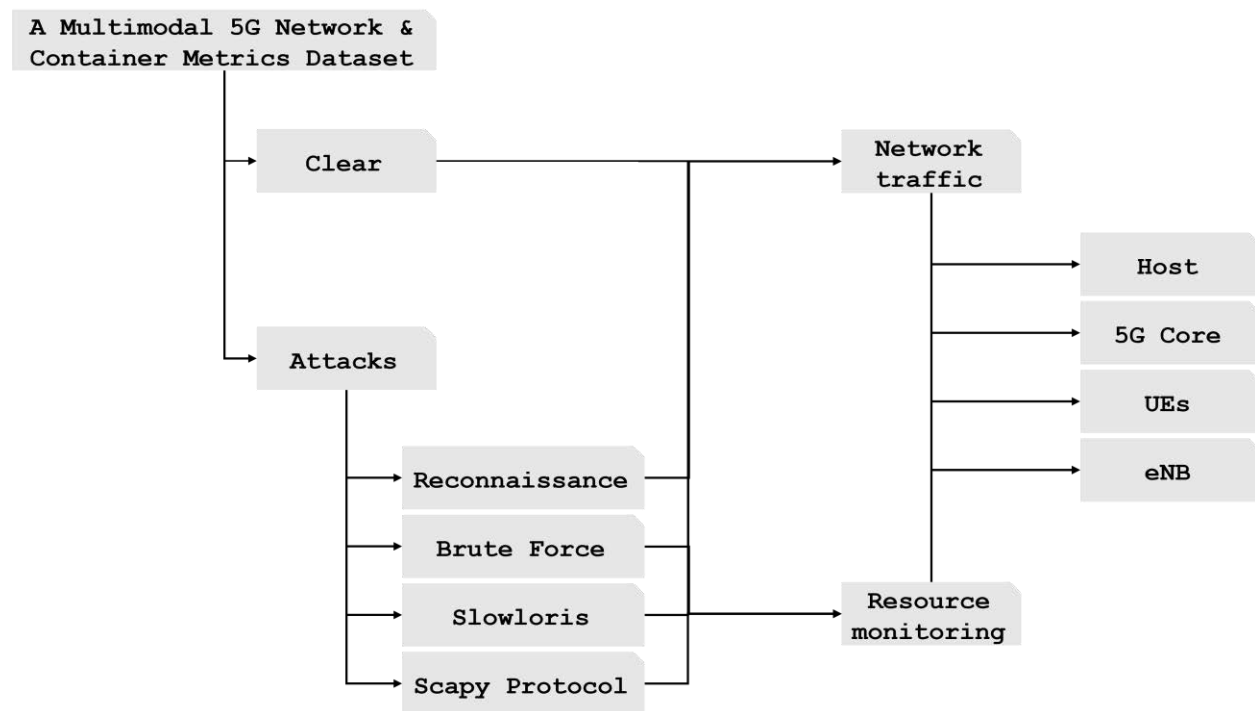


Figure 59 Structure of Multimodal 5G Network & Container Metrics Dataset

Building upon this experimental setup, the resulting dataset captures a rich set of multimodal measurements spanning container-level resource utilization and network-level traffic observations. To facilitate a structured understanding of the collected data, the key features are categorized into two main groups: (i) system and container metrics derived from Docker statistics (CSV), reflecting resource consumption and runtime behavior, and (ii) network traffic features extracted from packet capture (PCAP) files, representing communication patterns at both packet and flow levels. The following tables (Table 33 and Table 34) summarize the principal parameters included in each category, providing a clear overview of the dataset's structure and feature space.

Table 33: Summary of the main parameters captured in Resource monitor CSV

Parameter	Description	Unit/Format
Container ID	Unique identifier of the container	Alphanumeric
Name	Container name	Text
CPU %	Percentage of CPU usage by the container	%
Mem Usage / Limit	Current memory usage vs maximum allocated memory	MB / GB
Mem %	Percentage of memory usage relative to the limit	%
Net I/O	Network input and output traffic (received/transmitted data)	KB / MB / GB
Block I/O	Disk read/write operations performed by the container	KB / MB / GB
PIDs	Number of processes or threads running inside the container	Integer

Table 34: Summary of the main parameters captured in Network traffic PCAP

Parameter	Description	Unit/Format
Timestamp	Time at which the packet was captured	DateTime (e.g., UNIX / ISO 8601)
Frame Length	Total size of the captured packet	Bytes
Captured Length	Portion of the packet actually captured	Bytes
Source IP	IP address of the sender	IPv4 / IPv6
Destination IP	IP address of the receiver	IPv4 / IPv6
Source Port	Port number of the sending application	Integer
Destination Port	Port number of the receiving application	Integer
Protocol	Network/transport protocol used	Text / Numeric ID
Packet Length	Length of the packet payload	Bytes
TCP Flags	Control flags in TCP packets	Binary / Text
Sequence Number	TCP sequence number	Integer
Acknowledgment Number	TCP acknowledgment number	Integer
Window Size	TCP window size for flow control	Bytes
TTL	Maximum number of hops before packet is discarded	Integer
Flow Duration	Time duration of a network flow	Seconds / ms
Packets per Flow	Total number of packets within a flow	Integer
Bytes per Flow	Total number of bytes transferred in a flow	Bytes
Inter-Arrival Time	Time between consecutive packets	ms / μ s

The dataset spans a total duration of 65 minutes, during which normal operational traffic is systematically interleaved with multiple categories of cyberattacks. This structured approach enables the development and validation of AI-based detection mechanisms by providing both stable baseline conditions and diverse anomalous behaviors. In particular, it allows the models to learn typical microservice resource consumption patterns while simultaneously evaluating their ability to detect and distinguish between heterogeneous attack types.

The execution begins with a short initialization phase lasting approximately three minutes, during which baseline traffic is established and all system components reach steady-state operation. This period ensures that subsequent observations are not influenced by startup transients. Following initialization, a reconnaissance phase of roughly nine minutes is introduced. During this stage, the attacker performs systematic network discovery through probing and scanning activities targeting multiple internal nodes of the 5G Core. This behavior emulates an adversary attempting to map the network topology and identify potential entry points, generating low-to-moderate intensity traffic with distinct scanning signatures.

Subsequently, a five-minute brute-force attack phase is executed, characterized by high-frequency connection attempts against a specific service endpoint, in this case the login page of the SDN's controller. This phase produces a sharp increase in request rates and connection failures, placing stress on authentication mechanisms and exposing resource consumption anomalies at both the network and container levels. After the attack activity, the system transitions into an extended benign operation period lasting approximately 17 minutes. This interval serves as the primary baseline window, during which only legitimate traffic is present. It is particularly important for establishing reliable behavioral profiles of normal CPU usage, memory consumption, and network throughput across the microservices, in parallel of the normal network characteristics.

The next phase introduces a five-minute application-layer attack based on the Slowloris technique. Unlike volumetric attacks, this scenario focuses on maintaining a large number of partially open connections, gradually exhausting server resources. This phase is critical for evaluating the system's ability to detect low-rate, stealthy anomalies that primarily affect connection handling rather than raw traffic volume. Following a brief transition, the dataset concludes with a sequence of advanced protocol-level attacks lasting approximately ten minutes in total. These attacks are using packet manipulation techniques and are executed in three escalating stages. The first stage consists of MAC flooding, aiming to overwhelm network switching mechanisms by saturating forwarding tables. This is followed by a heartbeat-style attack involving malformed protocol messages, simulating targeted exploitation and potential data leakage behavior. The final stage is a high-intensity UDP flood, generating a significant surge

in traffic volume to stress both network bandwidth and processing capabilities of the deployed microservices.

A consolidated overview of all phases, including their duration and key characteristics, is provided in the Table 35 below, offering a structured summary of the dataset timeline and the progression from normal operation to increasingly complex attack scenarios.

Table 35: Attack timeline and short description

Phase	Description	Duration
Initialization	System startup and stabilization with baseline traffic generation	~3 minutes
Reconnaissance Phase	Network scanning and probing to discover topology and potential targets	~9 minutes
Brute Force Attack	High-frequency access attempts targeting a specific service endpoint	~5 minutes
Normal Traffic Baseline	Purely benign traffic used to model standard system behavior and resource usage	~17 minutes
Slowloris Attack	Application-layer attack maintaining partial connections to exhaust resources	~5 minutes
Protocol Attacks (Scapy)	Sequence of advanced attacks: MAC flooding, malformed requests, UDP flooding	~10 minutes total
Dataset	Total data collection	~65 minutes

The execution of both normal traffic scenarios and attacks has resulted in a comprehensive, multimodal dataset that captures system behavior across multiple layers of the 5G testbed. A summary of the generated dataset files, including their sizes and scope, is presented in the following, Table 36.

Table 36: Dataset File Summary

Type	File	Description	Size	Notes
Docker Metrics	docker_stats.csv	Aggregated container resource utilization metrics	4.79 MB	49,091 records
PCAP Trace	amf.pcap	Traffic related to AMF component	682 KB	≈ 5,000–15,000 packets
PCAP Trace	enb.pcap	Traffic from eNodeB	135 Bytes	≈ 1–2 packets
PCAP Trace	host.pcap	Host-level network traffic	5.8 GB	≈ 10–40 million packets
PCAP Trace	hss.pcap	HSS-related signaling traffic	412 KB	≈ 3,000–10,000 packets

Type	File	Description	Size	Notes
PCAP Trace	pcrf.pcap	PCRF policy control traffic	409 KB	≈ 3,000–10,000 packets
PCAP Trace	smf.pcap	SMF session management traffic	418 KB	≈ 3,000–10,000 packets
PCAP Trace	ue.pcap	User Equipment traffic	2.04 GB	≈ 2–8 million packets
PCAP Trace	upf.pcap	UPF data-plane traffic	12 MB	≈ 50,000–200,000 packets
PCAP Trace	webui.pcap	Web interface traffic	135 Bytes	≈ 1–2 packets

To conclude, the A Multimodal 5G Network & Container Metrics Dataset is further analyzed through a set of representative visualizations that illustrate the impact of the different attack scenarios on the system behavior. In particular, two key graphs (Figure 60 and Figure 61) are presented to highlight how the introduced attacks manifest across network and system-level metrics over time. These visualizations provide an intuitive understanding of traffic dynamics, resource utilization patterns, and anomaly characteristics, thereby complementing the previously described tabular and structural overview of the dataset.

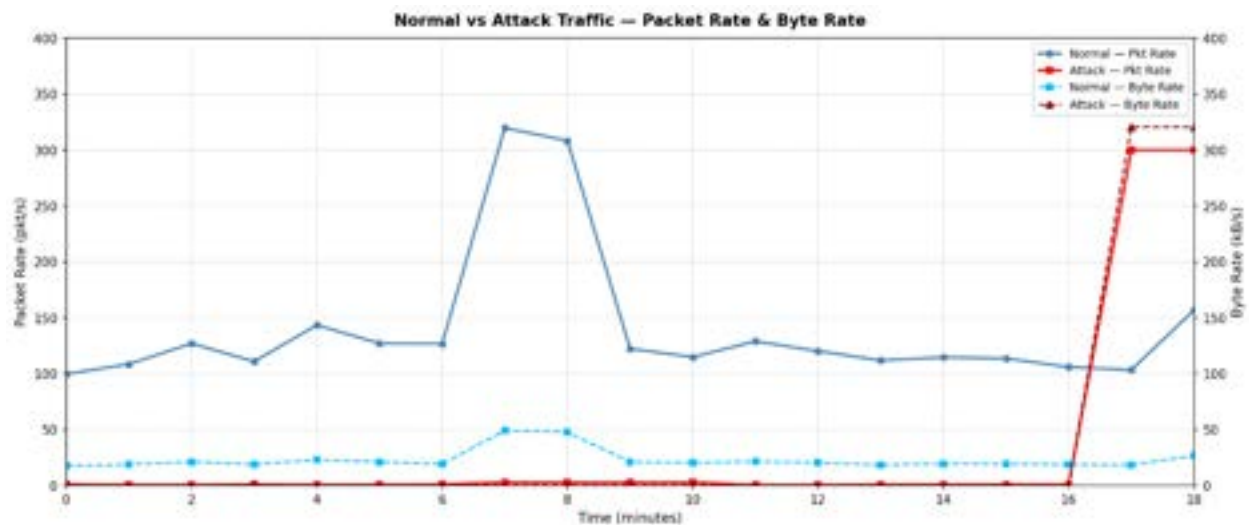


Figure 60 Normal vs. Attack Traffic — Packet Rate & Byte Rate

Figure 60 provides a clear temporal comparison of network throughput during benign conditions versus an active attack phase. Under normal conditions, the packet rate exhibits moderate fluctuations, generally hovering between 100 and 150 packets per second, with a brief, expected surge to approximately 320 pkt/s. Correspondingly, the normal byte rate remains consistently low, stable at around 20 to 50 kB/s. However, the onset of the attack introduces a dramatic anomaly at the 16-minute mark. Both the attack packet rate and byte rate spike vertically, surging

from near zero to roughly 300 pkt/s and 320 kB/s respectively, sustaining this extreme volume through the end of the capture window. This sudden, massive influx is highly characteristic of a volumetric assault, such as a UDP flood, designed to instantly overwhelm network bandwidth.

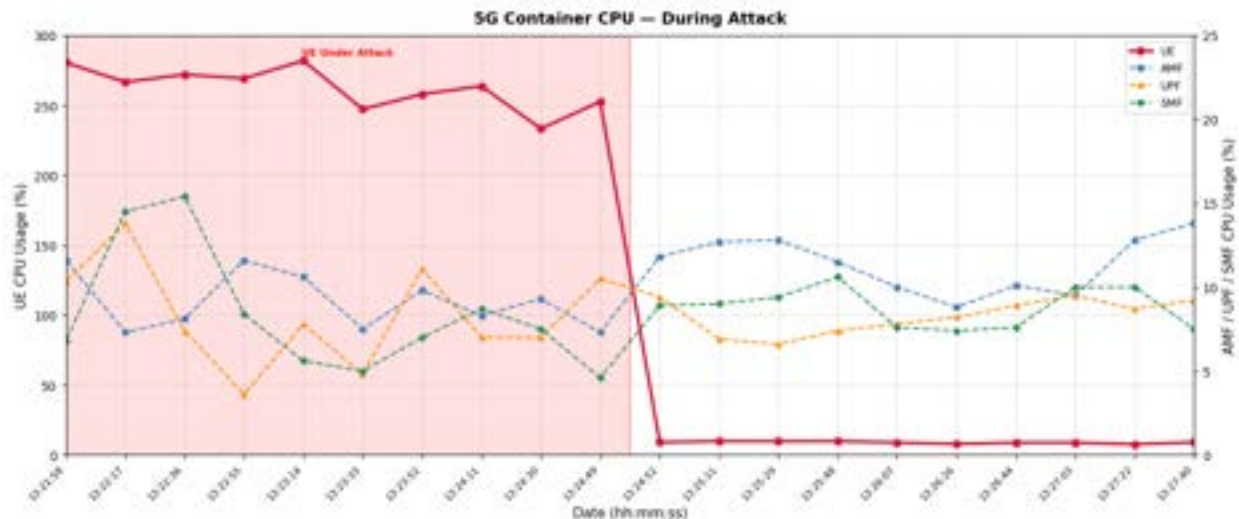


Figure 61: 5G Container CPU — During Attack

Figure 61 visualization demonstrates the severe, infrastructure-level impact of the attack on compute resources, specifically isolating the vulnerability of the Radio Access Network components compared to the 5G core. During the heavily shaded "UE Under Attack" window, one of the UE containers experiences extreme CPU exhaustion, sustaining a critical usage rate that fluctuates between 230% and 280%. In stark contrast, the core 5G microservices (AMF, UPF, and SMF) remain largely insulated from this resource drain, maintaining stable and low CPU utilization strictly under 15%. Immediately following the termination of the attack at approximately 13:24:52, the UE's CPU usage plummets to near 0%, definitively correlating the extreme compute stress with the adversarial behavior.

