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

D4.3 Intelligent networking, CTI & explainability.r1

Lead beneficiary	NEC	**Lead author**	Roberto González
Reviewers	Vincent Lefebvre (TSS), Maria Safianowska (ISRD)		
Type	R	**Dissemination**	PU
Document version	1.0	**Due date**	30/06/2025

Co-funded by the European Union

6GSNS

Project funded by

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

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

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)	T4.3, T4.4
Main Author(s)	Roberto González, Jaime Fúster (NEC)
Author(s)	Antonios Lalas, Virgilios Passas, Sarantis Kalafatidis, Asterios Mpatziakas, Evangelos Kopsacheilis, Ioanna Kapetanidou, Nikolaos Makris, Donatos Stavropoulos, Georgios Agrafiotis, Eleni Chamou, Evi Vogiatzi, Konstantinos Nikiforidis, Dimitrios Manolakis, Konstantinos Giapantzis, Thanasis Korakis, Anastasios Drosou (CERTH), Shankha Gupta , Mays Al-Naday, Sumeyya Birtane (UEssex), Wissem Soussi, Gökcan Cantali, Gürkan Gür (ZHAW), Nasim Nezhadsistani, Andy Aidoo (UZH), Joachim Schmidt, Eryk Schiller (HESSO), Sándor Laki, Mohammed Alshawki, Peter Voros (ELTE), Francesco Paolucci, Layal Ismail, Abdul Khan, Michelangelo Guaitolini (CNIT), Vinh Hoa La, Manh Nguyen, Edgardo Montes de Oca (MONT), Ioannis Markopoulos, Angelos Lampropoulos (NOVA)
Reviewers	Vincent Lefebvre (TSS), Maria Safianowska (ISRD)
Type	R - Report
Dissemination level	PU - Public
Due date	M18 (30/06/2025)
Submission date	02/07/2025

Document version history

Version	Date	Changes	Contributor (s)
v0.1	22/01/2025	Template ready	Roberto González (NEC)
v0.1.5	22/01/2025	Initial table of contents	Roberto González (NEC)
v0.2	25/02/2025	SotA Analysis on Explainable AI (XAI)	Gökcan Cantali (ZHAW)
v0.3	15/03/2025	Completed SotA for all technologies	Virgilios Passas, Sarantis Kalafatidis, Asterios Mpatziakas, Ioanna Kapetanidou, Nikolaos Makris, Donatos Stavropoulos, Georgios Agrafiotis (CERTH), Shankha Gupta, Mays Al-Naday, Sumeyya Birtane (UEssex), Wissem Soussi, Gökcan Cantali, Gürkan Gür (ZHAW), Nasim Nezhadsistani, Andy Aidoo (UZH), Joachim Schmidt, Eryk Schiller (HESSO), Sándor Laki, Mohammed Alshawki, Peter Voros (ELTE), Francesco Paolucci, Layal Ismail, Abdul Khan, Michelangelo Guaitolini (CNIT), Vinh Hoa La, Manh Nguyen, Edgardo Montes de Oca (MONT), Ioannis Markopoulos, Angelos Lampropoulos (NOVA)
v0.4	05/04/2025	Contribution for technical Sections	Antonios Lalas, Virgilios Passas, Sarantis Kalafatidis, Asterios Mpatziakas, Evangelos Kopsacheilis, Ioanna Kapetanidou, Nikolaos Makris, Donatos Stavropoulos, Georgios Agrafiotis, Eleni Chamou, Evi Vogiatzi, Konstantinos Nikiforidis, Dimitrios Manolakis, Konstantinos Giapantzis, Thanasis Korakis, Anastasios Drosou (CERTH), Shankha Gupta, Mays Al-Naday, Sumeyya Birtane (UEssex), Wissem Soussi, Gökcan Cantali, Gürkan Gür (ZHAW), Nasim Nezhadsistani, Andy Aidoo (UZH), Joachim Schmidt, Eryk Schiller (HESSO), Sándor Laki, Mohammed Alshawki, Peter Voros (ELTE), Francesco Paolucci, Layal Ismail, Abdul Khan, Michelangelo Guaitolini (CNIT), Vinh Hoa La, Manh Nguyen, Edgardo Montes de Oca (MONT)
v0.5	13/04/2025	Introduction and first draft of Conclusions ready.	Roberto González (NEC)

Version	Date	Changes	Contributor (s)
v0.6	15/05/2025	Contribution for technical Sections	All authors
v0.7	04/06/2025	Additional contribution for technical Sections	All authors
v0.8	10/06/2025	Ready for internal review	Roberto González (NEC)
v0.8.5	24/06/2025	Review complete and feedback to co-authors	Vincent Lefebvre (TSS), Maria Safianowska (ISRD)
v0.9	28/06/2025	Internal review results addressed	All authors
V0.9.5	29/06/2025	Quality control	Joachim Schmidt, Eryk Schiller (HESSO)
v0.9.8	01/07/2025	Quality control results addressed	Roberto González (NEC), Asterios Mpatziakas, Virgilios Passas, Evangelos V. Kopsacheilis (CERTH)
v1.0	02/07/2025	Final version ready for submission	Antonios Lalas (CERTH)

Disclaimer

Copyright message

Contents

List of acronyms and abbreviations

Abbreviation	Description
6G	Sixth Generation (of wireless networks)
AAR	After Action Review
ABE	Attribute-Based Encryption
AI	Artificial Intelligence
AIaaSecS	AI-as-a-Security-Service
API	Application Programming Interface
AMF	Access and Mobility Management Function
ASN	Autonomous System Number
AUSF	Authentication Server Function
B5G	Beyond 5G
BERT	Bidirectional Encoder Representations from Transformers
BNNs	Binarized Neural Networks
CAM	Class Activation Mapping
CKG	Cybersecurity Knowledge Graphs
CNF	Containerized Network Function
CRF	Conditional Random Field
CTI	Cyber Threat Intelligence
CVE	Common Vulnerabilities and Exposures
D2D	Device-to-Device
DARPA	Defense Advanced Research Projects Agency
DFE	Decentralized Feature Extraction
DLAU	Deep Learning Accelerator Unit
DPI	Deep Packet Inspection
DTs	Decision Trees
DLT	Distributed Ledger Technology
DN	Data Network
DoS	Denial of Service
E2E	End-to-End
ECC	Elliptic Curve Cryptosystem
FPGA	Field-Programmable Gate Array
GAT	Graph Attention Network
GDPR	General Data Protection Regulation
GNN	Graph Neural Network
IDS	Intrusion Detection System
INT	In-band Telemetry
IoD	Internet of Drones
IoMT	Internet of Medical Things
IP	Internet Protocol
JSON	JavaScript Object Notation

Abbreviation	Description
KPI	Key Performance Indicator
LIME	Local Interpretable Model-agnostic Explanations
LLMs	Large Language Models
LSTM	Long Short-Term Memory
MAPE-K	Monitor-Analyze-Plan-Execute-Know
MEC	Multi-access Edge Computing
MITM	Man-in-the-Middle
ML	Machine Learning
MOMDP	Multi-Objective Markov Decision Process
MONT	Montimage
MORL	Multi-Objective Reinforcement Learning
MTD	Moving Target Defense
NER	Name Entity Recognition
NFV	Network Functions Virtualization
NIC	Network Interface Card
NIST	National Institute of Standards and Technology
NMT	Neural Machine Translation
NS	Network Service
NSI	Network Slice
OODA	Orient-Observe-Decide-Act
OSINT	Open Source Intelligence
QoS	Quality of Service
PF	Pareto Front
PIRL	Programmatically Interpretable Reinforcement Learning
PUFs	Physical Unclonable Functions
RAN	Radio Access Network
RCA	Root-cause analysis
RIA	Research and Innovation Action
RNN	Recurrent Neural Networks
SHAP	SHapley Additive exPlanations
SDN	Software-Defined Networking
SIEM	Security Information and Event Management
SOM	Self-Organizing Maps
STIX	Structured Threat Information Expression
sTtl	source Time to Live
SUPI	Subscription Permanent Identifier
TEE	Trusted Execution Environment
TI	Threat Intelligence
TTP	Tactics, Techniques, and Procedures
UE	User Equipment
UPF	User Plane Function

Abbreviation	Description
VNF	Virtualized Network Function
VIM	Virtual Infrastructure Manager
vNIC	Virtual Network Interface Card
WP	Work Package
XAI	Explainable Artificial Intelligence
X-MORL	Explainable Multi-Objective Reinforcement Learning
XRL	Explainable Reinforcement Learning
ZT	Zero Trust
ZTN	Zero-Touch Networking

List of figures

List of tables

Executive summary

This deliverable, D4.3 – Intelligent Networking, CTI & Explainability.r1, presents the first results of Work Package 4 (WP4) of the NATWORK project, regarding the development of AI-powered security services for 6G networks. It focuses on integrating AI-based automation, cyber threat intelligence (CTI), and explainability into the orchestration and security management of distributed services. The document outlines the conceptual foundations, architectural models, and initial technical components developed to enable NATWORK's vision of secure, self-adaptive, and trustworthy service environments. It addresses the technical challenges that arise when delegating critical security decisions to AI-driven systems, particularly in terms of real-time responsiveness, human interpretability, and integration with heterogeneous infrastructures.

The work reported on this deliverable is the outcome of two complementary tasks. Task 4.3 introduces the concept of AI-as-a-Security-Service (AIaaSecS), a paradigm in which modular AI components operate across the orchestration layers to detect threats, enforce dynamic policies, and autonomously adapt service configurations in response to changing risk conditions. Task 4.4 focuses on two critical enablers: the integration of multi-source cyber threat intelligence and the design of mechanisms for explainability and observability in AI-based decision-making. Together, these efforts support the development of intelligent network services that are not only secure but also transparent, traceable, and responsive to evolving operational contexts.

Section 2 of the document provides a thorough state-of-the-art review, situating the NATWORK approach about ongoing research in zero-touch orchestration, threat detection, blockchain for trust, and explainable AI. Building on this foundation, Section 3 introduces the initial design of NATWORK's zero-touch solutions, which aim to automate the deployment and lifecycle management of services with minimal human input. Section 4 presents the architecture for real-time threat detection based on AI agents that process telemetry and contextual information to identify attacks and abnormal behaviour. Section 5 outlines a decentralized trust framework based on blockchain technology, supporting secure data sharing and authentication across service components. Section 6 develops the explainability dimension, proposing techniques and tools to ensure that AI decisions—especially those related to security—are understandable and verifiable. Finally, Section 7 describes the NATWORK CTI framework, which enables the ingestion, processing, and operational use of diverse threat intelligence sources.

The main conclusion of this deliverable is that NATWORK's combined use of AI, CTI, and explainability represents a viable and forward-looking approach to securing the next generation of networked services. The architectural designs and early components presented here demonstrate strong technical feasibility and alignment with the project's goals of building resilient, adaptive, and low-overhead cybersecurity solutions for 6G infrastructures. They also

provide a foundation for the deployment of intelligent orchestration platforms that can respond to threats in real-time while maintaining transparency and auditability.

The purpose of Deliverable D4.3 is to consolidate the results of the first phase of technical work within WP4 and to establish a coherent baseline for further development, integration, and testing. It provides a shared reference for partners working on related tasks. It prepares the ground for the implementation and validation activities that will follow in the second half of the project. The final results will be documented in Deliverable D4.4, where the complete system integration and performance evaluation will be presented based on NATWORK's use cases and scenarios.

1. Introduction

The transition toward 6G networks is driving a profound rethinking of how security is designed, integrated, and delivered across digital infrastructures. As services become more distributed, automated, and context-aware, the traditional approaches to cybersecurity—often centralized, static, and reactive—are no longer sufficient. Emerging architectures must embed security by design, leveraging real-time intelligence and AI-driven capabilities to ensure resilience, adaptability, and trustworthiness in an increasingly complex and dynamic environment.

Within this strategic vision, the NATWORK project proposes a bio-inspired, energy-aware, and self-adaptive framework that enables secure and autonomous orchestration of services across the 6G continuum. A core component of this vision is the ability to incorporate AI-powered security services that can proactively defend against threats while maintaining visibility, explainability, and operational accountability. Deliverable D4.3 contributes directly to this ambition, presenting the first major technical outcomes of Work Package 4 (WP4), which focuses on the research and development of such intelligent services.

The work captured in this deliverable, stems from two tightly interrelated tasks. Task 4.3 introduces the concept of AI-as-a-Security-Service (AIaaSecS)—a novel framework that leverages modular AI components to enable proactive and autonomous network defence. These AI agents operate within the orchestration layers of the network, dynamically assessing risks, enforcing security policies, and reacting to threat intelligence in real time. They are designed to interpret and integrate inputs from multiple sources, including CTI feeds, telemetry data, and contextual information, allowing them to anticipate vulnerabilities, detect ongoing attacks, and recommend or trigger appropriate mitigation measures. This marks a fundamental shift from post-facto security enforcement to preventive and adaptive protection embedded directly in service lifecycles.

Task 4.4 complements this work by addressing two equally critical dimensions: cyber threat intelligence (CTI) integration and explainability of AI-driven decisions. On one hand, it focuses on enriching the CTI pipeline by designing mechanisms for the collection, correlation, and contextualization of threat data at different layers of the system—ranging from infrastructure-level events to application and service-level indicators. This enables more precise and timely threat awareness, feeding both human operators and AI-based decision modules. On the other hand, Task 4.4 advances the explainability and observability of these intelligent systems. As AI agents gain control over sensitive decisions, it becomes essential to ensure that their actions are transparent, auditable, and understandable. The task therefore investigates and integrates explainable AI (XAI) techniques, observability frameworks, and tracing methods that can support accountability, foster user trust, and enable effective incident response and system debugging.

Together, these two tasks provide a robust foundation for embedding secure, intelligent, and interpretable functions into the NATWORK platform. Deliverable D4.3 represents the first major milestone in this process, consolidating the architectural designs, state-of-the-art analysis, and initial implementation activities related to both AIaaSecS and CTI-driven explainable intelligence. The components described herein will ultimately be integrated, tested, and validated in real-world scenarios in the later stages of the project, with final results to be reported in D4.4.

By aligning advanced AI technologies with cyber threat intelligence and explainability, this deliverable contributes to the overarching goals of NATWORK: to build resilient, energy-efficient, and trustworthy 6G infrastructures where services can self-adapt to threats and operate securely without constant human oversight. In this sense, D4.3 not only reflects significant technical progress but also lays the conceptual groundwork for the secure automation paradigm NATWORK seeks to achieve.

1.1. Purpose and structure of the document

This deliverable reports on the progress achieved within Work Package 4 (WP4) of the NATWORK project, which focuses on the design and development of intelligent secure services that integrate artificial intelligence, cyber threat intelligence, and explainability into next generation 6G network architectures. In particular, Deliverable D4.3 presents the outcomes of the initial research, architectural design, and early implementation activities conducted under Tasks 4.3 and 4.4. These tasks aim to enable proactive, autonomous, and trustworthy security mechanisms capable of operating within complex service orchestration environments.

The main objective of this document is to outline how AI-based modules can be used to enhance network security through self-adaptive mechanisms, real-time threat awareness, and transparent decision-making. It also highlights the importance of explainability and accountability in the deployment of such intelligent services, as well as the role of cyber threat intelligence in supporting robust and informed decision processes.

The structure of the document is as follows:

- Section 2 provides a state-of-the-art analysis covering AI-driven network security, zero-touch orchestration, threat intelligence, and explainable AI techniques, setting the foundation for the technical developments described in later sections.

- Section 3 presents the Zero-Touch Network Solutions, including the design of autonomous orchestration and AI-based secure service deployment mechanisms that reduce human intervention while maintaining operational integrity.
- Section 4 describes the AI-Driven Real-Time Threat Detection capabilities being developed within the project, focusing on the integration of machine learning agents and proactive defence strategies to identify and mitigate cyber threats.
- Section 5 introduces the Blockchain-Based Trust Establishment mechanisms that support data integrity, provenance tracking, and distributed trust for secure service orchestration.
- Section 6 focuses on the Explainability Framework, detailing the different technical approaches developed to provide visibility, interpretability, and auditability of AI-driven decisions across the NATWORK platform.
- Section 7 presents the Cyber Threat Intelligence (CTI) Framework, outlining how multi-source intelligence is collected, processed, and used to feed both human users and AI-based modules within the system.
- Section 8 concludes the document by summarizing the main achievements to date and outlining the next steps toward full system integration and validation.
- Section A.1 is an Annex that provides a classification of the examined attacks in NATWORK, that facilitate the development of intrusion detection mechanisms, while A.2 presents updates on an attack tool that generates some of these attacks.

This deliverable thus provides a comprehensive overview of the intelligent and secure service capabilities being developed in NATWORK, while preparing the path for further implementation and evaluation activities in the second phase of the project.

1.2. Intended Audience

This deliverable is classified as public and is intended for a broad audience that includes not only the members of the NATWORK consortium but also external stakeholders, such as researchers, practitioners, policy makers, and other European research and innovation projects working in the fields of network security, artificial intelligence, and 6G technologies.

Deliverable D4.3 serves as a comprehensive reference for those interested in the design and development of intelligent secure services, AI-driven threat detection, explainable AI, cyber threat intelligence, and trust establishment mechanisms within the context of next-generation network infrastructures. The content is particularly relevant for academic and industrial communities engaged in the advancement of autonomous orchestration, cybersecurity automation, and AI trustworthiness.

By sharing the architectural designs, conceptual models, and initial implementation strategies developed under Work Package 4, this document contributes to ongoing discourse in the fields

of secure and intelligent networking. It aims to foster knowledge exchange, cross-project collaboration, and alignment with related initiatives funded under the Horizon Europe framework.

As a public deliverable, D4.3 also supports the transparency and openness objectives of the NATWORK project, offering insight into its technical vision and intermediate results while encouraging further collaboration and feedback from the broader research and innovation community.

1.3. Interrelations

The NATWORK consortium integrates a diverse and complementary set of competencies from academia, research institutions, SMEs, and large industrial partners, covering critical domains such as user-centric service design, AI-driven orchestration, cybersecurity, trust mechanisms, and secure-by-design network architectures. With fifteen partners across ten EU member states and associated countries (including the UK and Switzerland), NATWORK ensures broad geographical and technical representation in addressing the complex security and intelligence challenges posed by emerging 6G Smart Networks and Services.

As a Horizon Europe Research and Innovation Action (RIA), the project is organized into seven interdependent work packages (WPs), each structured into focused tasks to facilitate specialization, cross-WP collaboration, and continuous alignment with overall project objectives. This approach ensures that knowledge and technologies developed in each part of the project are shared and leveraged across the consortium, enabling scientific and engineering innovation at scale.

Deliverable D4.3 is a central output of Work Package 4 (WP4) – Intelligent Secure Services. It captures the progress in designing and implementing intelligent AI-based modules for proactive threat detection, explainability, and cyber threat intelligence integration. These components represent foundational elements for NATWORK's overarching goal of enabling autonomous, secure, and trustworthy service orchestration in future 6G infrastructures.

This deliverable is closely interrelated with the following project components and deliverables:

- D4.5 – NATWORK Federated Repository for B5G/6G Networks: D4.3 contributes to the data generation processes that feed into the federated repository defined in D4.5, particularly datasets related to AI-based threat detection, CTI, and explainability. These data assets will support the training, evaluation, and reproducibility of NATWORK's security models and frameworks. Furthermore, Annex A presents a brief output of T4.1, related to AI-based attack generation, that populates the data repository, and facilitates development activities under T4.3.

- D4.1 – Payload Security per Runtime, Intelligent Runtime Selection and Attestation: The intelligent components developed in D4.3 complement the runtime security mechanisms defined in D4.1 by enhancing their adaptability to threat contexts through CTI analysis and AI-driven decision-making. Moreover, explainability mechanisms from D4.3 provide critical transparency into how secure runtime selections and reactions are made.
- All deliverables across the project involving AI-based decision-making: As explainability is a transversal requirement for building trust in AI-driven automation, D4.3 is inherently connected to every deliverable in NATWORK that integrates AI modules, particularly those involving orchestration, monitoring, actuation, and adaptive security. The methods, models, and tracing mechanisms defined here serve as a reference and technical input for ensuring that AI decisions are interpretable, auditable, and aligned with ethical and regulatory expectations across the project.

In addition, D4.3 provides input to future integration and validation work in WP6 and aligns with WP2's requirements and user-centric use cases. It supports coherence and reuse of developed assets while promoting a unified vision for intelligent and explainable cybersecurity in NATWORK's system architecture.

2. State of the art

2.1. Zero-Touch Networking

Intrusion Detection Systems (IDSs) are vital for the identification and mitigation of unauthorized network activities. The introduction of Artificial Intelligence (AI) has strengthened the IDS effectiveness. However, AI models are usually quite complex, and this often leads to a lack of transparency, making it hard for security professionals to trust and understand their AI-based decisions. To face this problem, Explainable AI (XAI) techniques have been developed that provide insights into AI-IDS actions [1].

Current XAI Techniques in IDS are the following:

- Local Interpretable Model-agnostic Explanations (LIME), which provides explanations for individual predictions by locally approximating the AI model with an interpretable model. In the context of IDS, LIME assists security specialists in understanding the reasons behind specific alerts, enabling them to make more informed responses [2].
- SHapley Additive exPlanations (SHAP) assigns an essential value to each feature for a given prediction, providing a clear understanding of how each input impacts the output. This is especially useful in IDS for pinpointing the features that play the most significant role in detecting anomalies [3].
- Self-Organizing Maps (SOM) are neural networks that create a low-dimensional representation of the input data while preserving its topological properties. They are able to visualize complex data forms aiding this way in the interpretation of network behaviours and anomalies [4].
- Decision Trees and Random Forests are interpretable by constructions since they provide well defined decision paths. For IDS applications, they can be used to identify patterns related to unauthorized activities and at the same time provide clear explanations for their detections [5].

Challenges and opportunities of XAI in IDS:

- Balancing accuracy and interpretability is a constant challenge, as there is often a trade-off between the complexity of an AI model and how easily it can be understood. AI models of higher complexity usually led to higher accuracy but this complexity negatively affects transparency. It is definitely a challenge to design accurate AI methods without sacrificing interpretability.
- Real-Time explanations are essential for sufficient responses to adversary actions. Fast generation of these explanations without accuracy degradation is a challenging task.

- User-Centric explanations should be pursued for the XAI-IDS output to be actionable and comprehensible, whether the users are security analysts, network administrators or stakeholders.

Future Research Directions:

- Integrating XAI with Large Language Models (LLMs) will strengthen the IDS interpretability by providing natural language explanations allowing users to access complex detections more easily [6].
- Development of comprehensive XAI frameworks specifically designed for intrusion detection systems (IDS) that can help standardize the generation and presentation of explanations, ensuring uniformity in how insights are delivered across different models and scenarios. This not only promotes interpretability but also enhances the reliability and trustworthiness of the system by reducing variability and ambiguity in the explanations provided [7] .

Conclusively, the integration of XAI into IDS is progressing, providing more transparent and reliable AI-driven security solutions. Current research aims to improve the balance between model performance and interpretability i.e. making the models more user-centric, enabling security professionals to effectively understand and respond to AI-generated insights.

2.2. AI-Driven Real-Time Threat Detection

As network technology continues to evolve, in-network Machine Learning (ML) is expected to transform network operations by enabling real-time processing of data streams, including packets and flows, while eliminating the need for intermediate processing stages. Integrating ML directly into network infrastructure creates new opportunities for enhancing efficiency, security, and resource management.

The application of Artificial Intelligence (AI) in real-time threat detection is an emerging field that utilizes in-network ML techniques to strengthen network security. Numerous research studies explore advancements in in-network functions and their implementations across various programmable architectures, such as P4-programmable [8] devices and Data Processing Units (DPUs).

In-Network ML for Threat Detection Using P4: The shift toward self-driven next-generation networks [9] highlights the growing role of ML algorithms as key enablers in solving complex network management and optimization challenges. For instance, the work in [10] introduces SwitchTree, a system that embeds a configurable and reconfigurable Random Forest model inside a programmable switch. This design enables real-time flow analysis by extracting flow-level stateful features for network monitoring and attack detection. Experimental results confirm that

SwitchTree operates at line rate and achieves real-time attack detection with minimal resource overhead.

Similarly, the study in [11] presents a machine learning technique that utilizes Decision Trees (DTs) to predict heavy network flows directly within the switch. Given the constraints of limited memory and computing power, the method relies on a specialized packet processing pipeline that integrates pre-trained DT models for in-network flow prediction, which has been evaluated on BMv2 and Tofino ASIC platforms.

The author in [12] introduced IIsy, a framework that enables programmable switches to efficiently run machine learning classification models using an optimized encoding algorithm. By adopting a hybrid strategy, IIsy processes lightweight models on the switch while offloading complex computations to a backend server, achieving near-optimal classification accuracy and reducing backend load by 70%.

Another recent work [13] introduced Planter, an open-source framework designed to integrate various trained ML models into different programmable network devices. Evaluations show that in-network ML using Planter achieves high performance in anomaly detection, operates at line rate with minimal latency impact, and efficiently manages resource constraints with negligible accuracy loss.

Finally, an architecture leveraging programmable data plane switches to implement Binarized Neural Networks (BNNs) as switch functions has been proposed in [14], enabling line-rate packet classification at the edge. To ensure efficient training with minimal communication overhead, even in large-scale scenarios, the architecture adopts a federated learning approach. Their P4-based prototype evaluation demonstrates significant latency and bandwidth improvements over conventional ML-based network architectures.

In-network ML for DPU: Programmable DPUs and smart NICs are revolutionizing networking and computing by introducing advanced programmability for edge applications. The work in [15] introduces three DPU-driven edge use cases: a distributed network monitoring system for 5G, a power-efficient edge-to-cloud continuum, and security mechanisms integrated within DPUs.

The work in [16] highlights Processing-in-Memory (PIM) as a promising accelerator for ML training, demonstrating significant performance gains over CPUs and GPUs in memory-intensive tasks. Like DPUs, PIM technology shifts computation closer to data, reducing bottlenecks and improving scalability for next-generation ML accelerators.

The authors in [17] propose DLAU, a deep learning accelerator unit optimized for scalable deep learning networks using an FPGA-based architecture. The design integrates three pipelined processing units and tile-based techniques to enhance efficiency, achieving high-speed computation with minimal power consumption compared to traditional CPU implementations.

Multimodal AI based approaches: The work in [18] introduces 5G-NIDD, a comprehensive and fully annotated dataset comprising both DoS attack traffic and normal traffic captured from a real 5G testbed. This dataset is specifically designed to facilitate the development and evaluation of AI-based security mechanisms. Their study demonstrates the dataset's utility in intrusion detection through extensive testing with standard machine learning (ML) models, achieving promising levels of detection accuracy. [19] investigated the application of representation learning for malware traffic classification in Network Intrusion Detection Systems (NIDS). Using raw network traffic from two open-source datasets, they implemented a preprocessing pipeline that transforms PCAP files into images. This process includes session extraction, duplicate removal, and input normalization to ensure uniform image dimensions. Their approach, encapsulated in the USTC-TK2016 toolkit, employs convolutional neural networks (CNNs) to perform traffic classification. In [20], PayloadEmbeddings where proposed an innovative IDS approach based on generating vector embeddings from packet payloads. Inspired by Word2Vec, this method captures contextual relationships between bytes within a payload, enhancing the system's ability to detect payload-based attacks such as SQL injection and cross-site scripting, which are often overlooked by traditional IDS techniques. [21] presented a technique for classifying Tor traffic using time-based flow features between clients and entry nodes. Unlike conventional methods that rely on packet size or port numbers, their model focuses exclusively on temporal patterns. This approach enables the differentiation of eight categories of Tor traffic, contributing a novel perspective to encrypted traffic analysis.

2.3. End to End Trust Establishment

As 6G networks will connect a large number of devices, many of which may not be reliable, there are a number of significant security challenges to address. Traditional trust management systems are deemed inadequate for 6G applications due to their poor attack resiliency, relying on central authorities, and not functioning efficiently in the increasing number of users and devices of 6G networks [22][23]. This highlights the vital need for improved end-to-end security and trust management solutions in 6G networks, utilizing various technologies and approaches such as blockchain technology and zero trust approach [24]. The zero trust refers to eliminating any implicit trust in the various components and entities of the system, following the rule of "never trust, always verify". It requires proper authentication during the trust establishment, and continuous verification of the involved entities and services [24]. Additionally, the involved entities are granted access to parts of the system, considering the minimal access policy to reduce the attack surface. The authentication, authorization, and attack detection need to be developed considering the trade-off between the high security level and system efficiency. Considering the trust management schemes, the main components and approaches of the trust management between entities in 6G environment are as follows:

Zero trust approach: It relies on the concept that no entity whether inside or outside the network should be trusted by default. An active authentication of all participating nodes is required before allowing access to resources. This feature is an important component considering the openness and diversity of 6G networks.

Blockchain-Based Trust Management: The decentralized nature of blockchain technology and without relying on centralized authorities provides safe and transparent trust evaluation procedures through the employing of smart contracts and immutable ledgers. This decentralization is crucial for maintaining trustworthiness among the vast number of devices in 6G networks.

Access Control Management: Malicious or compromised nodes represent a serious risk in the network. This risk can be minimized by implementing access control policies which can be based on predefined or real-time assessments of devices' security metrics.

Trust Evaluation and Security Monitoring: Ensuring security in 6G networks requires constant evaluation of the trustworthiness of network elements. This involves monitoring in real-time a number of variables, including compliance levels, in order to determine trust scores and respond effectively when trust thresholds are reached.

Advanced Authentication Mechanisms: Strong authentication approaches are essential given the large number of devices in 6G networks. Approaches such as authentication and key agreement protocols guarantee mutual authentication between various nodes in the network. These mechanisms help prevent unauthorized network access and ensure that only legitimate devices participate in the network.

