Future Generation Computer Systems 163 (2025) 107537

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

An intelligent native network slicing security architecture empowered by federated learning

Rodrigo Moreira [a,*], Rodolfo S. Villaça [b], Moisés R.N. Ribeiro [b], Joberto S.B. Martins [c], João Henrique Corrêa [d], Tereza C. Carvalho [e], Flávio de Oliveira Silva [f]

[a] *Federal University of Viçosa (UFV), Rio Paranaíba, Minas Gerais, Brazil*
[b] *Federal University of Espírito Santo (UFES), Vitória, Espírito Santo, Brazil*
[c] *Salvador University (UNIFACS), Salvador, Bahia, Brazil*
[d] *Federal University of Ceará (UFC), Itapajé, Ceará, Brazil*
[e] *University of São Paulo (USP), São Paulo, São Paulo, Brazil*
[f] *University of Minho (UMinho), Braga, Portugal*

ARTICLE INFO

Keywords:
Network slicing
Machine learning
Network analytics
Federated learning
Security

ABSTRACT

Network Slicing (NS) has transformed the landscape of resource sharing in networks, offering flexibility to support services and applications with highly variable requirements in areas such as the next-generation 5G/6G mobile networks (NGMN), vehicular networks, industrial Internet of Things (IoT), and verticals. Although significant research and experimentation have driven the development of network slicing, existing architectures often fall short in intrinsic architectural intelligent security capabilities. This paper proposes an architecture-intelligent security mechanism to improve the NS solutions. We idealized a security-native architecture that deploys intelligent microservices as federated agents based on machine learning, providing intra-slice and architectural operation security for the Slicing Future Internet Infrastructures (SFI2) reference architecture. It is noteworthy that federated-learning approaches match the highly distributed modern microservice-based architectures, thus providing a unifying and scalable design choice for NS platforms addressing both service and security. Using ML-Agents and Security Agents, our approach identified Distributed Denial-of-Service (DDoS) and intrusion attacks within the slice using generic and non-intrusive telemetry records, achieving an average accuracy of approximately 95.60% in the network slicing architecture and 99.99% for the deployed slice – intra-slice. This result demonstrates the potential for leveraging architectural operational security and introduces a promising new research direction for network slicing architectures.

1. Introduction

Over the last 40 years, mobile networks have evolved and benefited our society, changing the way we perform daily tasks with applications that are now indispensable, smarter, and more useful than before. This impact is measured when we look at the forecast of having 3.5 billion users consuming connectivity and supporting use cases with more than 70 different industrial segments [1,2]. The evolution of mobile networks has been marked by a paradigm shift from conventional networks to software-defined intelligent networks facilitated by softwarization, cloudification, and Network Slicing (NS) paradigms [3,4]. These advancements have been made to meet the needs of users and modern applications.

Applications such as Virtual Reality (VR), Internet of Everything (IoE), and Autonomous Vehicles are evolving, requiring, rather than network connectivity, a portion of network resources to cope with specific requirements [5,6]. NS involves tailoring a physical network to specific applications and services using three primary baselines: isolation, end-to-end connectivity, and application-driven requirements [7,8]. In this context, challenges arise from intelligent and secure network slicing, such as high automation, programmability, interoperability, data orchestration, and zero-touch management [9,10]. Disruptive paradigms, such as Artificial Intelligence as a Service (AIaaS) [11] or Machine Learning as a Service (MLaaS) [12], can evolve network slicing architectures that provide security skills, not as a feature to be developed apart from slicing architecture, but rather as native-aware

[*] Corresponding author.
E-mail addresses: rodrigo@ufv.br (R. Moreira), rodolfo.villaca@ufes.br (R.S. Villaça), moises.ribeiro@ufes.br (M.R.N. Ribeiro), joberto.martins@animaeducacao.com.br (J.S.B. Martins), joaocorrea@ufc.br (J.H. Corrêa), terezacarvalho@usp.br (T.C. Carvalho), flavio@di.uminho.pt (F. de Oliveira Silva).

https://doi.org/10.1016/j.future.2024.107537
Received 25 June 2024; Received in revised form 10 September 2024; Accepted 17 September 2024
Available online 2 October 2024

security defenses for network slicing, particularly in architectural core operations and transactions [3,13,14]. These technologies and findings are essential for realizing network slicing. The evolution of mobile networks is envisioned to integrate land, sky, and sea connectivity [15] because of the heterogeneity of technologies and domains conducive to security issues [16].

In the literature, we found the use of large-scale testbeds to build and validate network slicing architectures, algorithms, and methods. Additionally, advanced testbeds serve as testing grounds for developing and experimenting with innovative network slicing architectures [17–21]. Legacy testbeds [18,22] were integrated through a reference architecture, and functionalities were presented in the Slicing Future Internet Infrastructures (SFI2) reference Architecture [4,23,24]. However, incorporating these legacies and cutting-edge testbeds to support network-slicing security experimentation poses challenges. Given the heterogeneous integration scenario for network slicing architecture components, we believe a native security-aware architecture can effectively advance such methods.

Several security issues related to network slicing include impersonation, traffic injection, Denial-of-Service (DoS) tampering, eavesdropping, reply attacks, and interface monitoring [23,25,26]. This article introduces network slicing architecture enhancement with a native, distributed, and highly scalable security operation method through a combination of `Security Agents` and `ML-Agents`. Existing approaches do not fully explore the distributed `Security Agent` aligned with the `ML-Agent` as a countermeasure to improve the operational security of network slice architectures and intra-slice with the ability to enhance the attack-handling capabilities for each control-plane core entity [26,27].

In this context, FL is a type of distributed learning where multiple devices or systems train models locally on their own data [28], sharing model updates only (such as weights or gradients) with a central server at the `ML-Agent` in the SFI2 architecture. This server aggregates these contributions without accessing the raw data, allowing for the creation of a global model that preserves data privacy and security across distributed sources. Federated Learning offers several advantages over traditional centralized machine learning, particularly in attack detection within network slicing. In a scenario where each tenant within a network slice has access to their flow data and network attack metrics, federated learning becomes a powerful tool for collaborative model development without compromising tenants' privacy.

We validate the effectiveness of our method by embedding federated learning in `ML-Agents` to supply `Security Agents` handling Distributed Denial-of-Service (DDoS) and intrusion attacks on network slicing control-plane entities. Our experimentation and validation Proof of Concept (PoC) was based on microservices that provide cognitive services to architectural entities in a highly granular, independent, and customizable manner. `ML-Agents` and `Security Agents` are organized in Kubernetes sidecar containers that run within a separate service within a pod. The `Security Agent` works in the same data plane as the pod services of the SFI2 Architecture core entities that handle threat identification tasks. By contrast, other entities in the control plane of the architecture usually continue their management functions.