Overall, a comprehensive description of the A Multimodal 5G Network & Container Metrics Dataset is provided, covering its structure, composition, and underlying data sources. The combination of system-level Docker metrics and network-level PCAP traces enables a rich multimodal representation of the 5G testbed environment. Furthermore, the inclusion of both benign and diverse attack scenarios ensures that the dataset captures realistic operational variability as well as clearly identifiable anomalous patterns. The accompanying tables and visual analyses further clarify the dataset's scale, distribution, and temporal dynamics, offering a complete overview of how data is collected, organized, and represented.

6.16 MARLIN-5G: Multi-Agent Reinforcement Learning Intrusion Network dataset for 5G

The MARLIN-5G dataset is a labeled, multi-protocol network traffic dataset targeting 5G core network Intrusion Detection System (IDS) research. It is generated by running three protocol extensions of the Gorgo cooperative Multi-Agent Reinforcement Learning (MARL) framework against a free5GC software-defined 5G core deployment.

The dataset covers three 5G core protocol interfaces:

- HTTP/2 — Service-Based Architecture (SBA) interface between 5G Network Functions (NRF, AMF, SMF, etc.), targeted at TCP ports 80/8000 and 443.
- GTP-U — User-plane tunneling protocol on the N3 interface between the gNB and the UPF, targeted at UDP port 2152.
- PFCP — Packet Forwarding Control Protocol on the N4 interface between the SMF and the UPF, targeted at UDP port 8805.

The dataset uniquely combines RL-guided attack exploration (producing a non-uniform, impact-weighted strategy distribution), protocol-semantic ground-truth labels, and full-packet PCAP captures co-indexed with per-round label records. The MARLIN-5G dataset is organized into three components. Raw network traffic is stored as full-packet captures in the pcap/ subdirectory, with one PCAP file per protocol interface (http2.pcap, gtpu.pcap, pfcpc.pcap) plus a separate benign.pcap for UERANSIM-generated legitimate traffic. Ground-truth labels are provided in labels.jsonl, a newline-delimited JSON file containing one record per RL round, each carrying the wall-clock timestamp, protocol, traffic class, attack sub-category, and the complete joint agent action vector that produced the corresponding burst. Finally, flows.csv contains the extracted per-flow feature matrix, with one row per network flow and 52 columns covering flow metadata, nine protocol-agnostic statistical features, and up to 16 protocol-specific features depending on the interface. The label records and flow rows are co-indexed by timestamp: a flow is matched to the most recent label whose emission timestamp precedes the flow's start time, within a 120-second window.

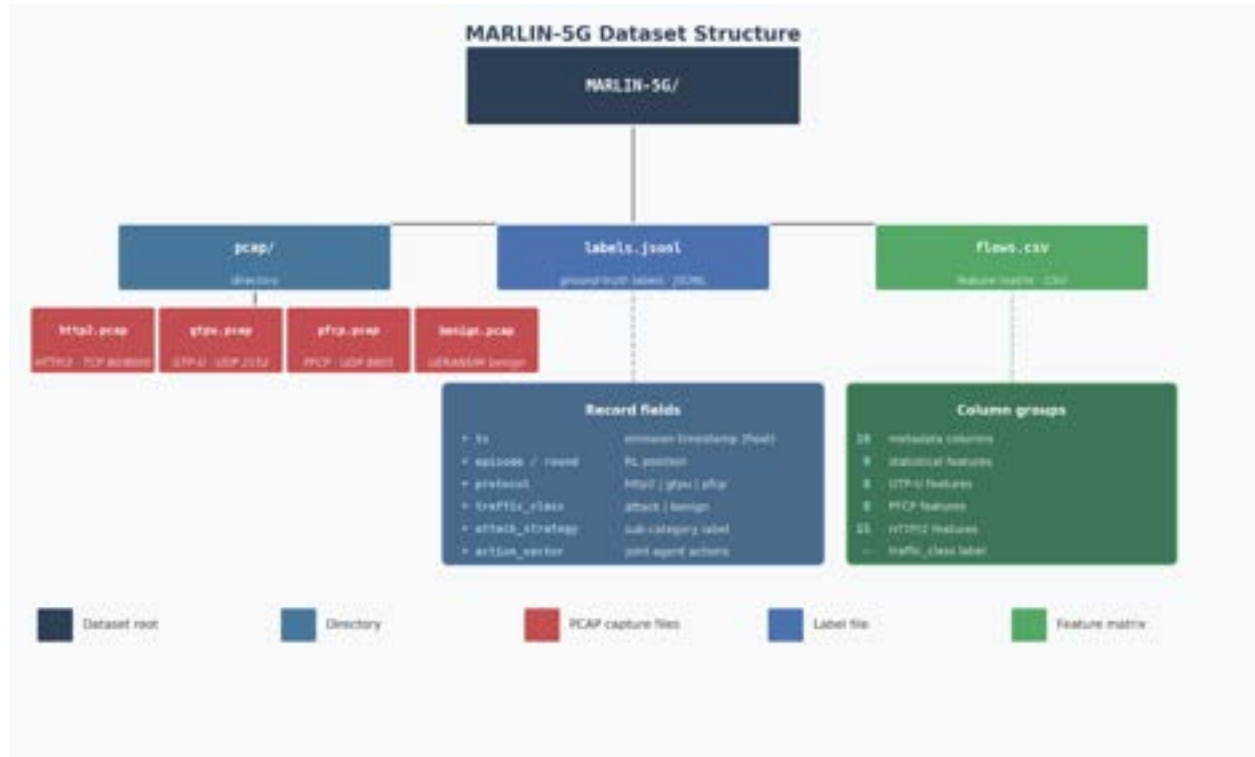


Figure 62: Structure of MARLIN-5G dataset

Table 37: MARLIN-5G dataset parameters

Parameter	Description
Dataset Name	MARLIN-5G
Total Size	1.24 GB
Information Format	PCAP · JSONL · CSV
Protocols	HTTP/2 (SBA) · GTP-U (N3) · PFCP (N4)
Feature Columns	52
Metadata columns	10
Statistical feature columns	9
HTTP/2 feature columns	16
GTP-U feature columns	8
PFCP feature columns	9
Possible Attack Sub-categories	29

Parameter	Description
Observed Attack Sub-categories	6
Episodes — HTTP/2	1
Episodes — GTP-U	1
Episodes — PFCP	1
Rounds per Episode	5
Total Attack Rounds	15
HTTP/2 Attack PCAP	23.2 MB
GTP-U Attack PCAP	892.3 KB
PFCP Attack PCAP	6.7 MB
Benign PCAP (UERANSIM)	1.20 GB
Total Flows	81,097
Attack Flows — Total	43,856
HTTP/2 Attack Flows	17,622
GTP-U Attack Flows	4,371
PFCP Attack Flows	21,863
Benign Flows	32,708
Unlabeled Flows	4,533

The MARLIN-5G dataset totals 1.24 GB across four PCAP captures, a JSONL label file, and a pre-extracted feature CSV. It was generated by running one RL episode of five rounds per protocol, yielding 15 attack label records in total. The raw captures comprise 23.2 MB of HTTP/2 attack traffic, 892.3 KB of GTP-U attack traffic, and 6.7 MB of PFCP attack traffic, complemented by 1.20 GB of legitimate traffic captured from a UERANSIM user equipment simulation. Feature extraction produces 81,097 labeled network flows across 52 columns: 10 metadata fields, 9 protocol-agnostic statistical features, and up to 16 protocol-specific features depending on the interface. Of the total flows, 43,856 are labeled as attack (17,622 HTTP/2, 4,371 GTP-U, and 21,863 PFCP), 32,708 are labeled as benign, and 4,533 remain unlabeled because they fall outside the 120-second label assignment window. Six distinct attack sub-categories were observed across the 15 attack rounds, drawn from a taxonomy of 29 possible sub-categories spanning technique-based and volumetric attack classes.

Table 38: Label record fields

Field	Type	Description
ts	float	Unix wall-clock timestamp (seconds since epoch) emitted before the burst.
episode	int	Episode index within the protocol block, 1-indexed.
round	int	Round index within the episode, 0-indexed (0–4).
protocol	string	"http2", "gtpu", or "pfcf".
traffic_class	string	"attack" for RL-generated; "benign" for UERANSIM rows.
attack_strategy	string	Attack sub-category label, null for benign.
action_vector	object	Flattened joint agent action dictionary

Each label record captures the full context of one RL round. The `ts` field records the wall-clock timestamp at which the label was emitted, immediately before the traffic burst was sent, establishing the temporal anchor used to co-index flows from the PCAP captures. The `episode` and `round` fields identify the record's position within the RL campaign, allowing reconstruction of the agent's learning trajectory across the dataset. The `protocol` field specifies which 5G interface was targeted in that round, i.e., HTTP/2, GTP-U, or PFCF, while `traffic_class` distinguishes between RL-generated attack traffic and UERANSIM-sourced benign traffic appended after capture. The `attack_strategy` field carries the derived sub-category label, determined by applying the two-tier taxonomy to the joint action vector; it is absent for benign records. Finally, `action_vector` contains the flattened joint selection of all nine cooperative agents for that round, encoding the complete set of transport and payload decisions, i.e., thread count, request rate, delay, protocol-specific message types, payload characteristics, and encoding strategies, that collectively defined the attack behaviour observed in the corresponding PCAP segment.

Table 39: Metadata features

Field	Type	Description
src_ip	string	Source IP address
dst_ip	string	Destination IP address
src_port	int	Source port
dst_port	int	Destination port
transport	string	"tcp" or "udp"

Field	Type	Description
flow_start	float	Unix timestamp of the first packet in the flow
protocol	string	"http2", "gtpu", or "pfc"p"
episode	int	RL episode number (–1 if unlabeled)
round	int	RL round within episode (–1 if unlabeled)
traffic_class	string	"attack", "benign", or "unlabeled"

Table 40: Statistical features

Field	Type	Description
pkt_count	int	Total packet count in the flow
byte_count	int	Total bytes in the flow
duration	float	Flow duration in seconds
iat_mean	float	Mean inter-arrival time between consecutive packets (s)
iat_std	float	Standard deviation of inter-arrival times
iat_min	float	Minimum inter-arrival time
iat_max	float	Maximum inter-arrival time
pkt_per_sec	float	Packets per second
byte_per_sec	float	Bytes per second

Table 41: GTP-U protocol features

Field	Type	Description
gtpu_teid_entropy	float	Shannon entropy of TEID field values. Low = sequential strategy.
gtpu_teid_unique	int	Number of distinct TEID values in the flow.
gtpu_msg_type_unique	int	Number of distinct GTP-U message type codes.
gtpu_inner_size_mean	float	Mean inner payload size (bytes).
gtpu_inner_size_std	float	Standard deviation of inner payload sizes.

Field	Type	Description
gtpu_ext_header_rate	float	Fraction of packets with Extension Header Present bit set.
gtpu_echo_req_rate	float	Fraction of packets with message type = Echo Request.
gtpu_gpdu_rate	float	Fraction of packets with message type = G-PDU.

Table 42: PFCP Protocol features

Field	Type	Description
pfcip_heartbeat_rate	float	Fraction of packets with message type = Heartbeat Request.
pfcip_assoc_setup_rate	float	Fraction of packets with message type = Association Setup Request.
pfcip_session_est_rate	float	Fraction of packets with message type = Session Establishment Request.
pfcip_session_del_rate	float	Fraction of packets with message type = Session Deletion Request.
pfcip_msg_type_unique	int	Number of distinct PFCP message type codes.
pfcip_ie_count_mean	float	Mean count of TLV Information Elements per packet.
pfcip_ie_count_std	float	Standard deviation of IE count per packet.
pfcip_seid_entropy	float	Shannon entropy of SEID field values. Low = sequential F-SEID strategy.
pfcip_seq_entropy	float	Shannon entropy of PFCP sequence numbers.

Table 43: HTTP/2 protocol features

Field	Type	Description
http2_frame_count	int	Total HTTP/2 frames observed in the flow.
http2_stream_count	int	Number of distinct non-zero HTTP/2 stream IDs.

Field	Type	Description
http2_max_stream_id	int	Highest stream ID seen; proxy for total streams opened.
http2_headers_rate	float	Fraction of frames of type HEADERS (0x1).
http2_data_rate	float	Fraction of frames of type DATA (0x0).
http2_continuation_rate	float	Fraction of frames of type CONTINUATION (0x9).
http2_rst_stream_rate	float	Fraction of frames of type RST_STREAM (0x3).
http2_settings_rate	float	Fraction of frames of type SETTINGS (0x4).
http2_ping_rate	float	Fraction of frames of type PING (0x6).
http2_window_update_rate	float	Fraction of frames of type WINDOW_UPDATE (0x8).
http2_goaway_rate	float	Fraction of frames of type GOAWAY (0x7).
http2_frame_type_entropy	float	Shannon entropy of frame type distribution.
http2_headers_size_mean	float	Mean payload length (bytes) of HEADERS frames.
http2_headers_size_max	int	Maximum payload length of HEADERS frames.
http2_data_size_mean	float	Mean payload length of DATA frames.
http2_data_size_std	float	Standard deviation of DATA frame payload lengths.

Table 44: Attack subcategories

Sub-category	Protocol	Type	Rounds	Trigger
stream_exhaustion	HTTP/2	Technique-based	3	HEAVY multiplexing +

Sub-category	Protocol	Type	Rounds	Trigger
				req/thread $\geq$ 100 concurrent streams
header_flood	HTTP/2	Volumetric	2	MANY/LARGE header strategy, high transport intensity
echo_burst	GTP-U	Volumetric	2	Echo Request message type, medium transport intensity
teid_exhaustion	GTP-U	Technique-based	3	Sequential TEID strategy
malformed_ie	PCFP	Technique-based	3	Malformed or duplicate Information Elements
session_exhaustion	PCFP	Technique-based	2	Session establishment + sequential F-SEID

The six observed sub-categories span all three protocol interfaces and reflect the early exploitation of high-impact strategies by the RL agents within a single episode of five rounds each. Four of the six are technique-based, meaning their labeling is independent of transport volume: stream_exhaustion and header_flood target the HTTP/2 server's stream management and header processing paths, respectively; teid_exhaustion exploits the cumulative nature of sequential TEID allocation to exhaust the UPF session table; malformed_ie stresses the PCFP TLV parser with structurally invalid Information Elements; and session_exhaustion depletes UPF session capacity through sequential F-SEID assignment across establishment requests. The two volumetric sub-categories, i.e., header_flood and echo_burst, carry an intensity suffix reflecting that the RL agents had already converged toward high request-rate configurations: header_flood at flood intensity and echo_burst at burst intensity. The dominance of technique-based categories over purely volumetric ones suggests that even in the early exploration phase, the agents preferentially discover and repeat actions that cause measurable degradation via semantic protocol exploitation rather than via raw packet volume.