Son et al. [25] introduced a zero-trust authentication scheme for 6G-enabled IoT environments. The proposed scheme provides continuous verification to ensure that all participating nodes within the network are authenticated independently of a secure channel. The scheme utilized the blockchain technology, which facilitates mutual authentication among network elements and provides the integrity and reliability of identity verification processes in a decentralized context. Additionally, a dynamic and fine-grained access control is achieved through the utilization of attribute-based encryption (ABE) [26]. The authors utilize a lightweight ABE method based on the elliptic curve cryptosystem (ECC), which minimizes computational overhead while improving the security. Through effective key management and access verification processes, the proposed approach guarantees effective defence against a range of security threats and provides adaptable access permissions based on real-time UE status. The authors provided a comprehensive security analysis using BAN logic and AVISPA [27] to validate the proposed approach, proving its resistance to potential attack vectors. A lightweight authentication scheme was proposed by the Rana et al. [28], designed for next-generation IoT infrastructures, specifically 6G networks. Mitigating

vulnerabilities such as user impersonation attacks was among the primary achievements of this work. The proposed scheme utilizes symmetric cryptographic algorithms to ensure mutual authentication between edge nodes and servers. This scheme ensures that only legitimate users can access services provided by the servers while protecting sensitive information transmitted over public channels.

An authentication scheme, named REHAS, presented in [29], was specifically designed for the Internet of Drones (IoD). The scheme employed Hyperelliptic Curve Cryptography (HECC) [30] and utilizes an 80-bit key size for strong security. It performs fuzzy extractors for biometric data processing, enhancing user authentication and safeguarding against unauthorized access in case of device theft or loss. The scheme also utilizes a hash function, which balances the security and computational efficiency. By generating unique session keys for each communication session through a base station, the scheme mitigates the risk of replay attacks. Considering the resource constraints of drones, the scheme adds low computational overhead (approximately 6.7171 ms.), communication overhead (1696 bits), and energy consumption (22.5 mJ.). Choi et al. [31] provided a trust management scheme, considering drone security through the use of physical unclonable functions (PUFs). The integration of the proposed scheme improves the system's resilience against specific attacks such as impersonation and stolen verifier threats. While schemes such as REHAS employ cryptography to ensure efficient trust management approach, the PUF-based scheme introduces a lightweight approach that addresses the vulnerabilities that such schemes may not fully cover, particularly concerning the physical security of drones and challenges posed by compromised cryptographic materials. By integrating PUF technology, the proposed scheme provides a more adaptable and responsive security solution that is better suited to the unique constraints and requirements of drone operations. However, increased complexity in the authentication phase which involves a higher number of message exchanges potentially lead to greater communication overhead and latency, which may affect the responsiveness of drone operations in critical scenarios.

The authors in [32] introduced a hierarchical architecture that integrates Multi-Access Edge Computing (MEC) and Device-to-Device (D2D) communications to enhance healthcare services in the 6G environment. This architecture comprises three layers: Sensing, Processing, and Storage, where Internet of Medical Things (IoMT) devices collect health data, Cluster Controller (CC) nodes process and relay the data to MEC servers, and MEC ensures secure storage. A key focus is on security and privacy, achieved through the development of a lightweight mutual authentication protocol named LiMAD, which employs strong encryption to defend against threats like replay and man-in-the-middle attacks. However, the reliance on Cluster Controller (CC) nodes to facilitate communication between IoMT devices and MEC servers introduces potential inefficiencies, particularly if the CCs become bottlenecks under high loads. Putra et al. [33] proposed a blockchain-based trust management framework that leverages the decentralized

nature of blockchain technology, which eliminates the reliance on a central trusted party and thus enhances overall security against malicious activities. Key cryptographic components of this framework include smart contracts [34], which automate the processes of trust evidence collection and score calculation, ensuring transparency and adherence to predefined rules. An immutable ledger stores all trust-related data securely, maintaining data integrity and enabling auditability. Furthermore, by employing pseudonymity through cryptographic identifiers, the system enhances user privacy, allowing participants to engage without revealing personal identities. The proposed framework continuously assesses and quantifies the trustworthiness of all network participants. However, the main drawback is the inherent challenges in ensuring privacy and security within a large-scale, decentralized system, particularly concerning the risk of de-anonymization attacks against users and potential vulnerabilities in complex smart contracts.

A decentralized framework introduced in [35] for secure end-to-end (E2E) communications in Large-Scale Heterogeneous Networks. This proposed approach addressed critical vulnerabilities found in traditional E2E security systems that often rely on centralized nodes, identity privacy breaches, and extensive communication costs. Key components of this scheme include a blockchain-enabled UE registration and key management protocol. Additionally, the framework incorporates a privacy-preserving mutual authentication protocol leveraging bilinear pairing which allows users and serving networks to authenticate each other securely while safeguarding their identities. Moreover, it employs a Trusted Execution Environment (TEE) for efficient session key generation and distribution, ensuring secure communication channels between data senders and receivers. The implementation of blockchain in similar protocols should be designed carefully with minimizing the necessity of frequent access to the blockchain, as it can lead to increased computational and communication costs [23]. Additionally, the complexity of integrating a TEE within various devices raises concerns about compatibility and accessibility, potentially limiting the widespread adoption of such frameworks across different network infrastructures.

2.4. Explainable AI

Ever since the emergence of Artificial Intelligence (AI), specifically neural networks, researchers have been curious about the reasoning behind the decisions made by the complex ML models. This curiosity has motivated various studies for providing human interpretations on the output of AI models, reaching back to 1980s with a focus on rule-based expert systems [36], [38]. At the time, numerous works were conducted to develop rule extraction techniques as a form of producing explanations for artificial neural network-based systems [39], [40], [41], along with survey papers to design taxonomies for such techniques [42], [43].

Over time, with the resurgence of deep learning and advancement in computation power, the focus of interpretability studies shifted into these highly accurate “black box” models, which are basically deep neural networks [44], [45]. Numerous research papers were conducted for human interpretations of such black box models, trying to explain the reasoning behind their predictions using various methods, such as additive structures [46], sensitivity analysis [47], randomization techniques [48] and so on.

While the studies on the interpretation of complex AI models continued, the focus of the relevant research shifted more into the concept of “Explainable AI (XAI)” after DARPA formulated their 4-year XAI program in 2015 [49]. Many researchers, tackling the trade-off between interpretability and accuracy, produced output on the explainability of different ML models, mostly working on inherently explainable models, such as generalized additive models [50], logistic regression [51] or linear integer models [52], which are **model-specific** explainability techniques. On the other hand, as a different approach, a novel and flexible **model-agnostic** technique called LIME (Local Interpretable Model-agnostic Explanations) [53] was proposed in 2016, which provided a formal way to explain specific predictions in addition to the global understanding of the model. Moreover, the enforcement of EU General Data Protection Regulation (GDPR) in 2018 has granted people the “right to explanation” for autonomous systems [54]. This legal binding has also been a propelling factor towards further research on XAI.

Following the impact of LIME, other model-agnostic explainability methods were developed throughout the years, some of them being widely adopted such as SHAP (Shapley Additive Explanations) [3] and counterfactual explanations [55]. In addition, some model-specific XAI methods were also proposed, namely saliency maps for neural networks [56], Grad-CAM for convolutional neural networks [57], and integrated gradients for deep networks [58]. These techniques are commonly utilized as the interpretability of many AI-based model in today’s problems.

Although previously mentioned XAI methods cover plenty of scenarios and support various ML algorithms, some techniques have also been developed to interpret specific models. For instance, **reinforcement learning** is a unique paradigm of machine learning which is significantly different than the supervised or unsupervised alternatives. In this approach, an agent is trained through the interaction with the environment and subsequent observations of events, enabling an autonomous learning process [59]. Although some model-agnostic methods such as LIME or SHAP can also be applied, special consideration is required for an enhanced **Explainable Reinforcement Learning (XRL)** concept, which is studied thoroughly in survey papers [60], [61].

To actualize XRL, PIRL (Programmatically Interpretable Reinforcement Learning) has emerged as an alternative to deep reinforcement learning paradigm and has become a widely used framework [62]. PIRL replaces the neural network-based policies in deep learning with high-level,

domain-specific programming language, and thus, providing an intrinsic XAI capability. Another intrinsic XRL method provides interpretability via fuzzy policies [63]. In addition to inherently explainable RL techniques, some *post-hoc* methods have also been proposed for the XRL concept, such as genetic programming [64], reward decomposition [65], expected consequences [66], policy distillation [67], and so on. All of these studies provide invaluable contributions to help reinforcement learning process to be humanly interpretable.

Methods explored so far have touched XRL for **global** explanations, since the interpretability of reinforcement learning mostly focuses on that aspect. Nevertheless, some work also has been conducted for the **local** interpretability of such models, to gain an insight into how an agent would perform under certain conditions. For instance, aiming to tackle a complex, multi-task reinforcement learning problem, a novel framework based on hierarchical policies was proposed [68]. In addition, other techniques such as interesting elements [69], autonomous self-explanation [70], and structural causal model [71] were also presented as a form of explaining specific actions performed by the agents. Furthermore, the recent proliferation of large language models (LLMs) empowers the reinforcement learning process in various ways [72]. Some studies propose to leverage LLM-based policy interpreters as a form of achieving XLR [73], [77]. Although the use of LLMs for XLR is quite limited, it seems like a promising future research direction.

Apart from reinforcement learning, another special group of ML models that could benefit from XAI techniques is **Graph Neural Networks (GNNs)**, which are a special type of neural network [78]. GNNs are similar to convolutional neural networks, however; unlike the latter models which work on images, GNNs operate on graph-structured data. This uniqueness results in a need for special XAI techniques for the interpretability of such models. One of the first developed methods for GNN explainability is called **GNNExplainer** [79], which is still widely adopted nowadays. GNNExplainer is a *post-hoc*, *model-agnostic* approach that works on any GNN-based model with any type of task, supporting both node classification and edge prediction. Inspired by GNNExplainer, many studies developed their own techniques for explainable GNN models. One such instance is **CFGExplainer** [80], which is a deep learning-based *model-agnostic* explainer, aiming to interpret control flow graph-based malware classification. Another work namely **RCExplainer** [81] focuses primarily on GNNs in the class of piecewise linear neural networks and provides robust counterfactual explanations. With an emphasis on cybersecurity, **ILLUMINATI** [83] is a post-hoc XAI method for GNNs, providing human-interpretable explanations by jointly considering nodes, edges, and attributes. Botnet detection is also a popular topic for the usage of GNN, motivating researchers to come up with XAI techniques for GNNs in this specific area. Both **XG-BoT** [87] and **BD-GNNExplainer** [88] are methods that tackle the issue of over-smoothing and high number of abnormal edges, respectively, in the botnet detection domain.

2.5. Cyber Threat Intelligence

The work presented in [89] introduces VinciDecoder, an automated approach that leverages provenance analysis, machine translation, and machine learning techniques to generate high-quality natural language reports from provenance graphs. VinciDecoder is not a CTI tool per se, but an attack forensic tool. VinciDecoder is designed to address the challenge of identifying the root cause of security incidents in large-scale cloud infrastructures, which is crucial for enhancing security awareness and strengthening threat detection and prevention capabilities. The system comprises two main phases: during the training phase, suspicious paths and their corresponding reports are collected, and these paths are transformed into primitive sentences in an intermediate language using a model trained with Neural Machine Translation (NMT). In the report generation phase, the trained model and the PILT (Translation to Intermediary Language) algorithm are used to create forensic reports based on suspicious paths associated with detected incidents. This approach enables analysts to quickly understand the sequence of operations during a security incident, significantly reducing the time and effort required to compile these reports.

The work in [90] provides a review of studies on the automatic extraction of cyber threat intelligence (CTI) from textual descriptions, highlighting its importance for proactive defence against cyber threats. CTI is defined as evidence-based knowledge that enables organizations to predict, prevent, or defend against cyberattacks, categorized into strategic, operational, tactical, and technical CTI. The authors emphasize the benefits of CTI, such as proactive and actionable defence, as well as the challenges, including the need for clean data and automated extraction methods to improve precision and relevance. Additionally, they address the importance of collaborations and automation in CTI exchange, building on a review of 34 previous studies and expanding their analysis to a larger number of publications, identifying three new purposes for CTI extraction and proposing a CTI extraction pipeline. The methodology involves searching six academic databases and selecting 20,922 relevant publications.

STIXnet, a modular and scalable solution designed to extract entities and relationships from unstructured cyber threat intelligence reports, is introduced in [91]. It uses natural language processing techniques and an interactive knowledge base to achieve high F1 scores in entity and relationship extraction, standing out as the first system to extract all STIX entities, including 18 entity types and over 100 relationships. The system consists of several modules, each specialized in different types of extraction, such as individual entities, novel entities, and Tactics, Techniques, and Procedures (TTPs), and employs a combination of rule-based and deep learning approaches to extract relationships between entities, producing a JSON file that can be processed by a graphical interface. Additionally, STIXnet includes a framework for submodule interaction that

avoids overlap during processing and merges results from different submodules into a single data structure, facilitating its integration into various information extraction scenarios.

The authors in [92] introduce CTI-BERT, a BERT model trained from scratch using a high-quality cybersecurity corpus, which outperforms other general and security-specific domain models in sentence and token-level classification tasks. They emphasize the importance of training domain-specific models with high-quality corpora to enhance precision in threat intelligence extraction. The authors compare their approach with models like CyBERT and SecureBERT and evaluate CTI-BERT in sentence and token classification tasks, demonstrating superior performance. Additionally, they tested CTI-BERT and SecRoBERTa's ability to classify malware-related sentences, achieving excellent results.

The study in [93] focuses on enhancing security professionals' ability to prioritize and defend against cyber-attacks by identifying temporal attack patterns from open-source CTI reports. The authors introduce ChronoCTI, a machine learning pipeline that employs a large language model to extract temporal relationships from CTI reports. Their evaluation on a large corpus of CTI reports revealed 124 temporal patterns across nine categories, with the most common involving tricking users into executing malicious code and evading malware protection systems. The study underscores the importance of educating users, implementing immutable operating systems, and requiring multi-user authentication to mitigate recurring attack patterns.

AttacKG [94] is an innovative technique for automatically extracting structured attack behaviour graphs from cyber threat intelligence reports. This system identifies attack techniques and their dependencies, incorporating cyber threat intelligence to create technique knowledge graphs. Evaluated on 1,515 real reports, AttacKG demonstrates superior precision in identifying attack techniques and IoCs, outperforming current approaches. By automating the analysis of intelligence reports, AttacKG enhances the detection of advanced cyberattacks and provides knowledge graphs that assist in attack reconstruction and APT detection. Additionally, it compares favourably with other methods like TTPDrill, ChainSmith, and EXTRACTOR, highlighting its performance and effectiveness in extracting cyber threat intelligence.

In [95] MALOnt is introduced as an open-source malware ontology that organizes extracted information into knowledge graphs centred around threat intelligence. This ontology provides a comprehensive dictionary encompassing attacks and correlated details, aiding security analysts in comprehending attack origins, goals, timelines, actors, and exploited vulnerabilities. MALOnt's structural design includes classes and properties that characterize various malware-related aspects, including behavior and affected targets, promoting efficient information extraction and the establishment of new connections. Through its flexible framework, MALOnt captures and analyses malware threat intelligence while offering pathways for extracting significant insights from threat reports.

The work in [96] presents GoodFATR, an innovative methodology developed for the comparative analysis of indicator extraction tools within cyber threat intelligence. Utilizing a majority voting scheme, GoodFATR offers a more thorough and reliable performance evaluation of these tools without necessitating a manually curated ground truth dataset. The effectiveness of GoodFATR was established through a series of rigorous experiments, underscoring its advancements over traditional methodologies. Moreover, the work introduces a new platform dedicated to systematically collecting and extracting indicators from threat reports, ensuring traceability throughout its operational pipeline and enabling analysts to accurately identify the underlying document sources for each extracted indicator.

[97] explores commercial threat intelligence (TI) services and contrasts their advantages and limitations vis-à-vis open-source alternatives. While commercial services tend to provide superior quality, contextualization, and expansive threat coverage, they can often be costly and not universally accessible. The study advocates for a hybrid approach that harnesses the strengths of both models, revealing an overlap of indicator feeds between competing providers. The qualitative assessment of TI is also noted as primarily based on informal heuristics rather than strict metrics, indicating a gap in academic scrutiny of commercial TI compared to the existing focus on open-source intelligence.

The study in [98] "Vulnerability Disclosure in the Age of Social Media" explores the role of Twitter to forecast and recognize real-world vulnerability exploitations. Through the analysis of Twitter data patterns, the authors employ natural language processing and machine learning algorithms to classify tweets in relation to vulnerability exploits. This work establishes methodologies for early exploit detection via Twitter, identifying features that mark useful indicators of ongoing exploits, while also assessing the robustness of their detection system against adversarial manipulation. The contributions encompass a characterization of the vulnerability disclosure landscape, an introduction of techniques for early exploit detection, and the formulation of a problem-specific threat model against competitive interference.

[99] introduces LogPrécis, a tool that employs language models (LMs) to scrutinize Unix shell attack logs, . By utilizing advanced LMs, LogPrécis effectively identifies attacker tactics associated with various components of shell sessions while condensing extensive logs into concise footprints conducive to identifying novel and similar attacks. The work validates that LogPrécis can enhance the defence response to cyber threats by employing pre-trained language models to evaluate Unix shell logs, classifying detection efforts based on attacker techniques and facilitating MITRE tactic identification. This comprehensive approach aims to enhance understanding of attackers' motives, improve threat identification capabilities, and streamline cybersecurity operations.

[100] proposes CyberRel, a joint entity and relation extraction model tailored for cybersecurity concepts, positioning the extraction challenge as a multiple sequence labelling task. By leveraging

techniques such as BERT, BiGRU, and attention mechanisms, CyberRel attains an impressive F1 score of 80.98% on Open-Source Intelligence (OSINT) data. The primary focus is on enhancing accuracy and efficiency in both entity and relation extraction within the cybersecurity domain while addressing overlaps between entities in the corpus. The paper discusses the construction of triples representative of cybersecurity knowledge and the model's ability to produce well-structured outputs through advanced deep learning frameworks.

The proposed method in [101], rcATT, aims to streamline the retrieval of ATT&CK tactics and techniques from cyber threat reports to bolster efficient threat hunting and risk assessment. This solution automates the extraction of Tactics, Techniques, and Procedures (TTPs) sourced from various references, organizing results within a STIX 2.0 structured format. The authors illustrate a machine learning model tailored for this task, encompassing processes such as data preprocessing, model selection, and performance evaluations, while accounting for challenges in text classification, including data rebalancing and augmentation strategies.

[110] proposes a comprehensive system for creating cybersecurity knowledge graphs (CKG) sourced from after-action reports (AAR) to enhance cyber threat intelligence. Employing named entity recognition (NER) and relation extraction methodologies, the system effectively identifies entities and relationships, representing this information within a CKG via an ontology known as Unified Cybersecurity Ontology (UCO). The authors detail the functionality of their Malware Entity Extractor (MEE), Relation Extractor (RelExt), and CKG module, aimed at improving cybersecurity analyses by automatically identifying relevant relations within AARs, subsequently facilitating the visualization of intricate malware details and enhancing overall security operations.

[111] introduces ATLAS, a novel sequential learning model tailored for investigating attacks within complex and interconnected systems, leveraging a combination of sequence mining techniques and machine learning to classify attack patterns in network traffic. Evaluation with real attack datasets displays ATLAS's proficiency in identifying attack entities, achieving a significant average precision of 91.06% and a recall of 97.29%. This flexible approach effectively addresses challenges related to contemporary network security assessments and enables a more streamlined understanding of attacks through causal graph generation, thereby enhancing operational investigation capabilities.

[112] presents CyNER, a Python library focused on named entity recognition (NER) specific to cybersecurity, adept at extracting entities and indicators of compromise from unstructured data. By integrating transformer models, heuristics, and publicly available NER solutions, CyNER presents a versatile threat intelligence discovery tool, tailored for efficient processing and analysis of cybersecurity information. The library employs high accuracy techniques using existing

malware ontologies, benchmark datasets, and feature combinations, facilitating insightful information extraction aimed at bolstering cybersecurity operations.

[113] presents an improved TTP (Tactics, Techniques, and Procedures) intelligence mining framework, termed TIM, enhanced by contextual threat factors to systematically extract and classify TTPs from unstructured threat datasets. By leveraging natural language processing in conjunction with threat context information, TIM meticulously organizes elements as STIX 2.1 formatted descriptions for effective sharing. The framework showcases the TCENet (Threat Context Enhanced Network) model, evaluated on annotated datasets, emphasizing superior classification performance in TTP analysis, aiming to arm defenders with robust long-term threat detection capabilities and realistic threat simulations to enhance security postures.

[114] compares various deep learning-based Named Entity Recognition (NER) algorithms using a cybersecurity dataset compiled from diverse sources such as the Microsoft Security Bulletin. The authors evaluate contemporary deep NER algorithms, including both established and novel methodologies, to identify the most effective model for recognizing entities within a cybersecurity corpus. Furthermore, the significance of embedding strategies in enhancing NER performance is discussed, providing a valuable resource for future researchers focusing on developing new cybersecurity information extraction algorithms.

[115] introduces a groundbreaking method for automatically extracting named entities from CTI reports using a deep learning approach. By defining security-related keywords, including malware and vulnerabilities, the authors leverage a Conditional Random Field (CRF) integrated with a bidirectional Long Short-Term Memory (Bi-LSTM) network to achieve exemplary performance, attaining an average F1 score of 75.05%. Moreover, a labeled dataset containing 498,000 entities is released to foster future research in the security domain, enhancing analysts' efficiency in scrutinizing CTI reports.

3. Zero-Touch Networking

With the advent of management and orchestration of virtualized NFs in NFV environments, the objective of automating such tasks to obtain a self-managed self-healing telco system is a natural continuation and evolution of such environments. ETSI targets the standardization of such management automation with the Zero-Touch Network & Service Management (ZSM) industry specification group (ISG) [102]. This ISG adds standards complementing the NFV and MEC standards, with a focus on the definition of a new, future-proof, horizontal, and vertical end-to-end operable framework and solutions to enable agile, efficient, and qualitative management and automation of emerging networks and services.

ETSI ZSM defines different management domains in various parts of a telco network, namely in the radio access edge, the transport, and the core domains. ETSI uses a closed-loop mechanism for each of these domains, then follows with a higher level End-to-End (E2E) management of all these domains in a bigger closed-loop management framework. Closed-loop mechanisms are described in various models such as the Orient-Observe-Decide-Act (OODA) and Monitor-Analyze-Plan-Execute-Know (MAPE-K) models [103]. Despite differences in step definitions, these models follow a similar high-level workflow: Monitoring, Analysing, Deciding, and Acting.

In alignment with the ETSI ZSM vision of achieving closed-loop, autonomous management across various telco domains, there is a growing need for intelligent, high-performance components capable of executing decisions at the data plane with minimal latency. NATWORK aims to provide solutions towards this direction through: 1) MERLINS which provides a ZSM-compliant methodology for selecting and executing MTD actions based on real-time network assessments, thereby closing the loop from observation to automated remediation and 2) Wirespeed traffic analysis in the 5G transport network, where threat detection, anomaly classification, and adaptive response must occur without human intervention.

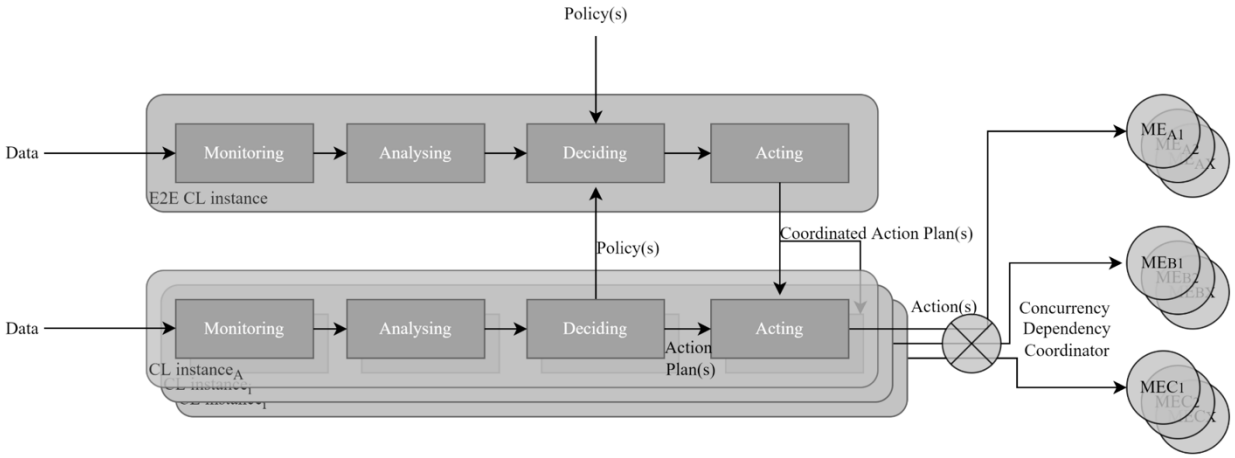


Figure 1: ETSI exemplary Closed Loop Coordination timeline [109].

3.1. AI-based MTD optimization

Following the ETSI ZSM approach, NATWORK provides a closed-loop methodology used to manage Moving Target Defense (MTD) enforcement and optimization. As MTD provides a set of operations that are applied to VNFs and CNFs at different levels of a telco network, such as NF live migration, NF reinstantiation, and network reconfigurations (*e.g.,* IP shuffling, port shuffling and dynamic vNIC), an automated system is required to select which MTD operation to perform, on what, when, where, and why, requiring a complex decision-making system that determines the MTD action based on what is observed in the network. This decision-making system is realized by designing and following a ZSM-compliant closed-loop security management methodology for MTD operations on NFV resources and over multiple edge domains, bridging ETSI NFV, ETSI MEC, and ETSI ZSM standards. This methodology is named MERLINS, and is composed of four chronological cyclic phases:

A. **Integration to the 5G/B5G network**: this phase consists of having a passive and active interaction with the network. The passive interaction is the consistent and real-time observation and monitoring of the network. In contrast, the active interaction consists of the ability to operate on the networks' components, *i.e.*, the VNFs/CNFs, NSs, NSIs, and VIMs of the different domains in the edge-to-cloud continuum, spanning from the multiple edge clusters or nodes to the core network.
B. **Network assessment and decision making**: using the data obtained from the passive interactions and monitoring in the previous phase, this phase focuses on analysing data such as performance metrics, resource consumption analysis, and security evaluations to assess risks or detect attacks. This analysis then results in a decision on whether to enforce an MTD operation or not. This is where the modelling of the network in near real-time to evaluate and assess its state is performed. AI/ML models then use such observations to evaluate and train its MTD strategies.
C. **MTD management/orchestration**: in the advent of the decision to perform an MTD operation, this phase goes through the validation process, analysing whether the operation can be performed, with respect to technical feasibility, *i.e.*, if the operation can be implemented on the specified target, and policy-based feasibility, *i.e.*, if there is no other orchestrator with a conflicting policy and a higher hierarchical priority.
D. **MTD enforcement**: at the validation of an MTD operation, this phase enforces and implements the MTD operation on the 5G network, also using the active interactions available in phase A, transitioning to this phase for the next iteration of the closed-loop methodology.

3.2. P4-based Network Analytics

The evolving landscape of 5G and beyond necessitates a shift toward autonomous, adaptive network architectures. Zero-Touch Networking (ZTN) emerges as a critical paradigm in this context, emphasizing self-configuring, self-optimizing, and self-healing network functions with minimal human intervention. In NATWORK, the Wirespeed traffic analysis in the 5G transport network aligns seamlessly with the vision of ZTN, enabling fully automated, intelligent management of the 5G data plane. Figure 2 presents the architecture of this approach and how the different components interact with each other. This solution can be applied either between the gNB and 5G CORE Network (O1) or between the UPF function and the DN (O2).