The contributions of this study are as follows: (1) A framework for embedding native security into network slicing architectures through `ML-Agent` and `Security Agents`; (2) a functional evaluation of a security-native network slicing architecture capable of handling lifecycle operations and intra-slice security threats; (3) an empirical evaluation of a self-adaptive learning architecture based on federated learning for network slicing architectures; and (4) an evaluation of the Security Agent's capacity to handle on-the-fly attack prediction in network slicing architectures on a nationwide testbeds.

The structure of the remainder of this paper is as follows. The topics that relate to this work are discussed in Session 2. In Section 3, we contextualize our work within a broader scope, highlighting the unique contributions of this research. The proposed method is presented in detail in Section 4, followed by a description of the experimental setup and results in Section 5. Section 6 discusses concluding remarks and future directions.

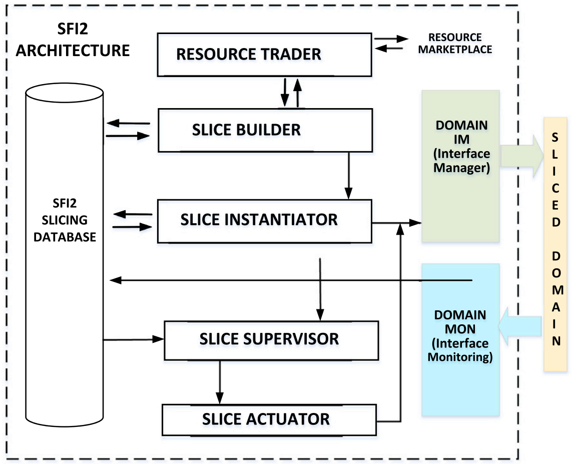


Fig. 1. Framework architecture of basic SFI2 building blocks.

2. Background

Network slicing. Resource sharing enabled by computing virtualization has influenced mobile networks, particularly in 5th Generation Mobile Network (5G) development mainstream. This has resulted in numerous initiatives from industry, Standards Development Organizations (SDOs), and academia to share network resources, known as Network Slicing (NS) [7,29]. Network slicing is defined by different standard bodies, such as Next Generation Mobile Networks (NGMN), as a division of a physical network into multiple networks with capabilities and characteristics oriented to a use case [30]. 3rd Generation Partnership Project (3GPP) defines network slicing as a technology that allows the operator to create and customize the network to meet different market demands [31]. Based on these pillars, the state of the art has been building and evolving solutions, architectures, and management approaches to offer tailored network resources to users despite the challenges of seamless isolation, performance, and security [32].

NS Arquitectures. A plethora of network slicing architectures for specific domains, industry verticals, and connectivity requirements have emerged [4,33–36]. While each architecture has its features, they share common aspects, such as the management and lifecycle control loop for network slicing. These architectures provide entities for network slices' preparation, commissioning, operation, and decommissioning phases. The control mechanisms for these phases may vary among architectures; however, coordination, privacy, energy efficiency, and security are still critical, especially for business verticals that utilize network slices [37]. Thus, we proposed a conceptual architecture of network slicing SFI2 focused on security, sustainability, and experimental network integration [23].

SFI2 Framework. The SFI2 architecture is illustrated in Fig. 1, in which the main functional blocks are highlighted [23]. The architecture includes a block slice requester, resource trader, slice builder, slice instantiator, slice supervisor, and slice actuator, as well as the interaction with different domains over which network slices can be instantiated. Thus, SFI2 becomes seamless to deploy network slices in heterogeneous testbeds with different technologies, such as Kubernetes, Docker, or Virtual Machines. The SFI2 architecture was built to be natively secure and intelligent [38]. In this paper, we introduce the `Security Agent`, `ML-Agent`, with native distributed learning, and `Monitoring Agent` components [23].

Federated Learning. Federated learning is a machine learning paradigm that trains a robust model by leveraging data spread across heterogeneous devices or servers. Federated learning offers several advantages over centralized approaches for attack detection. Primarily by preserving data privacy and security, tenants can train models locally and share only model updates without exposing sensitive data. It enables collaborative knowledge sharing, allowing tenants to benefit from a robust global model that captures a wider range of attack patterns without directly sharing their data. The approach enhances attack detection by continuously updating the global model with new information from tenants, improving adaptability to emerging threats. Additionally, it reduces bandwidth and computational overhead by minimizing data transfer requirements. It increases resilience to single points of failure by distributing data processing across multiple tenants, where data sensitivity and privacy are critical [39,40]. The general formula for the federated learning process can be expressed as $F(w) = \frac{1}{K} \times \sum_{k=1}^{K} F_k(w)$, where $F(w)$ is the global objective function to be minimized, w represents the model parameters, K is the total number of clients, and $F_k(w)$ is the local objective function of the kth client. Each client computes an update to the model based on its local data, and these updates are then aggregated to form a new global model. The process iterates until convergence or until a satisfactory model is obtained.

3. Related work

Previous studies have explored various network-slicing approaches, each tailored to specific requirements and applications [3], including aspects such as security or profit-based resource allocation [41, 42]. Moreover, federated learning has been applied to different network challenges, especially those focused on privacy [43–45]. Many of these architectures incorporate Artificial Intelligence (AI) capabilities, whereas others employ AI for orchestration and other purposes in network slices [46]. This section presents research on applying computational intelligence techniques in network experimental testbeds and using machine learning algorithms to enhance security in network-slicing architectures.

3.1. Federated learning for testbeds

Wijethilaka et al. [37] developed a federated learning strategy to enhance network slicing security, a technology that allocates dedicated logical networks to diverse applications. This manuscript introduces a framework called FLeSO, which utilizes federated learning to train machine learning models to detect anomalies and cyber-attacks in the control plane of sliced networks. The FLeSO framework was tested using an experimental testbed of sliced networks built with open-source tools and an NSL-KDD intrusion dataset. The study indicates that FLeSO is good at identifying attacks, keeping data private, and enabling proactive security measures.

Boualouache et al. [47] presented a solution for detecting inter-slice attacks in Vehicle-to-Everything (V2X) 5G networks, which pose a significant threat to the isolation and privacy of network slices. The proposed approach utilizes federated and deep learning to deploy virtual security functions within network slices, which cooperate to train and refine attack detection models. The effectiveness of this method was validated through extensive experimentation conducted on a testbed consisting of sliced networks and the CSE-CIC-IDS2018 intrusion dataset [48]. Our process results in high accuracy in detecting attacks in network slicing management transactions while ensuring data privacy.

Saad et al. [49] presents a timely contribution to the field of federated learning in Beyond Fifth Generation (B5G) networks with zero-touch management, focusing on the pressing issue of poisoning attacks that can prevent the optimal functioning and security of deep learning models utilized in the automated management and orchestration of network slices. To address this challenge, the authors introduce a novel framework called Trust Deep Q-learning Federated Learning (TQFL), which employs deep reinforcement learning to select a trusted participant in the federated learning process responsible for detecting and mitigating poisoning attacks using unsupervised learning and dimensionality reduction. The performance of TQFL was evaluated through an exhaustive experimentation campaign using the OpenAirInterface platform and a realistic dataset on the latency of the Access and Mobility Management Function (AMF) function. The results show the effectiveness of TQFL in mitigating poisoning attacks while preserving the accuracy and privacy of federated learning models.