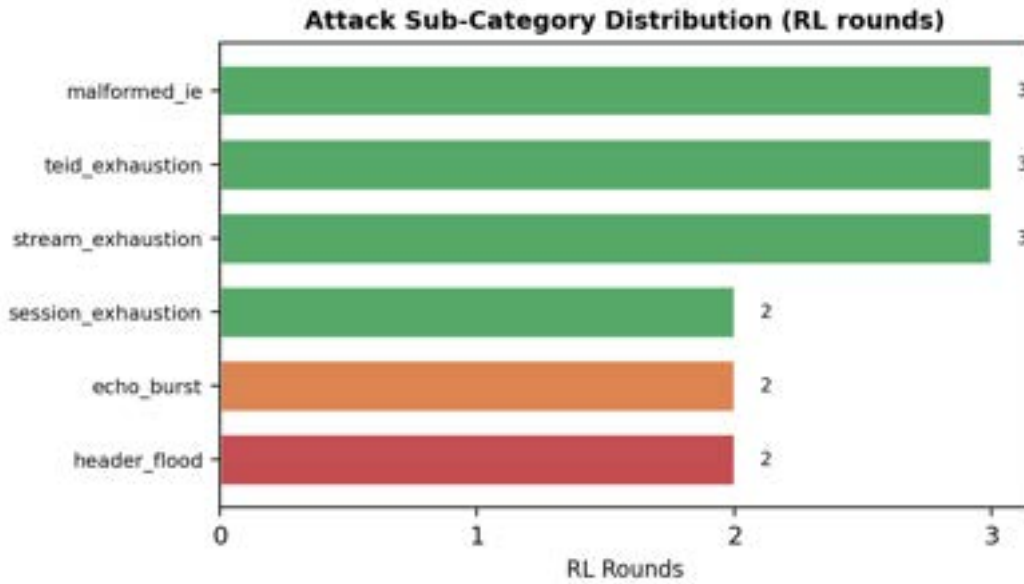


Figure 63: Attack sub-category distribution across all RL rounds. Colour coding: red = high-intensity (flood/storm), orange = medium (burst), blue = low (probe), green = technique-based.

7 Exploitation Aspects and Potentials

This section provides an overview of the exploitation aspects of the collected datasets in the frame of the NATWORK project. It also presents potential tactics for exploiting the cross-layer AI-based attacks datasets stored in the Federated Repository for b5G/6G networks.

This repository is one of the expected deliverables of NATWORK. The project’s Work Breakdown Structure outlines a well-defined schedule and the appropriate resources to build the repository and make it operational. The repository will store and provide cross-layer AI-based attack datasets as open data, available for scientific research. However, the Federated Repository is not merely an expected outcome of the project. The rapid development of artificial intelligence has significantly increased the demand for large-scale open datasets. Open scientific datasets, therefore, represent valuable resources for both academic research and commercial AI development. The consortium put in significant effort to make the Federated Repository a recognizable, high-impact focal point for searching and retrieving open data beyond NATWORK boundaries, both during the project and in the post-project period.

This approach sets high ambitions for the repository’s functionality, performance, and, mainly, the efficiency of the regulatory framework to be followed. These are fundamental requirements and prerequisites for achieving and retaining the target for cross-consortia recognition in the 6G, security, and native network AI realms.

7.1 Open science and open access to research data

Open science is an approach to research based on open, collaborative work that emphasizes the sharing of knowledge, results, and tools as early and widely as possible. Its purpose is to enhance the impact of research results and to foster greater transparency and trust for the benefit of scientific research and, indirectly, for the benefit of EU citizens. Open science is a legal obligation of participants under Horizon Europe⁴ research. It operates on the principle of being “as open as possible, as closed as necessary”. This principle means that results and data may be kept confidential if making them publicly available in open access would conflict with the researcher’s legitimate interests. Examples include commercial exploitation of the research results or obligations mentioned in the Grant Agreement (e.g., personal data protection).

Open access is the practice of providing online access to scientific information free of charge and in a reusable format for users. This includes both peer-reviewed publications and the data underlying them, as well as other datasets. Under Horizon Europe, researchers are not obliged

⁴ Open science https://rea.ec.europa.eu/open-science_en

to publish their results in publications; if they choose to do so, it should be in open access.

Research data is the information (facts or numbers) collected to be examined and considered, and to serve as a basis for reasoning, discussion or calculation. Open access to research data is the right to access and reuse digital research data under the terms and conditions set out in the Grant Agreement. The research data of EU-funded projects must be FAIR (Findable, Accessible, Interoperable and Re-usable).

The most extensive resource for accessing open data is the official portal⁵ for European data (data.europa.eu). This portal includes more than 1.8 million datasets from EU institutions and agencies, as well as from Member States. Datasets can be easily identified using the Quick search facility (<https://europa.eu/W4mvRb>) and results can be downloaded in various formats. More advanced searches are also supported through specific free database search tools.

7.2 Open science and Intellectual Property Rights (IPR) - Exceptions to open science

Open science in Horizon Europe follows the principle “as open as possible, as closed as necessary.” While open access to research data and scientific publications is the default approach, exceptions are allowed when openness could conflict with legitimate interests or legal obligations. Research data may remain closed when disclosure could harm commercial exploitation, violate data protection and privacy rules, breach confidentiality agreements, reveal trade secrets, compromise security, or conflict with intellectual property rights (IPR). In such cases, beneficiaries must clearly justify the restrictions in their Data Management Plan (DMP).

Open science requirements are not incompatible with the protection and exploitation of research results through IPR, including patenting. Beneficiaries are expected to retain sufficient intellectual property rights when publishing scientific results to ensure compliance with Horizon Europe open access obligations. Therefore, researchers should avoid signing publication agreements that restrict these rights or contradict the Grant Agreement.

Importantly, decisions regarding patent protection must precede open publication. Any invention intended for patenting must not be publicly disclosed before filing a patent application, since prior disclosure may compromise patentability. Consequently, open access publication should follow the filing of patent applications where relevant. Horizon Europe therefore seeks to balance openness with the protection of innovation, recognizing that certain restrictions are necessary to safeguard legitimate scientific, commercial, and legal interests.

⁵ Open data and high-value datasets: step-by-step access guide (<https://digital-strategy.ec.europa.eu/en/factpages/open-data-and-high-value-datasets-step-step-access-guide>)

7.3 Market Demand and Economic Outlook for AI-Native Telemetry and Datasets

The datasets generated by NATWORK are shown in the following table. Detailed descriptions of the datasets are given in Section 6 above.

1. Jamming Detection
2. Channel Estimation
3. Jamming Mitigation
4. Anomaly Detection on Kubernetes (GATAKU)
5. Stateful CNF Live Migration Dataset
6. Channel Estimation for sub-THz bands
7. DetAction jamming Dataset
8. Jamming Detection
9. DDoS Detection
10. Edge-IIoT Adversarial Dataset
11. 5G Network Traffic dataset
12. OT-ICS-attacks dataset
13. Denial of Sustainability Dataset
14. Cyber Threat Intelligence dataset
15. A Multimodal 5G Network & Container Metrics Dataset

The exploitation roadmap of the NATWORK dataset ecosystem is strategically aligned with an unprecedented macroeconomic surge in the demand for specialized, domain-specific training data.

According to an industry study by Grand View Research⁶, the global AI training dataset market was valued at €2.73 billion (\$3,195.1 million) in 2025 and is projected to reach €13.95 billion (\$16,320 million) by 2033, expanding at a compound annual growth rate (CAGR) of 22.6%. This exponential growth highlights a structural shift away from generic open-web data scraping toward high-fidelity, infrastructure-layer vertical datasets. This shift directly maximizes the inherent commercial and academic value of NATWORK's niche graph topologies, such as the GATAKU Kubernetes Dataset (4) and the Denial of Sustainability Dataset (13).

Concurrently, telecommunication operators are rapidly pivoting away from volume-based connectivity models toward insight-driven architectural frameworks. Market data published by

⁶ Grand View Research, Inc. "AI Training Dataset Market Size, Share & Trends Analysis Report by Data Type (Text, Image/Video, Audio), by Vertical (IT & Telecom, Automotive, Healthcare, BFSI, Retail, Government), by Region, and Segment Forecasts 2025 - 2033." Grand View Research. Available at: [grandviewresearch.com](https://www.grandviewresearch.com)

Mordor Intelligence⁷ demonstrates that the Telecom Data Monetization market is on track to expand from €4.76 billion (\$5.57 billion) in 2025 to €11.28 billion (\$13.20 billion) by 2031, progressing at a steady CAGR of 15.48%. This targeted sub-sector governs the exact multi-domain telemetry logs, sub-THz channel measurements, and containerized network functions generated across NATWORK's three functional bundles.

Furthermore, the operational rollout of the EU AI Act has introduced severe regulatory restrictions regarding cross-border transmission of raw infrastructure data, triggering a major demand spike for privacy-compliant synthetic data and hashed indicator schemas. Because NATWORK natively incorporates strict data minimization constraints—guaranteeing an optimized Exposed Information Ratio ($0 < x < 1$)—the consortium's output matches European requirements for secure, anonymized threat intelligence sharing.

Rather than diluting value through a premature corporate spin-off, this booming multi-billion-euro market confirms that the consortium's strategy of Dual-Licensing (CC BY-NC 4.0), Joint RTO Institutional Stewardship, and B2B/B2G Security Consultancies provides the most viable path to achieve long-term financial sustainability and assert European technological sovereignty.

7.4 NATWORK dataset exploitation potential strategies and models

The exploitation of NATWORK datasets encompasses multiple lines of action. We cannot consider solely the market exploitation since the datasets will become available as open research data. For a research project like NATWORK, exploiting Open Data requires balancing the "as open as possible, as closed as necessary" Horizon Europe principle⁸ with a focus on the Key Exploitable Results (KERs)⁹.

Core Exploitation Strategies

- **Value-Added Services** (Freemium Model): Provide the raw dataset for free while charging for premium features, such as advanced analytics, real-time API access, or cleaned/curated versions optimized for industrial use.
- **Product Wrapping**: Enhance existing software or physical products with embedded analytics derived from the dataset to increase their market value.
- **Dual Licensing**: Release data under a restrictive open-source license (e.g., CC BY-NC) that allows free academic use but requires a paid commercial license for for-profit

⁷ Mordor Intelligence. "Telecom Data Monetization Market Size & Share Analysis - Growth Trends & Forecasts (2026 - 2031)." Mordor Intelligence. Available at: mordorintelligence.com

⁸ European Commission, "Dissemination and exploitation of research results" https://research-and-innovation.ec.europa.eu/strategy/dissemination-and-exploitation-research-results_en

⁹ Future Needs, "How to Exploit Horizon EU Project Results: Steps for Maximum Impact", July 2025, <https://futureneeds.eu/how-to-exploit-horizon-eu-project-results-steps-for-maximum-impact/>

applications.

- **Knowledge Transfer & Consultancies:** Use the dataset to establish the project partners as subject-matter experts, offering paid training, certification, or bespoke consultancy services based on the data's insights.

Sustainability & Infrastructure Models

- **Public-Private Partnerships (PPPs):** Collaborate with industry players to co-fund data repositories or platforms that ensure the data remains live and usable beyond the project's funding cycle.
- **Data Marketplaces:** List high-value datasets on platforms like data.europa.eu or specialized research marketplaces to reach potential buyers in the policy or corporate sectors.
- **Institutional Stewardship:** Transfer the data to a sustainable RTO network that can maintain it as part of a long-term research infrastructure.

Scientific & Societal Impact

- **FAIR Principles:** Ensure data is Findable, Accessible, Interoperable, and Reusable to drive citations and scientific leadership.
- **Policy Support:** Package datasets into evidence-based policy recommendations to influence European standards and regulations.

7.5 Considerations on open dataset exploitation aspects

7.5.1 Commercial Exploitation

Open science aims to promote transparency, collaboration, and broad access to scientific knowledge, yet open datasets are increasingly exploited commercially by private companies. Large corporations, particularly in the pharmaceutical and technology sectors, can use publicly funded research data to develop profitable products and services without necessarily sharing benefits with the original data producers or the wider public. This raises concerns about whether public investments in research are indirectly subsidizing private profit. The unequal capacity to exploit open datasets also creates imbalances between actors. Large companies possess advanced computational infrastructure, skilled personnel, and financial resources that smaller institutions or researchers often lack. As a result, although data may be “open” in principle, the ability to extract value from it remains unevenly distributed. This can reinforce existing economic and scientific inequalities within the global research ecosystem. At the same time, commercial exploitation is not inherently incompatible with open science. Horizon Europe and other policy frameworks recognize that innovation, including patenting and commercialization, can coexist with openness when managed responsibly. The challenge lies in ensuring fair benefit-sharing,

transparency, and mechanisms that prevent the monopolization of knowledge derived from openly accessible scientific resources.

7.5.2 AI Training on Open Datasets

The use of open datasets for AI training may raise in some cases ethical and governance concerns. Data originally collected for scientific purposes may later be used in contexts not anticipated by participants or researchers, including commercial AI applications. In fields such as health or genomics, this creates questions about informed consent, privacy, and the acceptable scope of secondary data use.

Another major concern relates to the concentration of AI capabilities among a small number of powerful technology companies. Although datasets may be openly available, only a limited number of actors possess the computational resources necessary to train advanced AI systems. As a result, open scientific data can unintentionally contribute to market concentration and technological dominance, rather than democratizing innovation and knowledge production.

7.6 Operational Blueprint for Consolidated Dataset Exploitation

Based on the current status, the trends, and the data mentioned above in the previous sections, this section aims to sketch a roadmap towards the exploitation of project-generated datasets. Taking into account that a startup or a spin-off is not on the immediate horizon for the consortium, we estimate that the exploitation strategy should pivot fully toward RTO-driven Institutional Stewardship, Academic Leadership, and Policy-Driven Sustainability. This approach aligns to the collaborative nature of the NETWORK project, ensuring all partners benefit equally from the outputs without requiring commercial corporate restructuring. An outline of a realistic operational blueprint for this specific exploitation route follows:

7.6.1 Federated Institutional Stewardship Framework

Given the collaborative structure of the NETWORK consortium and the equal strategic value assigned to all fifteen (15) project datasets, asset management will be governed via a Federated Repository Agreement among the participating Research and Technology Organizations (RTOs) and academic institutions.

Infrastructure Integration: Rather than centralizing assets within a single commercial entity, datasets will be hosted across existing institutional nodes and interconnected via the project's decentralized, DLT-backed attestation registry.

Lifecycle and Sustainability: The hosting RTO partners will guarantee the operational availability, query capacity, and data curation of the repositories for a minimum post-project lifecycle of five (5) years beyond the official European Commission funding period.

7.6.2 Institutional Knowledge Transfer and B2B/B2G Consultancies

Consortium partners will individually and collectively leverage the unique data insights to expand their organizational portfolios, securing advanced research prestige and high-value industrial consultancy contracts.

Bespoke Security Auditing: University and RTO partners will utilize the structural graph profiles from the GATAKU Kubernetes Dataset (4), the Denial of Sustainability Dataset (13), and the OT-ICS-attacks dataset (12) as proprietary validation benchmarks to offer commercial security-auditing services to European critical infrastructure and telecom operators.

Professional Certification Modules: The consortium will design specialized, paid professional training curriculums (e.g., "6G Slice Reliability and Cloud-Native Network Security") targeting external network engineers, corporate SOC analysts, and security practitioners.

7.6.3 Policy Support and European Standardization Mechanisms

To maximize the project's societal and regulatory impact, the dataset bundles will be actively channeled into European digital sovereignty frameworks and international standardization bodies.

Standardization Contributions: Multi-domain telemetry, sub-THz channel measurements, and Jamming Mitigation (3) matrices will be packaged into evidence-based technical contributions submitted to active working groups within 3GPP (Release 19/20) and ETSI.