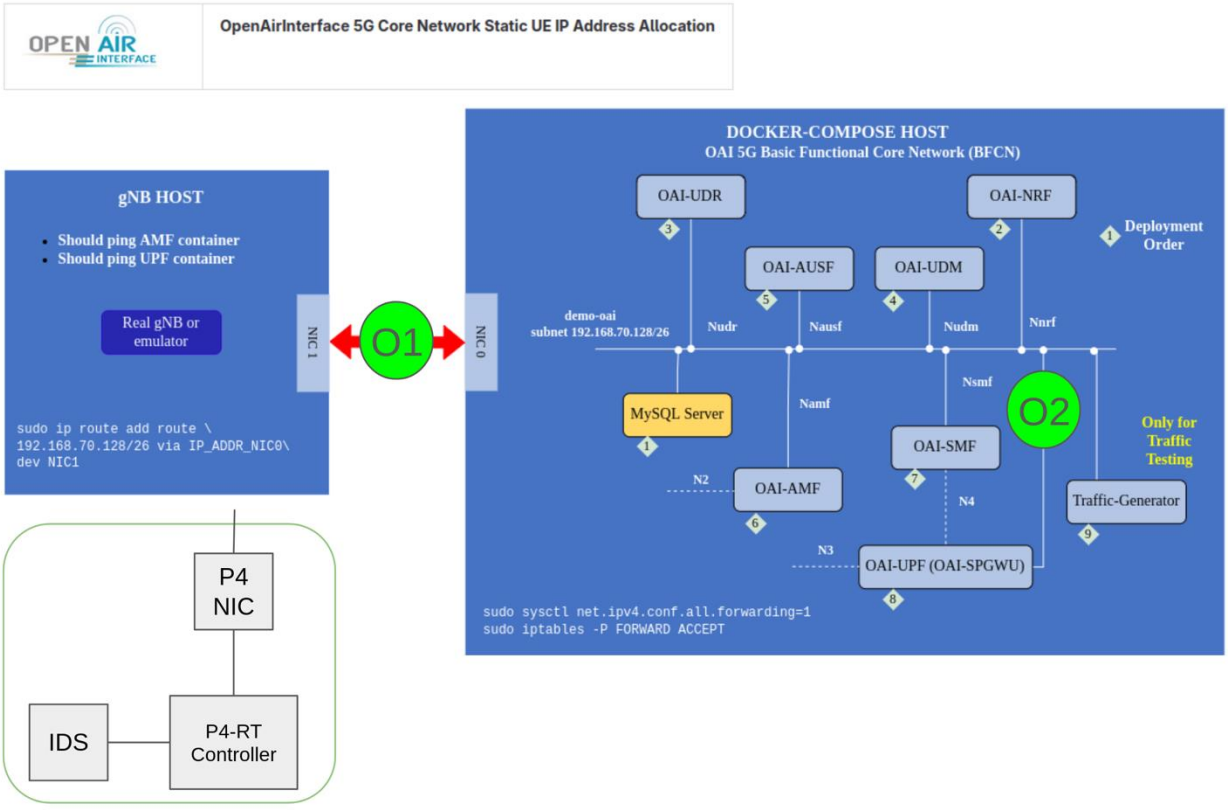


Figure 2: Wirespeed Traffic analysis Architecture

Our proposed approach introduces a programmable, intelligent pipeline for enhancing the security and visibility of 5G transport networks. By leveraging P4-enabled SmartNICs (Netronome Agilio CX25Gbps, as shown in Figure 3), we enable real-time parsing and data preparation for feature extraction directly at the network interface. Parsed data is forwarded to an AI-augmented Intrusion Detection System (IDS), which classifies the traffic and identifies anomalies or threats in real time. The IDS utilizes well-established Large Language Models (LLMs) and has been trained to detect potential attacks based on packet traces. The insights produced by the IDS are then relayed to a centralized Software-Defined Networking (SDN) controller, which applies adaptive

control policies over the P4 based network. Finally, the SmartNIC enforces these policies via a P4 match-action pipeline, ensuring low-latency, in-network mitigation of suspicious flows.

Figure 3: P4 SmartNIC - Netronome Agilio CX25

Figure 4 presents the internal architecture of O1 placement of the Wirespeed traffic analysis in 5G transport network, and its main functionalities are detailed below:

- Parse the packets in P4 SmartNIC: Leveraging a P4-programmable SmartNIC, 5G transport network packets are being parsed enabling the analysis of incoming network traffic and extraction of the relevant header fields, payloads, or metadata for further processing. Fine grained control of the data is possible, allowing at wirespeed to extract the data at different levels (e.g. specific host communicating over the telecom network, specific connection from a host, or the entire transport interface between the RAN and the Core Network).
- Send the packets to IDS for classification: Once parsed, the SmartNIC forwards either full packets or selectively extracted features (e.g., headers, flow keys, or metadata) to an Intrusion Detection System (IDS). The IDS conducts real-time deep packet inspection and behavioural analysis to classify traffic, identifying potential security threats such as anomalies, malicious payloads, or patterns indicative of attacks within the 5G network context.
- Send inference to SDN Controller: Upon completing its analysis, the IDS generates actionable insights, inferences or decisions which are transmitted to the SDN controller. The controller which orchestrates network behavior based on centralized control logic, interprets these inferences to dynamically update forwarding behavior, access controls, or mitigation strategies across programmable network elements. These decisions are enforced directly on the card, allowing the control over specific flows over the network (e.g. dropping a single connection).

- Apply match-action pipelines: Based on the inferences from the IDS and instructions from the SDN Controller, the P4 pipeline on the SmartNIC executes context-aware match-action rules. These pipelines consist of rules (match conditions) that determine how packets should be handled, such as forwarding, dropping, modifying, or mirroring them. This enables enforcement of fine-grained, stateful security and Quality of Service (QoS) policies at the data plane with minimal latency.

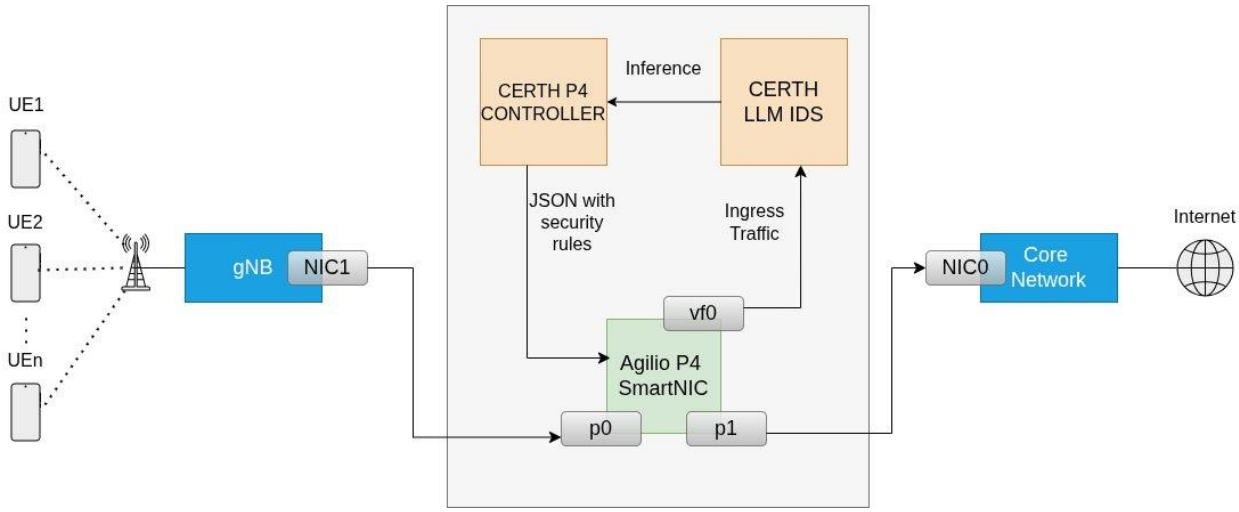


Figure 4: System architecture of O1 placement of Wirespeed traffic analysis in 5G

The communication between the IDS and the P4-RunTime has been defined, and a JSON file is created by the IDS after the completion of the inference. Figure 5 presents the template of the JSON file which is sent to the P4-RunTime in order to apply the match-actions to the smartNIC and consequently the 5G network.

```
{
    "tables": {
        "ingress::security_mac": {
            "rules": [
                {
                    "name": "drop_mac_0xbc24116bca8f",
                    "match": {
                        "ethernet.srcAddr": {
                            "value": "bc:24:11:6b:ca:8f"
                        }
                    },
                    "action": {
                        "type": "ingress::drop",
                        "data":{}
                    }
                }
            ]
        }
    }
}
```

Figure 5: JSON template for applying match-actions to P4 smartNIC

4. AI-Driven Real-Time Threat Detection

4.1. AI-based behavioural analysis

The increased complexity of modern computer networks has introduced significant challenges in ensuring performance, reliability, and security. These challenges are further amplified by the rapid growth of cloud computing, virtualization, and multi-tenant architectures, where diverse applications and users share infrastructure across multiple domains. In such environments, network monitoring plays a vital role in detecting anomalies, ensuring service quality, and defending against Cyber threats.

However, traditional monitoring techniques are no longer adequate to meet the growing demands of advanced infrastructures, particularly with the emergence of AI-based behavioural analysis. This cutting-edge approach leverages machine learning and artificial intelligence to detect patterns, predict anomalies, and enhance network security in real-time.

The effectiveness of AI models depends heavily on the availability of high-quality, fine-grained telemetry data from various points in the network. Emerging techniques, such as Postcard Telemetry and In-band Network Telemetry (INT), enable more detailed and real-time traffic analysis by leveraging programmable data planes, including those written in P4. These technologies allow the network to embed monitoring data within packets or generate trace messages at each hop. However, these methods often lack the flexibility required to adapt to the heterogeneous and rapidly evolving nature of modern networks.

Decentralized Feature Extraction (DFE) Telemetry, enabled by P4-based data plane programmability, has been proposed as a novel solution that provides a flexible mechanism for supplying AI models with only the required packet information. This is achieved by selectively extracting specific features from packets associated with a particular flow, thereby enabling real-time data processing, reducing bandwidth consumption, and preserving data privacy.

4.1.1. Decentralized Feature Extraction Telemetry (DFET):

The suggested DFE Telemetry module utilizes an offloaded data plane program, that can be deployed across multiple P4 switches of the monitored network, to configure and manage telemetry flows. Telemetry information can be dynamically tuned to adhere to specific monitoring purposes, enabling precise control over network visibility. This functionality of DFET is highly useful for AI-driven security techniques where the framework can recognize patterns and correlations in new attacks.

Behavioural Model (BMv2) software switch has been used to deploy the P4 program of DFET module. The model follows several key stages:

- The parser: Represented as a finite state machine, where each state extracts data from a specific header structure and stores it into runtime variables. Transitions between states are conditional, depending on the values of the parsed header fields.
- Ingress and Egress control blocks: These blocks include multiple match-action tables that inspect various header fields to trigger corresponding actions. A control function determines the sequence in which the tables are executed. The ingress block mainly handles packet forwarding, while the egress pipeline performs additional processing after the egress port has been selected.
- Traffic Manager: It is responsible for queueing and scheduling packets between the Ingress and Egress pipelines. It guarantees efficient packet flow by managing buffering and preventing congestion.
- The deparser: This final stage reconstructs the packet by serializing the modified headers, making the packet ready for forwarding to the next switch.

The processing procedure begins when a packet arrives at the P4 switch, where it first enters the parsing stage. As illustrated in Figure 6, the parser extracts the Ethernet header, followed by the IPv4 header. Then based on the value of the protocol field in the IPv4 header, the parser determines whether to extract the UDP or TCP header.

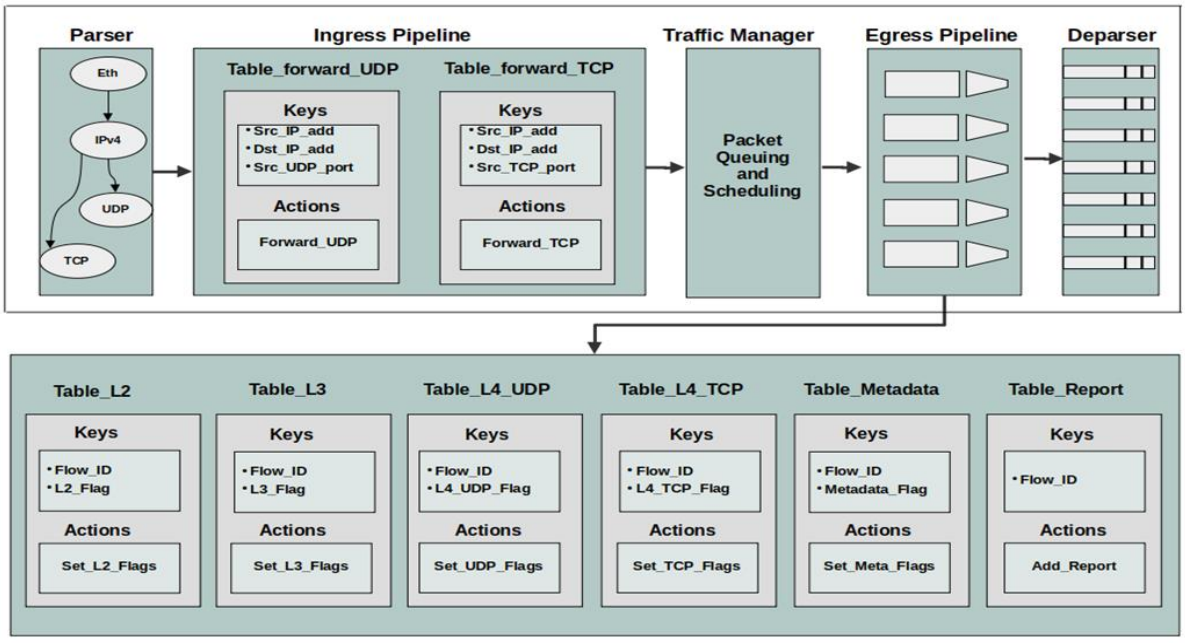


Figure 6: DFET pipeline

Once parsing is complete, the packet moves into the ingress pipeline, where it is directed to one of two tables (Table_forward_UDP / Table_forward_TCP) according to the transport protocol (UDP or TCP). These tables match on the packet header fields – specifically, IP addresses and source UDP/TCP port – and invoke the corresponding action (Forward_UDP / Forward_TCP) that

determines the output port and mirrors the packet to enable report generation in later stages. Flows are identified based on source IP address, destination IP address, and source UDP/TCP port.

It is essential to highlight that the DFET module enables the control plane to perform layer selection by specifying the header layers from which features should be retrieved. This selection is achieved by passing binary indicators (1 to activate extraction, 0 to deactivate extraction) as parameters to the relevant action associated with the forwarding table. The packet then proceeds through the egress pipeline, where only the cloned packet is subjected to additional processing in this stage, moving through a set of tables (Table_L1, Table_L2, Table_L3, Table_L4_UDP/Table_L4_TCP, Table_Metadata), one table for each layer. Each table checks whether the corresponding layer argument is activated for the current flow. If it is enabled, the DFET module allows the control plane to perform another level of flexibility by passing binary flags in the same order of the corresponding standard header fields to determine which fields are subjected to be retrieved from this header (by passing 1) and which not (by passing 0). A one field equivalent customer header is defined and activated for each field marked for extraction. Regarding the transport layer, two tables are defined for the extraction, and the packet is processed through one of them based on the transport layer protocol it carries. For the metadata extraction, the control plane provides a sequence of binary flags that determine which metadata fields to include. These flags are ordered as follows: ingress timestamp, egress timestamp, hop latency, enqueue timestamp, enqueue queue depth, dequeue time delta, and dequeue queue depth. At the end of the egress pipeline, the packet undergoes the stage of the report generation, where the destination of the report is defined by the control plane and the produced report is always a UDP packet despite the original transport protocol (TCP/UDP).

4.1.2. Functional Validation:

To validate the functionality of the proposed DFET module, the Mininet emulation environment was utilized for topology creation. Mininet enables the emulation of realistic network scenarios and facilitates the verification of the behaviour of inserted flow rules. Figure 7 demonstrates the network topology used for the operational verification, it consists of one P4 switch, three hosts and three collectors. Host H1 sends two flows (one UDP flow and one TCP flow) to host H2 and sends one UDP flow to H2. The P4 switch has been instructed to monitor different features for each flow and forward the monitoring data to a specific collector as it is illustrated in Table 1.

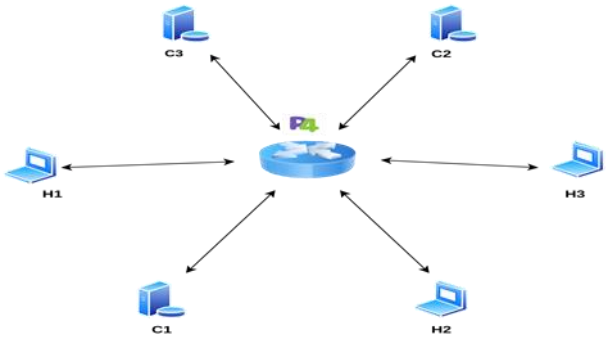
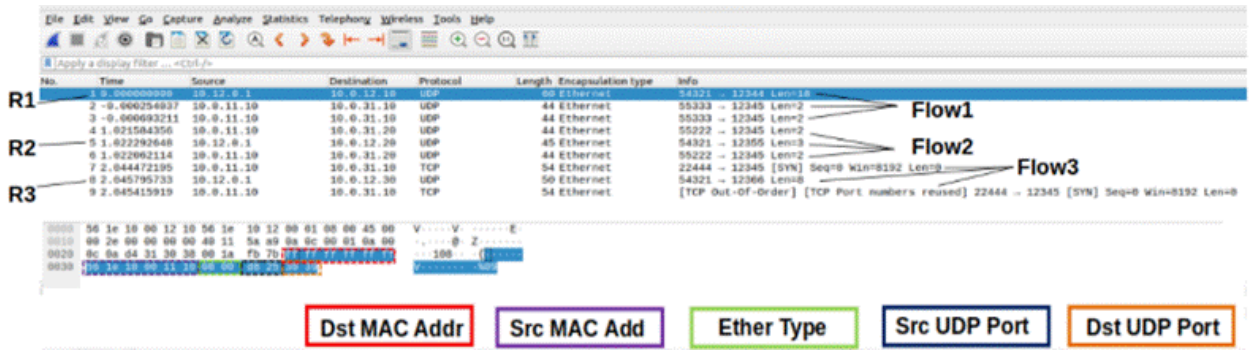


Figure 7: Network Topology in Mininet

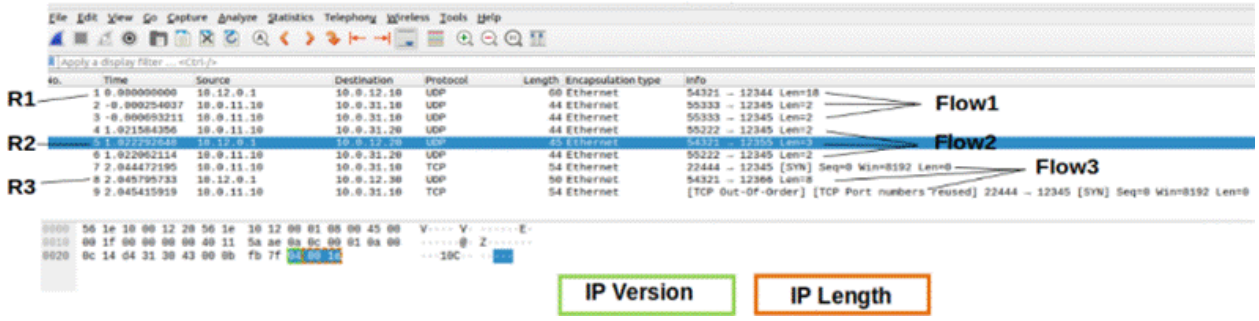
Table 1: Test flows and corresponding DFE extraction parameters

Flow	Report To	L2 Info	L3 Info	L4 Info	Internal Info
UDP	C1	Src MAC, Dst MAC Ether Type		Src Port, Dst Port	
UDP	C2		Version, Length		
TCP	C3			Src Port, Dst Port	Ingress Timestamp

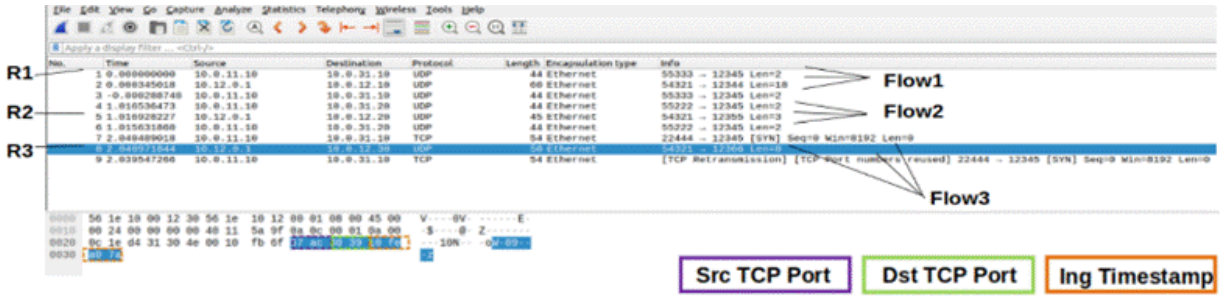
Figure 8 shows the packets captured by Wireshark at the P4 switch interfaces for each flow: the original received packet (length 2 bytes), the original forwarded packet (length 2 bytes), and three generated UDP reports. The sizes of the UDP reports vary depending on the amount of data extracted from each flow. As shown in Figure 8, the report length is 18 bytes for Flow 1, 3 bytes for Flow 2, and 8 bytes for Flow 3. Different colours are used to highlight the information contained in each report, which corresponds exactly to the information specified in Table 1.



a. DFET report for the first UDP flow.



b. DFET report for the second UDP flow.



c. DFET report for TCP flow.

Figure 8: DFET reports for the three generated flows.

The validation results demonstrate that the proposed Decentralized Feature Extraction Telemetry (DFET) mechanism functions as intended, successfully extracting varied information from multiple traffic flows and delivering it to designated destinations as configured by the control plane.

4.2. Multimodal Network IDS with PCAP Monitoring

This module offers a lightweight, AI-enabled Intrusion Detection System (IDS) for cloud-based services. It leverages Software-Defined Networking (SDN) for centralized data collection and control, in combination with artificial intelligence to deliver advanced threat detection and mitigation capabilities. This approach is motivated by the studies presented in section 2.2. It integrates three distinct representation methods through AI -Fusion methods.

The main objective is to create a resource-efficient IDS capable of promptly identifying and responding to security threats—particularly Denial-of-Service (DoS) attacks—while maintaining high performance and adaptability within modern cloud infrastructures.

The proposed IDS is grounded in a hybrid methodological approach that combines:

- Simple statistical machine learning models for rapid initial detection, using network data retrieved via the SDN infrastructure.
- AI-based analytical modules for in-depth threat characterization and attack profiling.

This dual-layer approach supports both speed and accuracy in detection while ensuring energy-efficient system operation. The OpenFlow protocol plays a central role in enabling communication and control between the SDN controller and network devices.

The proposed system will offer rapid Detection and Mitigation, which will be particularly effective in early-stage DoS attack scenarios along with detailed Threat Analysis. It is designed to provide insights into attack types and allows the identification of multiple malicious IP addresses. Moreover a lightweight design is proposed, optimized for minimal resource consumption, making it suitable for deployment in scalable cloud environments.

4.2.1. High Level overview

To enhance detection coverage and accuracy, multimodal architecture based on packet capture (PCAP) file analysis has been developed. The module aims to enable the extraction and evaluation of multiple data modalities from network flows. It offers deep packet inspection (DPI) alongside statistical session analysis. The module is protocol independent, meaning that it will support all standard communication protocols and is readily adaptable to new or evolving ones. It operates as an End-to-End AI Pipeline, eliminating the need for manual or domain-specific feature engineering.
The module processes network traffic captured in PCAP format that is pre-processed and filtered in a time efficient manner. Then it extracts three types of features from each flow detected:

- Image-based representations
- Feature Embeddings
- Statistical Attributes

Each feature type is processed by a dedicated AI model. Their outputs are then fused in a final decision module, allowing for robust detection of known and unknown attack patterns. Once a malicious flow is detected, the system triggers an alert and logs relevant details about the attack and its source. The process is shown graphically in Figure 9.

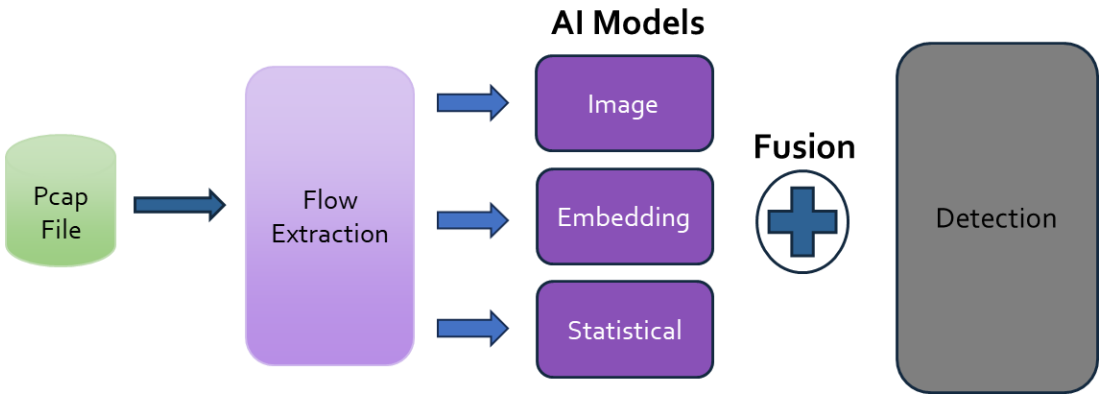


Figure 9: High level overview of the proposed approach.

Each type of feature is handled by a different AI model: Image-based representations are fed to a Convolutional Neural Network (CNN), Feature Embeddings are handled by a Long short-term memory Neural Network (LSTM) while Statistical Attributes are handled by a Multi-Layered Perceptron.

Currently, after an anomaly is detected, the user is notified via a terminal as shown in Figure 10. Along with the type of attack various other information is provided such as the IP address and Port of the Attack.

```
Detection of file37.pcap
Found HTTP/2 Large Window Flooding Content
File Captured at:     2024-05-24 10:16:50
File Detected at:     2024-05-24 10:17:03
Detection Time:       2.7 seconds
File Size:            0.39 MB
Total Sessions:       64
Malicious Sessions:   1
Src_IP:               10.42.1.143
Dst_IP:               172.17.0.3
Src_Port:             41996
Dst_Port:             7777
Sending Alarm
```

Figure 10 Example of attack detection result

4.2.2. Data Collection and preprocessing

The data collection phase will involve executing multiple distinct attacks e.g. Denial-of-Service (DoS) attacks across the CERTH testbed, capturing both malicious and normal traffic to create a representative and balanced dataset. Protocols such as TCP, UDP, and SCTP will be included to encompass a wide range of network applications, including those relevant to 5G technologies. Traffic data is captured in the PCAP format and pre-processed by removing identifying information to avoid training bias. Flows are then extracted based on 4-tuple identifiers. To address real-time detection challenges, PCAP files are segmented by time intervals—beginning

at one second and doubling with each subsequent segment—thus facilitating better file management and more practical flow identification, especially for UDP.

Once flows are identified, three categories of features are extracted per session: statistical features, embeddings, and image representations, as shown schematically in Figure 10.

Statistical features provide insights into traffic behaviour over time, highlighting anomalies through deviations from expected norms. These features, adapted from CICFlowMeter, are selected for their protocol independence to maintain adaptability. A total of 57 features were extracted for bidirectional flows and 19 for unidirectional flows. Embedding extraction, on the other hand, involves processing the hexadecimal packet data from each flow, converting byte pairs to integers, and standardizing the input to 1024 bytes through trimming or zero-padding. This transformation allows sequential models to interpret the hex stream as time-based input for AI-driven classification.

Image extraction followed a similar preprocessing path as embeddings. After converting the hex stream to integer values and ensuring a consistent length, the data were reshaped into 32×32 pixel grayscale images. These visual representations capture flow characteristics in a format conducive to convolutional neural networks (CNNs) or other image-based AI models. Each of the three feature types—statistical, embedding, and image—is processed through dedicated AI models. The final IDS decision is produced by fusing the outputs of these models using learnable parameters, providing a robust and flexible detection mechanism suitable for a range of cyber threats in both traditional and next-generation network environments.

4.3. AI-Driven Multi-Agent System for Real-Time Threat Intelligence and Automated Response in 5G Networks

Latest generation networks (b5G/6G) introduce complex security challenges stemming from their highly distributed, software-defined, and service-based nature. Addressing these challenges requires intelligent, scalable, and adaptive security systems that go beyond static rule-based models. The following section presents the high-level overview of an AI-driven, multi-agent architecture for real-time threat intelligence and automated response in 5G environments. The system is built around a secure, modular foundation using the Model Context Protocol (MCP) [104] and leverages advanced deep learning and large language model (LLM) mechanisms. Furthermore, the system is carefully mapped to the 3GPP security framework to ensure architectural compliance and operational synergy with established telecommunications standards.

4.3.1. System Architecture

The architecture consists of a collection of intelligent agents, each operating in a Dockerized environment to ensure high scalability, modular integration, and secure deployment. These agents are orchestrated using MCP, a communication protocol that enables the dynamic construction of agent workflows on top of LLMs while ensuring contextual data integrity and access control. The following subsections present details on the MCP and the functionalities of four different Agents.

4.3.1.1. Model Context Protocol

The Model Context Protocol (MCP) serves as the foundational communication and coordination layer that enables context-aware interactions among intelligent agents and large language models (LLMs) within a distributed AI system. It is particularly suited for high-stakes environments such as 5G cybersecurity, where secure, interpretable, and composable workflows are necessary for effective real-time decision-making [104][105].

At its core, MCP introduces a structured mechanism for maintaining and transmitting so called "context objects" i.e. semantically annotated data containers that encapsulate both the inputs and outputs of agent interactions. These objects persist across different stages of an AI-driven workflow, allowing downstream agents or models to reason with awareness of the full context in which prior decisions were made.

MCP ensures that all data shared between agents is bound to a secure, queryable context which is persistent and formally defined. MCP supports declarative workflow composition, enabling agents to be linked into arbitrarily complex configurations—such as chains, trees, or feedback loops—without losing context fidelity. This composability allows agents to collaborate on multi-step reasoning tasks, well suited to the tasks of the proposed system. An example of a multi-step reasoning in the discussed context is "receive information concerning threat → evaluate impact → generate report → select mitigation" in a traceable and consistent manner.

To protect these interactions, MCP incorporates a secure multi-agent messaging system where all communications are encrypted, time-stamped, and authenticated. Role-based access controls ensure that only authorized agents can read from or write to a given context, and temporal scopes define the lifetime and expiry conditions of context data. This ensures strict compliance with security and privacy policies, which are particularly critical in telecommunications environments governed by regulatory standards.

A key strength of MCP lies in its native integration with large language models. By treating LLMs as first-class agents, MCP allows dynamic retrieval and injection of context into prompts, as well as bidirectional reasoning between structured data and natural language outputs. This makes it

possible to support hybrid workflows that combine statistical learning (e.g., deep anomaly detection) with symbolic reasoning and narrative synthesis.