3.2. Sliced testbeds and security

Wichary et al. [50] presented a solution to safeguarding and isolating 5G network slices across multiple layers and domains. The proposed approach uses a security attribute model, which associates security controls with the specific requirements of each slice. The study assessed the effectiveness and practicality of various security measures, categorizing them into eight domains and six isolation classes.

Jiang et al. [56] proposed a DeepAR-based probabilistic forecasting model for admission control in network slicing within Software-defined Networks (SDNs). The model is designed for network segmentation and is compatible with 5G and Software-defined Wide Area Network (SD-WAN) standards. They incorporate AI and blockchain technologies to enhance network security. The training paradigm leverages DeepAR, a recurrent neural network model, using real-world historical traffic from the China Education and Research Network (CERNET) to proactively manage slice admissions and prevent congestion. In addition, a closed-loop parameter-update mechanism was employed to optimize resource allocation and improve defense strategies.

Khan et al. [51] presents a solution to the challenge of detecting DoS and DDoS attacks on 5G network slices, which can significantly impact the functionality and performance of the associated services. Their proposed method is based on a Recurrent Neural Network (RNN), which utilizes a recently collected dataset from a simulated 5G network slicing testbed. The model's efficacy was validated through accuracy tests, resulting in a remarkable value of 99.99%.

Niboucha et al. [52] addresses the problem of detection and mitigation of DDoS attacks on Massive Machine Type Communications (mMTC) network slices in 5G networks, which can connect many Internet-of-Things (IoT) devices. The proposed method is a zero-touch security management solution that uses machine learning to predict, identify, and block malicious devices that generate abnormal traffic. The measure used was the detection rate of the model, which was tested on a EURECOM 5G testbed.

Concerning testbeds for security, Wen et al. [53] presented VET5G, an end-to-end virtual testbed for experimenting with security in 5G networks. The proposed method is a container-based platform that emulates mobile devices, RAN, and 5G core networks, supporting programmability and isolation. The measures used were the performance and usability of the testbed, which was evaluated in two attack scenarios and a course project.

Wijethilaka et al. [37] focus on ensuring security in sliced networks, a critical component of future telecommunication systems. To address this issue, the authors propose an architecture for a security orchestrator that can provide tailored security services to different network slices and evaluate its feasibility and performance through experimentation using an OpenStack-based testbed, together with Open Source MANO (OSM), PyTorch, and an NSL-KDD intrusion detection dataset.

The paper by Silva et al. [54] focuses on the significant issue of DDoS attacks on the 5G control plane, which can compromise service availability and security. The authors propose a new approach, REPEL, an intelligent resource-scheduling strategy that utilizes game theory to

Table 1
Prior state-of-the-art works towards Security and AI for Network Slicing Architectures.

Approach	Network segment	Standards compatibility	Security level	Training paradigm	Dataset	Pro-active security/action	Defense/action strategy
Wijethilaka et al. [50]	Slicing Architecture	None	Architecture	Local	None	○	Manage the resources and services of each slice to guarantee data confidentiality, integrity, and availability.
Khan et al. [51]	RAN	3GPP	intra-Slicing	Local	CIC IDS 2017	○	A centralized controller that coordinates the agents in the slices collects and analyzes the network data, and sends alerts in the event of anomalies.
Niboucha et al. [52]	Core	3GPP	intra-Slicing	Local	Proprietary	○	A deep neural network is used to autonomously detect and mitigate DDoS attacks on the mMTC network slices.
Wen et al. [53]	Core	None	Arquitecture	None	None	○	Integrating different security solutions into the testbed, such as firewalls, IDS/IPS, and VPNs, on demand for the network slices that require it.
Silva et al. [54]	Core	3GPP	Arquitecture	None	None	●	Prevents DDoS attacks on the 5G control plane through intelligent resource scaling.
Chilukuri et al. [55]	Core	3GPP	Architecture	Local	Proprietary	○	Using Self-Organizing Network (SON) and Hierarchical Temporal Memory (HTM)-based learning, the approach detects and isolates malicious users trying to attack the 5G core using the XDP technique.
Wijethilaka et al. [37]	Slicing Architecture	None	Architecture	Distributed	NSL-KDD	○	Federated learning is used to train machine-learning models to detect anomalies and attacks in the control plane of sliced networks.
Boualouache et al. [47]	Slicing Architecture	None	intra-Slicing	Distributed	CSE-CIC-IDS2018	○	Using federated learning and deep learning to train attack detection models, virtual security functions are deployed in network slices that collaborate to update these models.
Saad et al. [49]	RAN	3GPP	Architecture	Distributed	Eurecom AMF Resource Consumption Dataset	●	Deep reinforcement learning was used to select a trusted participant in federated learning, which is responsible for proactively detecting and mitigating poisoning attacks in B5G networks.
Jiang et al. [56]	SDN	None	Architecture	Local	CERNET	○	It employs a closed-loop parameter update mechanism and uses DeepAR, a recurrent neural network model, for probabilistic forecasting in slicing admission.
Our Proposal	Slicing Architecture	Any	Architecture	Local & Distributed	CIC IDS 2017 & 5GAD	●	`ML-Agents` and `Security-Agents` deployed as daemonsets in slicing microservice architectures, providing security for each slicing control-plane entity and intra-slice.

combat these attacks. The study examines the efficiency and effectiveness of REPEL through a queuing model and an experimental testbed utilizing a virtualized evolved packet-core prototype.

Chilukuri et al. [55] present SENTINEL. This framework utilizes the Self-Organizing Network (SON) paradigm and learning based on Hierarchical Temporal Memory (HTM) to protect the 5G core control plane from DDoS attacks. The framework detects and isolates malicious users attempting to attack the 5G core using a slice aggregator and Multi-factor Authentication (MFA). The efficacy of SENTINEL is evaluated through experimentation on a 5G testbed and utilization of a semisynthetic dataset of anomalies. The results indicate that SENTINEL maintains high levels of service availability for legitimate users while avoiding the expenditure of additional resources.

We summarize the related works in Table 1, where the *Network Segment* column refers to the type of network slicing performed by the slicing architecture. The *Standards Compatibility* column refers to the compatibility of the slicing architecture with standardizing entities, such as 3GPP and European Telecommunications Standards Institute (ETSI). The *Security Level* column refers to the type of security feature the network-slicing architecture provides, whether for the network slice service (intra-slice) or the architecture as a whole. The *Training Paradigm* column refers to the method of training AI mechanisms supported by the slicing architecture. The ’*Dataset* column summarizes the datasets of the architectures used in the experimental evaluations. The *Pro-active Security/Action* column refers to how the security mechanisms act in the slicing architecture, being reactive and proactive.