ENISA Threat Mapping Alignment: Cyber threat indicators and anomaly signatures derived from the CTI Dataset (14) and the Edge-IIoT Adversarial Dataset (10) will be aligned with the European Union Agency for Cybersecurity (ENISA) frameworks, positioning the NATWORK consortium as a key architectural advisor to EU cybersecurity policy.

7.6.4 Dual-Licensing Legal Architecture

To safeguard the consortium's intellectual property from uncompensated commercial exploitation by external global vendors while adhering to Horizon Europe Open Science mandates, a strict dual-licensing legal model will be enforced:

The Academic/Open Track: In full compliance with FAIR principles, all 15 datasets will be made openly and freely accessible for non-commercial academic research via public platforms (e.g., Zenodo), maximizing global scientific citations and project visibility.

The Restricted Commercial Track: Under the terms of the Consortium Agreement, any for-profit entity wishing to ingest the datasets or embed their derived AI weights into a commercial product must secure a paid commercial license. The resulting revenues will be distributed among the generating partners based on the established Intellectual Property Rights (IPR) registry.

7.7 Legal and Licensing Issues

Open science depends heavily on legal and licensing frameworks that determine how data and publications can be accessed, reused, and redistributed. Open licenses are intended to facilitate sharing and collaboration, but inconsistencies between licensing systems can create uncertainty regarding permissible uses of scientific outputs. Researchers may also lack awareness of the legal implications associated with different licensing models.

Tensions frequently emerge between open access requirements and intellectual property rights. While funding frameworks such as Horizon Europe encourage openness, researchers and institutions may still seek patents or other protection for commercially valuable results. Balancing openness with intellectual property protection requires careful timing and legal coordination, particularly to avoid jeopardizing patent eligibility through premature public disclosure.

Additional legal challenges involve compliance with privacy regulations, confidentiality obligations, and international data protection laws such as the General Data Protection Regulation (GDPR). Sensitive datasets, especially in medical or social research, cannot always be fully shared openly. Consequently, open science policies increasingly emphasize the principle “as open as possible, as closed as necessary,” recognizing that legal and ethical restrictions are sometimes essential.

7.7.1 The regulatory framework for NETWORK datasets

Aiming at providing crucial and high value scientific research data to the key players in the EU funded research society, the NETWORK Federated Repository for b5G/6G networks needs to evaluate and adopt an effective internal regulatory framework. This should:

1. Provide a wide portfolio of the methods for data sharing and exploitation that may cover the needs of one of the various organizations that create datasets in order to encourage the potential providers of datasets
2. Become updated after a revision process conducted by the body that will operate and supervise the repository. This can be an action conducted on a periodic basis (i.e., yearly) or when an urgent need suggests to initiate it.

This approach will allow the repository to adapt to the varying external conditions and maximize its long-lasting positive impact to the EU research community.

8 Conclusions

This deliverable presents the final state of the NATWORK federated repository activities reported under T4.5, focusing on the collection, organization, documentation, and sharing of data generated within the project. Building on D4.5: “NATWORK Federated repository for B5G/6G networks (cross layer AI-based attacks Datasets). r1”, which introduced the initial repository concept, the first version of the CKAN-based platform, and the preliminary mapping of partner data practices, D4.6 consolidates the NATWORK Data Management Platform into an operational project asset.

The deliverable first revisits the main technologies, commercial tools, telecommunication repositories, and relevant EU initiatives related to federated data management. This analysis confirms the relevance of the adopted approach, where heterogeneous datasets are preserved in their native formats while being made discoverable, accessible, and reusable through a common repository layer.

The NATWORK ecosystem has also been updated to reflect the current project status. Consortium partners now act as data providers, the CKAN-based DMP acts as the common catalogue and access point, and the technical WPs act as consumers of the shared data for AI-based security, validation, integration, and exploitation activities. The repository supports dataset registration, metadata description, controlled access, preview, visualization, and download, while maintaining compatibility with heterogeneous formats such as CSV, PCAP, structured logs, ML-ready files, and JSON/STIX-based CTI information.

A major contribution of D4.6 is the detailed documentation of the datasets generated during the NATWORK project. These datasets cover a broad set of B5G/6G security and networking scenarios, including jamming detection, channel estimation, jamming mitigation, Kubernetes anomaly detection, CNF live migration, sub-THz channel estimation, DDoS detection, OT/ICS attacks, denial-of-sustainability attacks, CTI, 5G traffic analysis, and multimodal network/container monitoring. Their description includes dataset structures, parameters, statistical summaries, figures, and file-level information, improving traceability and supporting future reuse.

The deliverable further extends the cross-layer AI-based attacks analysis. In addition to the AI-DoS attack, D4.6 includes the jamming attack setup, its system architecture, experimental environment, and validation process. These attack-related activities demonstrate how the NATWORK repository can support both offensive security experimentation and defensive AI model development.

From a data governance perspective, the consortium's practices remain aligned with the principles established in D1.2. The datasets do not involve personal data, and no ethics or legal issues have been identified for the data currently described. Sensitive data, where applicable, is handled through controlled access, institutional restrictions, or partner-side repositories. Storage costs are mainly covered by partner facilities and the NATWORK DMP infrastructure.

Overall, D4.6 confirms that the NATWORK federated repository has progressed from an initial platform definition to a concrete data-sharing and exploitation environment. It provides the technical basis for secure and structured dataset exchange within the consortium, supports AI-powered B5G/6G security research, and establishes a reusable data asset that can contribute to future publications, demonstrations, validation activities, and selected dissemination within the SNS-JU community, subject to access and confidentiality constraints.

References

- [1] Luckham, D. C. (2005). *The power of events: An introduction to complex event processing in distributed enterprise systems*. ACM.
- [2] Cugola, G., & Margara, A. (2012). Processing flows of information. *ACM Computing Surveys*, 44(3), 1–62. <https://doi.org/10.1145/2187671.2187677>.
- [3] Roudjane, M., Rebaine, D., Khoury, R., & Halle, S. (2019, October). Predictive analytics for event stream processing. In *Proceedings of the 2019 IEEE 23rd International Enterprise Distributed Object Computing Conference (EDOC)* (pp. 171–182). IEEE. <https://doi.org/10.1109/EDOC.2019.00029>.
- [4] Zaharia, M., Das, T., Li, H., Hunter, T., Shenker, S., & Stoica, I. (2013, November). Discretized streams. In *Proceedings of the 24th ACM Symposium on Operating Systems Principles* (pp. 423–438). ACM. <https://doi.org/10.1145/2517349.2522737>.
- [5] Noghabi, S. A., Paramasivam, K., Pan, Y., Ramesh, N., Bringham, J., Gupta, I., & Campbell, R. H. (2017). Samza: Stateful scalable stream processing at LinkedIn. *Proceedings of the VLDB Endowment*, 10(12), 1634–1645. <https://doi.org/10.14778/3137765.3137770>.
- [6] Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012, August). Fog computing and its role in the Internet of Things. In *Proceedings of the First MCC Workshop on Mobile Cloud Computing* (pp. 13–16). ACM. <https://doi.org/10.1145/2342509.2342513>.
- [7] Li, M., Liu, M., Ding, L., Rundensteiner, E. A., & Mani, M. (2007). Event stream processing with out-of-order data arrival. In *27th International Conference on Distributed Computing Systems Workshops (ICDCSW '07)* (p. 67). IEEE. <https://doi.org/10.1109/ICDCSW.2007.35>
- [8] Król, M., & Psaras, I. (2017). NFaaS: Named function as a service. In *Proceedings of the 4th ACM Conference on Information-Centric Networking (ICN)*. ACM.
- [9] Psaras, I., Ascigil, O., Rene, S., Pavlou, G., Afanasyev, A., & Zhang, L. (2018). Mobile data repositories at the edge. In *USENIX Workshop on Hot Topics in Edge Computing (HotEdge 18)*.
- [10] Apache Hadoop. <https://ha-doop.apache.org/> (Accessed 2025/06/16)
- [11] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1), 107–113.
- [12] Apache Spark™ - Unified Analytics Engine for Big Data. <https://spark.apache.org/> (Accessed 2025/06/16)
- [13] Apache Storm. <https://storm.apache.org/> (Accessed 2025/06/16)
- [14] Apache Samza. <https://samza.apache.org/> (Accessed 2025/06/16)
- [15] Debauche, O., El Moulal, M., Mahmoudi, S., Boukraa, S., Manneback, P., & Lebeau, F. (2018). Web monitoring of bee health for researchers and bee-keepers based on the internet of things. *Procedia computer science*, 130, 991–998.
- [16] Apache Flume. <https://flume.apache.org/> (Accessed 2025/06/20)

- [17] Kumar, S., & Goel, E. (2018, April). Changing the world of Autonomous Vehicles using Cloud and Big Data. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT) (pp. 368-376). IEEE.
- [18] Samant, S. S., Chhetri, M. B., Vo, Q. B., Kowalczyk, R., & Nepal, S. (2018, July). Towards end-to-end qos and cost-aware resource scaling in cloud-based iot data processing pipelines. In 2018 IEEE International Conference on Services Computing (SCC) (pp. 287-290). IEEE.
- [19] Apache Flink 1.12 Documentation. <https://ci.apache.org/projects/flink/flink-docs-release-1.12/> (Accessed 2025/06/16)
- [20] Battulga, D., Miorandi, D., & Tedeschi, C. (2020, February). FogGuru: a Fog Computing Platform Based on Apache Flink. In 2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN) (pp. 156-158). IEEE.
- [21] Apache Kafka. <https://kafka.apache.org/> (Accessed 2025/06/16)
- [22] Amazon Web Services. AWS cloud computing services. <https://aws.amazon.com> (Accessed 2025/06/20)
- [23] CRAWDAD. <https://crawdad.org/> (Accessed 2025/06/12)
- [24] CAIDA Data Repository. <https://www.caida.org/catalog/datasets/> (Accessed 2025/06/12)
- [25] Measurement Lab Data. <https://www.measurementlab.net/data/> (Accessed 2025/06/20)
- [26] Ookla Open Data. <https://github.com/teamookla/ookla-open-data> (Accessed 2025/06/12)
- [27] OpenCellID. <https://www.opencellid.org/> (Accessed 2025/06/12)
- [28] Federal Communications Commission Broadband Data Collection. Broadband Data Collection. <https://www.fcc.gov/BroadbandData> (Accessed 2025/06/12)
- [29] NYU WIRELESS Datasets. <https://wireless.engineering.nyu.edu/> (Accessed 2025/06/20)
- [30] COSMOS Testbed Datasets. <https://wiki.cosmos-lab.org/wiki/Datasets> (Accessed 2025/06/12)
- [31] EdgeNet Project. <https://edge-net.org/> (Accessed 2025/06/12)
- [32] Shared Automation Operating Models for Worldwide Adoption (SHOW) Project, grant agreement No 875530. <https://show-project.eu/> (Accessed 2025/06/20)
- [33] Gaia-X: A Federated Secure Data Infrastructure, <https://gaia-x.eu/> (Accessed 2025/06/20)
- [34] Trustworthy. Energy-Aware federated DATA Lakes along the computing continuum. <https://doi.org/10.3030/101070186>
- [35] Data markets for IoT federations. <http://www.iotfeds.gr/> (Accessed 2025/06/20)
- [36] Federated Interoperable Semantic IoT/cloud Testbeds and Applications. <https://doi.org/10.3030/643943>
- [37] 5G-EPICENTRE Project. (2021). 5G ExPerimentation Infrastructure hosting Cloud-native Netapps for public protection and disaster RELief. <https://www.5gepicentre.eu/>
- [38] European Plate Observing System (EPOS). <https://www.epos-eu.org> (Accessed 2025/06/20)

- [39] Data Management in Distributed, Federated Research Infrastructures: The Case of EPOS. <https://datascience.codata.org/articles/1523/files/65b107f887544.pdf>
- [40] Data Infrastructure Capacity for EOSC (DICE). www.dice-eosc.eu (Accessed 2025/06/16)
- [41] Kousias, K., Proestakis, E., Kartsakli, E., Antonopoulos, A., & Verikoukis, C. (2024). A large-scale dataset of 4G, NB-IoT, and 5G non-standalone network measurements. *IEEE Communications Magazine*, 62(5), 44–49, <https://doi.org/10.1109/MCOM.011.2200707>
- [42] Raca, D., Leahy, D., Sreenan, C. J., & Quinlan, J. J. (2020, June 8–11). Beyond throughput, the next generation: A 5G dataset with channel and context metrics. In *Proceedings of the ACM Multimedia Systems Conference (MMSys)*, Istanbul, Turkey
- [43] Bouchelaghem, S., Boudjelaba, H., Omar, M., & Amad, M. (2022). User mobility dataset for 5G networks based on GPS geolocation. In *2022 IEEE 27th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)* (pp. 128–133). IEEE. <https://doi.org/10.1109/CAMAD55695.2022.9966906>
- [44] Samarakoon, S., Siriwardhana, Y., Porambage, P., Liyanage, M., Chang, S. Y., Kim, J., ... & Ylianttila, M. (2022). 5G-NIDD: A comprehensive network intrusion detection dataset generated over 5G wireless network. *arXiv preprint*, arXiv:2212.01298. <https://arxiv.org/abs/2212.01298>.
- [45] Priya, K., & Ranjan, R. (n.d.). AI/ML for 5G-energy consumption modelling by ITU AI/ML in 5G Challenge [GitHub repository]. GitHub. <https://github.com/ITU-AI-ML-in-5G-Challenge/5G-Energy-Consumption-Modelling-Solution-Team-Farzi-Data-Scientists>
- [46] Abdullahi, Abdijalil & Elmi, Abdikadir & Bare, Mohamed. (2024). A Comprehensive Analysis of 5G Network Deployment Using Machine Learning Techniques. *International Journal of Electrical and Electronics Engineering*. 11. 22-29. 10.14445/23488379/IJEEE-V11I10P103.
- [47] Chen, W., Lin, X., Lee, J., Toskala, A., Sun, S., Chiasserini, C. F., & Liu, L. (2023). 5G-advanced toward 6G: Past, present, and future. *IEEE Journal on Selected Areas in Communications*, 41(6), 1592-1619.
- [48] Uusitalo, M. A., Rugeland, P., Boldi, M. R., Strinati, E. C., Demestichas, P., Ericson, M., ... & Zou, Y. (2021). 6G vision, value, use cases and technologies from European 6G flagship project Hexa-X. *IEEE access*, 9, 160004-160020.
- [49] Björklund, M. (Ed.). (2017). *YANG - A data modeling language for the Network Configuration Protocol (NETCONF)* (RFC 7950). IETF. <https://tools.ietf.org/html/rfc7950>
- [50] Turnbull, J. (2014). *The Docker Book: Containerization is the new virtualization*. James Turnbull.
- [51] Egli, Peter. (2017). MQTT - Message Queueing Telemetry Transport Introduction to MQTT, a protocol for M2M and IoT applications. doi: 10.13140/RG.2.2.13210.54721.
- [52] Mosquito. <https://mosquitto.org/> (Accessed 2025/06/16)
- [53] PostgreSQL. <https://www.postgresql.org/about/> (Accessed 2025/06/16)