Furthermore, MCP maintains complete provenance for all context exchanges. Every decision, transformation and output are logged with agent identifiers, decision justifications (when extracted from LLM chains), and associated metadata. This audit trail is invaluable for post-incident analysis, model refinement, and regulatory compliance.

In the multi-agent system described in this section, MCP acts as the cohesive tissue that binds together specialized security agents—those performing threat detection, IOC correlation, response selection, and orchestration—into a coherent, resilient, and transparent decision-making system suitable for securing next generation networks.

- Threat Intelligence Agents

Two distinct AI-enabled agents will be developed to handle the following tasks: gathering, correlating, and interpreting threat intelligence data.

IOC Correlation Agent

This agent uses a two-stage AI pipeline to correlate indicators of compromise (IOC) across network functions to detected coordinated attacks or evolving threats.

The first stage utilizes a stacked autoencoder deep neural network (DNN) [106]. It has multimodal data input, including security logs, network traffic and resource related data. The model will be trained using data extracted using normal network operations. This type of DNN extracts and compresses relevant latent features from these inputs and uses these to recreate the original inputs. This characteristic allows the DNN to handle new inputs and utilizing them to classify the condition of the network as normal or abnormal at a given time.

Once abnormalities are discovered, the mechanisms of the second stage is activated: An LLM-based analytical layer then consumes resource consumption and security related data of network functions (e.g., AMF, SMF, UPF), to perform cross-domain correlation across the network, identifying indicators of compromise (IOCs) and uncovering potential patterns that e.g. might be associated with coordinated, multi-vector attacks.

Threat Reporting and Insight Agent

This agent utilizes two subsystems. The first is a retrieval-augmented generation (RAG) framework that connects a fine-tuned LLM to a domain-specific knowledge base. This knowledge base includes a) Historical incident data b) Cybersecurity whitepapers c) Relevant standards and best practices [107]. The LLM synthesizes this information to generate real-time, human-readable threat intelligence reports tailored to specific network zones (e.g., RAN vs. Core) or

operator roles (e.g., SOC analyst vs. compliance officer) of every threat or attacks detected. These reports will include multiple aspects of security related information such as severity scores, impacted assets, root cause analysis, and recommended actions.

A second LLM is utilized to receive the outputs from a) the IOC Correlation Agent and b) the RAG to create the final output for this agent, i.e. reports for different time granularities e.g. a daily or weekly digest etc.

- Automated Response Agents

The response subsystem consists of two AI-empowered agents responsible for determining and executing adaptive mitigative actions to handle threats and attacks against the system.

KPI-Driven Response Selector Agent

The first agent is a KPI-Driven Response Selector. It utilizes a pointer neural network [108] that picks the optimal selection against a set of predefined response actions (e.g., rerouting, quarantine, rate limiting) by performing multi-objective optimization based on the values of several key performance indicators (KPIs). These KPIs are evaluated in real time based on network telemetry and risk metrics. The agent will interface with the appropriate network endpoints to trigger enforcement actions via secure APIs.

Orchestration Coordination Agent

The second agent handles Orchestration Coordination. This agent employs an LLM that interacts with Security Orchestration, Automation, and Response (SOAR) tools/platforms to perform complex mitigation tasks such as a) Patching vulnerable services, b) Updating firewall configurations c) Adjusting access control lists (ACLs) and d) Modifying slice-level security policies. It ensures end-to-end execution traceability and feedback incorporation into the agent network for closed-loop adaptation. A second LLM is utilized to document all decisions in a human readable format.

Figure 11 presents the high-level architecture of the proposed solution.

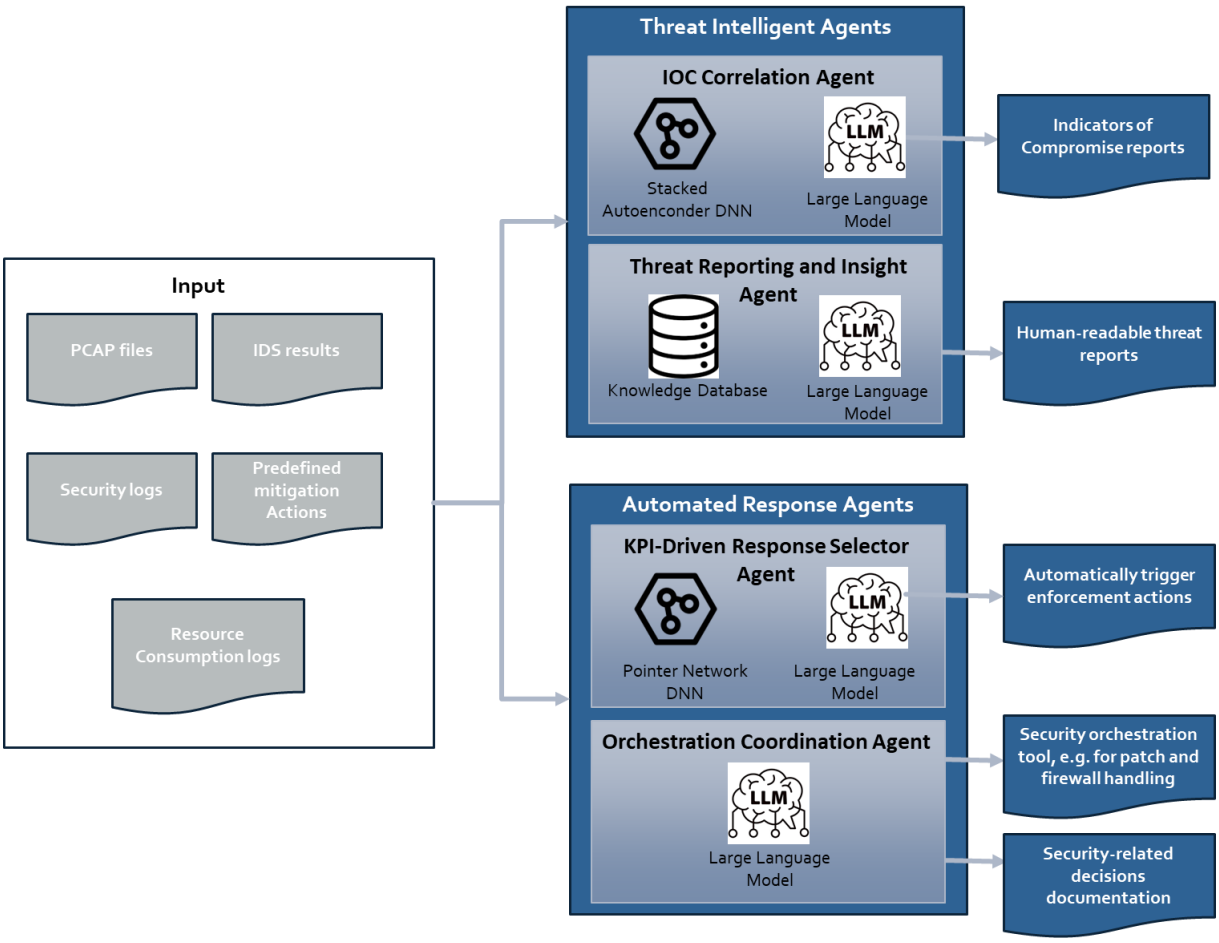


Figure 11 High level overview of the proposed Architecture

4.4. Real-time monitoring and centralised response to network threats

4.4.1. Technology summary

The device is based on FPGA technology and comes in the form of a PCIe SmartNIC card, designed to be integrated into existing network infrastructure. Its primary role is to detect Distributed Denial of Service (DDoS) attacks in real time by analysing network traffic at very high speed. To achieve this, it combines several advanced technologies such as FPGA-accelerated hardware processing, machine learning algorithms to identify malicious behaviours, and fine-grained customisation enabled by P4 programming. This combination ensures intelligent and adaptable packet inspection with minimal latency.

This device features multiple interfaces, for instance a 10 Gbps Ethernet interface for intercepting traffic, a PCIe interface for communicating with the host machine via an API, and a 1 Gbps Ethernet interface for management and configuration. Through its API, it can also provide alerts, detailed measurements and actionable recommendations to management systems or the host machine.

4.4.2. Benefits for the network

In a distributed network architecture, multiple FPGA-based devices can be deployed at strategic points in the infrastructure, each performing local threat detection and analysis. These devices operate autonomously to inspect traffic in real time where they are installed, but their full potential is realised when they are integrated into a centralised monitoring system. Each device exposes an API that allows data to be sent to a global dashboard responsible for collecting, aggregating, and analysing information about the entire network.

This centralised dashboard plays a key role in correlating security events: it can detect distributed attack patterns, such as coordinated DDoS campaigns, by comparing suspicious flows observed on different devices. Using the API, the dashboard receives real-time alerts, detailed metrics, and event logs, giving network administrators a unified dynamic view of the overall security level of the network.

Beyond simple monitoring, the dashboard can also play an active role in incident response. Using APIs exposed by each device, it can emit commands to update detection rules, isolate compromised network segments, or block malicious flows close to the source. This centralised coordination enables a rapid and consistent response to threats, significantly enhancing the overall resilience and security of the infrastructure.

4.4.3. Implementation in the field

Devices are deployed across the network, each equipped with real-time traffic analysis capabilities. These devices generate security alerts based on suspicious activity and expose an API through their PCIe interface. This API allows for the dynamic configuration of detection rules and machine learning models, enabling rapid response and adaptation to emerging threats.

A centralised monitoring system, as shown schematically in Figure 11, hosted securely, collects and consolidates data from all deployed devices. This dashboard regularly queries each device through its API and is also capable of receiving push webhooks when critical events occur. Local alerts and identified suspicious traffic are transmitted to the central system in structured message formats. The dashboard then aggregates, stores, correlates, and visualises these events using graphical tools such as Grafana.

The system offers more than just visualisation. Through the same device APIs, the dashboard can issue remote commands for orchestration and response. For instance, it can block suspicious flows across multiple devices, update detection models or thresholds in real-time, and even isolate network segments where infected containers are detected. It can also disseminate blacklists to all devices through the central controller. Every command sent and event received is logged, ensuring full traceability and enabling comprehensive security audits and efficient incident response.

Security and scalability are integral to the system's design. All API communications are secured using mutual TLS authentication and encrypted over HTTPS. The architecture supports horizontal scalability through load balancers, allowing it to manage from dozens to hundreds of devices efficiently. Auto-discovery features and integration with orchestrators further enhance the system by automating the enrolment and configuration of new devices.

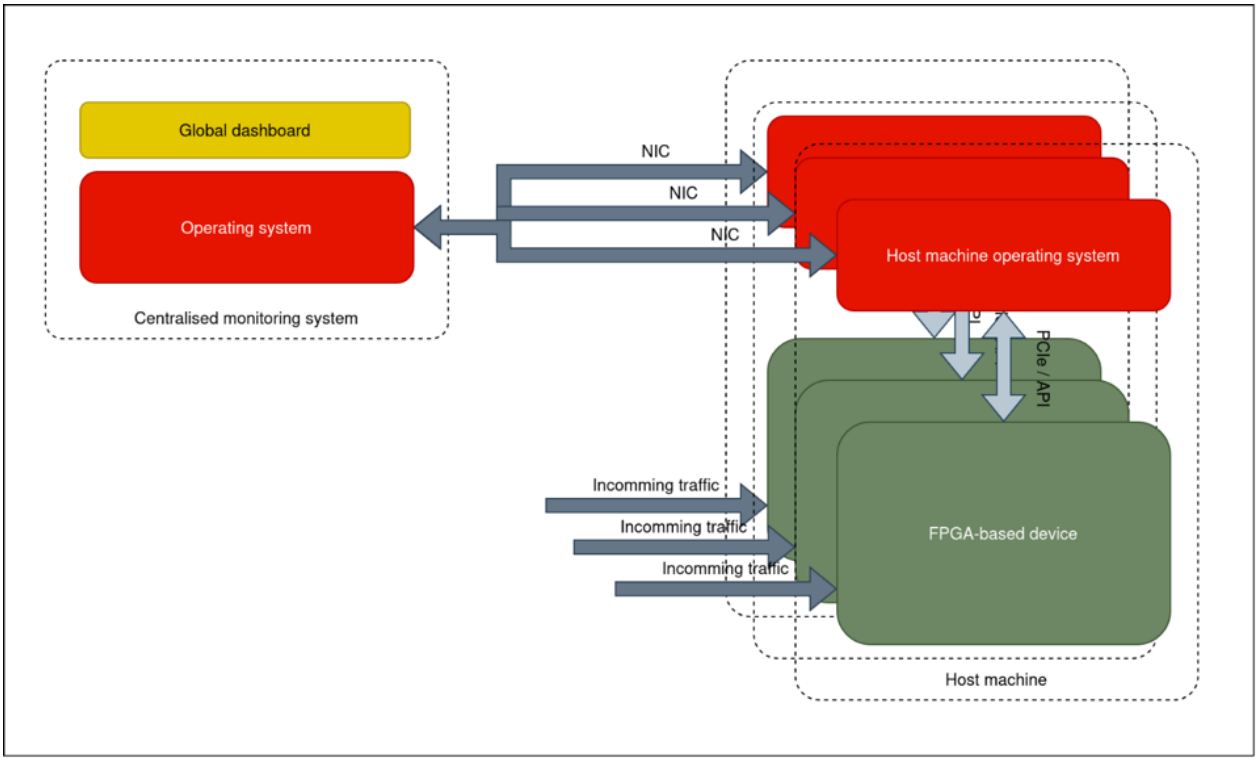


Figure 12: Example of centralized monitoring.

5. Blockchain-based Trust Establishment

The rapid utilization of IoT devices in their integration in 5G networks has introduced new demands for scalable, low-latency, and secure trust establishment approaches. In modern 5G environments, devices often require access not just to network services but also to third-party service providers across diverse ecosystems. This shift calls for an authentication model that goes beyond traditional, centralized approaches—one that can offer trust, privacy, and interoperability at scale. The ZT and blockchain-based trust management approaches are two critical strategies for securing 5G networks. zero trust architecture removes implicit trust from any system component, mandating active authentication for all network entities, both internal and external. This method improves security by enforcing continuous verification and minimizing attack surfaces through access control. However, its downside is the potential performance degradation in ultra-low latency applications due to the overhead of continuous verification.

Blockchain-based trust management, on the other hand, uses decentralized ledgers to ensure transparency and immutability, offering a trust framework without relying on central authorities. This decentralization is beneficial for the dynamic nature of 5G networks with many devices but introduces challenges related to the scalability and privacy of a large, decentralized network. The complexity of managing such a system and ensuring secure, efficient smart contracts can also impact system performance and increase the risk of attacks, such as de-anonymization. Several additional security strategies complement these approaches. Advanced authentication mechanisms, such as elliptic curve cryptosystem-based attribute-based encryption (ABE), help minimize computational overhead while maintaining robust security for IoT devices in 5G networks. However, these methods must balance security with efficiency to avoid excessive energy consumption or processing delays, particularly in resource-constrained environments like drones. Moreover, the integration of Multi-Access Edge Computing (MEC) and Device-to-Device (D2D) communications enhances specific use cases like smart manufacturing but can introduce bottlenecks if intermediate nodes become overloaded.

Current 5G authentication mechanisms during trust establishment rely heavily on involving network functions within the core network, such as the AMF and AUSF. While these network functions provide robust security for network access, they are not optimized for repeated or federated end to end validation when devices interact with multiple external service providers. This centralization introduces potential bottlenecks and unnecessary latency—especially in trust-sensitive and end to end IoT use cases such as smart manufacturing and smart cities. To address these challenges, NATWORK includes a security management service that integrates blockchain technology with the 5G authentication process, enabling decentralized and transparent trust establishment. This service allows devices to prove their authenticity directly to service providers—without needing repeated interaction with the 5G Core. This approach aligns with the

principles of zero-trust networking and supports scalable, trust less access control, particularly in distributed IoT environments.

The service leverages both standard and newly introduced components to enable blockchain-based trust establishment. It is built on the standard 5G Core architecture, preserving its native functions and the 5G Core Network as the baseline infrastructure). As shown in Figure 13, the main players are as follows:

- **UE Device**: The UE device is an IoT node that aims to join the network and utilizes the services provided by the IoT service provider. It initiates the process by interacting with the gNB node and requesting to join the network and later accessing services.
- **gNB Node:** The gNB nodes reside in access plane and act as an intermediary node between the UE on one side, the core network, and the service provider on the side. The node facilitates part of the UE registration and trust establishment.
- **Core Network**: Through the involved NFs, including the AMF and AUSF, it provides the initial registration of UE in the network. Additionally, it generated the tokens and stored in the blockchain.
- **Service Provider**: The node provides services to the authenticated and authorized UE in which the trust with them has been established and grants access based on authorization properties.

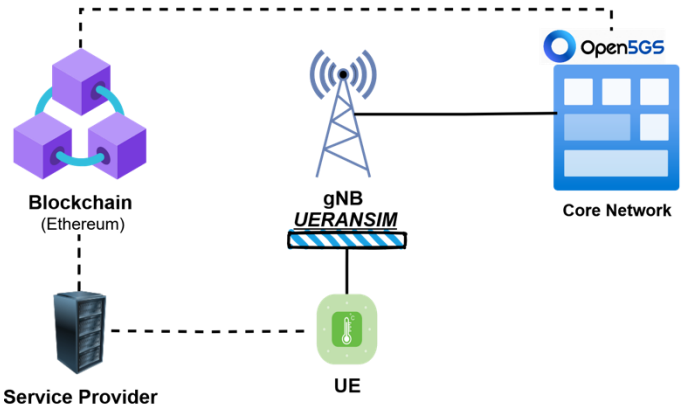


Figure 13 High level overview

Additionally, the following two other framework components enable the decentralized authentication and access control in an IoT environment:

- **Blockchain**: The traditional authorization database is replaced with an Ethereum-compatible permissioned blockchain. This provides a decentralized, transparent, and integrity-safeguarded mechanism for device authentication management. It consists of a permissioned Ethereum Blockchain and a smart contract.

- **Bridge:** It is a vital component of the trust establishment which acts as a communication bridge between the 5G core network and the blockchain. The main function of this bridge is to listen to the log of the AMF function inside the 5G core, derive the pseudonym associated with the registration, and to write authentication and access control status to the blockchain via Web3 interfaces.

UE registers with the 5G network and, simultaneously, a pseudonym is generated from its SUPI and recorded on the blockchain. RAN node that manages the wireless link, facilitating UE registration and service requests. AMF oversees device registration and mobility and triggers the blockchain-based process by emitting logs, which are monitored externally. AUSF and UDM conduct standard identity checks and enable initial network-level trust. SMF and UPF handle session setup and data flow, routing traffic to the data network (DN), where external services reside. DN is the logical endpoint for external services. Here, further authentication happens through blockchain mechanisms.

The Service Provider Module simulates an application or external service. Though centralized in its design, it uses the blockchain for offloading identity verification. Its key operations include: Receiving authentication requests from UEs, verifying pseudonyms through a blockchain smart contract (Auth5G), and performing challenge-response with the UE using cryptographic signatures, and issuing short-term tokens for low-latency access without repeated authentication

Instead of relying on internal databases for identity storage, this service utilizes a permissioned blockchain (Ethereum-Compatible with Smart Contracts) to manage device credentials. Auth5G Smart Contract manages access control, stores pseudonym records, validates service provider identities, and handles access policy checks. Device pseudonyms, timestamps, validity periods, and access control hashes are stored in the network. The service guarantees verifiable, immutable identity assertions with reduced reliance on centralized control.

The 5G-Blockchain Bridge in this service facilitates the interaction between the 5G Core and the blockchain. It monitors AMF logs for successful device registrations, and extracts SUPIs and derives pseudonyms using time windows and deployment-specific salts [75]. Finally, it submits authentication records to the blockchain via Web3 interfaces. This bridge ensures seamless communication between otherwise separate infrastructures, while respecting existing 5G standards.

To maintain user privacy while supporting trust, UEs are identified on-chain via pseudonyms. These are generated as follows: First it combines the UE's SUPI, a X-hour time window, and a deployment-specific salt. Then, it hashes the result and produces an unlinkable identifier. Finally, it stores the pseudonym on the blockchain with metadata such as expiry time and access policy reference. This allows the service provider to validate the UE without learning or storing the

original identity. When a UE requests access to a service, the service provider queries the blockchain to check the pseudonym's status, verifies the UE's cryptographic signature, and issues a service token for future access without repeating blockchain queries. This design reduces latency, enhances trust decentralization, and supports stateless verification aligned with zero-trust principles.

5.1. Blockchain Authentication Mechanism

This service integrates a blockchain-assisted authentication mechanism, where both the UE and the service provider are represented through verifiable identities recorded on-chain, aiming to enable secure and transparent trust decisions. The blockchain authentication process contains the following.

5.1.1. Pseudonym Generation

For representing UE identity on-chain, a pseudonym-based UE identity is employed. Each UE is identified with a pseudonym after a successful registration with the Core network. The pseudonym is the hashed long-term Subscription Permanent Identifier (SUPI) and serves as a privacy-preserving on-chain identity, being deterministically generated by integrating also a time window, and a salt. The combination is then hashed with the Keccak256 algorithm [76] to obtain a fixed-length pseudonym that is unlinkable to the original SUPI and is compliant with any blockchain identity representation.

Once the pseudonym has been generated, it is recorded on-chain via a transaction to the smart contract. In this case, the transaction includes: the pseudonym being used as the identity reference, a validity duration of 12 hours after which the pseudonym needs to be regenerated and re-authenticated, and a hashed access policy that is derived from the UE's network slice and session parameters that are retrieved from the 5G subscription database.

5.1.2. Service Provider Registration:

Similarly to UE authentication, each service provider must be explicitly registered on the blockchain to take part in the authentication process. This is done in parallel to UE identity management, and each service provider is identified by its blockchain address, and registration is performed by the smart contract deployed by the network operator. The smart contract will maintain an indexed list of accepted service providers, and for each service provider, it stores the metadata, including the registration time and current status. This is to allow the service provider to be approved before it can issue an authentication challenge or verify UE credentials. Only the service provider recognized on-chain can call the authentication records decentralized.

5.1.3. Pseudonym Verification

When a UE interacts with a service provider, the provider needs to verify the UE's pseudonym on the blockchain. In the verification process, the service provider checks if the pseudonym has been authenticated and is currently active, at the same time, the authentication record of the pseudonym to make sure it is still within the valid lifetime, then, the authentication record of the pseudonym to make sure the service scope of the service provider includes the associated access policy.

The end-to-end trust establishment in NATWORK delivers several key advantages for IoT and 5G security. First, it shifts trust from centralized 5G Core entities to blockchain-enabled mechanisms. Also, it uses cryptographic pseudonyms and verifiable smart contracts to enforce strict access controls, and minimizes repeated interactions with the 5G Core, improving response times for external services. This component works alongside existing 5G infrastructure without modifying native components. By combining blockchain's integrity with 5G's flexibility, this component enables a scalable and resilient trust layer for IoT services.

5.2. Main Phases

Generally, the component has three main phases. Figure 14 illustrates the process in more details.

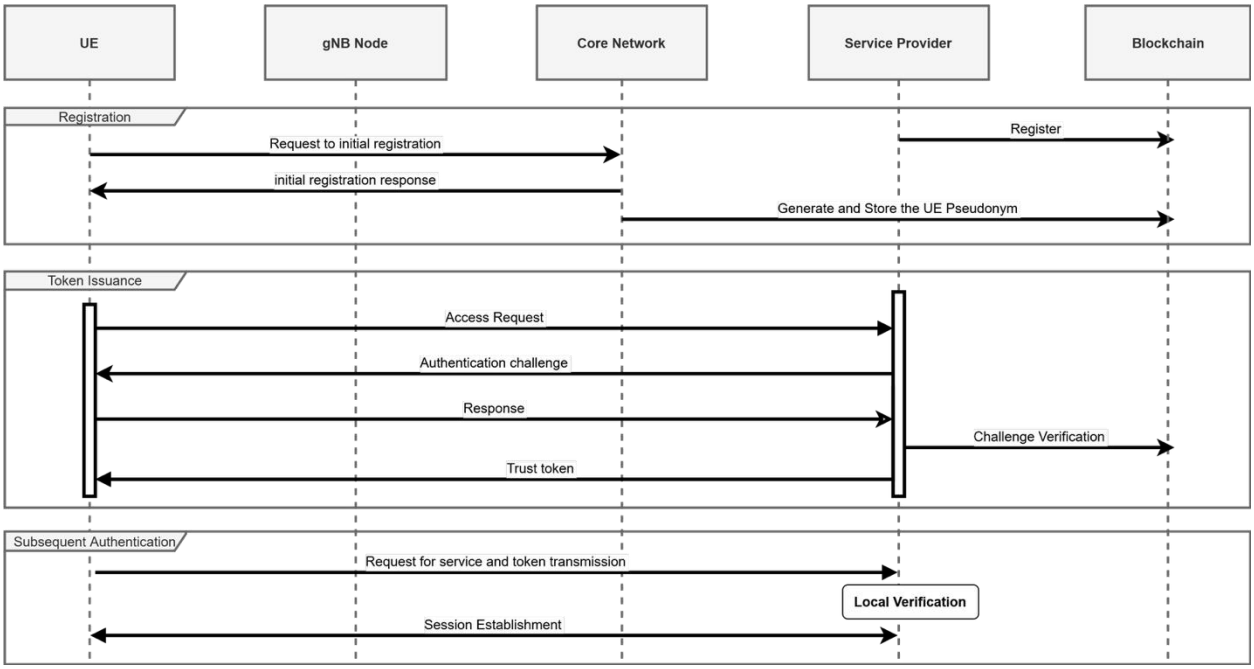


Figure 14 Main phases.

In the following, the phases have been briefly described.

UE Registration: The UE initiates the process by sending a request to register with the gNB node, which will be directed to AMF in the Core for initial authentication. AMF performs the authentication and security procedures with UE. Simultaneously, the registration will be observed by the Bridge and based on that the pseudonym will be generated and stored in the Blockchain. Finally, the Core network responds with an initial registration response, confirming that UE is registered in the network.

Token Issuance: When a UE tries to access a service provider for the first time within the validity window of the pseudonym, the UE can be fully verified through the registered pseudonym. The service provider issues a random cryptographic challenge to the UE. The UE signs the challenge with its own private key of the blockchain wallet. The service provider then sends the challenge and the signature to the blockchain, where the verify˙Authentication function is called on the smart contract to verify the authenticity of the UE. The function will check if the pseudonym exists and has been recorded on-chain, the current timestamp is below the expired time, and the pseudonym's UE's associated access policy hash equals the one required by the service provider.

Subsequent Authentication: During the subsequent authentication and access, the UE will present the service token issued in the previous step to the service provider, the service provider then performs an offline, local verification of the token, without contacting the blockchain again. This design significantly reduces authentication latency and supports scalable, high-frequency access patterns. The service provider recomputes the token hash from the cached payload and compares it with the token received to ensure the integrity and authenticity. The service provider checks the expiration timestamp embedded within the token to make sure its validity. To optimize the performance of the authentication, the service provider maintains a lightweight in-memory token cache with efficient lookup and automatic cleanup of expired entries. This allows for rapid and reliable access control for trusted UE, without any additional blockchain cost, thereby ensuring low-latency authentication in time-sensitive environments such as IoT-based services.

6. Explainable AI

With the growing complexity of ML models, it is becoming more crucial to be able to understand the decisions or predictions made by AI-based systems. The black-box nature of such models becomes even more of a serious concern when there is an automated AI-powered system that can take actions without requiring human input. In these cases, the system owners or developers would prefer to have a means to interpret (i.e., "explain") why a certain decision is made by an AI model. Such transparency not only allows system owners to foresee future decisions under similar circumstances but also enables them to adjust their current knowledge depending on the past actions. Moreover, by providing Explainable AI (XAI) solutions, the decision-making processes can be made accountable, ensuring the compliance to certain legal standards.

In the scope of NATWORK project, XAI has a significant importance for delivering high-quality and trustworthy solutions. The envisioned 6G architecture relies heavily on AI-powered components, leveraging advanced ML methods across edge-to-cloud continuum. By following the seamless orchestration and integration approach provided by 5G standards, mainly NFV and ZSM, 6G networks will also decrease the need for human input for continuous operation and allocation of resources. Therefore, achieving reliable, accountable, and transparent decision-making processes is critical for every service of the project. In addition, these beneficial features would help the operators to investigate potential issues in case of unexpected decisions performed by automated models. For this purpose, XAI is the key concept to unravel the black-box nature of the underlying AI mechanisms, by either making the models intrinsically interpretable (i.e., intrinsic XAI models) or developing separate explainability components (i.e., post-hoc XAI models), as previously explained in 2.4. This section explains the specific XAI techniques utilized by each applicable NATWORK component, which differ based on the underlying mechanisms or the requirements of the relevant service.

6.1. XAI extension for Multimodal Network IDS with PCAP Monitoring

This sub-module aims to enhance the transparency of the AI-based Intrusion Detection Systems (IDS) module presented in section 4.2. It will introduce explainability features aimed at making the model's decision-making process more interpretable for users. The complexity of many deep learning models, often referred to as their "black box" nature, limits user trust and hinders adequate validation. Addressing this issue is particularly important in cybersecurity contexts, where understanding the rationale behind alerts is crucial for operational response and model improvement.

A major challenge in explainability arises from the nature of payload data. Payloads contain complex and often noisy patterns that are inherently difficult to interpret. This complexity is compounded by the high volume of data processed by IDS models, making it challenging to

isolate meaningful features or causes behind detections. As a result, raw payload analysis frequently falls short in providing transparent or actionable insights into the system's behaviour.