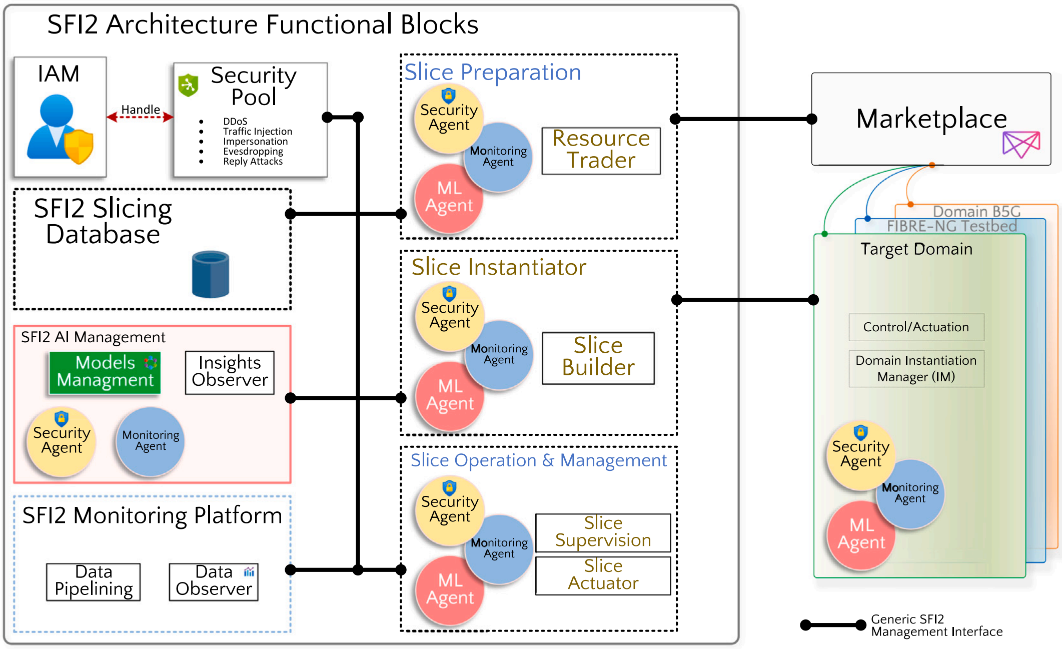


Fig. 2. Architecture Building Blocks.

4. Federated learning empowering sliced testbeds security

Network slicing architectures are designed to meet a specific set of connectivity requirements, based mainly on reference models such as 3GPP and ETSI, among others. In previous work, we presented and validated a reference model for network slicing architectures that addressed aspects not fully covered in the state-of-the-art, such as intrinsic security in the functional blocks of architecture and slicing energy efficiency [23,24,57].

In the literature, there are security approaches for network slices from a connectivity or service perspective, with a predominantly DoS family of attacks [13,25,27]. Furthermore, existing security approaches for network slicing focus on a single type of attack owing to the coupled nature of the slicing architecture design. In this paper, we addressed the operational security of slicing architecture. Our approach allows a slice service provider to handle multiple attacks if the trained ML model is embedded in the `Security Agents`. We applied intrinsic security to slicing architecture through self-adaptive learning techniques in microservices to provide security for architecture operations regarding slicing life-cycle management.

Fig. 2 shows the functional blocks of the SFI2 reference architecture. This architecture envisioned the implementation of network slices considering energy-efficient slicing and integrating experimental networks and testbeds while providing slicing-tailored security functionalities. The functional blocks on the left (Identity And Access Management (IAM), Database, AI Management, Monitoring, Slice Preparation, Slice Instantiator, Slice Operation & Management) cooperate at different stages of the life cycle of a network slice to operationalize the connectivity service. The blocks on the right (Control/Actuation and Instantiation Manager) refer to the different target domains where the SFI2 architecture can deploy network slices in the B5G domains, Future Internet Brazilian Environment for Experimentation (FIBRE) new generation domains [58], well-known experimental testbeds, and others.

The Marketplace maintains and aggregates different target domains and their resources to enable the deployment of network slices during the slice commissioning phase. In all functional layers of the SFI2 architecture, there is provision for the coexistence of two daemon agents, the Security-Agent and the `ML-Agent`. These agents act independently and with other functional blocks of the architecture using a microservices approach.

As mentioned, each functional block of the SIF2 architecture has two complementary services: the `ML-Agent` and the `Security Agent`, which work asynchronously with the slicing architecture. The `ML-Agent` performs passive and active functions in the functional block. The passive role involves the prompt and local response to requests for information from AI or analytics associated with the network traffic of the functional block. On the other hand, the active role involves the distributed processing and training of AI models on data common to the functional block it serves, with the `ML-Agent` periodically reporting the performance of the local model to the SFI2 AI Management for aggregation. In line with [59] findings, to avoid malicious manipulation of system behavior, our approach assumes that all entities in the SFI2 Architecture that support the operation of the network slice life-cycle management have zero trust.

The `Security Agent` is a critical component of the SFI2 architecture, working as a composite service that operates in parallel with slicing architecture entities. Its main responsibility is actively or passively monitoring the functional block for any security threats, including intrusion and DDoS attacks. The `Security Agent` works in close collaboration with the `ML-Agent` microservice (which embeds trained AI models into the `Security Agent`) to detect and prevent malicious traffic patterns in architecture entities, ensuring the overall security and integrity of the SFI2 Architecture. For our implementation, we used the NetData monitoring platform (in `Monitoring Agent`), which can monitor the statistics of both computational nodes and the microservices that run on them; in our case, the Kubernetes DaemonSets.

Our primary aim was to develop a `Security Agent` that operates as a distributed and asynchronous microservice within our network-slicing architecture. In 5G networks, Network Data Analytics Function (NWDAF) provides subscription and notification services to core entities that consume analytics information from this function. In this work, we extended this indirect approach to providing and consuming services for SFI2 security enforcement. Additionally, we propose a novel method for updating threat defense models using federated learning, enabling entities in the SFI2 Architecture in different phases of the network slice life cycle to handle security actions on the fly.