- [54] Open Knowledge Foundation. CKAN: Comprehensive Knowledge Archive Network (Version 2.11) [Computer software]. <https://ckan.org/> (Accessed 2025/06/16)
- [55] S. Samarakoon, Y. Siriwardhana, P. Porambage, M. Liyanage, S.-Y. Chang, J. Kim, J. Kim, M. Ylianttila (2022). 5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network. IEEE Dataport. <https://dx.doi.org/10.21227/xtep-hv36>
- [56] AbdulGhaffar, A., Mahyoub, M., & Matrawy, A. (2024, May). On the Impact of Flooding Attacks on 5G Slicing with Different VNF Sharing Configurations. In *2024 20th International Conference on the Design of Reliable Communication Networks (DRCN)* (pp. 136-142). IEEE.
- [57] Caccavale, F. G., Nguyen, H. N., Cavalli, A., Montes De Oca, E., & Mallouli, W. (2023, August). HTTP/2 Attacks Generation using 5Greplay. In *Proceedings of the 18th International Conference on Availability, Reliability and Security* (pp. 1-7).
- [58] Loureiro, L., Pereira, V., Cruz, T., & Simões, P. (2024, May). Enhancing 5G Core security with eBPF/XDP. In *NOMS 2024-2024 IEEE Network Operations and Management Symposium* (pp. 1-6). IEEE.
- [59] Mamolar, A. S., Salvá-García, P., Chirivella-Perez, E., Pervez, Z., Calero, J. M. A., & Wang, Q. (2019). Autonomic protection of multi-tenant 5G mobile networks against UDP flooding DDoS attacks. *Journal of Network and Computer Applications*, 145, 102416.
- [60] Singh, V. (2023, November). Brutector: A Probabilistic Detection Model for Brute-force Attacks in SSH Server. In *2023 16th International Conference on Security of Information and Networks (SIN)* (pp. 1-8). IEEE.
- [61] Everson, D., Cheng, L., & Zhang, Z. (2022, February). Log4shell: Redefining the web attack surface. In *Proc. Workshop Meas., Attacks, Defenses Web (MADWeb)* (pp. 1-8).
- [62] Domb, M., Balaji, C. G., Menaka, S., Gayathri, A., & Joshi, S. (2025). Securing 5G Networks by Mitigating Cybersecurity Risks for Transformative Applications. *International Journal of Interactive Mobile Technologies*, 19(11).
- [63] Lu, N., Li, D., Shi, W., Vijayakumar, P., Piccialli, F., & Chang, V. (2021). An efficient combined deep neural network based malware detection framework in 5G environment. *Computer Networks*, 189, 107932.
- [64] Alwaisi, Z. (2025). Memory-efficient and robust detection of Mirai botnet for future 6G-enabled IoT networks. *Internet of Things*, 101621.
- [65] Savadatti, S., Kuldeep Dhariwal, S., Krishnamoorthy, S., & Delhibabu, R. (2024). An Extensive Classification of 5G Network Jamming Attacks. *Security and Communication Networks*, 2024(1), 2883082.
- [66] Vidal, J. M., Monge, M. A. S., & Villalba, L. J. G. (2018, August). Detecting Workload-based and Instantiation-based Economic Denial of Sustainability on 5G environments. In *Proceedings of the 13th International Conference on Availability, Reliability and Security* (pp. 1-8).

- [67] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning," *IEEE Access*, vol. 10, pp. 40281-40306, Apr. 2022. DOI: 10.1109/ACCESS.2022.3165809
- [68] Chowdhury, F. Z., Kiah, L. B. M., Ahsan, M. M., & Idris, M. Y. I. B. (2017, August). Economic denial of sustainability (EDoS) mitigation approaches in cloud: Analysis and open challenges. In *2017 International Conference on Electrical Engineering and Computer Science (ICECOS)* (pp. 206-211). IEEE.
- [69] Turnbull, J. (2018). *Monitoring with Prometheus*. Turnbull Press.

Annexes

A.1 Data generated per partner in the project consortium

Table 44 provides a summary of the data generated by each partner according to the deliverable D1.2.

Table 44: Data generated per partner in the project consortium, input from D1.2

Partner	Data Sources	Doc. & Data Quality	Research outcomes	Cost	Data Security	Data ethics	Personal Data	Other procedures for DM
CERTH	Generated and public datasets	Readme files & Doc.	Release of data via publications	Cost for storage in partner facilities	Federated repository	No ethics/legal issues	No personal data	No national, funder, sectorial, or departmental procedures for data management.
GRAD	Generated and public	Readme files	Release of data via publications	Cost for storage in partner facilities	No sensitive data	No ethics/legal issues	No personal data	Same
TSS	Generated	Readme files	Publicly available	Cost for storage in partner facilities	No sensitive data	No ethics/legal issues	No personal data	Same

Partner	Data Sources	Doc. & Data Quality	Research outcomes	Cost	Data Security	Data ethics	Personal Data	Other procedures for DM
CNIT	Public and Generated (if it is needed)	Readme files	Publicly available	Cost for storage in partner facilities	Sensitive data to CNIT repos	No ethics/legal issues	No personal data	Same
ZHAW	Generated and public	Readme files & Doc.	Release of data via publications	Cost for storage in partner facilities	Sensitive data to ZHAQ repos	No ethics/legal issues	No personal data	Same
Uessex	Generated	Doc.	Publicly available	Cost for storage in partner facilities	No sensitive data	No ethics/legal issues	No personal data	Same
ISRD	Generated and public datasets	Readme files	Release of data via publications	Cost for storage in partner facilities	No sensitive data	No ethics/legal issues	No personal data	Same
ELTE	Generated and public datasets	Readme files & Doc.	Release of data via publications	Cost for storage in partner facilities	Sensitive data to ELTE repos	No ethics/legal issues	No personal data	Same
MONT	Generated and public datasets	Doc.	Publicly available	Cost for storage in partner facilities	No sensitive data	No ethics/legal issues	No personal data	Same

Partner	Data Sources	Doc. & Data Quality	Research outcomes	Cost	Data Security	Data ethics	Personal Data	Other procedures for DM
IMEC	Generated	Doc.	Release of data via publications	Cost for storage in partner facilities	No sensitive data	No ethics/legal issues	No personal data	Same
UZH	Generated	Doc.	Publicly available	Cost for storage in partner facilities	Sensitive data to UZH repos	No ethics/legal issues	No personal data	Same
NEC	Public	STIX format	Release of data via publications	Cost for storage in partner facilities	No sensitive data	No ethics/legal issues	No personal data	Same
NOVA	Data produced within NATWORK	NA	NA	Cost for storage in partner facilities	No sensitive data	No ethics/legal issues	No personal data	Same
PNET		Readme files	Release of data via publications	Cost for storage in partner facilities	No sensitive data	No ethics/legal issues	No personal data	Same

A.2 Collection of information by each data related partner

In order to identify the precise technologies required to ensure that the proposed architecture of the Data Repository meets NATWORK's needs, the following information was requested:

1. Identification of suitable database technologies and data management tools.
2. Definition of real-time streaming frameworks and interfaces.
3. Proposals for interoperability standards and data format compatibility.
4. Definition of the main data attributes' names.
5. Indicative data format (either real-time or CSV), specifying structure and metadata requirements.

This information was gathered using Table 45 as a template.

Table 45: Template for gathering data handling information related to NATWORK's Data repository

ID	Partner Feedback
#1	MongoDB, PostgreSQL, Lenses.
#2	Apache Kafka, RabbitMQ
#3	Rest APIs, MQTT
#4	Timestamp, I,Q, SNR, BER
#5	CSV format <pre>timestamp,I,Q,SNR,BER 2025-03-10T08:15:00Z,0.123,-0.456,25.7,0.00002</pre> JSON format <pre>{ "timestamp": "2025-03-10T08:15:00Z", "I": 0.123, "Q": -0.456, "SNR": 25.7, "BER": 0.00002 }</pre>

The information/feedback gathered by relevant partners are presented in the following Tables:

Table 46: CERTH data handling information

ID	Input
1	MongoDB/Influx DB
2	N/A
3	Rest APIs, MQTT
4	-
5	Flow Identifier • Source IP • Source Port • Destination IP • Destination Port Flow features • Flow Duration • Total Forward Packets • Total Backward Packets • Total Forward Bytes • Total Backward Bytes • Forward Packet Length Mean • Forward Packet Length Standard Deviation • Forward Packet Length Minimum • Forward Packet Length Maximum • Backward Packet Length Mean • Backward Packet Length Standard Deviation • Backward Packet Length Minimum • Backward Packet Length Maximum • Forward IAT Mean • Forward IAT Standard Deviation

ID	Input
	• Forward IAT Maximum • Forward IAT Minimum • Backward IAT Mean • Backward IAT Standard Deviation • Backward IAT Maximum • Backward IAT Minimum • Flow Bytes per Second • Flow Packets per Second • Packet Length Mean • Packet Length Standard Deviation • Packet Length Variance • Header Length • Payload Bytes • Packet Size Mean • Packet Size Minimum • Packet Size Maximum • Packet Size Standard Deviation • Flow IAT Mean • Flow IAT Standard Deviation • Flow IAT Minimum • Flow IAT Maximum • Fwd/Rev Packet Ratio • Fwd Bytes per Packet Ratio • PSH Flag Count • URG Flag Count • Subflow Count • Subflow Packet Count • Subflow Byte Count • Active Flow Mean Duration

ID	Input
	• Active Flow Maximum Duration • Active Flow Minimum Duration • Active Flow Standard Deviation • Idle Flow Mean Duration • Idle Flow Maximum Duration • Idle Flow Minimum Duration • Idle Flow Standard Deviation • Init_Win_bytes_forward • Init_Win_bytes_backward • Act_data_pkt_fwd • Min_seg_size_forward • Forward Header Length • Backward Header Length Flow ID: { "Src_IP ":192.0.2.15, "Src_Port ": 54123, "Dest_IP ": 198.51.100.45, "Dest_Port ":443} Flow Information: { "flow_duration_ms ": 800, "total_fwd_packets ": 4, "total_bwd_packets ": 2, "total_fwd_bytes ": 512, "total_bwd_bytes ": 64, "fwd_pkt_len_mean ": 128, "fwd_pkt_len_std ": 0, "fwd_pkt_len_min ": 128, "fwd_pkt_len_max ": 128, "bwd_pkt_len_mean ": 32,

ID	Input
	"bwd_pkt_len_std ": 0, "bwd_pkt_len_min ": 32, "bwd_pkt_len_max ": 32, "fwd_iat_mean ": 266.67, "fwd_iat_std ": 5, "fwd_iat_max ": 270, "fwd_iat_min ": 260, "bwd_iat_mean ": 150, "bwd_iat_std ": 10, "bwd_iat_max ": 160, "bwd_iat_min ": 140, "flow_bytes_per_sec ": 720.0, "flow_pkts_per_sec ": 7.5, "pkt_len_mean ": 96, "pkt_len_std ": 44.8, "pkt_len_var ": 2008, "header_len ": 84, "payload_bytes ": 576, "pkt_size_mean ": 96, "pkt_size_min ": 32, "pkt_size_max ": 128, "pkt_size_std ": 44.8, "flow_iat_mean ": 200, "flow_iat_std ": 70, "flow_iat_min ": 100, "flow_iat_max ": 300, "fwd_bwd_ratio ": 2.0, "fwd_bytes_per_pkt_ratio ": 4.0, "psh_flag_count ": 0, "urg_flag_count ": 0,

ID	Input
	<pre> "subflow_count ": 1, "subflow_fwd_pkt_count ": 4, "subflow_fwd_byte_count ": 512, "active_mean ": 300, "active_max ": 400, "active_min ": 200, "active_std ": 50, "idle_mean ": 250, "idle_max ": 300, "idle_min ": 200, "idle_std ": 40, "init_win_bytes_fwd ": 8192, "init_win_bytes_bwd ": 8192, "act_data_pkt_fwd ": 4, "min_seg_size_fwd ": 32, "fwd_header_len ": 42, "bwd_header_len ": 42 } </pre>

Table 47: ZHAW data handling information

ID	Input
1	PostgreSQL, MongoDB
2	Apache Kafka, FastAPI
3	Rest API, OpenTelemetry
4	Timestamp, Source IP, Source Port, Destination IP, Destination Port, ACK Flag, PSH Flag
5	CSV Format:

ID	Input
	ip_source,ip_destination,is_anomaly,source_pod_label,source_namespace_label,source_port_label,destination_pod_label,destination_namespace_label,destination_port_label,ack_flag,psh_flag,timestamp,anomaly_class 10.42.2.35,10.42.3.3,False,color-app-magenta-2-6b7954cd47-xf2,magenta,48468.0,color-app-magenta-1-5c9d47cd95-zj956,magenta,3000.0,True,False,2024-08-08T13:56:53.050069205Z,0 JSON Format: <pre>{ "ip_source": "10.42.2.35", "ip_destination": "10.42.3.3", "is_anomaly": "False", "source_pod_label": "color-app-magenta-2-6b7954cd47-xf2", "source_namespace_label": "magenta", "source_port_label": 48468, "destination_pod_label": "color-app-magenta-1-5c9d47cd95-zj956", "destination_namespace_label": "magenta", "destination_port_label": 3000, "ack_flag": "True", "psh_flag": "False", "timestamp": "2024-08-08T13:56:53.050069205Z", "anomaly_class": 0 }</pre>

Table 48: NEC data handling information

ID	Input
1	MongoDB
2	N/A
3	Rest APIs, MQTT
4	IoC, Malware, vulnerability, relationship

ID	Input
5	<pre>STIX format (JSON) _id: ObjectId('679a4a65b4bccca4538db8f4') type: "malware" id: "malware--c51cfc0e-7ff7-43df-9e2f-42f305d2abb2" created_by_ref: "identity--65fc0e98-f40f-4617-9faa-c9ab0322a5af" created: "2023-09-02T02:59:58.739Z" modified: "2023-09-02T02:59:58.739Z" name: "trojan.floxif/agentwdcr" ▶ kill_chain_phases: Array (1) ▶ labels: Array (1)</pre>

Table 49: MONT data handling information

ID	Input
1	MongoDB
2	Apache Kafka, RabbitMQ, Redis, NATS
3	Rest APIs, MQTT
4	ip.session_id,meta.direction,ip.pkts_per_flow,duration,ip.header_len,ip.payload_len,ip.avg_bytes_tot_len,time_between_pkts_sum,time_between_pkts_avg,time_between_pkts_max,time_between_pkts_min,time_between_pkts_std
5	CSV format