To overcome these limitations, statistical features are employed to support model explainability. These features, such as bytes per second or flow duration, enhance human readability and offer a clearer decision path that can be logically traced. Their structured and interpretable nature also facilitates model debugging and the identification of erroneous or misleading behaviours. In this context, SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) are proposed as algorithmic tools to identify explainability patterns within flow payloads, further contributing to a more interpretable and trustworthy IDS framework.

The submodule will provide the following explainability elements based on the statistical features:

- Graphic depictions of feature contributions
- A set of decision rules that underly the AI tools utilized
- Traces of the path followed for each decision of the IDS

6.2. X-MORL – Explainable Multi-Objective deep-RL

The security of 6G/NextG networks can be strengthened by MTD, increasing the uncertainty for attackers and reducing their chances of success. However, enforcing MTD operations can also impact network performance and come with additional operational costs and energy consumption. Therefore, smart and dynamic control of MTD following a cognitive paradigm (i.e., following the ETSI ZSM closed-loop methodology as previously described in Section 3.1) considering security requirements, security gains, overhead, and feasibility is crucial. These are multiple objectives to be considered that often do not overlap, and conflicts might arise when performing MTD operations, favouring one goal to the detriment of the other.

For instance, moving a VNF from a remote Virtual Infrastructure Manager (VIM) to an edge node's VIM for communication optimization may be a poor choice security-wise, since an attacker can easily predict that action. A purely random placement, instead, improves security by reducing its predictability but can hinder the network's performance and QoS of the moved service.

In the scenario of MTD operations in the telco edge-to-cloud continuum, moving a VNF to a closer edge VIM may improve latency, but it may also weaken security since its position and movement becomes predictable to attackers following traffic loads. Conversely, a completely random move aimed at enhancing security could negatively affect the network and service performance.

Consequently, **the AI-based MTD service provided in this project** uses Multi-Objective Markov Decision Process (MOMDP) to monitor and model the state of a Telco Cloud network (*i.e.,* B5G

ant NextG networks) and train a deep-RL model to tackles the multi-objective optimization problem, in which three main objectives are quantified and considered:

1. To find the optimal balanced strategy to maximize security (*i.e.*, minimize threats and reduce their likelihood to succeed)
2. To minimize its operational cost (overhead in the consumption of physical resources)
3. To alleviate the impact on QoS and service availability (*i.e.,* reducing service downtime and network overhead)

An important requirement is for the decisions made by such ML model to be humanly explainable, as it makes decisions affecting critical infrastructures and potentially moving and reconfiguring critical services running on the telco network. However, the integration of explainable AI into MTD remains an open and unexplored research question. In this context, the X-MORL (eXplainable Multi-Objective RL) module is designed and implemented to provide explainable MORL models using reward decomposition [74]. With this method, rewards can be classified according to semantically meaningful reward types, which fits well with the multi-objective nature of the MTD optimization problem.

6.2.1. Deep-RL and MORL

RL agents learn by interacting with their environment, observed and modelled as a Markov Decision Process (MDP) -- a tuple *(S, A, P, R, γ)* where *S* is the set of states of the environment, *A* is the set of actions that the agent can take, *P* is the transition probability matrix defining the probability that an action a_i changes a state s_i to a new specific state s_j, *R* is a set of reward values for all (a_i, s_i) pairs and *γ* is the discount factor defining the importance of the immediate rewards over the future rewards.

The agent's goal is to learn an optimal policy that maximizes the cumulative reward. RL has seen evolutionary advances through the usage of deep neural networks (DNN) leading to deep-RL algorithm [116]. Conventional deep-RL algorithms, however, are designed for single-objective optimization and used with the scalarization of the different rewards corresponding to the different objectives into one reward value. This scalarization can be part of the missing information we want to learn, *i.e.*, the best trade-off among objectives to maximize the overall return. If the optimization occurs for only one fixed weighted sum, the result produced would be suboptimal as other weight sums are not explored.

Specific to the explainability of the deep-RL model, the single scalarized value can be semantically meaningless as multiple objectives be fundamentally different in nature. For instance, one of the three MTD objectives is to reduce an economic cost metric, measured in a monetary unit, while another objective is to improve proactive security, which is measured in terms of the attack

success probability (ASP) reduction of a threat. Merging both measures gives a value that is hard to interpret and leads to decisions that are also hard to explain.

MORL is a new category of RL algorithms that keeps different interpretable reward functions, one for each objective, and iterates the optimization process on different weighted sums, avoiding suboptimal solutions and approximating the set of optimal policies for all scalarizations (see Figure 15). This solution set is defined as the coverage set (CS), which, for monotonically increasing reward functions, is reduced to the Pareto Front (PF). PF is the set of undominated solutions, where each solution is optimal with respect to a specific scalarization. MORL's interactions to retrieve the PF occurs with a MOMDP, where the main difference with respect to the MDP's definition is that R is now a vector $\bar{R}$ comprising the reward values of the multiple objectives defined in the model.

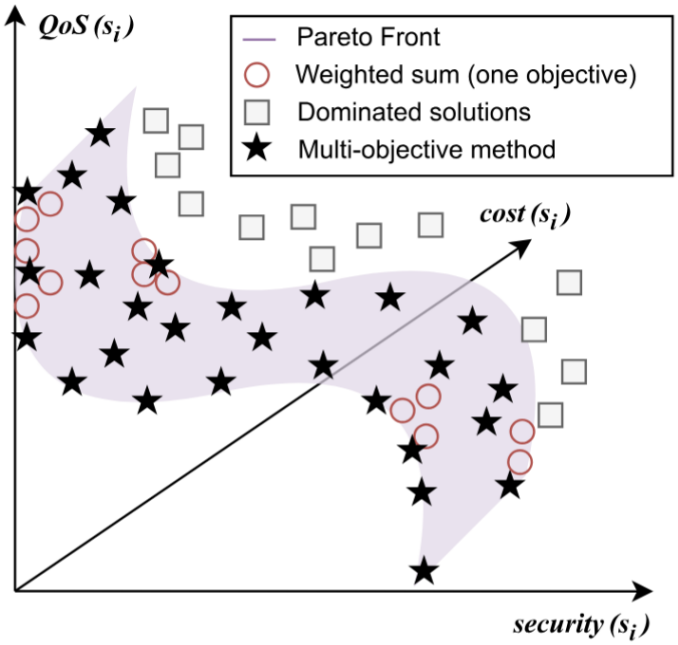


Figure 15: Pareto front for three objectives showing the benefits of MO methods over weighted sum optimization methods.

6.2.2. MORL reward decomposition

The deep-RL training brings the model to define a value function $v_{(s,a)}$, or Q-function, estimating the value of performing an action a at a state s in terms of reward acquisition, and following the policy thereafter. Reward decomposition decomposes the reward function into a vector $\bar{R}$ and then calculates the decomposed value function, which sum leads to the original Q-function when summing the value functions based on a scalar defined to unify the rewards of the different objectives. The decomposed Q-function then provides statistical explanations on which objective affected a specific action a at state s the most.

In order to do that, X-MORL defines three Q-functions: Q_c, Q_a, and Q_s, where Q_c represents the Q-function related to the objective of reducing the operational cost of MTD actions, Q_a the Q-function related to the objective of reducing the impact on the availability of the protected NFs, and Q_s the Q-function related to increasing the proactive security of MTD operations (measured as reducing the likelihood of exploitation of NFs attack surface). The three Q functions are calculated with the decomposed reward Q-learning (drQ) algorithm [117], which guarantees the convergence of the estimated values towards the value function of a learned policy. Finally, to understand why the MORL agent took an action a_1 instead of other actions a_i, we calculate the difference between values $Q_{c,a,s}(a_1)$ and $Q_{c,a,s}(a_1)$, *i.e,* the values $\Delta_{c,a,s}(a_1, a_i)$, and then derive a reward difference explanation (RDX), that shows the objectives that a_1 improves over the other actions leading to the MORL agent's decision.

6.3. Explainable Ensemble Graph Attention Networks

Cell-level Key Performance Indicator (KPI) monitoring has an importance to ensure reliability in future networks. Cell-level KPIs are not independent of each other: the behavior of one cell is strongly conditioned by what happens in the neighbouring cells. Flattening this structure into a tabular form discards precisely the interactions that a root-cause analysis (RCA) needs. For instance, two adjacent cells that share spectrum or whose coverage areas overlap can degrade one another's throughput, yet such cross-cell effects vanish once the features are aggregated.

Graph-based learning avoids this pitfall by treating each cell as a node and each inter-cell relation (e.g., interference, hand-over adjacency, shared feeder) as an edge. Among the many flavours of Graph Neural Networks (GNNs) [119], Graph Attention Networks (GATs) [120] are especially attractive for telecom data because their attention mechanism assigns content-dependent weights to every neighbour. Unlike spectral GCNs, where aggregation weights are fixed by the Laplacian, a GAT can learn that, for example, a high-load neighbour matters more than an idle one.

6.3.1. Ensemble GAT model

Our data consists of daily snapshots of the same physical network taken at different times. Peak-hour snapshots resemble each other (high load, many degradations) and differ markedly from off-peak snapshots. A single GNN trained on the union of all snapshots must compromise between these regimes, and it quickly becomes compute-heavy as the number of snapshots grows. Aggregating the snapshots first is faster but erases temporal diversity. Therefore, we have implemented the following ensemble learning steps: 1) Partition the snapshots into homogeneous subsets (e.g., morning, afternoon, and night). 2) Train one GAT on each subset. 3) Combine the base estimators in a gradient-boosted meta-model (XGBoost). This Ensemble GAT retains snapshot-specific knowledge while benefiting from the bias–variance reduction typical of

ensembles. Figure 16 illustrates the prediction and explainability process of the Ensemble GAT model.

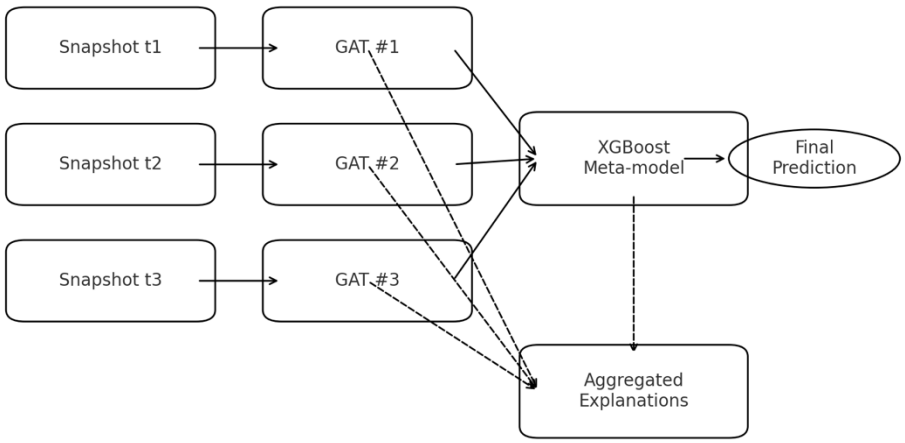


Figure 16: High-level overview of the Ensemble GAT model

The detailed description of the method can be found in [35].

6.3.2. Explaining the ensemble GAT model

The proposed explainability and root cause analysis method has two key components, answering two complementary questions:

- Ensembled GraphLime quantifies how each feature contributes,
- Neighbour Perturbation isolates which cells in the vicinity drive the prediction.

We tackle each question at the level of the base GATs and then fuse the explanations using the same XGBoost gains that are used to fuse the predictions.

To determine the feature importances, we apply GraphLime [144] that builds a Hilbert–Schmidt Independence Criterion Lasso (HSIC-Lasso) surrogate on the N-hop ego-subgraph of the target node, returning a coefficient vector $\beta^{(i)} = \left(\beta_1^{(i)}, \dots, \beta_k^{(i)}\right)$ for the i-th base GAT. Let g_i be that model's gain in the XGBoost combiner. The ensemble-level importance is then the weighted average $\beta_{Ens} = \sum_{i=1}^{n} g_i \beta^{(i)}$. Weighting by g_i (rather than using an unweighted mean) emphasizes the explanations of the more influential base models, yielding more stable attributions. We refer to this procedure as Ensembled GraphLime.

While Ensembled GraphLime tells us which features are essential, it cannot reveal which neighbors matter. To handle this orthogonal aspect, we propose the Neighbour Perturbation method illustrated in Figure 17.

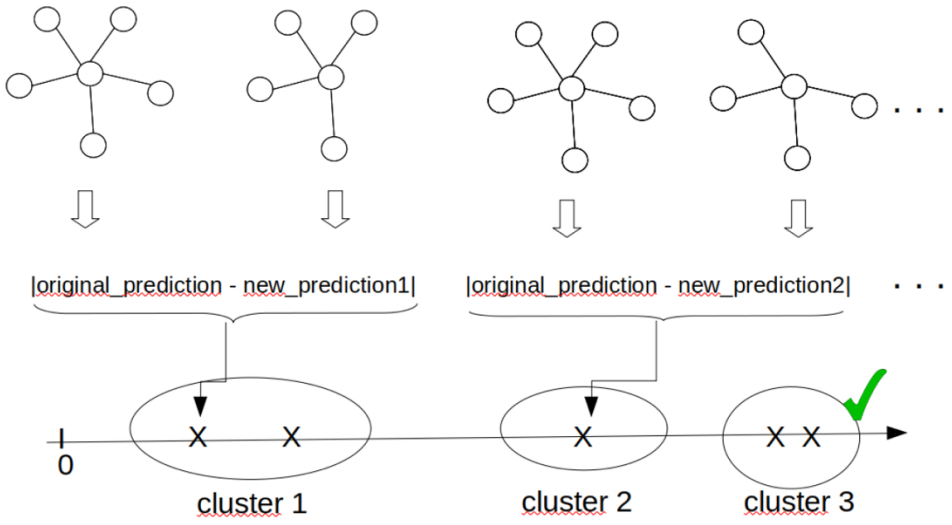


Figure 17: Main steps of the neighbour perturbation method

The method consists of the following main steps:

Edge deletion: For every neighbour v of the target node u, create a perturbed graph by removing an edge (u, v).

Prediction difference: Pass the perturbed graph through the model and record the prediction difference: $\Delta_v = \left| y_{full} - y_{perturbed}^{(v)} \right|$, where y_{full} is the prediction based on the complete graph and $y_{perturbed}^{(v)}$ is the prediction based on the perturbed graph, where the links to the neighbour v have been removed.

Clustering: Finally, we apply the clustering method X-means [82] to the vector $\{\Delta_v\}_{v \in neighbors(u)}$. Then the automatically selected cluster with the highest mean Δ is deemed the critical neighbor set. For the ensemble, we again compute $\Delta_v^{(i)}$ for every base GAT and aggregate them with the XGBoost gains: $\Delta_v^{Ens} = \sum_{i=1}^{n} g_i \Delta_v^{(i)}$. We then run clustering on $\{\Delta_v^{Ens}\}$ delivering a concise, gain-aware summary of the neighbour influence.

Graph-structured telemetry is not limited to radio access networks; the same idea applies to cybersecurity, where hosts (or IoT devices, user accounts, files, etc.) form nodes and their interactions (flows, authentications, API calls) form edges. In the next phase of the project, we will investigate how the Ensemble GAT model and its XAI approach can be applied to different security scenarios like intrusion detection and botnet/malware detection, in addition to KPI prediction.

6.4. Random Forest and XGBoost in FPGA context

Previous sections of this deliverable have discussed the main concepts related to the explainability of artificial intelligence (XAI), including approaches such as SHAP and LIME, which enable the transparent interpretation of model decisions. In the current project, artificial intelligence is applied at multiple levels, combining offline and online processing in demanding contexts, such as 5G networks.

In the first phase, AI is employed offline for model training using representative datasets such as CIC-DDoS201 [121], renowned for its diversity of attacks (SYN flooding, UDP flooding, DNS amplification, etc.) and its structuring by network flow. This phase facilitates the extraction and selection of relevant features using pre-processing techniques, including dimensionality reduction (PCA), variable encoding, and, in some cases, oversampling to balance classes.

The AI is then utilized online once the models are deployed on-site, as part of a detection-action mechanism (DetAction), to identify and neutralize malicious attacks in real-time. This phase imposes strict constraints in terms of latency, processing capacity, and hardware integration. Although artificial intelligence is a vast field, it would be a mistake to assume that increasingly complex models inherently guarantee superior performance. For example, exploratory techniques such as PCA, while effective at visualising models or reducing complexity, are ill-suited to production environments where every microsecond is critical. Similarly, specific deep learning models such as RNN or LSTM, although powerful on sequential data, have inference times and computing requirements that are incompatible with the constraints of embedded systems, particularly in an FPGA context.

In this context, several areas of AI research are currently being explored at the HES-SO, with a particular focus on lightweight, efficient, and explainable solutions that can be deployed in infrastructures with limited resources. The objective is clear: to combine detection accuracy, low energy consumption, and ease of hardware integration. With this in mind, we are developing a network traffic monitoring tool based on a PCIe SmartNIC equipped with an FPGA, positioned in parallel with the central server. This SmartNIC functions as an active probe, capable of analysing a duplicated traffic flow in real-time.

The solution incorporates two key components:

- A programmable P4 packet processing unit, enabling rapid, modular analysis of network headers.
- A module dedicated to AI model inference, responsible for detecting anomalies or suspicious behaviour online.

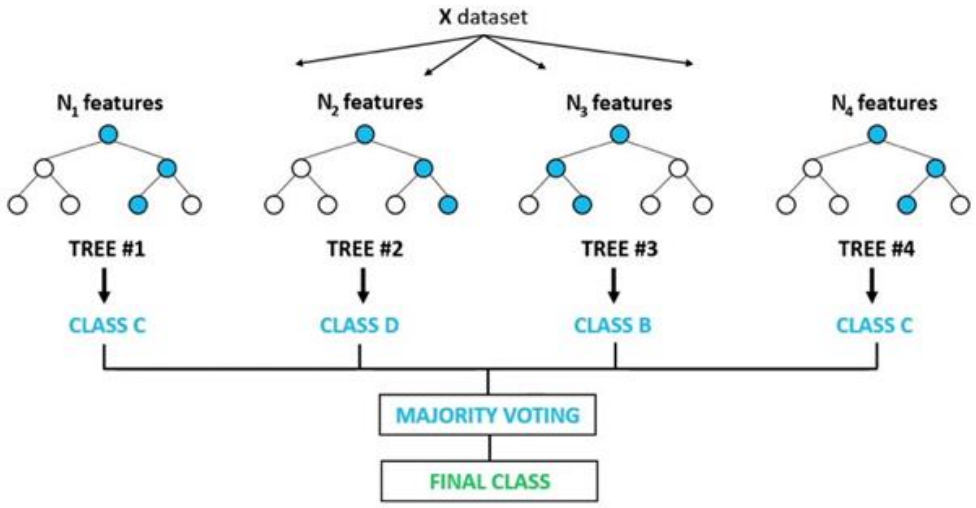


Figure 18 Random Forest Classifier tree

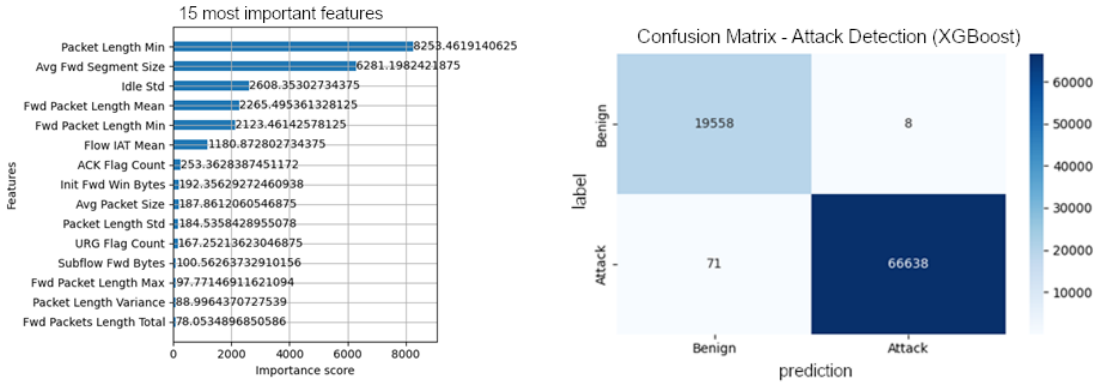


Figure 19 The left image shows the top 15 features used by the XGBoost classifier for attack detection, ranked by their importance scores. The left image presents the confusion matrix, illustrating the model's classification performance in distinguishing between benign and malicious network packets.

In this demanding context, where speed and responsiveness are crucial, our choice fell on the Random Forest model (cf. Figure 18), recognised for its real-time efficiency, small memory footprint, and ease of implementation in embedded architectures such as FPGAs. This model was trained on the CIC-DDoS2019 dataset, after transforming the PCAP files into flow data using CICFlowMeter, and cleaning up the features to reduce complexity while preserving relevance.

The tests showed that Random Forest met the system's requirements with an overall accuracy of 93%, although performance was lower on rare classes. To improve robustness, XGBoost was also evaluated. The latter model, which is known to handle unbalanced datasets more effectively, achieved 94% accuracy, with only 79 misclassified flows out of 86,275, while maintaining a relatively compact architecture (cf. Figure 18). By contrast, more complex models, such as deep neural networks (DNN, LSTM, etc.), despite their effectiveness in the laboratory, do not guarantee significant gains in real-life conditions and are too resource-intensive (in terms of computing time, memory, and energy consumption) to be integrated into an embedded

environment. Additionally, their limited explicability makes auditing and validation more challenging in a critical context. Random Forest and XGBoost, therefore, appear to be well-balanced solutions, offering good accuracy, satisfactory responsiveness, and easy integration into an embedded system dedicated to real-time traffic supervision. Both XGBoost and Random Forest provide a degree of explainability by default. They offer global feature importance measures and allow inspection of individual decision trees within the ensemble. This enables the understanding of which features most significantly influence the model's predictions overall. However, while they support some level of interpretation natively, neither model provides detailed per-sample explanations out of the box. For more granular, local explainability—such as understanding why a specific prediction was made—external tools like SHAP are commonly used and well-supported for both algorithms.

6.5. Explainable IDS via SHAP

The evolution of 5G and the emergence of 6G networks are enabling groundbreaking applications such as autonomous vehicles, smart manufacturing, and e-healthcare. These innovations are made possible by features like ultra-low latency, massive device connectivity, and significantly higher data rates. However, the rapid advancement of these technologies also introduces new and complex cybersecurity challenges. Threats such as Distributed Denial-of-Service (DDoS) attacks, Man-in-the-Middle (MITM) attacks, and Advanced Persistent Threats (APT) are becoming more sophisticated and severe. Artificial Intelligence (AI)-based Intrusion Detection Systems have shown great promise in bolstering network security by identifying and responding to these threats. Nevertheless, a major limitation of many AI-based systems is their lack of interpretability. This opacity raises critical concerns regarding trust, accountability, and regulatory compliance, especially in a high-stakes environment like a 5G network [84]. XAI addresses this issue by providing tools and techniques that make AI decision-making processes more transparent and understandable. XAI is particularly crucial in cybersecurity, where stakeholders must trust and audit the system's responses to potential threats. Techniques such as SHAP, LIME, and Grad-CAM (Gradient-weighted Class Activation Mapping) offer promising solutions by elucidating how AI models reach specific conclusions. We aim to investigate and integrate XAI methods into AI-driven intrusion detection systems tailored for the 5G environment. By enhancing not only the detection accuracy but also the interpretability of these systems, we can significantly improve real-time response, system transparency, and user trust in the face of evolving cyber threats.

The integration of SHAP into the intrusion detection pipeline not only demystifies the internal decision-making process of the model but also yields actionable insights that enhance the overall understanding of traffic behaviours in both legacy and modern 5G environments. Through local interpretability, SHAP enables per-sample analysis identifying which features, and in what magnitude, influenced the classification of an individual network flow as benign or malicious. For

instance, in the 5G-NIDD dataset [85][85], features such as Offset and Sum exhibited strong negative contributions to the classifier's output, effectively steering the prediction toward a benign label. Conversely, features like sHops were shown to positively contribute to malicious classification, albeit not strongly enough to override the dominant benign indicators. This type of granular insight allows security analysts to understand not only what the model predicted but why is an invaluable asset when validating alerts or conducting forensic analysis. At a broader scale, global SHAP summary plots offer a cumulative view of feature importance across the dataset, effectively revealing the most influential variables in shaping the model's overall behavior. In the case of the 5G-NIDD dataset, top predictors included sTtl (source Time to Live), State, and sMeanPktSz (source mean packet size). These features align well with known characteristics of legitimate 5G control-plane traffic, where a higher TTL value typically signifies longer, valid packet traversal. On the other hand, unusually low TTLs or irregular state transitions may signal anomalies, such as packet injection or spoofing attempts. In the CIC-IDS2017 dataset [86][86], which primarily contains IP-based traffic, features like Min Packet Length, ACK Flag Count, and Bwd Packet Length Min surfaced as highly indicative of malicious behavior. These findings highlight how explainability tools can adaptively distinguish attack patterns that are specific to different networking paradigms, be it traditional IP or emerging 5G protocols.

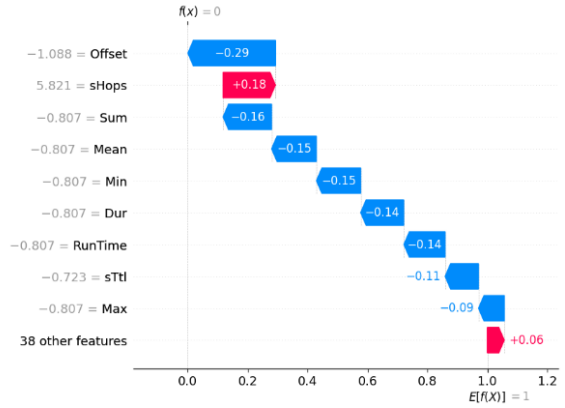
As Figure 20a and Figure 20b show, local explanations unveil how a single feature influences the classifier's prediction for sample flows. For the 5G-NIDD example (Figure 20a), features like Offset and Sum have a strong negative impact, pushing the prediction towards the benign class. While the sHops feature makes a positive contribution, it is not enough to counteract the negative contributions to produce a final classification of normal traffic. This result confirms the significance of timing and routing features in defining 5G control-plane traffic. By contrast, the CIC-IDS2017 instance (Figure 20b) displays a model of consistently positive contributions. Avenues such as Min Packet Length, Average Packet Size, and Bwd Packet Length Max all point prediction sharply in the attack-class direction, meaning payload-size abnormalities are critical in legacy IP traffic. The behavior of global models is exhibited by Figure 20c and Figure 20d, with SHAP summary plots that cumulate feature impacts across the initial 30 test-set flows.

Beyond model validation, the insights offered by SHAP explanations can directly inform system-level improvements. Feature importance metrics derived from SHAP can guide the selection or engineering of input features, reducing dimensionality while retaining high-informative attributes. Additionally, anomalous patterns repeatedly flagged by SHAP across samples may hint at previously unknown or under-documented threat signatures, prompting further investigation or updates to the threat detection policies. From a practical standpoint, the ability to visually communicate why a specific flow was classified as an attack builds trust with stakeholders and supports compliance with regulatory frameworks that demand transparency and accountability in automated decision systems. In this light, SHAP serves not only as a diagnostic tool but as a

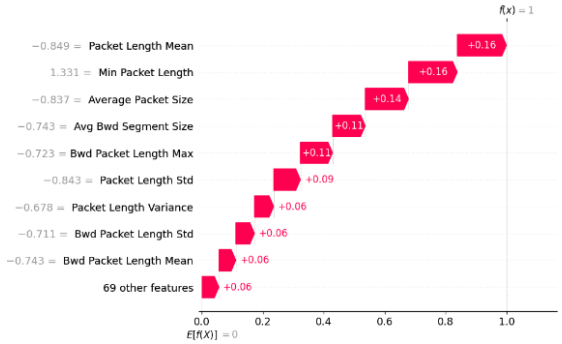
cornerstone for deploying intelligent, explainable, and human-aligned cybersecurity solutions in next-generation networks.

In the case of 5G-NIDD (Figure 20c), top predictors are sTtl, State, and sMeanPktSz. High values of sTtl will tend predictions towards benign, as would be expected with greater traversal by legitimate control packets, while unusually low TTLs or high hop counts indicate suspicious traffic behavior. In the CIC-IDS2017 dataset (Figure 20 d), the strongest features are Min Packet Length, ACK Flag Count, and Bwd Packet Length Min. High values in these statistics are strongly correlated with attack streams, i.e., port scanning or DoS, while normal traffic is described by small, regular packet lengths and normal TCP flag patterns.

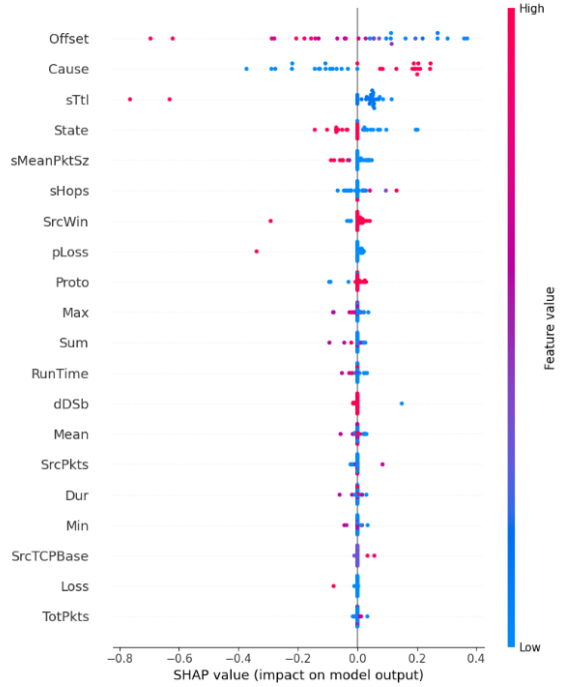
Beyond model validation, the insights offered by SHAP explanations can directly inform system-level improvements. Feature importance metrics derived from SHAP can guide the selection or engineering of input features, reducing dimensionality while retaining high-informative attributes. Additionally, anomalous patterns repeatedly flagged by SHAP across samples may hint at previously unknown or under-documented threat signatures, prompting further investigation or updates to the threat detection policies. From a practical standpoint, the ability to visually communicate why a specific flow was classified as an attack builds trust with stakeholders and supports compliance with regulatory frameworks that demand transparency and accountability in automated decision systems. In this light, SHAP serves not only as a diagnostic tool but as a cornerstone for deploying intelligent, explainable, and human-aligned cybersecurity solutions in next-generation networks.