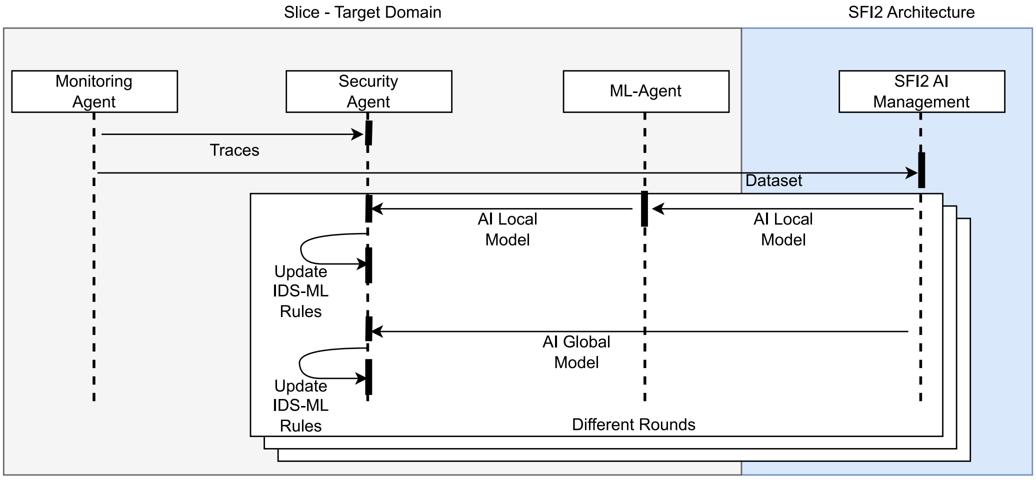


Fig. 3. Secure Flow Interactions.

An example of this cooperation between the Security Agent and the ML-Agent is constructing an IDS (Intrusion Detection System) based on ML (IDS-ML). Fig. 3 illustrates the flow of the interaction between the Security Agent, the ML-Agent, and the SFI2 Architecture. As there will be a Security Agent in each slice created by the SFI2 architecture, it will be responsible for checking any transaction (communication between entities in the slicing architecture) and checking the traffic pattern according to the embedded AI model.

The Monitoring Agent is capable of offering or directing anonymized network flows (traces) such that SFI2 AI Management can store them as a dataset to allow updates to AI models in the future. The SFI2 AI Management entity relates to the ML-Agents by inducing these agents to train AI models with global data from SFI2 AI Management or with local data, where each ML-Agent is embedded. These traces are transformed into an ML-based input and fed ML-Agent, which generates a local AI model related to detecting attacks in the context of that specific slice. The Security Agent updates the IDS-ML rules from this model.

Our slicing architecture extends the literature by allowing AI models of each slice (or core entity) to be used to generate a global ML model for the entire SFI2 architecture. Our architectural design decision was based on the scalability requirement of the NS architecture, so we envisioned a microservice-based architecture with different actuator agents. Each ML-Agent sends its ML model to SFI2 AI Management, thereby enabling the slicing architecture to handle different DDoS threats. The SFI2 AI Management then creates a global AI model that considers what has been learned by all ML-Agents scattered by the architecture. Using the global AI model, each slice (or Security Agents) updates the rules in IDS-ML, effectively mitigating potential attacks on the core entity of the slicing architecture. Finally, the AI models (local and global) are updated in different rounds and continuously improved over time.

It should be noted that the SFI2 architecture does not have access to the data used to create the AI model, only to the AI model generated by the ML-Agent coupled with the specific entities of the microservice-based network slicing architecture. Access to this data is limited to the Security Agent and ML-Agent for each slice (or core entity). What is shared with SFI2 AI Management in the SFI2 architecture is only the AI model, without containing the data generated by the AI model. However, the overall AI model comprises contributions from all slices (or core entities) created with information about threats that the core entity itself did not find. In addition, using native federated learning allows the proposal to be placed in the context of Edge Computing with fewer computational resources. This is because the slice (or core entity) can have a more complex AI model at the edge without requiring significant computational power to train and generate the model.

5. Experimental evaluation

This paper examines the potential for collaboration between the ML-Agent, Security Agent, and Monitoring Agent to improve the functionality and operational security of network slicing architectures and intra-slice. To achieve this goal, we assessed two (2) experimental perspectives. The first involved evaluating the ML-Agent's ability to recognize reconnaissance and DoS attacks in a sliced 5G core network, using generic non-intrusive monitoring metrics provided by the Monitoring Agent feeding the ML-Agent to identify those threats. The second perspective focuses on detecting anomalies within network slicing architectures' operational components (building blocks), spanning slice preparation, slice implementation, and slice operation and management entities using federated learning.

We have developed a progressive experimental framework, beginning with centralized and classical algorithms to address intra-slice security threats in instantiated and active services. Subsequently, we elevate the analysis of defense mechanisms against security threats by leveraging the unprecedented generalization capabilities of FL. This approach enables the construction and validation of slice architectures capable of addressing threats from two perspectives: (1) within the deployed service (intra-slice); and (2) from the operational standpoint of the control plane empowered by FL.

5.1. Intra-slice anomaly detection

In this first experiment, we validate the ability of our architecture to deal with threats involving the running service or deployed network slice. In state-of-the-art, network-slicing architectures, security solutions deal predominantly with the operational security of the architecture, striving to maintain the confidentiality, availability, and integrity of operational components. On the other hand, our proposal sheds light on the security improvement for the service in operation, or intra-slice, by guaranteeing the security aspects for the running network slices on the tenant.

This experiment is based on monitoring a network slice during its operation. We built a Monitoring Agent to collect metrics from the network slice and feed the ML-Agent and the Security Agent. Among the advances in this study, when monitoring the running network slice, we protect it from privacy when the Monitoring Agent inspects the packets' contents, only volumetric and statistical aspects [60]. We collected metrics such as network consumption, Central Processing Unit (CPU), and memory of running network slice.

Table 2
Generic slice metrics and their descriptions.

Metric	Description
cpu	CPU usage by the entity.
cpu_limit	Maximum allowed CPU usage for the entity.
throttled	Number of times the CPU was throttled or restricted.
throttled_duration	Total duration of CPU throttling.
mem	Total memory usage by the entity.
writeback	Amount of data being written back to the disk.
mem_activity	Activity related to memory usage, such as accesses and modifications.
pgfaults	Number of page faults that occurred.
mem_usage	Current amount of memory in use.
mem_usage_limit	Maximum allowed memory usage.
mem_utilization	Percentage of memory utilization.
mem_failcnt	Count of failed memory allocation attempts.
net_eth0	Network traffic on the eth0 interface.
net_carrier_eth0	Carrier (signal) status of the eth0 network interface.
net_packets_eth0	Number of network packets transmitted and received on the eth0 interface.
net_errors_eth0	Number of network errors on the eth0 interface.
net_drops_eth0	Network packets dropped on the eth0 interface.
net_fifo_eth0	Number of FIFO errors on the eth0 interface.
net_events_eth0	Network-related events on the eth0 interface.
throttle_io	Rate of I/O (input/output) throttling.
throttle_serviced_ops	I/O operations that were throttled.
pids_current	Current number of active processes.