ID	Input

Table 50: ELTE data handling information

ID	Input
1	Influx DB
2	N/A
3	Rest APIs
4	Flow ID, Source IP, Source Port, Destination IP, Destination Port, Protocol, Timestamp, Flow Duration, Total Fwd Packets, Total Backward Packets, Total Length of Fwd Packets, Total Length of Bwd Packets, Fwd Packet Length Max, Fwd Packet Length Min, Fwd Packet Length Mean, Fwd Packet Length Std, Bwd Packet Length Max, Bwd Packet Length Min, Bwd Packet Length Mean, Bwd Packet Length Std, Flow Bytes/s, Flow Packets/s, Flow IAT Mean, Flow IAT Std, Flow IAT Max, Flow IAT Min, Fwd IAT Total, Fwd IAT Mean, Fwd IAT Std, Fwd IAT Max, Fwd IAT Min, Bwd IAT Total, Bwd IAT Mean, Bwd IAT Std, Bwd IAT Max, Bwd IAT Min, Fwd PSH Flags, Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, Fwd Header Length, Bwd Header Length, Fwd Packets/s, Bwd Packets/s, Min Packet Length, Max Packet Length, Packet Length Mean, Packet Length Std, Packet Length Variance, FIN Flag Count, SYN Flag Count, RST Flag Count, PSH Flag Count, ACK Flag Count, URG Flag Count, CWE Flag Count, ECE Flag Count, Down/Up Ratio, Average Packet Size, Avg Fwd Segment Size, Avg Bwd Segment Size, Fwd Header Length, Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk, Bwd Avg Bulk Rate, Subflow Fwd Packets, Subflow Fwd Bytes, Subflow Bwd Packets, Subflow Bwd Bytes, Init_Win_bytes_forward, Init_Win_bytes_backward, act_data_pkt_fwd, min_seg_size_forward, Active Mean, Active Std, Active Max, Active Min, Idle Mean, Idle Std, Idle Max, Idle Min, Label
5	CSV format Flow ID, Source IP, Source Port, Destination IP, Destination Port, Protocol, Timestamp, Flow Duration, Total Fwd Packets, Total Backward Packets, Total Length of Fwd Packets, Total Length of Bwd Packets, Fwd Packet Length Max, Fwd Packet Length Min, Fwd Packet Length Mean, Fwd Packet Length Std, Bwd Packet Length Max, Bwd Packet Length Min, Bwd Packet Length Mean, Bwd Packet Length Std, Flow Bytes/s, Flow Packets/s, Flow IAT Mean, Flow IAT Std, Flow IAT Max, Flow IAT Min, Fwd IAT Total, Fwd IAT Mean, Fwd IAT Std, Fwd IAT Max, Fwd IAT Min, Bwd IAT Total, Bwd IAT Mean, Bwd IAT Std, Bwd IAT Max, Bwd IAT Min, Fwd PSH Flags, Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, Fwd Header Length, Bwd Header Length, Fwd Packets/s, Bwd Packets/s, Min Packet Length, Max Packet Length, Packet Length Mean, Packet Length Std, Packet Length Variance, FIN Flag Count, SYN Flag Count, RST Flag Count, PSH Flag Count, ACK Flag Count, URG Flag Count, CWE Flag Count, ECE Flag Count,

ID	Input
	Down/Up Ratio, Average Packet Size, Avg Fwd Segment Size, Avg Bwd Segment Size, Fwd Header Length,Fwd Avg Bytes/Bulk, Fwd Avg Packets/Bulk, Fwd Avg Bulk Rate, Bwd Avg Bytes/Bulk, Bwd Avg Packets/Bulk,Bwd Avg Bulk Rate,Subflow Fwd Packets, Subflow Fwd Bytes, Subflow Bwd Packets, Subflow Bwd Bytes,Init_Win_bytes_forward, Init_Win_bytes_backward, act_data_pkt_fwd, min_seg_size_forward,Active Mean, Active Std, Active Max, Active Min,Idle Mean, Idle Std, Idle Max, Idle Min, Label 192.168.10.12-192.168.10.50-60148-22-6,192.168.10.12,60148,192.168.10.50,22,6,6/7/2017 1:00,166,1,1,0,0,0,0,0,0,0,0,0,0,0,0,12048.19277,166,0,166,166,0,0,0,0,0,0,0,0,0,0,0,0,32,32,6024.096386,6024.096386,0,0,0,0,0,0,0,0,1,1,0,0,1,0,0,0,32,0,0,0,0,0,0,1,0,1,0,290,243,0,32,0,0,0,0,0,0,0,0,BENIGN JSON format [{ "Flow ID": "192.168.10.12-192.168.10.50-60148-22-6", "Source IP": "192.168.10.12", "Source Port": 60148, "Destination IP": "192.168.10.50", "Destination Port": 22, "Protocol": 6, "Timestamp": "6/7/2017 1:00", "Flow Duration": 166, "Total Fwd Packets": 1, "Total Backward Packets": 1, "Total Length of Fwd Packets": 0, "Total Length of Bwd Packets": 0, "Fwd Packet Length Max": 0, "Fwd Packet Length Min": 0, "Fwd Packet Length Mean": 0, "Fwd Packet Length Std": 0, "Bwd Packet Length Max": 0, "Bwd Packet Length Min": 0, "Bwd Packet Length Mean": 0, "Bwd Packet Length Std": 0, "Flow Bytes/s": 0, "Flow Packets/s": 12048.19277, "Flow IAT Mean": 166, "Flow IAT Std": 0, "Flow IAT Max": 166, "Flow IAT Min": 166, "Fwd IAT Total": 0, "Fwd IAT Mean": 0, "Fwd IAT Std": 0, "Fwd IAT Max": 0, "Fwd IAT Min": 0, "Bwd IAT Total": 0, "Bwd IAT Mean": 0, "Bwd IAT Std": 0, "Bwd IAT Max": 0, "Bwd IAT Min": 0, "Fwd PSH Flags": 0, "Bwd PSH Flags": 0, "Fwd URG Flags": 0, "Bwd URG Flags": 0, "Fwd Header Length": 32, "Bwd Header Length": 32, "Fwd Packets/s": 6024.096386, "Bwd Packets/s": 6024.096386, "Min Packet Length": 0, "Max Packet Length": 0, "Packet Length Mean": 0, "Packet Length Std": 0, "Packet Length Variance": 0, "FIN Flag Count": 0, "SYN Flag Count": 0, "RST Flag Count": 0, "PSH Flag Count": 0, "ACK Flag Count": 1, "URG Flag Count": 1, "CWE Flag Count": 0, "ECE Flag Count": 0, "Down/Up Ratio": 1, "Average Packet Size": 0, "Avg Fwd Segment Size": 0, "Avg Bwd Segment Size": 0, "Fwd Header Length__1": 32, "Fwd Avg Bytes/Bulk": 0, "Fwd Avg Packets/Bulk": 0, "Fwd Avg Bulk Rate": 0, "Bwd Avg Bytes/Bulk": 0, "Bwd Avg Packets/Bulk": 0, "Bwd Avg Bulk Rate": 0, "Subflow Fwd Packets": 1, "Subflow Fwd Bytes": 0, "Subflow Bwd Packets": 1, "Subflow Bwd Bytes": 0, "Init_Win_bytes_forward": 290, "Init_Win_bytes_backward": 243, "act_data_pkt_fwd": 0, "min_seg_size_forward": 32, "Active Mean": 0, "Active Std": 0, "Active Max": 0, "Active Min": 0, "Idle Mean": 0, "Idle Std": 0, "Idle Max": 0, "Idle Min": 0, "Label": "BENIGN" }]

Table 51: TSS data handling information

ID	Input
1	MongoDB
2	Apache Kafka, RabbitMQ
3	Rest APIs, MQTT
4	Timestamp Execution time Call frequency
5	STIX format (JSON)

Table 52: UESSEX data handling information

ID	Input
1	Prometheus Time Series Database (TSDB)
2	node-exporter, kube-state-metrics, cAdvisor
3	REST API (via Prometheus HTTP endpoints)
4	Timestamp, Node/Pod, CPU Usage, Memory Usage, NIC Bandwidth, Metric Name, Value
5	CSV format pod,timestamp,metric_type,value moving-service-67ffb6698d-c4kg2,2025-06-17T16:30:00,cpu,0.08698116307089052 moving-service-67ffb6698d-c4kg2,2025-06-17T16:31:00,cpu,0.13177884526604955 moving-service-67ffb6698d-c4kg2,2025-06-17T16:32:00,cpu,0.00047354625559259736 moving-service-67ffb6698d-c4kg2,2025-06-17T16:33:00,cpu,0.1898272392196254 moving-service-67ffb6698d-c4kg2,2025-06-17T16:34:00,cpu,0.15414587822884146 moving-service-67ffb6698d-c4kg2,2025-06-17T16:35:00,cpu,0.08910558546071423

ID	Input
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:36:00,cpu,0.2069602118827793
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:53:00,cpu,0.07532518697581864
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:54:00,cpu,0.3008671402329336
	moving-service-67ffb6698d-c4kg2,2025-06-17T17:00:00,cpu,0.02937473121738031
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:30:00,memory,30560256
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:31:00,memory,30105600
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:32:00,memory,30093312
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:33:00,memory,29679616
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:34:00,memory,30060544
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:35:00,memory,29429760
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:46:00,memory,30138368
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:47:00,memory,30072832
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:48:00,memory,29696000
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:49:00,memory,30097408
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:33:00, network_bandwidth,188358.21566687652
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:34:00,network_bandwidth,227793.92461542453
	-17T16:40:00,network_bandwidth,67408.7198270156
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:41:00,network_bandwidth,96667.60730729243
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:42:00,network_bandwidth,68978.6849622775
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:43:00,network_bandwidth,0
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:44:00,network_bandwidth,20130.923302393745
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:45:00,network_bandwidth,71019.77771444856
	moving-service-67ffb6698d-c4kg2,2025-06-17T16:46:00,network_bandwidth,87556.77857273893
	JSON format
	{ "status": "success", "data": { "resultType": "matrix", "result": [{ "metric": { "metric_type": "cpu", "pod": "moving-service-67ffb6698d-c4kg2" }, "values": [[1750177800, "0.08698116307089052"], [1750177860, "0.13177884526604955"], [1750178400, "0.07012665826742365"], [1750178460, "0.10734235688918459"], [1750178520, "0.06893488341433096"], [1750178580, "0.00048247

ID	Input
	685336090827"],[1750178640,"0.03489403768054382"],[1750178700,"0.056895962709008666"],[1750178760,"0.08685963660520701"],[1750178820,"0.043811828835137345"],[1750178880,"0.09456284615096958"],[1750178940,"0.13619819585820445"],[1750179000,"0.090580267034894"],[1750179060,"0.11671101993515906"],[1750179120,"0.08177971413349122"],[1750179180,"0.07532518697581864"],[1750179240,"0.3008671402329336"],[1750179300,"0.166221041984382"],[1750179360,"0.03732648476392271"],[1750179420,"0.2367184401807441"],[1750179480,"0.15138559519834377"],[1750179540,"0.027195596728652068"],[1750179600,"0.02937473121738031"]]],{"metric":{"metric_type":"memory","pod":"moving-service-67ffb6698d-c4kg2"},"values":[[1750177800,"30560256"],[1750177860,"30105600"],[1750177920,"30093312"],[1750177980,"29679616"],[1750178040,"30060544"],[1750178100,"29429760"],[1750178160,"29818880"],[1750178220,"29958144"],[1750178280,"29794304"],[1750178340,"30834688"],[1750178400,"29728768"],[1750178460,"31117312"],[1750178520,"30179328"],[1750178580,"30171136"],[1750178640,"27762688"],[1750179420,"30396416"],[1750179480,"30588928"],[1750179540,"30593024"],[1750179600,"29925376"]]]},"metric":{"metric_type":"network_bandwidth","pod":"moving-service-67ffb6698d-c4kg2"},"values":[[1750177800,"73222.32378084844"],[1750177860,"91267.5892143952"],[1750177920,"0"],[1750177980,"188358.21566687652"],[1750178040,"227793.92461542453"],[1750178100,"87071.53123948743"],[1750178160,"181683.8527142828"],[1750178220,"255204.63880698156"],[1750178280,"15311.870305916835"],[1750178340,"176060.34461628963"],[1750178400,"67408.7198270156"],[1750178460,"96667.60730729243"],[1750179420,"232442.46765249537"],[1750179480,"167584.22841604354"],[1750179540,"41824.25747920942"],[1750179600,"0"]]]]]}

For any additional assistance, please refer to the NATWORK DMP support team.

Organizations are the primary way to control who can see, create, and update datasets in CKAN. Each dataset can belong to a single organization, and each organization controls access to its datasets. Datasets can be marked as public or private. The **organization** field ensures you are associated with the correct organization, and the role field designates your authorities within that organization.

When a user joins an organization, an organization admin gives them one of three roles:

- member
- editor
- admin.

A member can:

- View the organization's private datasets.

An editor can do everything a member can plus:

- Add new datasets to the organization.
- Edit or delete any of the organization's datasets.
- Make datasets public or private.

An organization admin can do everything as editor plus:

- Add users to the organization, and choose whether to make the new user a member, editor, or admin.
- Change the role of any user in the organization, including other admin users.
- Remove members, editors, or other admins from the organization.
- Edit the organization itself (for example: change the organization's title, description, or image)
- Delete the organization.

When a user creates a new organization, they automatically become the first admin of that organization.

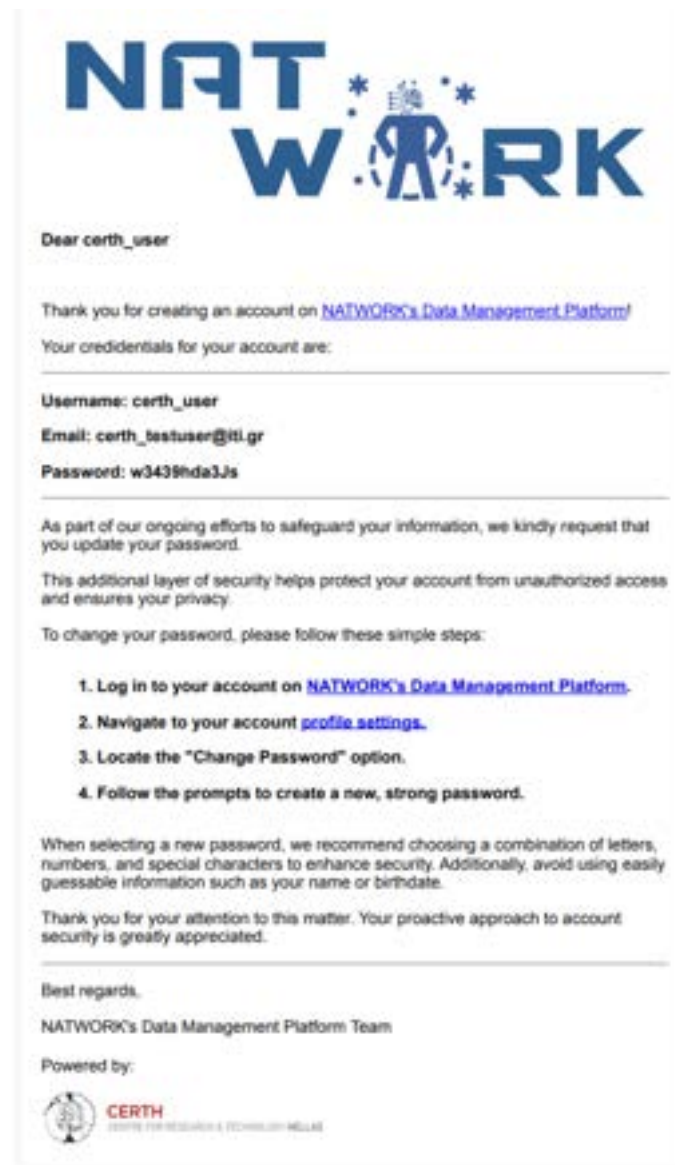


Figure 64: Confirmation e-mail for user registration

Create and Upload Dataset

Once you have obtained access in the NATWORK DMP you can navigate in the landing page of the platform in this URL: <https://NATWORK-data-portal.iti.gr/> . After adding your credentials and login you will navigate in this page as shown below in Figure 65. In this page you can see your uploaded datasets.