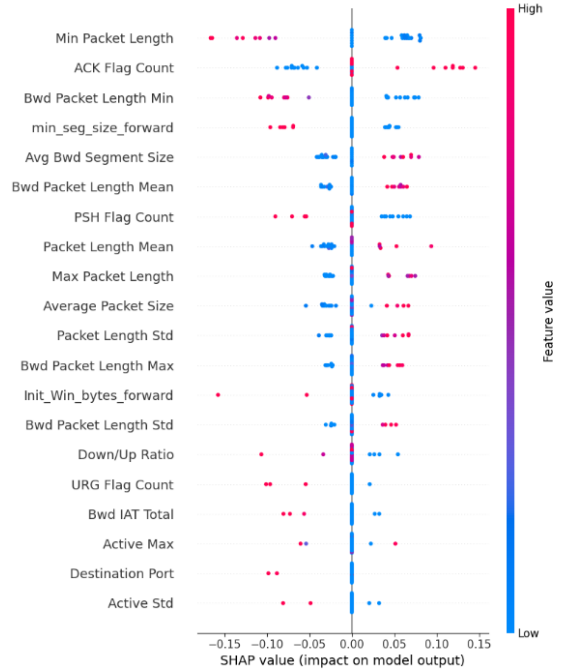
(a) Contribution of Features for 5G-NIDD



(b) Contribution of Features for CIC-IDS2017



(c) SHAP summary plot for the first 30 samples from the test set of the 5G-NIDD dataset. The selected samples include both benign and attack instances.



(d) SHAP summary plot for the first 30 samples from the test set of the CIC-IDS2017 dataset. The selected samples include both benign and attack instances.

Figure 20: SHAP Results

7. Cyber Threat Intelligence

The rapid evolution of 6G networks promises unprecedented levels of connectivity, intelligence, and automation. However, this technological advancement also introduces a wide array of new cybersecurity risks, fuelled by the scale, heterogeneity, and complexity of emerging infrastructures. In this context, Cyber Threat Intelligence (CTI) becomes a foundational element for the secure operation of 6G systems. CTI refers to the systematic collection, analysis, and dissemination of information about existing and emerging cyber threats—including malicious actors, their capabilities, and their attack strategies. When operationalized effectively, CTI enables organizations to anticipate, detect, and respond to threats in a timely and informed manner.

In traditional networks, CTI has often played a reactive role—focused on analysing incidents after they occur. However, the distributed, software-defined, and highly dynamic nature of 6G networks demands a proactive, automated, and context-aware CTI architecture. The convergence of telecommunications, cloud-native infrastructures, edge computing, and AI-driven services amplifies the attack surface and creates conditions where manual or static threat intelligence processes are no longer sufficient. This calls for CTI systems capable of operating autonomously at scale, adapting to constantly shifting threat environments, and integrating seamlessly with other network defence components.

Within the scope of the NATWORK project, CTI is positioned as a strategic enabler of self-resilient and self-adaptive network security. It serves as a critical input to several core capabilities envisioned in NATWORK, including intent-based orchestration, generative AI for system adaptation, and autonomous service resilience. By embedding CTI at the heart of the architecture, NATWORK aims to create an intelligence-driven framework in which threat information is not only collected and stored, but also continuously analysed, contextualized, and acted upon across the edge-cloud continuum.

The project's CTI framework is built on three foundational pillars:

- Multi-source threat data collection: Integrating diverse sources—including honeypots, darknet traffic, OSINT, social media, and intrusion detection systems—to build a broad and deep threat visibility layer.
- LLM-powered intelligence automation: Using generative AI to convert unstructured threat data into actionable intelligence, produce STIX-compliant bundles, and support automated reasoning and documentation.
- Edge-cloud observability and integration: Ensuring that infrastructure monitoring, telemetry, and behavioural analytics from Kubernetes-based 6G environments feed into the CTI pipeline, enabling real-time and context-rich insights.

These components work together to enhance the security posture of 6G networks by enabling continuous threat monitoring, early warning systems, and dynamic security policy enforcement. NATWORK's approach to CTI also contributes to broader cybersecurity objectives at the European level by addressing long-standing challenges such as the automation of threat report analysis, the standardization of threat knowledge using STIX, and the reduction of manual overhead in threat intelligence workflows.

This section presents the specific contributions of NATWORK to the field of CTI, ranging from architectural frameworks and threat data processing engines to generative AI-based CTI extraction and infrastructure monitoring. Together, these innovations lay the foundation for an advanced, scalable, and explainable CTI infrastructure tailored to the needs of secure and trustworthy 6G systems.

7.1. Multi-Source CTI Framework for Proactive Network Defense and LLM-Powered Intelligence

In today's rapidly evolving cyber landscape, organizations face persistent and sophisticated threats originating from a wide array of sources, including botnets, attackers, and malicious actors lurking in hidden corners of the internet. To address this challenge, we propose a Comprehensive Cyber Threat Intelligence (CTI) Solution that leverages a diverse set of threat detection mechanisms to ensure robust protection for individual users and enterprise networks alike.

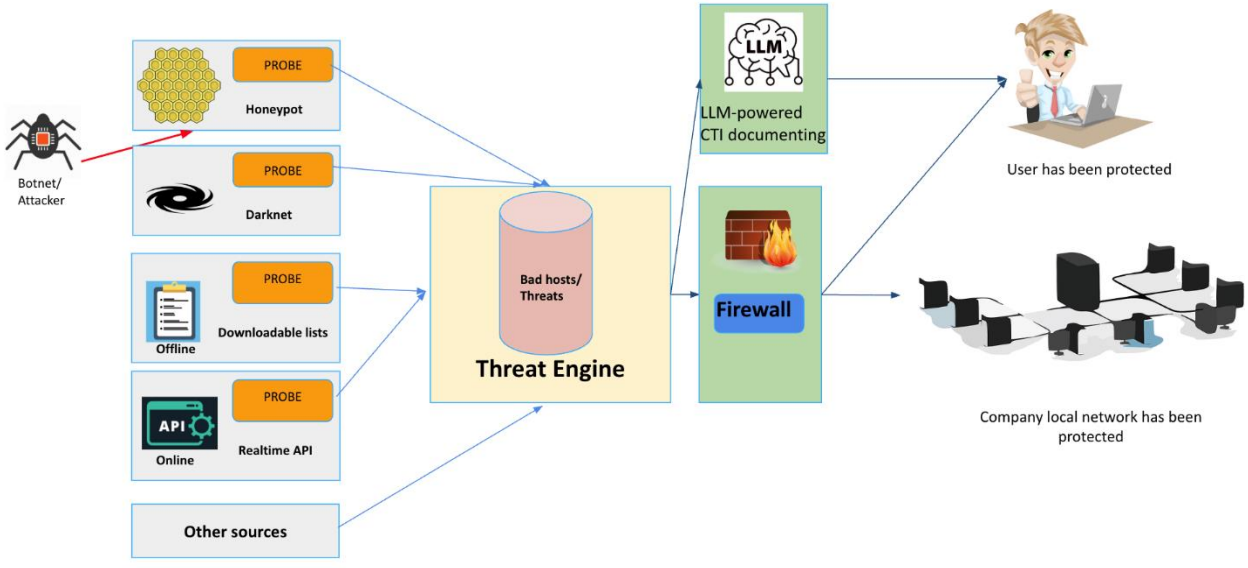


Figure 21: Overall architecture of CTI Framework for Proactive Network Defense and LLM-Powered Intelligence

The proposed architecture in Figure 21 provides a comprehensive solution for detecting, analysing, and mitigating cyber threats in real time. It integrates multiple Cyber Threat Intelligence (CTI) sources into a unified Threat Engine, which processes and prioritizes threat data to feed defence mechanisms like firewalls and an LLM-powered CTI documenting component. This layered defence strategy ensures that both individual users and enterprise networks are protected from malicious actors such as botnets and attackers. The architecture not only blocks threats but also generates actionable intelligence to improve security awareness and response capabilities across the organization.

7.1.1. CTI Collection

A variety of CTI sources feed threat information into the system. Figure 22 illustrates the different types of CTI sources, which we discuss in detail below.

- Honeypots: Decoy systems or networks are deliberately deployed to lure attackers and observe their behaviour. It can take various forms, such as a counterfeit website, a simulated server, or a virtual environment designed to resemble real systems, often configured with intentionally unpatched vulnerabilities to entice malicious actors. When attackers engage with a honeypot, they inadvertently reveal critical information about their tactics, techniques, and procedures (TTPs), offering valuable intelligence to defenders. Our threat intelligence framework employs over 20 distinct types of honeypots, each strategically deployed to capture a wide variety of malicious activities across different protocols and threat vectors. Among these are well-established and widely used honeypots such as Cowrie (SSH/Telnet interaction honeypot), Glastopf (web application honeypot designed to detect web attacks), Heralding (credential capturing honeypot for various services), and Dionaea (designed to capture malware).
- Darknet Probes: A darknet or network telescope refers to an Autonomous System Number (ASN), a segment of an ASN not allocated by IANA, or an otherwise unused portion of the IP address space. Any traffic directed to these addresses is generally unsolicited and typically malicious, often consisting of scanning attempts, DDoS backscatter, or other nefarious activities. As part of the H2020-SISSDEN project, MONT has collected approximately 1GB of sample darknet traffic, providing valuable insights into global malicious activities. In contrast, the dark web (sometimes also referred to as "darknet" in casual use) represents a hidden segment of the internet not indexed by conventional search engines and accessible only through specialized tools, such as the Tor browser [122]. This environment often serves as a marketplace for illicit trade and other illegal operations. Monitoring both darknet traffic and dark web activities can be highly useful for gaining intelligence on illegal behaviour, emerging threats, and attacker methodologies.

Figure 22: Examples of CTI sources

- Downloadable Lists (Offline Sources) and Realtime API (Online Sources): Offline sources typically consist of precompiled lists, curated and updated periodically by trusted security organizations. These lists include the most active attacking subnets and IP addresses, often identified through global traffic monitoring initiatives. They also provide specialized datasets focused on particular types of threats, such as botnet command-and-control servers, compromised hosts distributing malware, and known spam sources. The inclusion of curated drop and block lists from reputable entities further strengthens the CTI system's ability to detect and block malicious actors effectively. Whilst, real-time and near-real-time online feeds are crucial for maintaining an updated defence posture. Threat intelligence feeds generated by intrusion detection systems (IDS) and security monitoring platforms provide dynamic information about ongoing threats. For instance, platforms like the SANS Internet Storm Center [123] distribute feeds containing details on malware activities, spam campaigns, and network scanning attempts. Similarly, blacklists maintained by IDS solutions such as Suricata[124] offer valuable data on botnet-related communications, unsolicited network traffic, and emerging threats. Furthermore, community-driven databases, including resources like AbuseIPDB [125] and Onyphe

[126], contribute user-reported information on abusive IP addresses, further enriching the CTI ecosystem with relevant and timely indicators.

- Other Sources: These include Intrusion Detection Systems (IDS) such as MMT, Suricata, and Snort, which continuously analyse network traffic and report security issues or anomalies in real time. They inspect packets for suspicious patterns, malicious payloads, or policy violations. Unlike external CTI feeds that report on global threat activities, IDS solutions offer localized intelligence, detecting attacks as they unfold within the monitored network perimeter. MMT, Suricata, and Snort employ advanced detection techniques, including signature-based analysis, anomaly detection, and protocol decoding, to identify a wide range of threats. These may include malware infections, unauthorized access attempts, reconnaissance scans, and exploitation of vulnerabilities. The alerts and logs generated by IDS systems are then fed into the CTI framework, enriching it with highly relevant and contextual data about current threats targeting the specific environment.

These sources ensure the CTI system has both breadth and depth in threat visibility, capturing diverse and evolving threat indicators.

7.1.2. Processing CTI Reports in Threat Engine

Once ingested, the cyber threat intelligence (CTI) data is processed by the Threat Engine, a core component responsible for transforming raw information into actionable insights. This module performs several essential functions that enable the CTI system to operate efficiently and effectively. The first of these is aggregation, which involves collecting and consolidating threat data from a diverse array of sources. These may include honeypots, darknet probes, intrusion detection systems, and external threat feeds. By bringing together intelligence from various origins, the Threat Engine ensures comprehensive visibility across the threat landscape.

Following aggregation, the system undertakes correlation to identify meaningful relationships between seemingly disparate data points. This step is crucial for detecting patterns and linking indicators that may signify coordinated or distributed attack campaigns. Through correlation, the Threat Engine is able to move beyond isolated alerts and provide a more holistic understanding of complex threat activities, such as multi-stage attacks or widespread scanning efforts.

The next function is analysis, where advanced analytical techniques are applied to interpret the aggregated and correlated data. This process involves contextualizing the detected threats, assessing their potential relevance, and determining whether they pose a genuine risk to the protected environment. By examining factors such as the nature of the attack, historical behaviours, and known tactics, techniques, and procedures (TTPs), the Threat Engine can distinguish between benign anomalies and serious security incidents.

Finally, the system performs prioritization, which ranks the identified threats based on several critical criteria, including severity, likelihood of exploitation, and potential impact on organizational assets. This prioritization process is essential for optimizing incident response efforts, ensuring that security resources are focused on the most pressing and potentially damaging threats first.

Through these integrated processes, the Threat Engine plays a pivotal role in refining raw CTI into validated and relevant threat indicators. Only after undergoing aggregation, correlation, analysis, and prioritization are these indicators forwarded to enforcement mechanisms and intelligence-sharing components. This approach reduces unnecessary noise, enhances operational efficiency, and ensures that the organization's defensive posture is based on accurate and actionable intelligence.

7.1.3. Application

- **Firewall/ CTI Portal**

The processed and prioritized threat indicators are systematically fed into a Firewall, which acts as the primary enforcement mechanism within the security architecture. This firewall dynamically updates its ruleset based on the intelligence generated by the Threat Engine, allowing it to block identified malicious hosts and prevent harmful traffic from reaching end users and critical corporate assets. By proactively filtering threats at the network perimeter, the system ensures that both individual users and the company's local network remain safeguarded against a wide spectrum of known attacks, ranging from malware distribution and phishing attempts to command-and-control communications and reconnaissance scans.

Beyond its fundamental role in traffic filtering, the firewall component is integrated into a broader threat management platform offering advanced features designed to enhance usability and responsiveness. One key capability is the subscription model, which allows users to configure personalized alerting rules. Through this mechanism, subscribers can receive timely notifications related to specific IP addresses, Autonomous System Numbers (ASNs), or other relevant indicators of interest. This ensures that security teams and stakeholders remain informed about potential threats targeting their organization or critical infrastructure in near real time.

Another significant functionality is the platform's advanced search interface, which enables users to query the threat database with precision. Users can search for information related to IP addresses or prefixes, domain names, hostnames, URLs, ASNs, countries, organizations, or even upload files in CSV format containing multiple entries for batch processing. This powerful search capability provides flexible access to historical and current threat data, supporting incident investigations, threat hunting activities, and compliance requirements.

To facilitate seamless integration with external systems and automate security workflows, the platform also offers a comprehensive Application Programming Interface (API). Through this API, users can programmatically retrieve threat intelligence, submit queries, and receive updates, which significantly enhances the scalability and adaptability of the solution across various organizational environments.

Moreover, the system delivers real-time notifications to alert subscribers when activities associated with specific IPs or ASNs are detected. This ensures that defenders are immediately informed of emerging threats and can respond without delay. To support informed decision-making, each reported entity is assigned a reputation score, which is calculated by aggregating intelligence from multiple independent sources. This scoring mechanism provides valuable context, helping users to assess the risk level associated with a given indicator and prioritize their mitigation efforts accordingly.

Through this comprehensive set of capabilities, the firewall and its supporting platform not only offer automated threat blocking but also serve as an interactive and intelligent interface for managing and responding to evolving cyber threats in real time.

- **LLM-powered CTI Documenting**

In parallel to threat detection and enforcement mechanisms, the curated threat data is leveraged by the LLM-powered CTI documenting component, which plays a critical role in transforming raw and often complex threat indicators into actionable and comprehensible intelligence. This advanced module harnesses the capabilities of large language models to automatically interpret technical details and produce clear, human-readable reports. These reports are tailored to meet the needs of various stakeholders, including technical teams who require in-depth analysis, management who benefit from executive summaries, and broader security communities seeking situational awareness.

In addition to reporting, the component serves an essential function in supporting cybersecurity training and preparedness. By summarizing both ongoing and historical threat activities, it provides valuable input for cyber ranges, which are controlled environments designed to simulate real-world cyberattacks. These simulations are used to train cybersecurity professionals, helping them to develop and refine their defensive skills in response to realistic scenarios derived from actual threat intelligence.

Furthermore, the LLM-powered system enables intuitive interaction through natural language querying. This feature allows security analysts and incident responders to obtain contextual explanations and insights on demand, improving decision-making and reducing the time required to understand complex threat landscapes. By offering accessible and context-rich intelligence, this LLM-driven layer effectively bridges the gap between technical threat data and the diverse

information needs of its users, facilitating clear and efficient communication across all levels of an organization.

7.2. Advanced generative AI powered CTI data collection for 6G Networks

A substantial volume of valuable Cyber Threat Intelligence (CTI) is disseminated in unstructured formats. These include open-source intelligence (OSINT), social media posts, dark web forums, industry whitepapers, news reports, government-issued threat bulletins, and detailed incident response documentation. While these sources are rich in context and threat-related information, their unstructured nature presents significant challenges for efficient storage, classification, and automated analysis. The lack of standardized formatting prevents direct ingestion by security systems and forces human analysts to painstakingly read and interpret lengthy texts, which is both time-consuming and error prone.

As a result, one of the core tasks of security analysts is to manually extract relevant intelligence from these heterogeneous data sources and convert it into a structured format that can support automated correlation, querying, reasoning, and integration with existing threat detection or response systems. This manual translation process, however, is increasingly unsustainable in modern threat environments characterized by high-volume, high-velocity data and the growing complexity of multi-vector attacks—especially in highly dynamic environments such as 6G networks.

To address this challenge, the cybersecurity community has increasingly turned toward structured representation formats, such as the **Structured Threat Information eXpression (STIX)** standard [127]. STIX has emerged as one of the most widely adopted standards for representing CTI in a machine-readable form. In the STIX framework, each individual report—referred to as a *bundle*—is modelled as a **knowledge graph**, comprising interconnected entities and their relationships. These entities encapsulate key concepts such as threat actors, malware samples, system vulnerabilities, and tactics or techniques, while relationships express interactions between them (e.g., a threat actor *uses* a specific piece of malware or *targets* a particular organization).

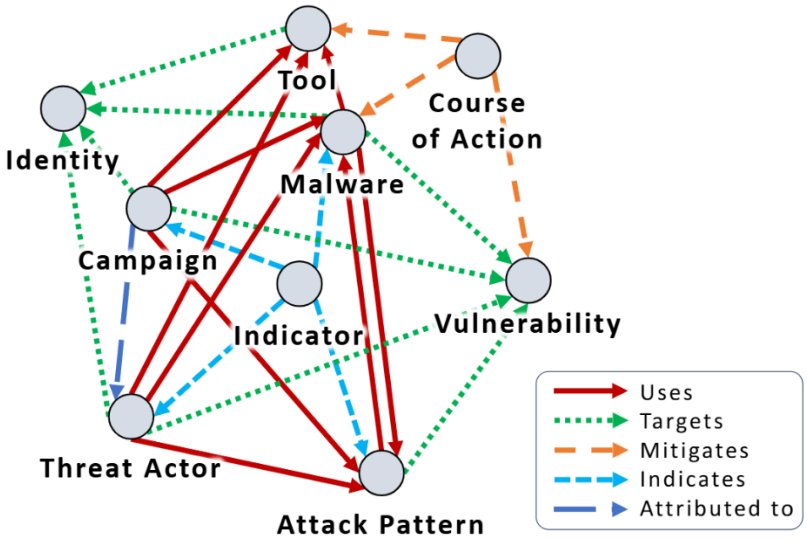


Figure 23: A subset of the STIX ontology, including all entities

Figure 23 illustrates a subset of the STIX ontology as applied to our dataset, capturing the core entities and relationships that appear in real-world CTI records. Among the primary entity types included in the ontology are:

- **Threat Actor**: Individuals or groups responsible for cyber attacks.
- **Malware**: Malicious software or code used to carry out attacks.
- **Vulnerability**: Known weaknesses in software, hardware, or configurations that attackers can exploit.
- **Attack Pattern**: The method or strategy employed during the attack.
- **Indicator**: Observables or signals that point to malicious activity (e.g., IP addresses, file hashes, domain names).

The ontology also supports semantic relations between entities, such as uses, targets, exploits, and attributed-to, enabling a detailed and contextualized representation of threat intelligence.

To demonstrate how analysts extract STIX-compliant bundles from unstructured reports, we present in the following subsection a representative example and outline the core extraction tasks typically performed. In particular, analysts focus on identifying and structuring the most commonly reported aspects of an incident:

- **Who** conducted the attack (e.g., the Threat Actor entity),
- **Against whom** the attack was carried out (e.g., the Identity entity, linked through a targets relationship),
- **How** the attack was executed (e.g., via Malware and Attack Pattern entities).

This subset of the STIX ontology covers the majority of critical information found in practical CTI reports. For instance, in our dataset, 75% of reports contain at least one Malware entity, while 54% reference a Threat Actor. This selection of entity types is also consistently supported across state-of-the-art information extraction tools and previous work, providing a stable foundation for benchmarking, evaluation, and integration efforts.

As we explore in the next part of this section, the advent of generative AI—particularly large language models (LLMs) and multi-agent frameworks—holds immense potential for automating the extraction of these entities and relations directly from unstructured CTI sources. By leveraging these technologies, the NATWORK architecture aims to close the gap between high-volume raw threat intelligence and actionable, structured insights tailored for self-resilient 6G environments.

7.2.1. Structured CTI Extraction

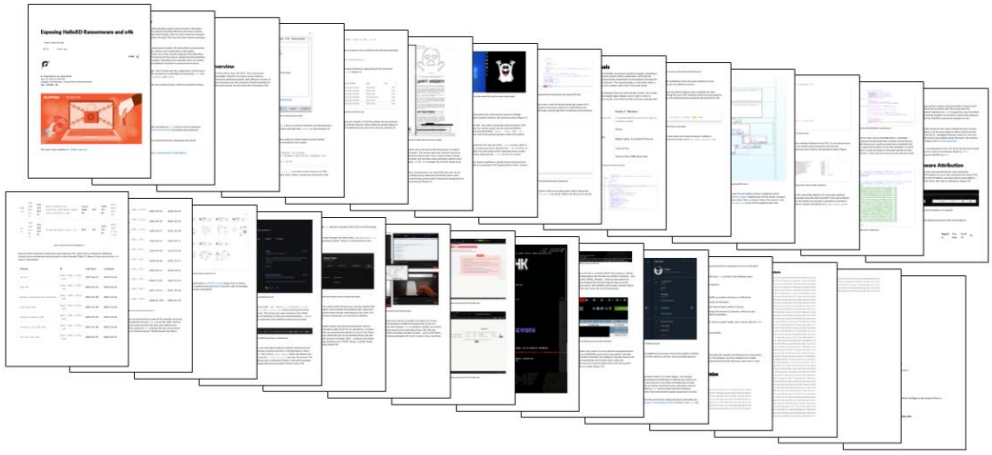


Figure 24: An example of a report published by Palo Alto Networks.

(While Indicators of Compromise are easy to extract being collected at the end of the report, extracting Threat Actor, Malware, Attack Pattern and the other STIX's entities requires security experts to perform manual analysis)

To concretely illustrate the task of **structured CTI extraction**, we examine a technical blog post published by Palo Alto Networks on the *HelloXD* ransomware campaign [128]. A snapshot of this report is shown in Figure 24. Like many industry-grade threat intelligence reports, it presents a dense and information-rich narrative: approximately 3,700 words, 24 figures, three detailed tables, and a dedicated section summarizing Indicators of Compromise (IoCs).

The report discusses the attribution of the *HelloXD* ransomware to a threat actor known as *x4k* and outlines the tactics, techniques, and procedures (TTPs) observed in related campaigns. It provides an in-depth analysis of the malware's functionality, explores its behavioural indicators, and presents several clues linking the malware to the threat actor. It also details aspects of the adversary's infrastructure and operational approach.

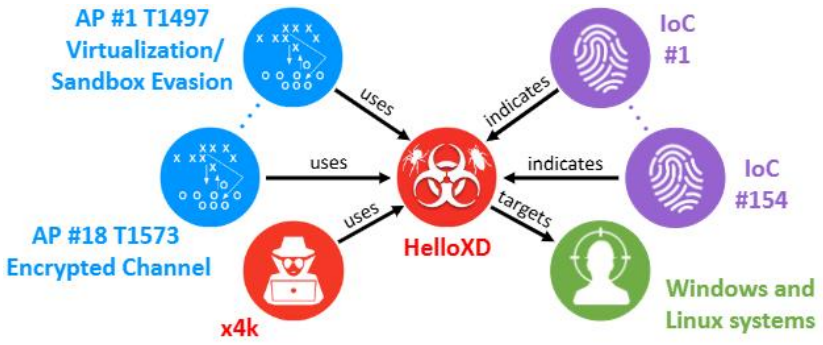


Figure 25: A STIX bundle describing the report from previous figure.

The aim of structured CTI extraction is to convert such a report into a **STIX-compliant bundle**, suitable for automated analysis and integration into cyber defence systems. Figure 25 depicts the resulting STIX bundle derived from this report. It includes key entities such as a Threat Actor node for *x4k*, a Malware node for *HelloXD*, multiple Attack Pattern entities describing the adversary's TTPs, and several Indicator entities from the IoC section of the report.

Generating this structured representation is far from trivial. Producing a complete and accurate STIX bundle typically requires between three and ten hours of dedicated analysis by experienced CTI professionals. This is supported by previous studies; for instance, Park et al. [129] report that annotating just 133 reports required three full-time annotators working over a five-month period. Similarly, the annotation of the 204 reports used in our evaluation took several months of sustained work by our team of CTI analysts.

One reason for this high annotation cost is the nuanced and implicit nature of much of the information contained in CTI reports. Even basic tasks—such as identifying malware names or linking threat actors to attack campaigns—require careful interpretation and contextual reasoning. The challenges include ambiguities in terminology, the use of aliases, and uncertainty in attribution.

A typical first step in the extraction process involves identifying the malware involved, the responsible threat actor, and any targeted identities. This may appear straightforward, but CTI reports often describe threats in subtle, context-dependent ways. For instance, entities may share names (e.g., a threat actor and a malware family both named similarly), or a single malware may be referenced using multiple aliases. Attribution is also frequently qualified or speculative, requiring analysts to distinguish between confirmed and hypothetical associations.

In our example report, *HelloXD* is clearly the central malware being described. However, the text also mentions other ransomware families—*LockBit 2.0* and *Babuk/Babyk*—which are not part of the *HelloXD* campaign. Their inclusion in the report is purely illustrative, used to draw comparisons or highlight common tactics.

Consider the following excerpt:

```
The ransom note also instructs victims to download Tox and provides
a Tox Chat ID to reach the threat actor. Tox is a peer-to-peer
instant messaging protocol that offers end-to-end encryption and
has been observed being used by other ransomware groups for
negotiations. For example, LockBit 2.0 leverages Tox Chat for
threat actor communications.
```

Although *LockBit 2.0* is referenced here, it is not directly connected to the *HelloXD* ransomware or the actor *x4k*. As such, it should not be included in the corresponding STIX bundle. This type of disambiguation is critical to ensure accurate modelling of the threat landscape and to avoid polluting CTI databases with unrelated or tangential information.

These examples highlight why manual annotation is so time-consuming and why fully automating this process remains a challenge. However, advances in natural language processing—particularly in generative AI and large language models—offer promising capabilities to support or accelerate this task.

A second critical step in constructing structured CTI is the identification of **attack patterns**—that is, the tactics, techniques, and procedures (TTPs) employed by the threat actor during the execution of the attack. This process introduces another layer of complexity: unlike discrete entities such as malware names or indicators, attack patterns are typically **descriptive behaviours**, often embedded across multiple paragraphs within a report.

These behaviours must not only be detected but also classified according to standardized taxonomies, such as the **MITRE ATT&CK® Matrix** [130], which is widely adopted for mapping adversarial behaviour. The ATT&CK framework includes over 190 techniques and more than 400 sub-techniques, covering a broad spectrum of activities across different stages of an attack lifecycle—from initial access and execution to exfiltration and impact. As a result, mapping natural language descriptions from threat reports to specific MITRE techniques requires both deep reading comprehension and extensive domain knowledge.