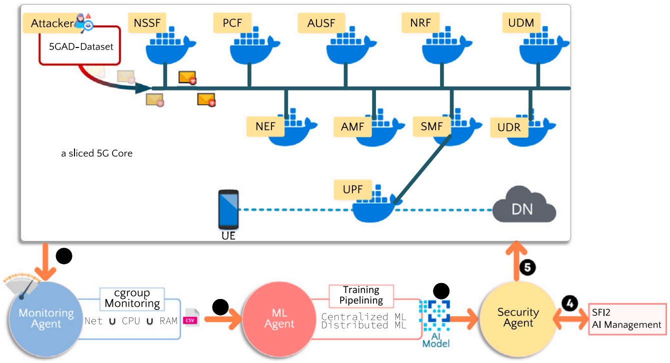


Fig. 4. `Monitoring Agent`, `ML-Agent` and `Security Agent` pipelining for intra-slice anomaly detection.

5.1.1. Description of test

We validated the feasibility of using generic non-intrusive metrics to assess anomaly detection using basic ML algorithms. We employed K-Nearest Neighbors (KNN), Decision Tree (DT), and Random Forest (RF) to handle the resource consumption dataset to predict anomalies in a running network slice containing a 5G core. Our test aimed to validate the performance of these algorithms for anomaly detection in a 5G core. In contrast, validate the collaboration of `Security Agent`, `ML-Agent`, and `Monitoring Agent`.

Fig. 4 shows the experimental scenario. Initially, we instantiated an "Attacker" container equipped with *Packet Captures (PCAPs)* with traces of attacks on 5G core entities and with the premise of being connected to the 5G core control plane network. Packets are injected into the deployed core using the TCPReplay tool [61]. In phase one (1), the `Monitoring Agent` is an instance of NetData running in a container that collects different CPU, memory, and networking metrics based on the Docker Control Group (cgroup), as detailed in Table 2. The metric records were transformed into the features of the running slice resource consumption dataset. Some monitored metrics in Table 2, such as *net_packets_eth0*, have additional attributes, including *received*, *sent*, and *multicast*, resulting in a final dataset with 42 features and labels.

Following, as Fig. 4 we perform the union operation (∪) based on the timestamp, taking the resulting Comma-Separated Values (CSV) to the `ML-Agent` in step two (2). The resulting CSV in step two (2) is a new resource behavioral dataset employed to validate our method. Different algorithms and ML are applied to the data in this phase. The `ML-Agent` service instance responds to the remote procedure call of SFI2 AI Management that starts the model training life cycle. We employed classic algorithms such as KNN, RF, and DT to validate our contribution. In phase three (3), when the training of the AI models is complete, the Security-Agent receives the trained model. In phase four (4), the trained model weights integrate the SFI2 AI Management model pool, which can serve this model for further requests. In phase five (5), the `Security Agent` can now perform anomaly detection based on the current running network slicing.

5.1.2. Dataset

The 5G Attack Detection (5GAD-2022) dataset consists of intercepted 5G network data, including both normal and malicious traffic in Packet Capture (PCAP) files. The normal data was generated by simulating typical user activities like streaming videos, making web requests, and joining video conferences. The malicious data includes ten types of attacks categorized into reconnaissance, DoS, and network reconfiguration. These attacks exploit vulnerabilities in the 5G Core network [62].

The dataset was collected in a simulated environment using open-source software free5GC and User Equipment (UE) simulator [63]. Network traffic was captured using Wireshark, focusing on the application layer to ensure that attack packets were fully included. Each packet was truncated or padded to 1024 bytes to standardize the data for the machine learning model training [62]. In our experiment, the *PCAPs* were previously processed by changing the source and destination Internet Protocol (IP) to enable correct forwarding to core entities.

The dataset consists of CPU, memory, and network metrics of all free5GC core entities. Here, AMF, Authentication Server Function (AUSF), Charging Function (CHF), Non-3GPP Interworking Function (N3IWF), N3 Interface for Untrusted non-3GPP User Equipment (N3IWUE), Network Repository Function (NRF), Network Slice Selection Function (NSSF), Policy Control Function (PCF), Session Management Function (SMF), Unified Data Management (UDM), Unified Data Repository (UDR), UE and User Plane Function (UPF).

This *PCAP* dataset was reinforced in our experimental testbed, leading the `Monitoring Agent` to record new behavioral resource consumption. We labeled our slice resource consumption dataset according to the original *PCAP* dataset. Empirically, we have established

Table 3
Security Agent performance in anomaly detection: Values highlighted within the rectangle represent the highest average F1-scores among entities directly impacted by intra-slice attacks.

Sliced 5G Core Entity	Accuracy (%)			F1-score (%)			Recall (%)			Precision (%)		
	DT	KNN	RF	DT	KNN	RF	DT	KNN	RF	DT	KNN	RF
AMF	99.04	99.00	99.04	$\boxed{97.14}$	97.00	97.13	95.27	94.87	95.08	99.22	99.40	99.42
AUSF	92.58	92.59	92.58	67.10	67.02	67.00	61.83	61.75	61.74	94.74	95.36	95.14
CHF	100	99.98	100	100	99.94	100	100	99.90	100	100	99.99	100
N3IWF	94.45	94.49	94.50	78.47	78.51	78.62	71.51	71.44	91.58	96.30	97.01	96.78
N3IWUE	92.75	92.76	92.78	68.14	68.01	68.20	62.58	62.45	62.60	95.49	96.29	96.10
NRF	100	99.92	100	$\boxed{100}$	99.77	100	100	99.58	100	100	99.96	100
NSSF	92.60	92.58	92.58	67.24	66.74	$\boxed{66.95}$	61.94	61.51	61.69	94.76	96.21	95.35
PCF	96.44	96.56	96.53	87.99	$\boxed{88.27}$	88.19	82.47	82.35	82.34	96.65	97.87	97.63
SMF	100	99.96	100	100	99.88	$\boxed{100}$	100	99.79	100	100	99.98	100
UDM	92.58	92.61	92.58	67.00	$\boxed{67.00}$	67.00	61.74	61.71	61.74	95.14	96.00	95.14
UDR	97.32	97.29	97.36	91.26	91.10	$\boxed{91.38}$	86.41	86.07	86.48	98.10	98.34	98.32
UE	100	99.99	100	100	99.97	$\boxed{100}$	100	99.95	100	100	99.99	100
UPF	99.95	99.89	99.99	99.86	99.68	$\boxed{99.97}$	99.79	99.61	99.95	99.93	99.75	99.99

a new dataset derived from 5GAD with a proportion of 90% benign instances and 10% malignant instances with precision of one second. This was done to simulate a real-world scenario where benign traffic is more prevalent than malignant traffic.

5.1.3. *Evaluation*

For our evaluation, we used the Fabric testbed, a nationwide testbed on which we deployed a virtual machine with 32 GB of memory and 16 cores with an Ubuntu 20.04 operating system, containing scikit-learn, Docker 27.1 and Python 3.11. We started the architecture containers (available here https://github.com/romoreira/SFI2_B5G_Security), instantiated the "Attacker" node and the sliced 5G core based on free5GC.