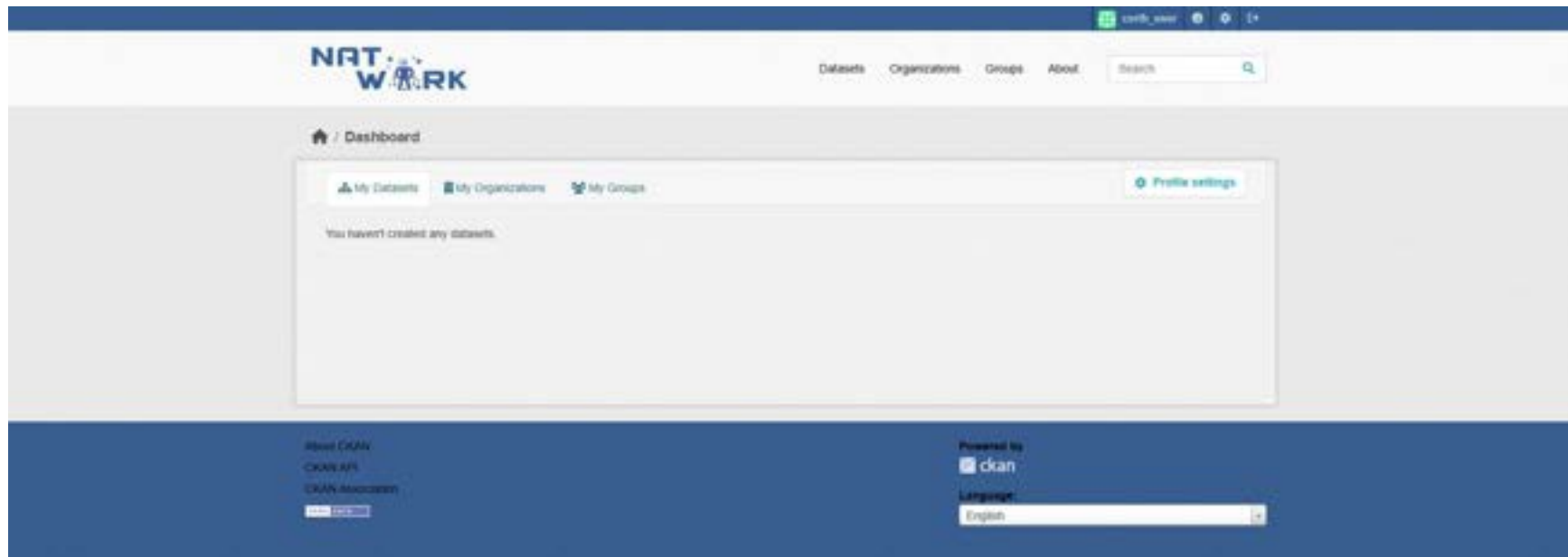


Figure 65: Dashboard Page (Member role user)

If you have a member role you cannot add a new dataset, this functionality is available only for editor and admin roles as stated above. Below in Figure 66, the dashboard page is illustrated for a user with editor privileges.

In the Datasets section, you can view the datasets you have access to, as well as add new datasets. You can add a new dataset by clicking the “Add Dataset” button.

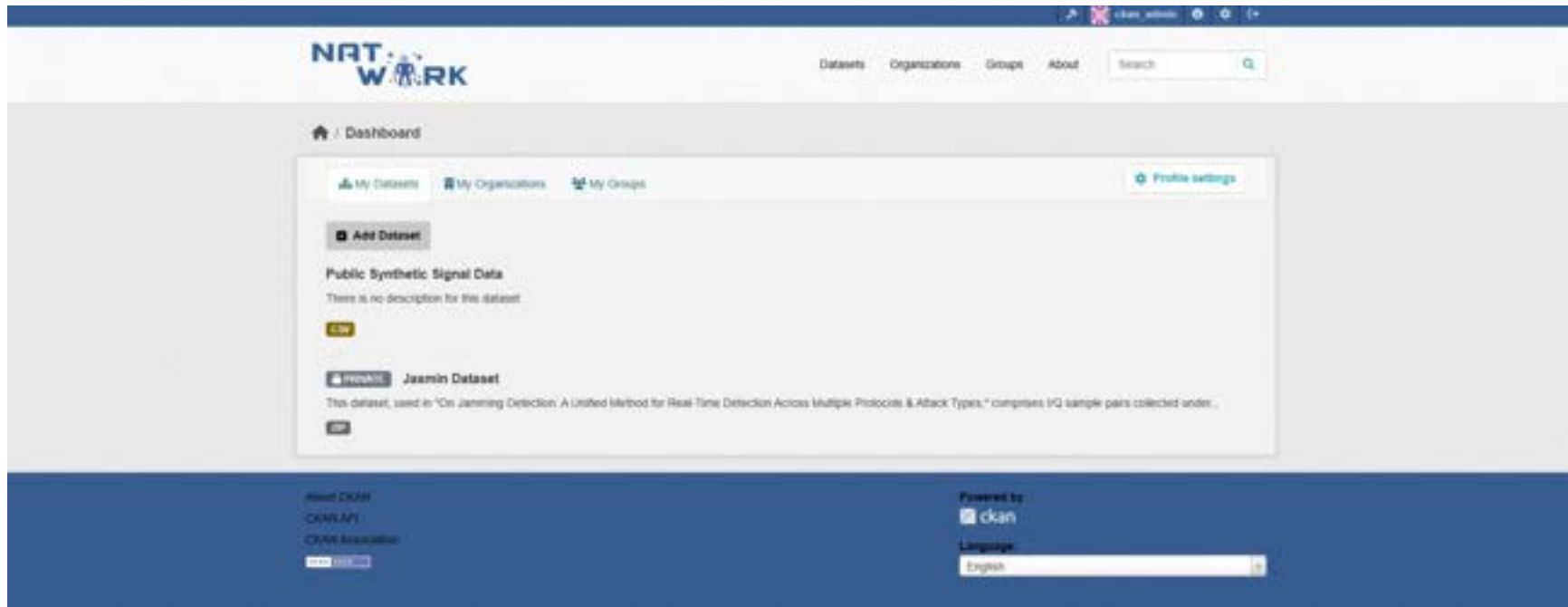


Figure 66: Dashboard Page (Editor role user)

To create a dataset, you must complete and provide specific information. First, you need to fill out a form with details about your dataset, such as the title, description, and URL. While you are only required to complete the mandatory fields, it is advisable to provide as much information as possible. An example of the first form is depicted in Figure 67.

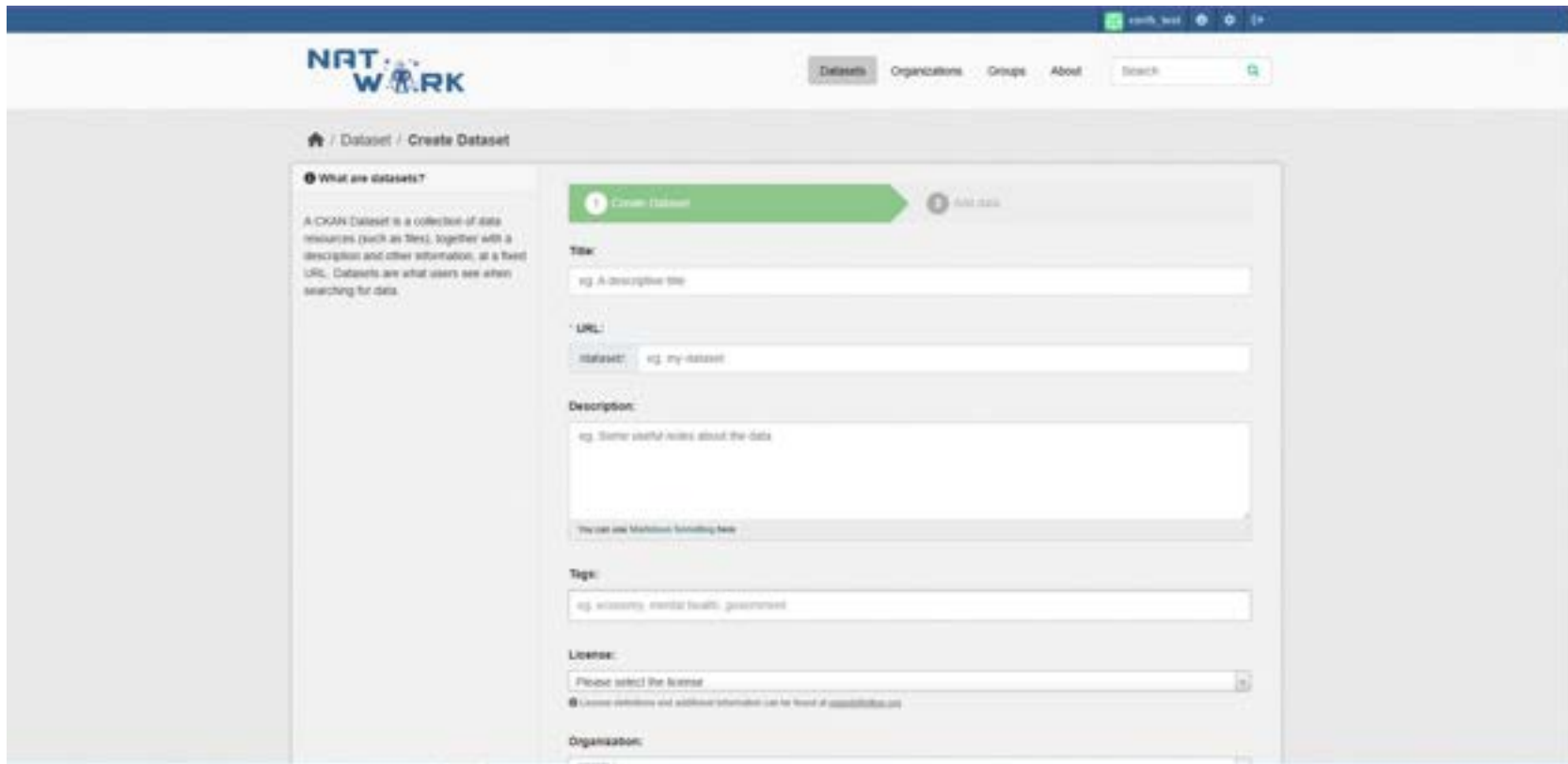


Figure 67: Create Dataset initial form

After completing the initial form, you will be prompted to fill out a subsequent form, as shown in Figure 68. It is important to specify the format type of your dataset in the field named “format.” Once this process is complete, your dataset will be created.

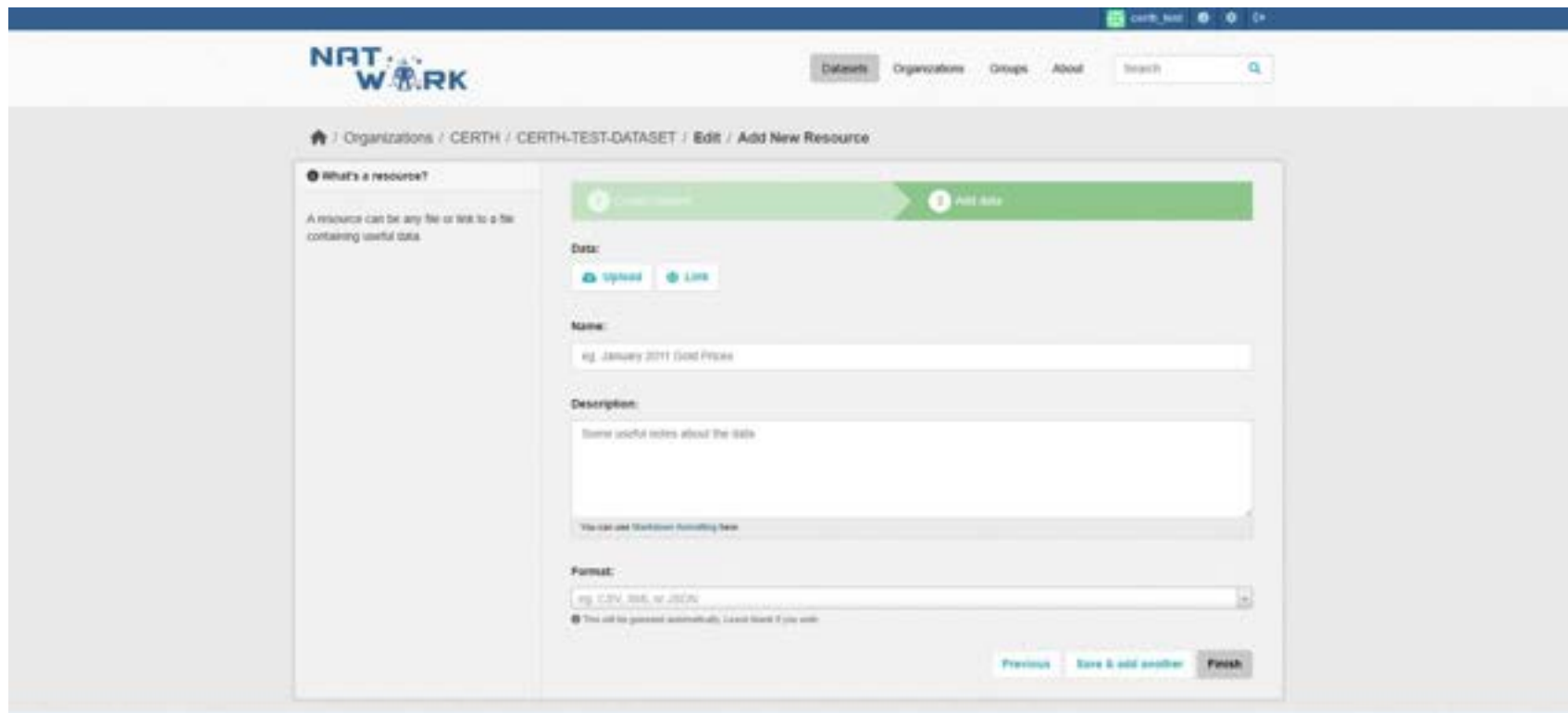


Figure 68: Create Dataset subsequent form

Preview Dataset

In the Organizations section, you can view the organizations within the platform, as shown in Figure 69. You have access only to the organization in which you are registered. By clicking on your organization, you can view the uploaded datasets.

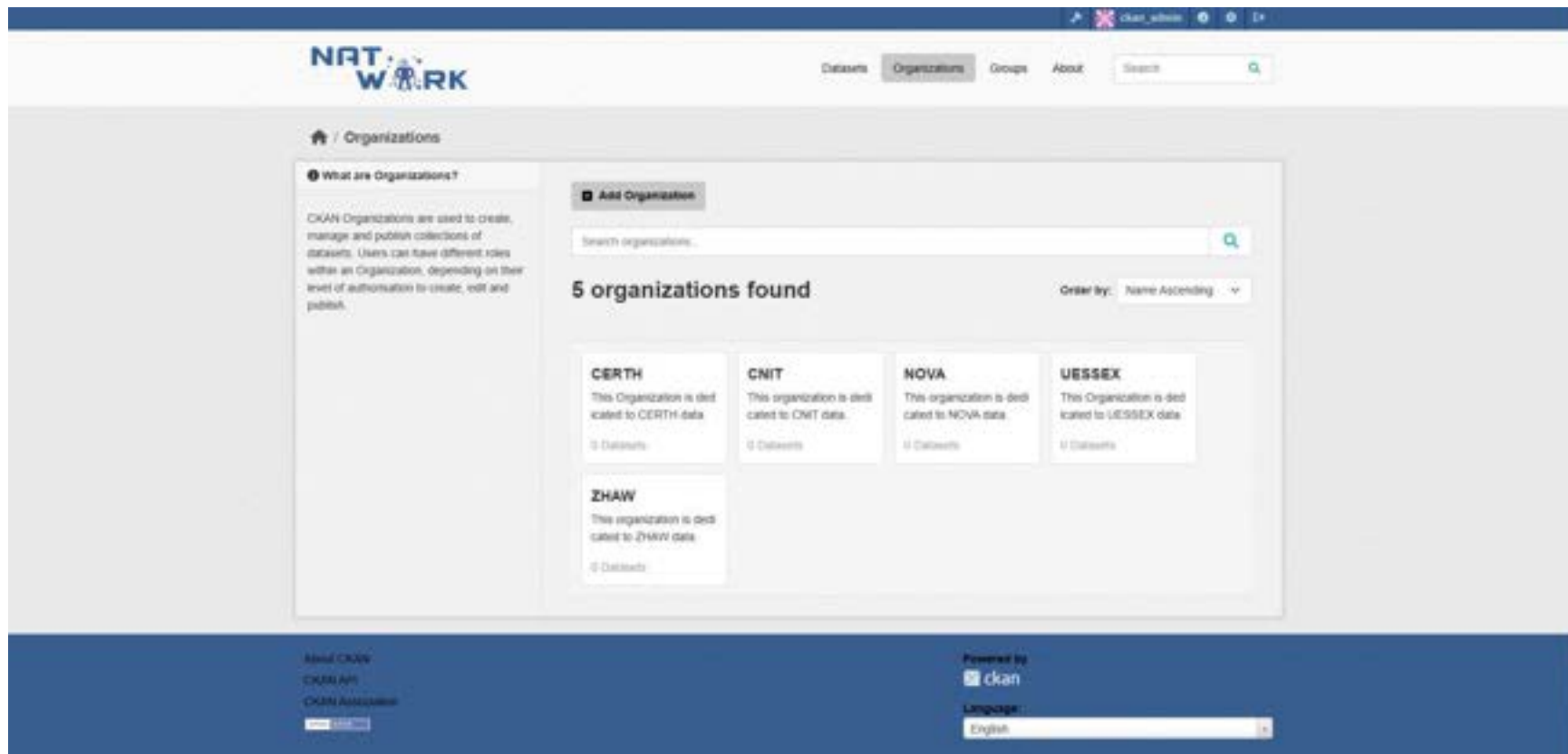


Figure 69: Organizations Page

You can click on and navigate through a dataset within your organization. There, you can view additional information about the selected dataset, as shown in Figure 70 and Figure 71. To see a preview of the dataset, click the “explore” button and select “preview” from the dropdown menu.