For example, the following excerpt, taken from a report by Proofpoint [131], illustrates how an attack pattern may be embedded in narrative text:

```
TA416 has updated the payload by changing both its encoding method
and expanding the payload's configuration capabilities.
```

An analyst must first detect this behaviour, and then correctly map it to a relevant MITRE technique. In this case, the appropriate mapping is **T1027: Obfuscated Files or Information**, described in the MITRE ATT&CK framework as:

```
Adversaries may attempt to make an executable or file difficult to
discover or analyze by encrypting, encoding, or otherwise
obfuscating its contents on the system or in transit.
```

This example demonstrates the layered reasoning required in structured CTI extraction: identifying the behavioural action, resolving its technical implications, and linking it to an appropriate standardized concept. Errors in this process can lead to incorrect or incomplete STIX bundles, weakening the ability to correlate CTI across sources or automate detection.

In the case of our *HelloXD* report, the analysis resulted in the extraction of 18 distinct attack patterns, each requiring close reading and interpretation. Some were explicit (e.g., the use of a peer-to-peer communication channel for ransom negotiation), while others required inferring intent or behaviour from context. Notably, this stage is particularly challenging to automate with classical rule-based systems or shallow machine learning approaches, making it a promising application area for generative AI systems capable of deeper semantic understanding.

A further consideration during the extraction process is the **relevance** of the information. Analysts must make expert decisions about which elements to include in the final bundle and which to omit. This requires evaluating not only the technical accuracy of the information, but also its **salience to the core narrative** of the report and the **confidence** with which the information is presented.

For instance, the *HelloXD* report includes tangential references to other activities associated with the threat actor *x4k*, such as the deployment of **Cobalt Strike Beacon** and the development of custom **Kali Linux distributions**. While potentially interesting, these activities are not discussed in detail and are not central to the campaign under analysis. Therefore, they are omitted from the final STIX bundle to maintain a clear and focused representation.

This judgment-based filtering is essential to ensure that the resulting structured intelligence remains precise, actionable, and free from noise. However, it further contributes to the time and expertise required for high-quality CTI annotation.

7.2.2. Existing solutions

Given the significant complexity and time investment required for manual structured CTI extraction, a range of automated solutions has been proposed in recent years [132][133][134][135][136].These efforts span from narrowly focused systems targeting specific subtasks—such as the identification of attack patterns or indicators—to more ambitious approaches that aim to automate the entire pipeline of CTI extraction from unstructured sources. Despite these advances, the practical utility of such tools remains limited: most still require considerable human oversight and post-processing to produce high-quality, actionable threat intelligence.

This gap between theoretical capabilities and practical usability is reinforced by our own experience. The empirical results from our CTI analyst team confirm that none of the existing tools offer a fully reliable or scalable solution, particularly when applied to realistic, high-variability datasets. The persistence of this challenge suggests that the limitations are not purely technical, but also methodological.

One contributing factor is the absence of robust benchmarks that truly reflect the nature and complexity of the structured CTI extraction task. Many existing solutions rely on machine learning techniques—particularly from the natural language processing (NLP) domain—but are evaluated using **standard NLP metrics** that do not align with the specific requirements and goals of CTI practitioners.

To illustrate this misalignment, consider the task of **Named Entity Recognition (NER)**. In the NLP field, a model is typically evaluated based on its ability to correctly identify every mention of an entity in the text. For example, if the malware *HelloXD* is mentioned ten times in a report and correctly labelled on each occasion, a word-level (or more precisely, token-level) evaluation metric would count this as ten correct outputs. This leads to what we refer to as **word-level labelling**, which may overstate a system's performance when applied to the CTI domain.

From a CTI perspective, however, the goal is to extract the **unique entities** that are relevant to the security event being described—regardless of how many times they are mentioned. Whether *HelloXD* appears once or ten times, it constitutes a single relevant malware entity in the context of a structured STIX bundle. Furthermore, as shown in our earlier example involving *LockBit 2.0*, not all entities identified by a generic NER tool are contextually relevant for structured CTI. A named entity may be correctly labelled from an NLP standpoint but should be excluded from a CTI bundle if it is not directly connected to the campaign or threat being analysed.

These discrepancies become even more pronounced in the evaluation of **Attack Pattern extraction**. Most current approaches rely on sentence-level classification tasks, where the goal is to determine whether a given sentence contains an attack pattern and, if so, assign it to the appropriate category. This **sentence-level labelling** strategy, while useful for training classifiers, does not capture the full complexity of real-world CTI extraction. In practice, what matters is the ability to identify all **relevant** attack patterns scattered across a document, accurately classify them according to taxonomies like MITRE ATT&CK, and correctly attribute them to the associated entities (e.g., malware or threat actor).

In essence, NLP-derived metrics often assess **syntactic accuracy**, while structured CTI extraction demands **semantic relevance** and **contextual correctness**. The failure to differentiate between these two levels of evaluation risks inflating perceived system performance and conceals the true limitations of current approaches.

This misalignment highlights a broader need for domain-specific metrics—what we refer to as **CTI-metrics**—which evaluate a system's ability to reconstruct accurate, coherent, and relevant threat intelligence bundles from unstructured inputs. These metrics must account not just for precision and recall at the sentence or token level, but also for the correctness and relevance of the resulting structured knowledge graph.

Table 2: Overview of manually annotated CTI datasets.

(In some cases, the annotated reports represent a small, labelled subset of a much larger corpus (total size in parentheses). Asterisks (*) indicate that annotations are made at the sentence level rather than across entire reports. A checkmark (✓) in the 'Public' column denotes datasets that are only partially released as open-source.)

Dataset	Entities & Relations	Attack Patterns	CTI Metrics	Public
SecIE	133	133	–	
CASIE	1k	–	–	✓
ThreatKG	141 (149k)	141 (149k)	–	
LADDER	150 (12k)	150 (12k)	5	(✓)
SecBERT	–	14.4k*	6	✓
TRAM	–	1.5k*	–	✓
TTPDrill	–	80 (17k)	80	
AttacKG	–	16 (1.5k)	16	
rcATT	–	1.5k	–	✓

Table 2 summarizes a selection of datasets used in prior work to evaluate structured CTI extraction methods. These datasets vary in scope, granularity, and coverage, particularly with respect to the annotation of complex entity types such as Attack Pattern. For this reason, we distinguish attack pattern extraction from simpler entity types (e.g., Malware, Threat Actor, Identity), given its higher semantic complexity and the additional requirement of mapping behavioural descriptions to formal taxonomies such as MITRE ATT&CK.

Several studies report the use of large datasets to evaluate their models using conventional **natural language processing (NLP) metrics**, such as the frequency and accuracy of extracted entity mentions. However, these large corpora are often only partially annotated and are typically accompanied by much smaller manually labelled subsets when evaluations are performed using **CTI-specific metrics**. The principal reason cited for the limited size of these subsets is the high cost associated with manual annotation, which requires expert input from trained CTI analysts. To contextualize the limitations of current datasets, we analyse them through two evaluation frameworks:

NLP-metrics. Datasets such as those used in SecIE[137], ThreatKG [135], and LADDER [134] provide word-level or sentence-level annotations, making them suitable for traditional NLP evaluation tasks like NER or sentence classification. Similarly, CASIE [132] offers a large corpus

annotated at the word level, though it does not include attack pattern annotations. The TRAM [147] and rcATT [136] datasets are focused on attack patterns but are also limited to sentence-level labelling, which does not support full structured extraction workflows.

CTI-metrics. Only a handful of works provide annotations suitable for evaluating CTI extraction methods from an operational, graph-based perspective. TTPDrill [133] and AttacKG [94] both include manually labelled datasets of 80 and 16 full reports, respectively, and adopt a CTI-centric evaluation approach. However, neither dataset is publicly released, and both are restricted to attack pattern extraction. SecBERT [135] performs a two-stage evaluation: it first assesses model performance on a large sentence-level dataset, then conducts a CTI-metrics evaluation on only six annotated reports. Likewise, LADDER [134] includes an evaluation of attack pattern extraction using CTI metrics on five reports, but these are also not shared publicly.

7.2.3. Creating a new dataset

The absence of a sufficiently large, open-access dataset specifically designed for structured CTI extraction has long hindered the ability to evaluate and compare existing approaches in a consistent and meaningful way. Without such a benchmark, it becomes difficult to assess progress beyond surface-level NLP metrics or to ensure that proposed methods are applicable in real-world CTI workflows.

To address this limitation, we rely on a dataset that was previously created by our team, consisting of 204 manually annotated CTI reports collected over a 12-month period starting in February 2022. Each report has been paired with a corresponding STIX-compliant bundle, meticulously constructed by expert CTI analysts to capture the relevant entities, relationships, and attack patterns described in the original text. This dataset, developed independently prior to the NATWORK project, is designed specifically to support evaluation based on CTI metrics, offering a realistic and high-quality benchmark for structured threat intelligence extraction. Its use in this work enables a more rigorous and operationally relevant assessment of generative AI-based approaches for CTI automation.

The remainder of this section provides information about the dataset creation methodology and introduces high-level statistics about the data.

7.2.3.1. Methodology

Our organization includes a dedicated team of Cyber Threat Intelligence (CTI) analysts whose primary responsibility is the structured extraction of threat intelligence from publicly available sources. Leveraging their expertise and established internal methodology, we use a dataset previously created by this team, comprising **204 manually annotated CTI reports**, each paired with its corresponding **STIX bundle**. These reports, collected over a 12-month period starting in

February 2022, originate from well-known and reputable sources (cf. Section 7.2.3.2) and have been further reviewed to ensure classification quality.

Structured CTI extraction in this context is performed manually, following a rigorous multi-step process, carried out by **three independent analyst groups**, each with clearly defined roles:

Group A is responsible for selecting the reports to be processed. These reports are chosen based on analyst expertise and awareness of global threat trends. This group typically consists of two to four people and focuses on ensuring the relevance and diversity of the collected intelligence.

Group B carries out the core extraction process. Their workflow involves several sequential steps:

1. **Initial Parsing**: The selected report is processed using custom-built parsers to extract raw text from the original web source. These parsers are developed in-house by the CTI team to handle new formats or sources as needed.
2. **Text Segmentation**: The extracted text is segmented into blocks of sentences, which are manually refined - analysts may merge, split, or discard segments to improve clarity and focus.
3. **Entity Extraction**: Pre-labelling is performed using automated Named Entity Recognition (NER) tools. Analysts then manually verify and adjust these labels, adding or removing entities based on contextual relevance. Automated tools speed up the process by visually highlighting candidate entities, allowing analysts to focus on semantic validation.
4. **Attack Pattern Extraction**: The same text blocks are processed using a logistic regression model from TRAM [147], which flags likely attack-pattern-related sentences. These are quickly verified by the analyst, while remaining ambiguous or uncovered text is reviewed manually. This tiered approach reduces the volume of text requiring deep manual analysis and helps disambiguate edge cases.
5. **STIX Bundle Assembly**: Using a custom-built graphical interface, the analyst assembles the final STIX bundle. This includes not only verified entities but also correct attribution through STIX relationships (e.g., uses, targets).

Group B consists of two analysts who alternate roles, each working on different reports.

Group C independently reviews the STIX bundles produced by Group B. This group inspects the intermediate steps and either accepts or requests revisions to the submitted bundle. Group C also consists of two analysts, and all members of Groups B and C rotate roles between analyst and reviewer across different reports to ensure impartiality and reduce annotation bias.

This entire process is supported by a web-based software infrastructure specifically developed to streamline structured CTI annotation. Analysts access a unified toolchain via this platform,

which tracks their interactions, roles (e.g., analyst vs reviewer), and time spent on each stage of the process. For the dataset used in this study, the average time required to complete the full structured CTI extraction for a single report (excluding selection by Group A) was approximately **4.5 hours**, with **Group B responsible for the majority of the workload (~3 hours per report)**.

To further ensure annotation quality and consistency, an additional validation phase was conducted. A team of two independent researchers reviewed a subset of the 204 reports previously processed by Groups B and C. They re-labelled the selected reports from scratch and compared results with the existing annotations. This process confirmed unanimous agreement across both analyst teams and researchers. During validation, researchers accessed the original web sources directly—bypassing the automated parsers used during Group B's initial processing—to eliminate even the slightest possibility of parser-induced errors.

This multi-tiered approach ensures that the resulting dataset not only reflects real-world intelligence extraction practices, but also meets a high standard of annotation quality, providing a robust foundation for evaluating structured CTI extraction techniques based on CTI-specific metrics.

7.2.3.2. Dataset Summary

The dataset used in this work comprises **204 CTI reports**, each manually annotated with a corresponding **STIX bundle**. The reports are sourced from **62 well-known public entities**, including organizations such as **Palo Alto Networks** [148], **Trend Micro** [149], and **Fortinet** [150]. On average, each source contributes 3.3 reports (see Table 3). Importantly, approximately **79%** of the sources are referenced by the **MITRE ATT&CK® framework** as external citations, confirming the representativeness and relevance of the selected materials within the global CTI landscape.

Table 3: 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 4. 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 4: 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%

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** [153], 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 23, covering **9 STIX entity types** and **5 types of relations**, and providing a structured foundation for knowledge graph-based CTI processing.

Table 5 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 5: 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%
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%

Metric / Type	Min	Avg	95p	Max	Quota
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 5 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.

7.3. Edge-Cloud Infrastructure Monitoring for CTI

Monitoring and observability are crucial for ensuring the resilience and security of distributed, containerized systems, particularly in edge-cloud environments that support network slices. In recent years, the shift toward cloud-native architectures—including microservices, containers, and orchestration platforms like Kubernetes—has brought significant advantages in scalability and flexibility. However, these architectures also introduce complexity, especially when deployed across geographically distributed and highly dynamic edge-cloud infrastructures. This complexity makes it more challenging to understand system behaviour and detect faults or threats in real time. Traditional monitoring tools designed for monolithic or centralized systems fall short in this context, such as Zabbix and Nagios [154] [155].

To address this challenge, as part of our contribution, we focus on developing and maintaining an infrastructure for monitoring the health and behaviour of Kubernetes-based clusters and the microservices within these clusters, supporting Cyber Threat Intelligence (CTI) and AI frameworks. Our primary objective is to monitor workloads over Software-Defined Networking (SDN) and to observe how microservices behave in virtualized ecosystems under different conditions. To this end, we use Prometheus, a widely used open-source monitoring tool designed for cloud-native systems [155][156]. Prometheus is integrated with our experimental testbeds to collect metrics from various layers of the infrastructure. These include data on CPU and memory consumption, disk I/O, container lifecycle events, pod-level statistics, bandwidth and latency, and error rates across services and interfaces.

To make the collected metrics more accessible and actionable, we integrate Prometheus with Grafana, a powerful open-source analytics and visualization platform. By configuring Prometheus as a data source in Grafana [135], we can query, visualize, alert, and explore key metrics from across our experimental VM and container testbeds. This integration significantly enhances operational insight by providing interactive dashboards that display metrics in real time and

across historical windows. As shown in Figure 26 and Figure 27, the system is designed to monitor the overall health of the cluster as well as focus on service-level granularity.

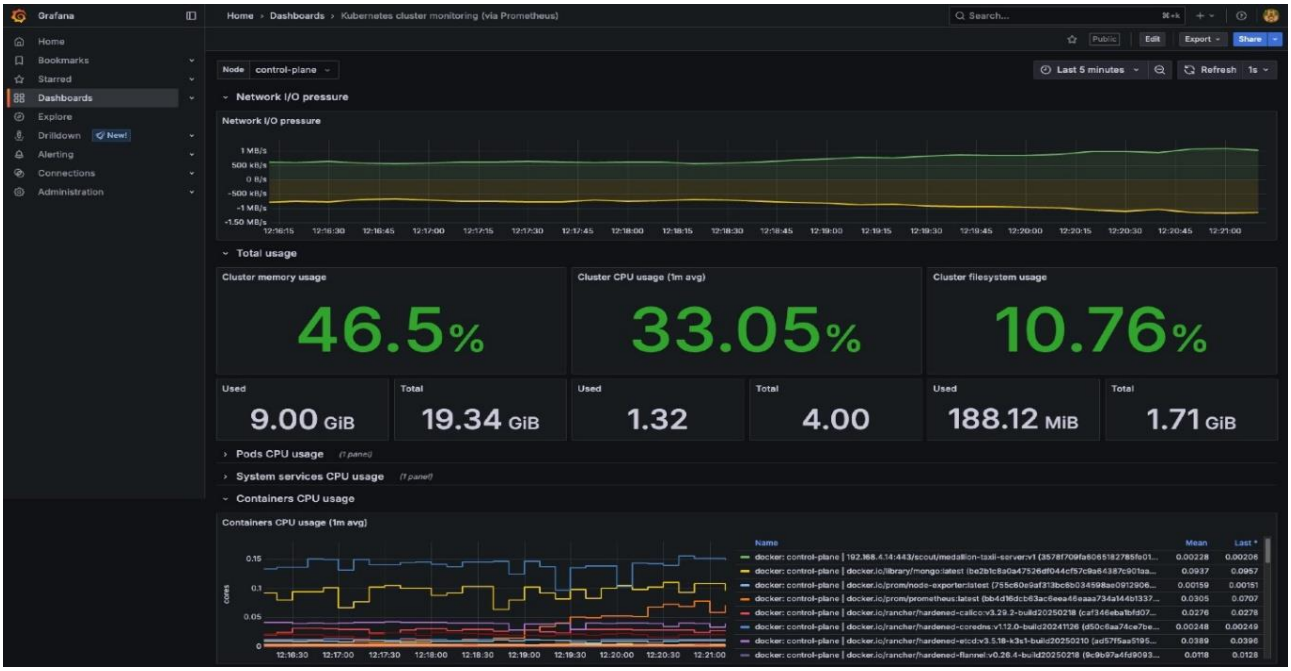


Figure 26: Cluster-Level Monitoring

Figure 27: Service-Level Monitoring

At the container level, we can monitor fine-grained performance metrics for individual microservices. At the node level—including both control plane and worker nodes—we track broader metrics, such as CPU load averages, memory usage, filesystem pressure, etc. We also attempt to understand how the K8s scheduler on the control plane responds under unstable load conditions, where oscillating request patterns cause rapid and repeated scaling decisions. These fluctuations lead to increased control plane activity, driving up CPU usage and energy consumption, affecting the service sustainability without any actual service failure. The term for this kind of attack is called Denial of Sustainability, wherein rapid and repetitive pod scaling in response to fluctuating workloads leads to increased energy consumption and operational costs, degrading the Quality of Service [158].

Furthermore, we focus on the structured generation of time-series datasets that can feed AI-based threat detection and CTI systems. Prometheus serves as the backbone for collecting operational telemetry, enabling us to create labelled datasets that capture both routine and anomalous system behaviours under varied workloads.

This work complements the broader architecture by ensuring the availability of accurate, time-aligned, and context-rich observability data. This data serves long-term analytics and modelling required by CTI frameworks and AI systems in 6G environments.

8. Conclusions

Deliverable D4.3 has presented the foundational contributions of the NATWORK project in the field of intelligent secure services for 6G networks. It brings together advancements in AI-driven orchestration, real-time cybersecurity, trust mechanisms, explainability, and cyber threat intelligence, developed within the scope of Work Package 4.

The document began in Section 2 with a comprehensive state-of-the-art analysis, establishing the scientific and technological context for NATWORK's innovations. It reviewed existing approaches in zero-touch network management, AI-based threat detection, explainable artificial intelligence (XAI), cyber threat intelligence (CTI), and blockchain for trust establishment, identifying key challenges and research gaps that NATWORK addresses.

In Section 3, the deliverable introduced the first version of zero-touch network solutions, focusing on the design of AI-driven orchestration mechanisms that minimize human intervention while enabling autonomous, secure, and context-aware service deployment. This section outlined the proposed architecture and the early design of modules capable of managing dynamic security policies and adapting to runtime conditions.

Section 4 detailed the AI-driven real-time threat detection capabilities being developed in the project. It presented the conceptual design of intelligent agents that process telemetry data and threat indicators in real time, enabling proactive identification and response to malicious activities across different network layers.

In Section 5, the document described the initial design of blockchain-based trust establishment mechanisms, proposing a decentralized approach to support trust, data integrity, and secure interactions among distributed network components. This section introduced the foundational architecture and demonstrated how blockchain can enhance the transparency and reliability of service orchestration.

Section 6 addressed one of the project's key cross-cutting challenges: explainability. It explored different models and technical strategies to ensure that the decisions made by AI modules—particularly those related to security enforcement and service orchestration—are understandable and interpretable by operators and auditors. Several explainability mechanisms, ranging from visual tools to traceable decision models, were introduced.

Finally, Section 7 presented the first version of the Cyber Threat Intelligence (CTI) framework, outlining the approach for collecting, analysing, and operationalizing multi-source threat data. The CTI framework is designed to support both human analysts and automated agents, enhancing situational awareness and enabling adaptive security responses across the network.

Overall, this deliverable consolidates a wide range of technical contributions toward building an intelligent, secure, and explainable orchestration environment for 6G networks. The components and designs presented in D4.3 will serve as the basis for further integration and validation activities in the next phase of the NATWORK project.

References

[1] S. Neupane *et al*., "Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities," in *IEEE Access*, vol. 10, pp. 112392-112415, 2022.

[2] D. Gaspar, P. Silva and C. Silva, "Explainable AI for Intrusion Detection Systems: LIME and SHAP Applicability on Multi-Layer Perceptron," in IEEE Access, vol. 12, pp. 30164-30175, 2024

[3] Scott M. Lundberg and Su-In Lee. (2017) A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 4768–4777.

[4] J. Ables, T. Kirby, S. Mittal, I. Banicescu, S. Rahimi, W. Anderson and M. Seale, ``Explainable Intrusion Detection Systems Using Competitive Learning Techniques," https://arxiv.org/abs/2303.17387

[5] Patil, S.; Varadarajan, V.; Mazhar, S.M.; Sahibzada, A.; Ahmed, N.; Sinha, O.; Kumar, S.; Shaw, K.; Kotecha, K. Explainable Artificial Intelligence for Intrusion Detection System. *Electronics* **2022**, *11*, 3079.

[6] T. Ali and P. Kostakos, ``HuntGPT: Integrating Machine Learning-Based Anomaly Detection and Explainable AI with Large Language Models (LLMs)," https://arxiv.org/abs/2309.16021

[7] Arreche, O.; Guntur, T.; Abdallah, M. XAI-IDS: Toward Proposing an Explainable Artificial Intelligence Framework for Enhancing Network Intrusion Detection Systems. *Appl. Sci.* **2024**, *14*, 4170.

[8] P4 consortium. https://p4.org/

[9] M. Usama et al, "Unsupervised machine learning for networking: Techniques, applications and research challenges," IEEE Access, vol. 7, pp. 65579–65615, 2019.

[10] J.-H. Lee and K. Singh, "Switchtree: In-network computing and traffic analyses with random forests," Neural Comput. Appl., pp. 1–12, Nov. 2020.

[11] X. Zhang, L. Cui, F. P. Tso, and W. Jia, "pHeavy: Predicting heavy flows in the programmable data plane," IEEE Trans. Netw. Service Manag., vol. 18, no. 4, pp. 4353–4364, Dec. 2021.

[12] C. Zheng et al., "IIsy: Hybrid in-network classification using programmable switches," IEEE/ACM Trans. Netw., early access, Feb. 16, 2024, doi: 10.1109/TNET.2024.3364757.

[13] C. Zheng et al., "Automating in-network machine learning," 2022, arXiv:2205.08824.

[14] Q. Qin, K. Poularakis, K. K. Leung, and L. Tassiulas, "Line-speed and scalable intrusion detection at the network edge via federated learning," in Proc. IFIP Netw. Conf. (Netw.), 2020, pp. 352–360.

[15] Barsellotti, Luca, et al. "Introducing data processing units (DPU) at the edge."2022 International Conference on Computer Communications and Networks (ICCCN). IEEE, 2022.

[16] Gómez-Luna, Juan, et al. "An experimental evaluation of machine learning training on a real processing-in-memory system." arXiv preprint arXiv:2207.07886 (2022).

[17] Wang, Chao, et al. "DLAU: A scalable deep learning accelerator unit on FPGA." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 36.3 (2016): 513-517.

[18] Samarakoon, S., Bandara, D., Wijayasekara, C., Rupasinghe, T., & Marasinghe, A. (2022). *5G-NIDD: A comprehensive network intrusion detection dataset generated over 5G wireless network* (arXiv:2212.01298). arXiv. https://arxiv.org/abs/2212.01298

[19] Wang, W., Zhu, M., Zeng, X., Ye, X., & Sheng, Y. (2017). Malware traffic classification using convolutional neural network for representation learning. In *2017 International Conference on Information Networking (ICOIN)* (pp. 712–717). IEEE. https://doi.org/10.1109/ICOIN.2017.7899588

[20] Hassan, M., Devendran, V., Shereen, F., Muthanna, A., & Choudhury, N. (2021). Intrusion detection using payload embeddings. *IEEE Access, 10*, 4015–4030. https://doi.org/10.1109/ACCESS.2021.3139534

[21] Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., & Ghorbani, A. A. (2017). Characterization of Tor traffic using time-based features. In *Proceedings of the 2017 International Conference on Information Systems Security and Privacy (ICISSP)* (Vol. 2, pp. 253–262). SciTePress. https://doi.org/10.5220/0006110102530262

[22] Ziegler, Volker, Peter Schneider, Harish Viswanathan, Michael Montag, Satish Kanugovi, and Ali Rezaki. "Security and trust in the 6G era." Ieee Access 9 (2021): 142314-142327.

[23] Li, Wenjuan, and Weizhi Meng. "BCTrustFrame: enhancing trust management via blockchain and IPFS in 6G era." IEEE Network 36, no. 4 (2022): 120-125

[24] Fei, S., Yan, Z., Xie, H., & Liu, G. (2023). Sec-e2e: End-to-end communication security in ls-hetnets based on blockchain. *IEEE Transactions on Network Science and Engineering*, *11*(1), 761-778.

[25] Son, Seunghwan, Deokkyu Kwon, Sangwoo Lee, Hyeokchan Kwon, and Youngho Park. "A Zero-Trust Authentication Scheme With Access Control for 6G-enabled IoT Environments." IEEE Access (2024)

[26] Yan, Y., Alshawki, M. B., & Ligeti, P. (2020). Attribute-based encryption in cloud computing environment. In *2020 International Conference on Computing, Electronics & Communications Engineering (iCCECE)* (pp. 63-68). IEEE.

[27] Vigano, L. (2006). Automated security protocol analysis with the AVISPA tool. *Electronic Notes in Theoretical Computer Science*, *155*, 61-86.

[28] Rana, Minahil, Akasha Shafiq, Izwa Altaf, Mamoun Alazab, Khalid Mahmood, Shehzad Ashraf Chaudhry, and Yousaf Bin Zikria. "A secure and lightweight authentication scheme for next generation IoT infrastructure." Computer Communications 165 (2021): 85-96

[29] Pratap, B., Singh, A., & Mehra, P. S. (2025). REHAS: Robust and Efficient Hyperelliptic Curve-Based Authentication Scheme for Internet of Drones. *Concurrency and Computation: Practice and Experience*, *37*(3), e8333.

[30] Jacobson, M. J., Menezes, A., & Stein, A. (2003). *Hyperelliptic curves and cryptography*. Faculty of Mathematics, University of Waterloo.

[31] Choi, J., Son, S., Kwon, D., & Park, Y. (2025). A PUF-Based Secure Authentication and Key Agreement Scheme for the Internet of Drones. *Sensors*, *25*(3), 982.

[32] Suraci, C., Pizzi, S., Molinaro, A., & Araniti, G. (2021). MEC and D2D as Enabling Technologies for a Secure and Lightweight 6G eHealth System. *IEEE Internet of Things Journal*, *9*(13), 11524-11532.

[33] Putra, G. D., Dedeoglu, V., Kanhere, S. S., & Jurdak, R. (2022). Toward blockchain-based trust and reputation management for trustworthy 6G networks. *IEEE Network*, *36*(4), 112-119.

[34] Zou, W., Lo, D., Kochhar, P. S., Le, X. B. D., Xia, X., Feng, Y., ... & Xu, B. (2019). Smart contract development: Challenges and opportunities. *IEEE transactions on software engineering*, *47*(10), 2084-2106.

[35] Hajdú-Szücs, K., Vaderna, P., Kallus, Z., Kersch, P., Szalai-Gindl, J. M., & Laki, S. (2024). Ensemble Graph Attention Networks for Cellular Network Analytics: From Model Creation to Explainability. IEEE Transactions on Network and Service Management.

[36] Abdel Hakeem, S.A., Hussein, H.H., Kim, H. Security Requirements and Challenges of 6G Technologies and Applications. Sensors. 2022; 22(5):1969. https://doi.org/10.3390/s22051969

[37] Buchanan, B. G., & Shortliffe, E. H. (1984). Rule based expert systems: The MYCIN experiments of the Stanford heuristic programming project, Boston: Addison-Wesley Longman Publishing Co., Inc.

[38] Mitchell, T. M., Keller, R. M., & Kedar-Cabelli, S. T. (1986). Explanation-based generalization: A unifying view. Machine Learning, 1(1), 47–80.

[39] Gallant, S. (1988) Connectionist expert systems, Communications of the ACM, Vol 31, No 2 pp 152-169.

[40] Fahlman, S. and Lebiere, C. (1991) The cascade-correlation learning architecture, in Lippman, R., Moody, J. and Touretzky, D. (Eds.) Advances in Neural Information Processing Systems - Vol 3: San Mateo pp 190-196.

[41] Towell, G. and Shavlik, J. (1993) The extraction of refined rules from knowledge based neural networks, Machine Learning, Vol 131 pp 71-101.

[42] R. Andrews, J. Diederich, and A. B. Tickle. (1995) “A survey and critique of techniques for extracting rules from trained artificial neural networks,” Knowledge-Based Syst., vol. 8, no. 6, pp. 373–389.