We performed 10-*fold* cross-validation on the dataset to ensure that all training data were used as test instances to avoid overfitting. As shown in Table 3, the classical ML models embedded in the Security Agent can identify anomalies in the running network slice (intra-slice). In Table 3, we compare the performance of the algorithms according to different metrics such as accuracy, F1-Score, recall, and precision.

Accuracy is the proportion of correctly classified instances: Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$. Precision measures true positives among predicted positives: Precision $= \frac{TP}{TP+FP}$. Recall, or Sensitivity, measures true positives among actual positives: Recall $= \frac{TP}{TP+FN}$. The F1-score, the harmonic mean of precision and recall, is given by F1-Score $= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$. These metrics offer a comprehensive view of model performance, with F1-score being especially useful for imbalanced datasets. Our results is presented in Table 3.

In this experiment, we reinjected packets into a sliced 5G core to trigger reconnaissance, network reconfiguration, and DoS attacks. Specifically, we cause AMF to generate fraudulent requests to UDM while impersonating AMF. We also injected the "Get All Network Functions" attack without specifying the network function, causing NRF to behave in a Byzantine manner. We also triggered a "Random Data Dump" that refers to deliberately requesting nf-instances from NRF. We triggered "Automatic Redirect with Timer" attacks, which caused UE traffic to be temporarily redirected by changing policies in UPF.

We inject packets leading to the network reconfiguration phenomenon, thus causing "Fake AMF Delete" causing the sliced 5G core to lose connectivity with the AMF. Similarly, we provoke "Random AMF Insert" scenarios that generate new fraudulent instances of the AMF. These events triggered a change in the consumption pattern of the CPU, memory, and network resources of the running network slice. Thus, the Security Agent was able to identify these intra-slice nuances, leading to the performance presented in Table 3.

Table 4
Intra-Slice security defense mechanism comparison.

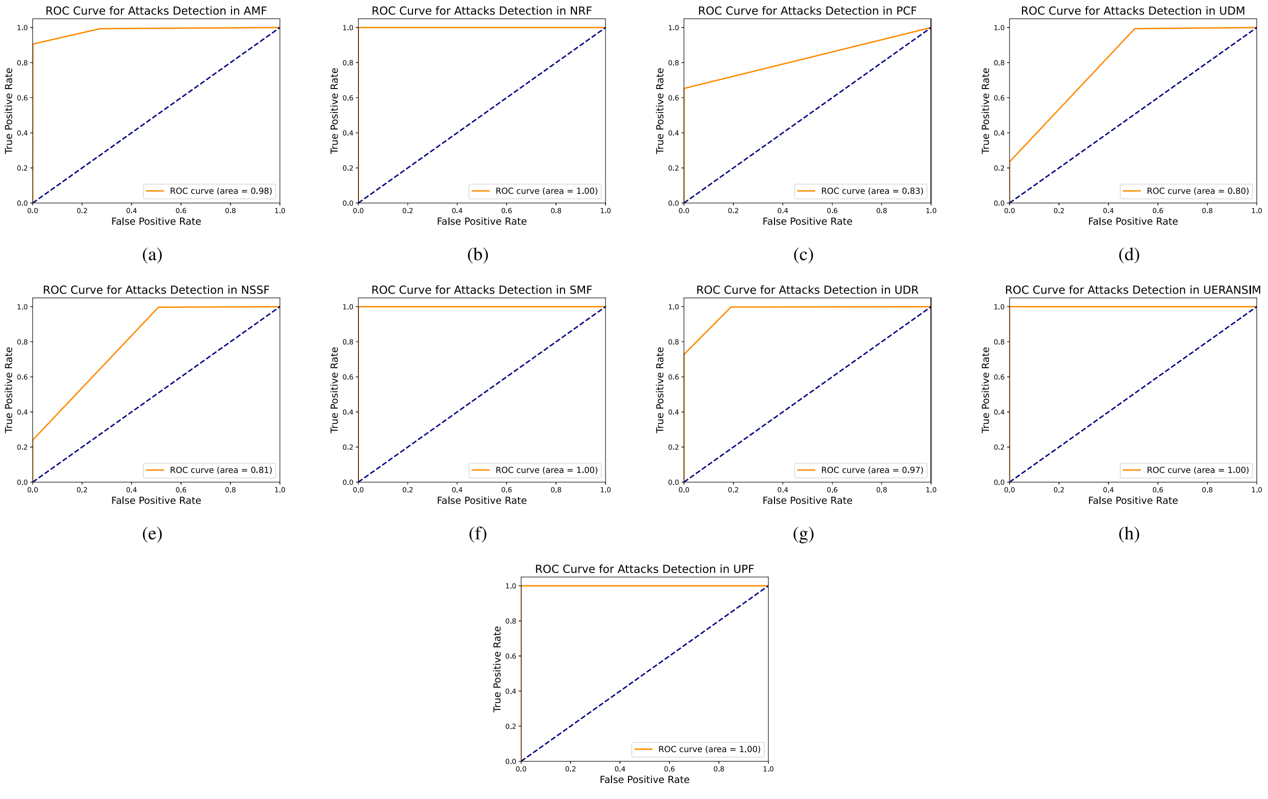
Approach	Dataset	Security threat	Employed method	On-time detection	Low-overhead monitoring	Defense efficiency
Boualouache et al. [47]	CSE-CIC-IDS 2018 [64]	DoS	Deep Learning (DL)	○	○	Accuracy: 99.00%
Hossain et al. [65]	VeReMi [66]	DDoS	DL with Knowledge Destilation (KD)	○	○	Accuracy: 99.00%
Majeed et al. [67]	CTU-13 [68]	BotNet	DL	○	○	Accuracy: 97.74%
Boualouache et al. [69]	5G-NIDD [70]	DoS	FL	●	○	F1-Score 88.00%
Our Approach	5GAD [62]	DoS	Classic ML	○	●	Accuracy: 100%

Finally, we cause a DoS attack, specifically the "Crash NRF Attack" behavior, where malformed requests are deliberately sent to the entity, causing it to fail and triggering abnormal behaviors in the resources of the other entities of the sliced 5G core. In addition to the "automated drop with timer" and "automated redirect with times" behavior that alternates between redirecting or dropping the traffic of the UE in the UPF, this leads to anomalous resource consumption in the control entity.

Specifically, the attacks we simulated to evaluate our method directly affected the entities AMF, NRF, NSSF, PCF, SMF, UDR, UDM, UE, and UPF, as listed in Table 3. Therefore, we observed in more detail the performance of the embedded AI model generated by the ML-Agent combined with the Security Agent on these entities. As shown in Fig. 5, the graphs contain the ROC curve, showing a trade-off between sensitivity and specificity.