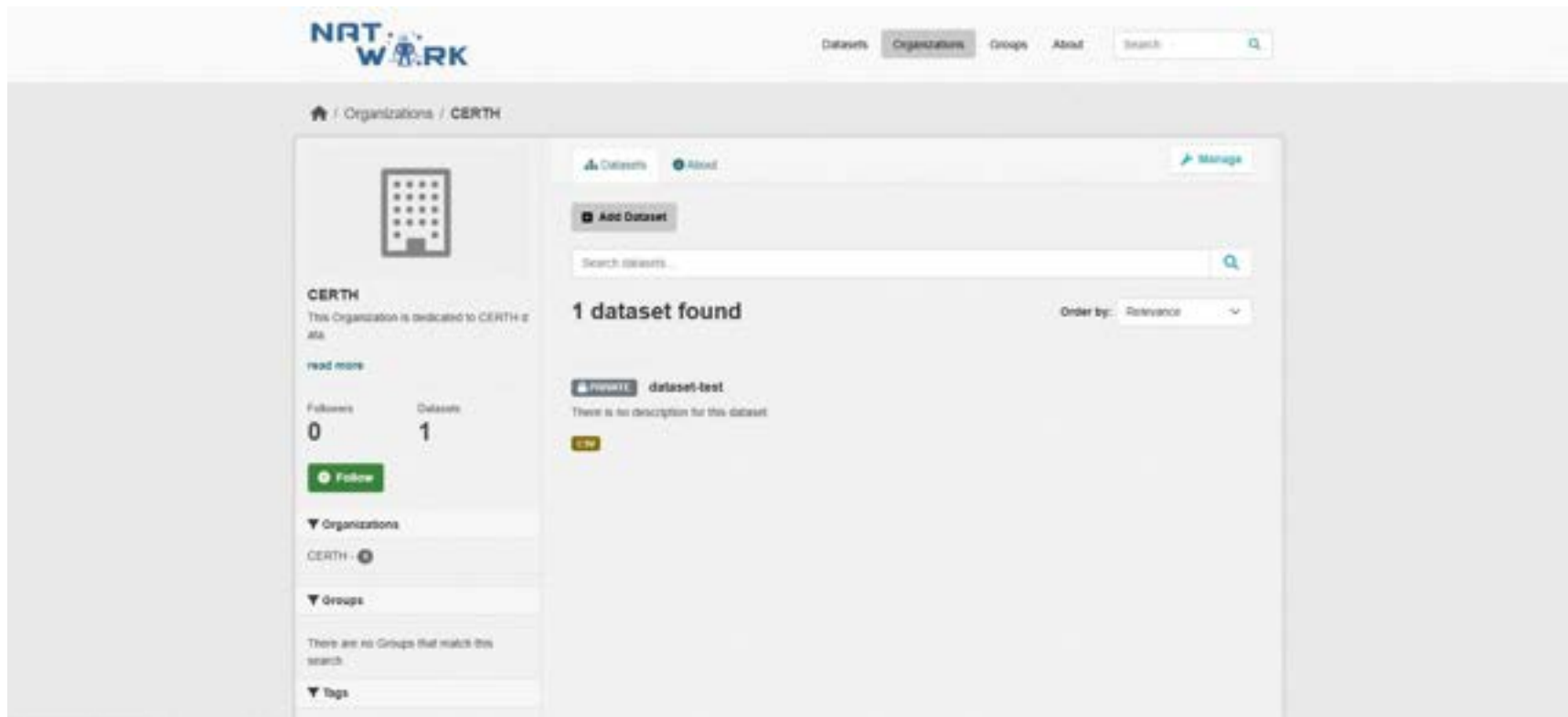


Figure 70: Organization Page

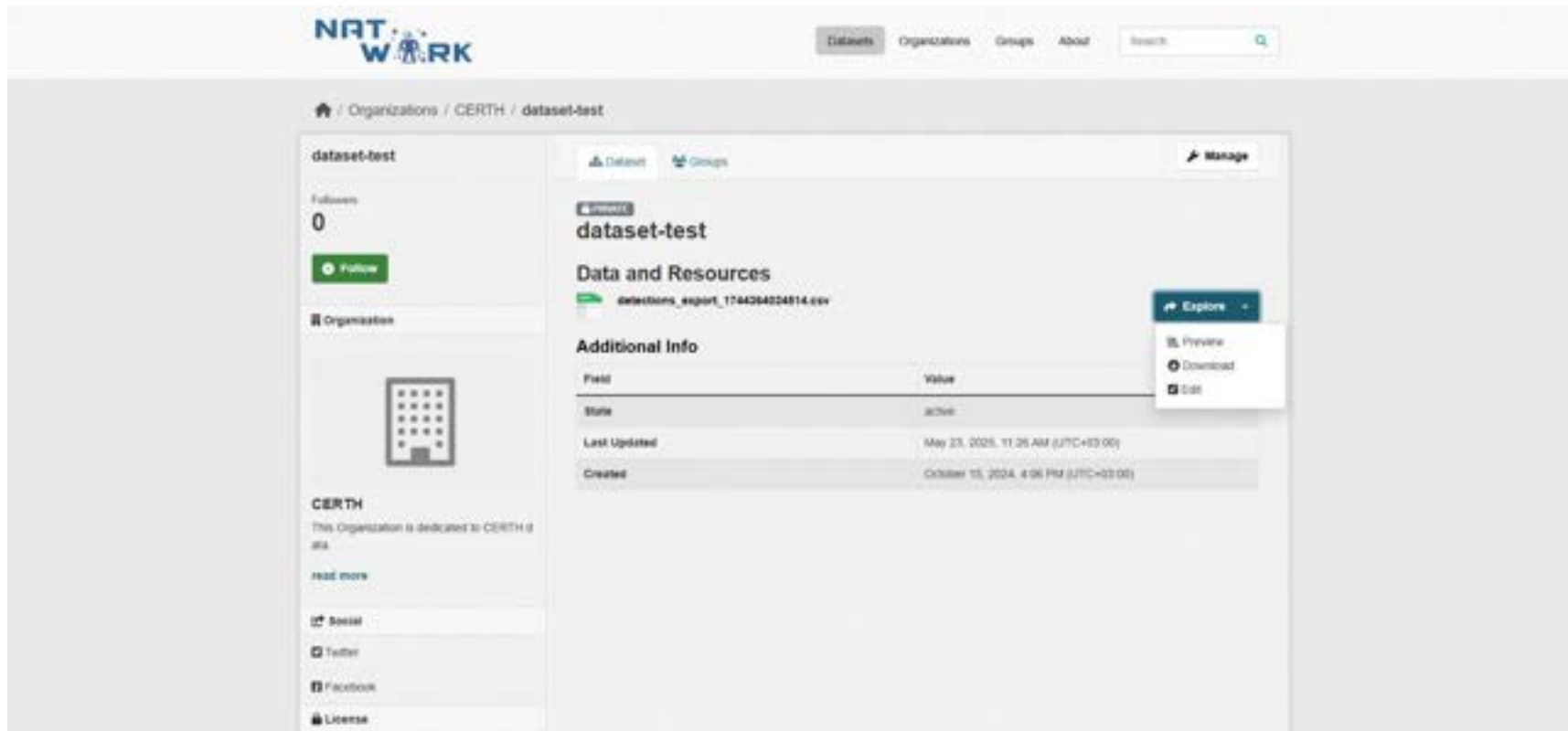


Figure 71: Dataset info through the Organization Page

On the preview page of the dataset, you can view the dataset in the form of a grid, graph, or map. Additionally, you can apply column filters and download the dataset in various file formats such as CSV or JSON, as depicted in Figure 72.

The screenshot shows the NATWORK web interface. At the top, there's a navigation bar with 'Datasets', 'Organizations', 'Groups', and 'About'. Below this, the breadcrumb path is 'Organizations / CERTH / Synthetic Signal Data / synthetic_signal_data.csv'. The main title is 'synthetic_signal_data.csv' with buttons for 'Manage', 'Download', and 'Data API'. A URL is provided: 'https://network-data-portal.it-gr-4131dabw88f344e-6547-4565-95e1-a6f4c500b9e9/resource/7901ac3-7bad-4460-806a-85136ae2ef4/download/synthetic_signal_data.csv'. There's a 'Data Explorer' section with 'Fullscreen' and 'Embed' buttons. Below that, an 'Add Filter' button is present. The table has columns: 'id', 'timestamp', 'I', 'Q', 'SNR', and 'BER'. It shows 15 records with values for each column. At the bottom right, there's a search bar and a 'Filters' button.

id	timestamp	I	Q	SNR	BER
1	2025-07	0.62944	0.62563	19.3295	0.00000
2	2025-07	0.91158	0.58856	11.3583	0.00053
3	2025-07	0.78602	0.37756	24.3474	0.00000
4	2025-07	0.62678	0.05706	15.9647	0.00000
5	2025-07	0.26471	0.66872	10.5218	0.00012
6	2025-07	-0.80494	0.20396	29.1750	0.00000
7	2025-07	0.44300	0.47400	26.2782	0.00187
8	2025-07	0.09075	0.30815	16.5046	0.00186
9	2025-07	0.91301	0.31842	16.6142	0.00077
10	2025-07	0.92977	0.49630	17.6113	0.00000
11	2025-07	-0.68477	0.09891	6.23226	0.00043
12	2025-07	0.94118	0.83235	9.00738	0.00011
13	2025-07	0.91433	0.54204	14.1277	0.00779
14	2025-07	-0.02924	0.82667	6.91464	0.00039
15	2025-07	0.62006	0.62634	25.3067	0.00158

Figure 72: Preview Dataset Page

Data API

As shown in Figure 73, there is a green button labeled “Data API” located above and to the right. The Data API allows for reading, searching, and filtering data without the need to download the entire file first. Further documentation for the Data API can be found at this [URL](#).

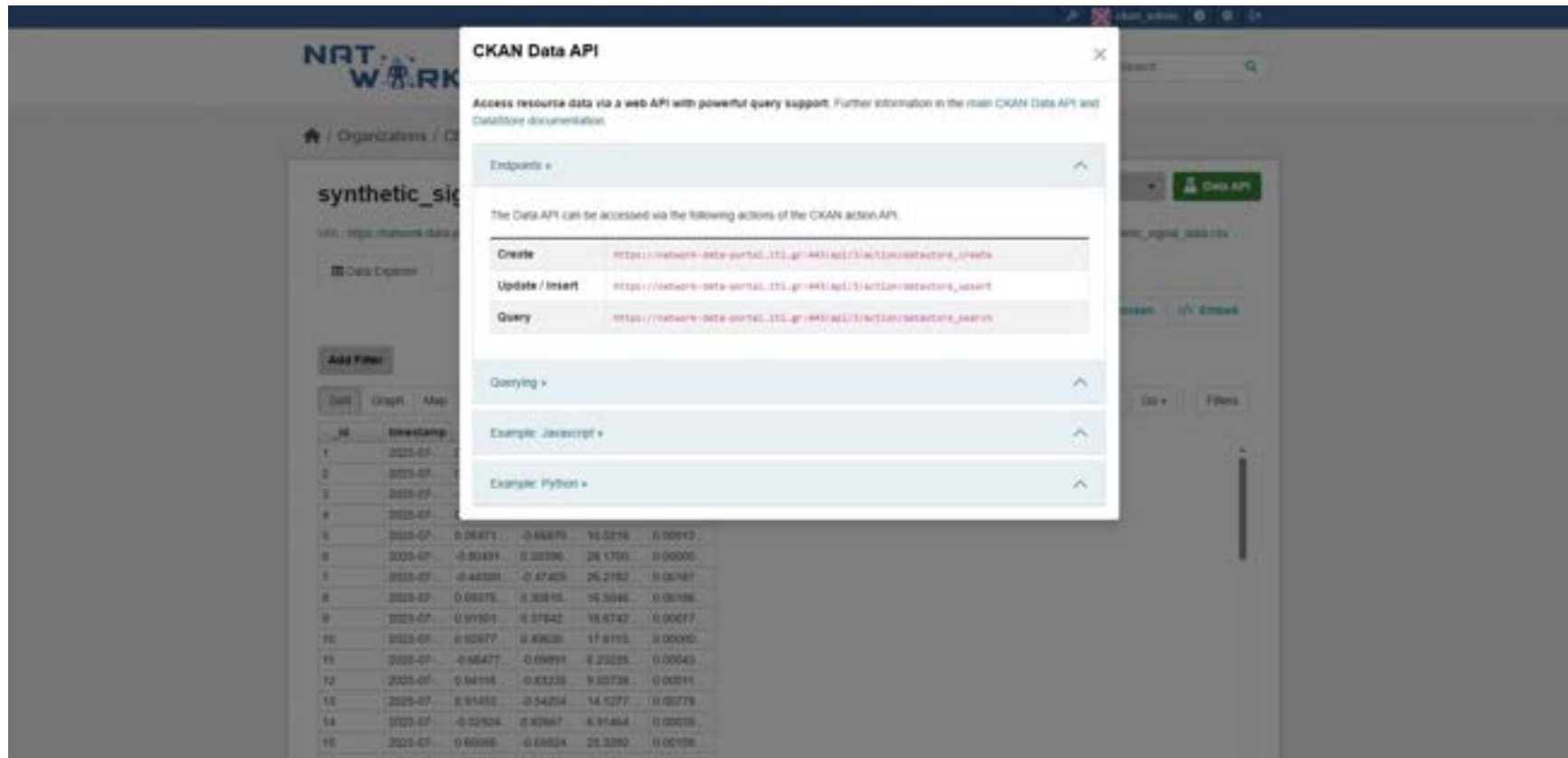


Figure 73: Data API