[43] A. B. Tickle, R. Andrews, M. Golea and J. Diederich. (1998) "The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks," in IEEE Transactions on Neural Networks, vol. 9, no. 6, pp. 1057-1068.

[44] J.M. Benitez, J.L. Castro, I. Requena. (1997) Are artificial neural networks black boxes? IEEE Trans. Neural Networks 8 (5) 1156–1164.

[45] Olden, J.D., Jackson, D.A. (2002) Illuminating the “black box”: understanding variable contributions in artificial neural networks. Ecol. Model. 154, 135–150.

[46] G. Hooker. (2004) Discovering additive structure in black box functions, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 575–580.

[47] P. Cortez, M.J. Embrechts. (2011) Opening black box data mining models using sensitivity analysis, in: 2011 IEEE Symposium on Computational Intelligence and Data Mining (CIDM), IEEE, pp. 341–348.

[48] A. Henelius, K. Puolamäki, H. Boström, L. Asker, P. Papapetrou. (2014) A peek into the black box: exploring classifiers by randomization, Data mining and knowledge discovery 28 (5-6) 1503–1529.

[49] David Gunning. (2019) DARPA's explainable artificial intelligence (XAI) program. In Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI '19). Association for Computing Machinery, New York, NY, USA.

[50] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad. (2015) Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In Knowledge Discovery and Data Mining (KDD).

[51] B. Letham, C. Rudin, T. H. McCormick, and D. Madigan. (2015) Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. Annals of Applied Statistics.

[52] B. Ustun and C. Rudin. (2015) Supersparse linear integer models for optimized medical scoring systems. Machine Learning.

[53] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. (2016) “Why should I trust you?: Explaining the predictions of any classifier”. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM. 2016, pp. 1135–1144.

[54] European Commission (2019), Parliament: Regulation (EU) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). OJ L 119, 1–88.

[55] Wachter, Sandra & Mittelstadt, Brent & Russell, Chris. (2018). Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. Harvard journal of law & technology. 31. 841-887.

[56] Mundhenk, T. & Chen, Barry & Friedland, Gerald. (2019). Efficient Saliency Maps for Explainable AI.

[57] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra. (2017) "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, pp. 618-626.

[58] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. (2017) Axiomatic attribution for deep networks. In Proceedings of the 34th International Conference on Machine Learning - Volume 70 (ICML'17). JMLR.org, 3319–3328.

[59] Buffet, Olivier & Pietquin, Olivier & Weng, Paul. (2020). Reinforcement Learning. DOI: 10.1007/978-3-030-06164-7_12

[60] Stephanie Milani, Nicholay Topin, Manuela Veloso, and Fei Fang. (2024) Explainable Reinforcement Learning: A Survey and Comparative Review. ACM Comput. Surv. 56, 7, Article 168, 36 pages. https://doi.org/10.1145/3616864

[61] Qing, Yunpeng & Liu, Shunyu & Song, Jie & Song, Mingli. (2022). A Survey on Explainable Reinforcement Learning: Concepts, Algorithms, Challenges. 10.48550/arXiv.2211.06665.

[62] Verma, A., Murali, V., Singh, R., Kohli, P., Chaudhuri, S. (2018) Programmatically interpretable reinforcement learning. PMLR 80:5045-5054.

[63] Hein, D., Hentschel, A., Runkler, T., Udluft, S. (2017) Particle swarm optimization for generating interpretable fuzzy reinforcement learning policies. Engineering Applications of Artificial Intelligence 65, 87–98, https://doi.org/10.1016/j.engappai.2017.07.005

[64] Hein, D., Udluft, S., Runkler, T.A. (2018) Interpretable policies for reinforcement learning by genetic programming. Engineering Applications of Artificial Intelligence 76, 158–169.

[65] Juozapaitis, Z., Koul, A., Fern, A., Erwig, M., Doshi-Velez, F. (2019) Explainable reinforcement learning via reward decomposition. In: Proceedings of the IJCAI 2019 Workshop on Explainable Artificial Intelligence. pp. 47–53.

[66] van der Waa, J., van Diggelen, J., van den Bosch, K., Neerincx, M. (2018) Contrastive explanations for reinforcement learning in terms of expected consequences. IJCAI-18 Workshop on Explainable AI (XAI). Vol. 37. 2018.

[67] Rusu, A.A., Colmenarejo, S.G., Gulcehre, C., Desjardins, G., Kirkpatrick, J., Pascanu, R., Mnih, V., Kavukcuoglu, K., Hadsell, R. (2015) Policy distillation.

[68] Shu, T., Xiong, C., Socher, R. (2017) Hierarchical and interpretable skill acquisition in multi-task reinforcement learning.

[69] Sequeira, P., Gervasio, M. (2019) Interestingness elements for explainable reinforcement learning: Understanding agents' capabilities and limitations.

[70] Fukuchi, Y., Osawa, M., Yamakawa, H., Imai, M. (2017) Autonomous selfexplanation of behavior for interactive reinforcement learning agents. In: Proceedings of the 5th International Conference on Human Agent Interaction - HAI '17. ACM Press.

[71] Madumal, P., Miller, T., Sonenberg, L., Vetere, F. (2019) Explainable reinforcement learning through a causal lens.

[72] Y. Cao et al., "Survey on Large Language Model-Enhanced Reinforcement Learning: Concept, Taxonomy, and Methods," in IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2024.3497992.

[73] D. Das, S. Chernova, and B. Kim (2023) “State2explanation: Concept-based explanations to benefit agent learning and user understanding,” in Thirty-seventh Conference on Neural Information Processing Systems.

[74] Stuart Russell and Andrew L. Zimdars. 2003. Q-decomposition for reinforcement learning agents. In Proceedings of the Twentieth International Conference on International Conference on Machine Learning (ICML'03). AAAI Press, 656–663.

[75] Gauravaram, P. (2012, November). Security Analysis of salt|| password Hashes. In 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT) (pp. 25-30). IEEE.

[76] Liu, J. (2023, May). Digital signature and hash algorithms used in Bitcoin and Ethereum. In Third International Conference on Machine Learning and Computer Application (ICMLCA 2022) (Vol. 12636, pp. 1302-1321). SPIE.

[77] (2023) “Understanding Language in the World by Predicting the Future” in The Twelfth International Conference on Learning Representations.

[78] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini. (2009) "The Graph Neural Network Model," in IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61-80, DOI: 10.1109/TNN.2008.2005605.

[79] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. (2019) GNNExplainer: generating explanations for graph neural networks. Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA, Article 829, 9244–9255.

[80] J. D. Herath, P. P. Wakodikar, P. Yang and G. Yan. (2022) "CFGExplainer: Explaining Graph Neural Network-Based Malware Classification from Control Flow Graphs," 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Baltimore, MD, USA, pp. 172-184, doi: 10.1109/DSN53405.2022.00028.

[81] Mohit Bajaj, Lingyang Chu, Zi Yu Xue, Jian Pei, Lanjun Wang, Peter Cho-Ho Lam, and Yong Zhang. (2021) Robust counterfactual explanations on graph neural networks. In Proceedings of the 35th International Conference on Neural Information Processing Systems (NIPS '21). Curran Associates Inc., Red Hook, NY, USA, Article 431, 5644–5655.

[82] D. Pelleg and A. W. Moore, “X-means: Extending k-means with efficient estimation of the number of clusters,” in Proc. ICML, 2000, pp. 727–734.

[83] H. He, Y. Ji and H. H. Huang. (2022) "Illuminati: Towards Explaining Graph Neural Networks for Cybersecurity Analysis," 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P), Genoa, Italy, pp. 74-89, doi: 10.1109/EuroSP53844.2022.00013.

[84] K. Gai, M. Qiu, L. Tao, and Y. Zhu, “Intrusion detection techniques for mobile cloud computing in heterogeneous 5G,” Security and Communication Networks, vol. 9, no. 16, pp. 3049–3058, 2016

[85] S. Samarakoon, Y. Siriwardhana, P. Porambage, M. Liyanage, S.-Y. Chang, J. Kim, J. Kim, and M. Ylianttila, “5G-NIDD: Comprehensive network intrusion detection dataset generated over 5G wireless,” IEEE Dataport, 2022.

[86] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization,” in Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP), 2018, pp. 108–116. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

[87] W. W. Lo, G. Kulatilleke, M. Sarhan, S. Layeghy, and M. Portmann. (2023) “XG-BoT: An explainable deep graph neural network for botnet detection and forensics,” Internet Things, vol. 22, Art. no. 100747.

[88] X. Zhu, Y. Zhang, Z. Zhang, D. Guo, Q. Li and Z. Li. (2022) "Interpretability Evaluation of Botnet Detection Model based on Graph Neural Network," IEEE INFOCOM 2022 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), New York, NY, USA, pp. 1-6, DOI: 10.1109/INFOCOMWKSHPS54753.2022.9798287.

[89] Tabiban, A., Zhao, H., Jarraya, Y., Pourzandi, M., & Wang, L. (2022). VinciDecoder: Automatically Interpreting Provenance Graphs into Textual Forensic Reports with Application to OpenStack. In NordSec 2022.

[90] Md Rayhanur Rahman, Rezvan Mahdavi Hezaveh, and Laurie Williams.2023. What Are the Attackers Doing Now? Automating Cyberthreat Intelligence Extraction from Text on Pace with the Changing Threat Land-scape: A Survey. ACM Comput.Surv. 55, 12, Article 241 (March 2023), 36 pages.

[91] Marchiori, F., Conti, M., & Verde, N. V. (2023). STIXnet: A Novel and Modular Solution for Extracting All STIX Objects in CTI Reports. In Proceedings of ARES 2023, Woodstock, NY, USA, June 03–05, 2023. ACM, New York, NY, USA, 11 pages.

[92] Park, Y., & You, W. (2023). A Pretrained Language Model for Cyber Threat Intelligence. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track, 113-122.

[93] Rahman, M. R., Wroblewski, B., Matthews, Q., Morgan, B., Menzies, T., & Williams, L. (2023). Mining Temporal Attack Patterns from Cyberthreat Intelligence Reports. TRANSACTION OF SOFTWARE ENGINEERING, 1(1), 1-10.

[94] Li, Z., Zeng, J., Chen, Y., & Liang, Z. (2022). AttacKG: Constructing Technique Knowledge Graph from Cyber Threat Intelligence Reports. In ESORICS 2022 (pp. 1-18).

[95] Nidhi Rastogi, Sharmishtha Dutta, Mohammed J. Zaki, Alex Gittens, and Charu Aggarwal. 2020. MALOnt: An Ontology for Malware Threat Intelligence. In MLHat: The First International Workshop on Deployable Machine Learning for Security Defense, August 24, 2020, San Diego, CA. ACM, New York, NY, USA, 8 pages.

[96] J. Caballero, G. Gomez, S. Matic et al., The rise of GoodFATR: A novel accuracy comparison methodology for indicator extraction tools, Future Generation Computer Systems (2023), doi: https://doi.org/10.1016/j.future.2023.02.012

[97] Bouwman, X., Griffioen, H., Egbers, J., Doerr, C., Klievink, B., & van Eeten, M. (2020). A Different Cup of TI? The Added Value of Commercial Threat Intelligence. In Proceedings of the 29th USENIX Security Symposium (pp. 1-10). USENIX. https://www.usenix.org/conference/usenixsecurity20/presentation/bouwman

[98] Sabottke, C., Suciu, O., & Dumitraş, T. (2015). Vulnerability Disclosure in the Age of Social Media: Exploiting Twitter for Predicting Real-World Exploits. In Proceedings of the 24th USENIX Security Symposium (pp. 1-10). Washington, D.C.: USENIX Association. doi: 10.1145/2814113.2814123

[99] Boffa, M., Valentim, R. V., Vassio, L., Giordano, D., Drago, I., Mellia, M., & Houidi, Z. B. (2023). LogPrecis: Unleashing Language Models for Automated Shell Log Analysis. arXiv preprint arXiv:2307.08309.

[100] Guo, Y., Liu, Z., Huang, C., Liu, J., Jing, W., Wang, Z., & Wang, Y. (2021). CyberRel: Joint Entity and Relation Extraction for Cybersecurity Concepts. In D. Gao et al. (Eds.), ICICS 2021, LNCS 12918, pp. 447–463. Springer Nature Switzerland AG. https://doi.org/10.1007/978-3-030-86890-1_25

[101] Caselli, M., Legoy, V., Peter, A., & Seifert, C. (2020). Retrieving ATT&CK tactics and techniques in cyber threat reports. In FIRST CTI Symposium 2020 (pp. 0).

[102] Industry Specification Group Zero Touch Network and Service Management https://www.etsi.org/committee/1431-zsm

[103] Chollon, G., Ayed, D., Garriga, R.A., Zarca, A.M., Skarmeta, A., Christopoulou, M. Soussi, W., Gür, G. and Herzog, U. . ETSI ZSM driven security management in future networks. In 2022 IEEE Future Networks World Forum (FNWF), pages 334–339, 2022.

[104] X. Hou, Y. Zhao, S. Wang, and H. Wang, “Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions,” *arXiv preprint arXiv:2503.23278*, 2025. [Online]. Available: https://arxiv.org/abs/2503.23278

[105] Ray, Partha Pratim. "A survey on model context protocol: Architecture, state-of-the-art, challenges and future directions." *Authorea Preprints* (2025).

[106] Z. Liu, J. Gaddam, S. Huang, I. Bandara, M. Liu, S. Rajasegarar, M. U. Hassan, L. Yang, G. Li, and M. Angelova, “Large Language Model and Variational Autoencoder Based Deep Neural

Framework for Cyber Attack Detection," in Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining, 2025, pp. 91–102.

[107] Asterios Mpatziakas, Ioannis Schoinas, Antonios Lalas, Anastasios Drosou, Nestor Chatzidiamantis , Dimitrios Tzovaras - Deciphering Standards for cybersecurity in Industry 4.0: Advisory AI for Cybersecure IIoT, 2025 IEEE CSR Workshop on Cyber-Physical Resilience and Security Against Digital Breakdowns (CYPRES), Chania August 2025 (Accepted, to be published)

[108] A. Mpatziakas, A. Drosou, S. Papadopoulos, and D. Tzovaras, "IoT threat mitigation engine empowered by artificial intelligence multi-objective optimization," J. Netw. Comput. Appl., vol. 203, p. 103398, Jul. 2022.

[109] European Telecommunications Standards Institute (ETSI). Zero-touch network and service management (zsm); closed-loop automation; part 1: enablers. Technical Specification ETSI GS ZSM 009-1 V1.1.1, ETSI, 2021. Available at: https://www.etsi.org/deliver/etsi_gs/ZSM/001_099/00901/01.01.01_60/gs_ZSM00901v010101p.pdf.

[110] Piplai, A., Mittal, S., Joshi, A., Finin, T., Holt, J., & Zak, R. (2020). Creating cybersecurity knowledge graphs from malware after action reports. IEEE Access, 8, 211691-211703.

[111] Alsaheel, A., Nan, Y., Ma, S., Yu, L., Walkup, G., Celik, Z. B., Zhang, X., & Xu, D. (2021). ATLAS: A Sequence-based Learning Approach for Attack Investigation. In Proceedings of the 30th USENIX Security Symposium (pp. 1-18). USENIX. https://www.usenix.org/conference/usenixsecurity21/presentation/alsaheel

[112] Alam, M. T., Bhusal, D., Park, Y., & Rastogi, N. (2022). Cyner: A python library for cybersecurity named entity recognition. arXiv preprint arXiv:2204.05754.

[113] You, Y., Jiang, J., Jiang, Z., Yang, P., Liu, B., Feng, H., Wang, X., & Li, N. (2022). TIM: Threat Context-Enhanced TTP Intelligence Mining on Unstructured Threat Data. Cybersecurity, 5(3), 1-18.

[114] Dasgupta, S., Piplai, A., Kotal, A., & Joshi, A. (2020). A Comparative Study of Deep Learning based Named Entity Recognition Algorithms for Cybersecurity. In BigData 2020 (pp. 0).

[115] Kim, G., Lee, C., Jo, J., & Lim, H. (2020). Automatic extraction of named entities of cyber threats using a deep Bi-LSTM-CRF network. International Journal of Machine Learning and Cybernetics, 11(2), 2341-2355. doi: 10.1007/s13042-020-01122-6

[116] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." *nature* 518.7540 (2015): 529-533.

[117] Juozapaitis, Zoe, et al. "Explainable reinforcement learning via reward decomposition." IJCAI/ECAI Workshop on explainable artificial intelligence. 2019.

[118] Seungwon Shin and Guofei Gu, "CloudWatcher: Network security monitoring using OpenFlow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?)," 2012 20th IEEE International Conference on Network Protocols (ICNP), Austin, TX, 2012, pp. 1-6, doi: 10.1109/ICNP.2012.6459946.

[119] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. IEEE transactions on neural networks, 20(1), 61-80.

[120] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.

[121] DDoS evaluation dataset (CIC-DDoS2019) https://www.unb.ca/cic/datasets/ddos-2019.html

[122] TOR project https://www.torproject.org/, 2025, [Online; accessed 30-June-2025]

[123] Internet Storm center https://isc.sans.edu/index.html, 2025, [Online; accessed 30-June-2025]

[124] Suricata https://suricata.io/, 2025, [Online; accessed 30-June-2025]

[125] AbuseIPDB https://www.abuseipdb.com/, 2025, [Online; accessed 30-June-2025]

[126] Onyphe, BigData for Cyber Defense https://www.onyphe.io/, 2025, [Online; accessed 30-June-2025]

[127] S. Barnum, "Standardizing cyber threat intelligence information with the structured threat information expression (stix)," Mitre Corporation, vol. 11, pp. 1–22, 2012.

[128] PaloAlto Network Unit42 https://unit42.paloaltonetworks.com/helloxd-ransomware/, 2025, [Online; accessed 30-June-2025]

[129] Y. Park and T. Lee, "Full-stack information extraction system for cybersecurity intelligence," in Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track, 2022, pp. 531–539.

[130] "MITRE ATT&CK," https://attack.mitre.org/, 2023, [Online; accessed 13-April-2023].

[131] Proofpoint blog https://www.proofpoint.com/us/blog/threat-insight/good-bad-and-web-bug-ta416-increases-operational-tempo-against-european, 2025, [Online; accessed 30-June-2025]

[132] T. Satyapanich, F. Ferraro, and T. Finin, "Casie: Extracting cyber-security event information from text," in Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 05, 2020, pp. 8749–8757

[133] G. Husari, E. Al-Shaer, M. Ahmed, B. Chu, and X. Niu, "Ttpdrill: Automatic and accurate extraction of threat actions from unstructured text of cti sources," in Proceedings of the 33rd annual computer security applications conference, 2017, pp. 103–115.

[134] M. T. Alam, D. Bhusal, Y. Park, and N. Rastogi, "Looking beyond iocs: Automatically extracting attack patterns from external cti," arXiv preprint arXiv:2211.01753, 2022.

[135] P. Gao, X. Liu, E. Choi, S. Ma, X. Yang, Z. Ji, Z. Zhang, and D. Song, "Threatkg: A threat knowledge graph for automated open-source cyber threat intelligence gathering and management," arXiv preprint arXiv:2212.10388, 2022.

[136] V. Legoy, M. Caselli, C. Seifert, and A. Peter, "Automated retrieval of att&ck tactics and techniques for cyber threat reports," arXiv preprint arXiv:2004.14322, 2020.

[137] Wang, M., et al. (2020) Security and privacy in 6G networks: New areas and new challenges. *Digital Communications and Networks* 6(3): 281-291.

[138] Kang, Hongzhaoning, Gang Liu, Quan Wang, Lei Meng, and Jing Liu. "Theory and application of zero trust security: A brief survey." Entropy 25, no. 12 (2023): 1595

[139] Wlodarczak, P. (2017) Cyber Immunity: A Bio-Inspired Cyber Defense System. *Bioinformatics and Biomedical Engineering: 5th International Work-Conference, IWBBIO*, 2017 Granada, Spain, April 26–28, 2017, Proceedings, Part II 5. Springer International Publishing.

[140] Porambage, P., et al. (2021) 6G security challenges and potential solutions. *2021 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. IEEE.

[141] Yu, Q., et al. (2020) An immunology-inspired network security architecture. *IEEE Wireless Communications* 27(5): 168-173.

[142] United Nations, (2023) Goal 9: Build resilient infrastructure, promote sustainable industrialization and foster innovation. The Sustainable Development Goals Report 2023: https://www.un.org/sustainabledevelopment/infrastructure-industrialization.

[143] Li, Shan, Muddesar Iqbal, and Neetesh Saxena. "Future industry internet of things with zero-trust security." Information Systems Frontiers 26, no. 5 (2024): 1653-1666

[144] Huang, Q., Yamada, M., Tian, Y., Singh, D., & Chang, Y. (2022). Graphlime: Local interpretable model explanations for graph neural networks. IEEE Transactions on Knowledge and Data Engineering, 35(7), 6968-6972.

[145] Hayes, Conor F., et al. "A practical guide to multi-objective reinforcement learning and planning." Autonomous Agents and Multi-Agent Systems 36.1 (2022): 26.

[146] K. Kivanç Eren, K. Küçük, F. Özyurt and O. H. Alhazmi, "Simple Yet Powerful: Machine Learning-Based IoT Intrusion System With Smart Preprocessing and Feature Generation Rivals Deep Learning," in *IEEE Access*, vol. 13, pp. 41435-41455, 2025, doi: 10.1109/ACCESS.2025.3547642.

[147] "TRAM GitHub repository," https://github.com/center-for-threat-informed-defense/tram/ , 2023, [Online; accessed 13-April-2023].

[148] PaloAlto Networks https://www.paloaltonetworks.com/, 2025, [Online; accessed 30-June-2025]

[149] Trend Micro https://www.trendmicro.com/, 2025, [Online; accessed 30-June-2025]

[150] Fortinet https://www.fortinet.com/, 2025, [Online; accessed 30-June-2025]

[151] Z. Li, J. Zeng, Y. Chen, and Z. Liang, "Attackg: Constructing technique knowledge graph from cyber threat intelligence reports," in Computer Security–ESORICS 2022: 27th European Symposium on Research in Computer Security, Copenhagen, Denmark, September 26–30, 2022, Proceedings, Part I. Springer, 2022, pp. 589–609.

[152] V. Orbinato, M. Barbaraci, R. Natella, and D. Cotroneo, "Automatic mapping of unstructured cyber threat intelligence: An experimental study:(practical experience report)," in 2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE). IEEE, 2022, pp. 181–192.

[153] "Picus Labs - The Red Report 2023," https://www.picussecurity.com/resource/report/the-red-report-2023, 2023, [Online; accessed 04-May-2023].

[154] Olups, R. (2010). *Zabbix 1.8 network monitoring*. Packt Publishing Ltd.

[155] Barth, W. (2008). *Nagios: System and network monitoring*. No Starch Press. Prometheus Documentation https://prometheus.io/docs/introduction/overview/

[156] Turnbull, J. (2018). Monitoring with Prometheus. Turnbull Press.

[157] Chakraborty, M., & Kundan, A. P. (2021). Grafana. In Monitoring cloud-native applications: Lead agile operations confidently using open source software (pp. 187-240). Berkeley, CA: Apress.

[158] 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.

Annex A

A.1 Annex – Classification of attacks

As part of the task 4.1, the NATWORK project continues to develop the attack datasets. Different partners are working on different attacks, on HTTP2, TCP, UDP, AMF, among others. The partners have classified their attacks so that to facilitate data collection activities, through uploading the generated datasets to a common repository provided by T4.5. A glimpse on the structure can be seen in the following table:

Table 6: Classification of attacks

Attack Name	Type of attack	Target Protocol	Target
DoS attacks and port scans	Denial of Service	ICMP UDP TCP HTTP	Server deployed in the 5GTN MEC environment
AI-DoS attack	Denial of Service	TCP/SCTP/HTTP2/UDP	AMF and SMF 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)	SSH brute force	SSH	MONT's testbed
	Log4Shell	TCP / HTTP	
Data Exfiltration	Data Exfiltration	Depends on the attack type HTTP might be a target	CNFs/VNFs on 5G testbed
Malware Infection	Malware Infection	Depends on the attack type HTTP might be targets	CNFs/VNFs on 5G testbed

Attack Name	Type of attack	Target Protocol	Target
DoSt Attack	App HTTP	HTTP	CNFs/VNFs on 5G testbed
Mirai botnet attack	UDP generic	UDP	HES-SO's testbed
	TCP SYN	TCP	
	App HTTP	HTTP	
Jamming Attack	Jamming	IEEE 802.11.p	CERTH testbed

A.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 Denial-of-Service (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.

A.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 using TCP, UDP, or SCTP, 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 the Access and Mobility Function (AMF) or an On-Board Unit (OBU) in vehicular networks, GORGO adjusts its tactics accordingly.

The tool can conduct protocol-level fuzzing: It generates and manipulates network packets autonomously to expose vulnerabilities that may not be detectable through 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 Quality-of-Service metrics, such as latency and throughput, GORGO refines its strategy to enhance the overall effectiveness of the attack.

A.2.2 Experimental Setup and validation

An initial Validation and performance testing of GORGO was conducted using the CERTH 5G testbed, which provides a cloud-native, containerized network environment. This infrastructure includes a complete 5G core network implemented with Free5GC, along with simulated User Equipment and eNodeB elements. Network functions are deployed across Docker containers and interconnected through a software-defined networking architecture managed by Open vSwitch and the Floodlight controller.

We performed the following experiment: GORGO targeted the AMF component using SCTP over port 38412, leading to a complete disruption of core services. The final results of this attack are shown in Figure 28. The tool managed to disrupt the AMF functionalities after training on its environment after performing unsupervised learning that required 2 days, 21 hours, 56 minutes.

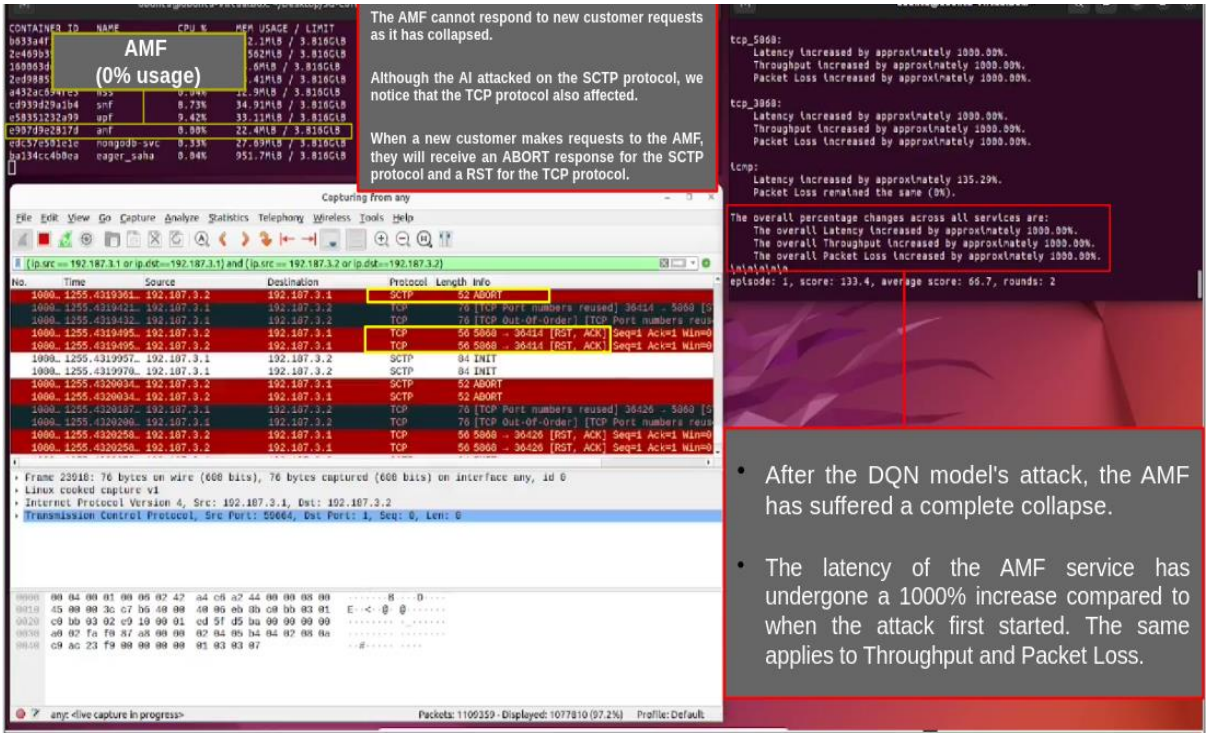


Figure 28: Results of DoS attack against AMF component in real 5G testbed environment.

The validation strategy for the first evaluation experiments of GORGO was structured so that can be extended to fit the scope of Use Case 3.2 of the NATWORK project. This use case focuses on AI-enabled penetration testing for 5G and 6G infrastructures. The evaluation was conducted through a three-stage scenario. Initially, GORGO launched a DoS attack while continuously monitoring the impact on network performance. As the attack unfolded, the system received feedback and dynamically optimized its approach to increase the level of disruption. The final phase involved a complete breakdown in communication between key network functions, effectively resulting in a full denial of service.

A.2.3 Future Steps

Further improvements to GORGO are planned to enhance its application before the end of the project. These include expanding support to additional protocols such as HTTP/1.1 and HTTP/2.0, which will allow GORGO to operate in a broader range of network environments. Techniques to evade intrusion detection systems are also being explored, with the aim of making GORGO's behaviour more difficult to detect. Recorded data from GORGO's activities will also be used to train intelligent intrusion detection systems, creating a feedback loop between offensive testing

and defensive development. These data collection activities will facilitate also the population of the common NATWORK data repository, provided by T4.5, with attack generated datasets.