The Area Under the ROC Curve (AUC) quantifies the overall ability of the model to discriminate between positive and negative classes. An AUC value closer to 1 indicates good classification, while a value of 0.5 suggests random guessing. A higher AUC value represents a better model performance in distinguishing between classes. The results in Fig. 5 suggest that the Security Agent combined with the ML-Agent can adequately identify intra-slice anomalies. A higher area under the curve (AUC) indicates better performance in distinguishing between legitimate traffic and DoS attacks for the Security Agent, reflecting the system's effectiveness in accurately identifying attacks while minimizing false alarms.

Table 4 presents a thorough comparison of existing intra-slice security defense mechanisms, positioning our proposed approach within the current research landscape. The comparison covers key aspects,



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

(i)

Fig. 5. ROC curve for the main entities directly impacted by the intra-slice attacks.

including the dataset used, type of security threat addressed, employed method, on-time detection capabilities, low-overhead monitoring, and overall defense efficiency. By comparing various approaches, such as those based on DL, FL, and traditional ML, our method achieved 100% accuracy in detecting DoS attacks on the 5GAD dataset through non-intrusive monitoring. This comprehensive analysis underscores the robustness of our approach, particularly in achieving superior detection accuracy without increasing monitoring overhead, thereby contributing to the advancement of security solutions in network slicing.

5.2. Anomaly detection in architecture building blocks

In this second experiment, we propose a formal evaluation of security-native new advances, showcasing it on the SFI2 Architecture through experiments in which each federated client processes a local dataset [48] consisting of network flows generated by FlowMeter [71]. This tool creates tuples of network flows based on the statistical grouping of network packets. Once the packet capture file is generated (*PCAP*), it is converted into a *CSV* containing 78 features for each network flow. In this experiment, we validated the “Security Level” feature (Table 1) of our architecture based on its ability to handle security threats from an architectural perspective. We have advanced the SFI2 Architecture by empowering each functional entity to manage security threats in the lifecycle control flow on the fly.

We assume that all SFI2 Architecture microservices are ready to handle the network slice lifecycle. In our experimental evaluation, we used `SFI2 AIManagement` to trigger a federated learning scenario using *non*–Informally, Identically Distributed (IID) data. Each federated client (see Fig. 2 as `ML-Agent` and associated with microservices as a daemon set) had access to a dataset of a specific type of intrusion and DDoS attack. The data in each `ML-Agent` may not follow the same distribution nor may have the same characteristics as the data on other devices or the overall population, referred to as *non*-IID data. This presents challenges for federated learning, such as slow convergence, poor accuracy, and model divergence.

We analyzed the optimal hyperparameters for each participant, considering their respective local training sets. In addition, we used the Bayesian approach with the help of the Optuna hyperparameter optimization framework [72]. As depicted in Fig. 6, the neural network comprises distinct structures, layers, and input and output data. We sought to minimize loss by considering the number of neural network layers, optimizer, learning rate, and epochs. This neural network architecture was chosen after a previous hyperparameter optimization process, which sought to determine the optimal number of layers to minimize the loss.

We embedded the Long Short-Term Memory (LSTM) model shown in Fig. 6 designed for the efficient processing of sequential data in our `ML-Agent`. The model starts with an input layer that maps 78 input features (detailed in Section 5.2.2) to 16 units, followed by a hidden layer with the same dimensionality, employing Rectified Linear Unit (ReLU) activation, and a dropout layer with a probability of 0.4 to prevent overfitting. The output layer reduces the feature space to two units for binary classification (DDoS or *non*-DDoS). During training, our model has nodes such as *AccumulateGrad*, *TBackward0*, *AddmmBackward0*, and *SoftmaxBackward0*, which represent operations and gradient accumulations, respectively. For example, *input_layer.weight* and *output_layer.weight* with shapes (16, 78) and (2, 16), respectively, are crucial in forward passes, whereas backward operations such as *ReluBackward0* and *SoftmaxBackward0* ensure accurate gradient computation during the backward pass, culminating in the final output of the shape (32, 2).

5.2.1. Description of test

We have formalized the evaluation of our SFI2 reference architecture security extension by conducting experiments on a testbed that replicated the production network conditions. We deployed a

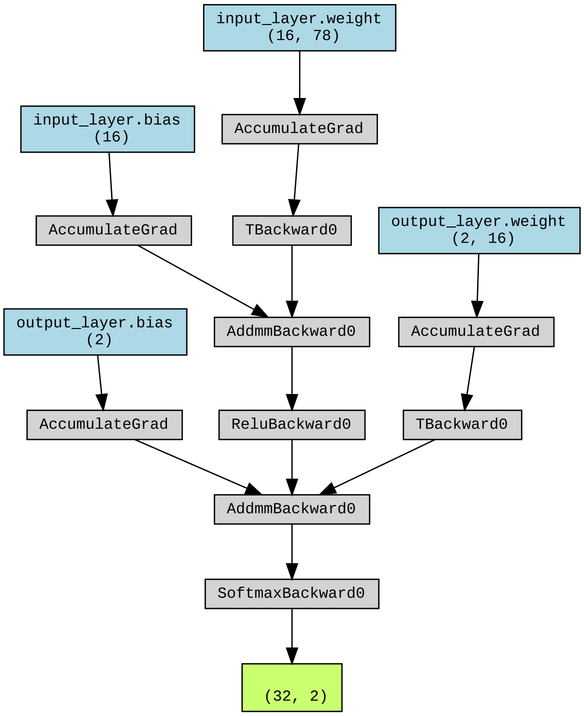


Fig. 6. The Neural Network used by each ML-Agent.

virtual machine with 32 GB of memory, an 8-core CPU, and a GPU RTX 4060 Ti with 8 GB of memory, Ubuntu 20.04 operating system. Flower federated learning framework, cuDNN 12.0 toolkit combined with Torch 2.3. Our dataset consisted of benign and malicious network traffic [48]. The testing process was divided into two phases. The first phase involves offline training of deep neural network algorithms in a local and federated manner. In the second phase, we implemented the learned models in the SFI2 architecture, specifically in the distributed scenario of the testbed, where each ML-Agent runs as microservices in different architectural blocks.

The second experiment involved running the functional blocks of the SFI2 Architecture on different testbed nodes. The SFI2 prediction API can receive a network flow in tuple format and judge its traffic class, benign or malignant. Therefore, we measured the API response time capacity to assess the API readiness regarding the response of our architecture when running production slices.

Finally, validation was performed on a nationwide physical testbed in the Future Internet Brazilian Environment for Experimentation New Generation testbed, which is a microservice-based testbed with many compute nodes spread across educational institutions in Brazil and is designed to be an evolution of the previous FIBRE testbed supported by the National Education and Research Network (RNP) [58]. This network is geographically distributed and has an interconnection between Kubernetes nodes via an Internet Protocol (IP) network that connects different research institutions in Brazil.

In our current implementation, we considered only a centralized coordinator within the testbed. This decision was made because the centralized coordinator is located in the AI agent of the SFI2 architecture, which is protected by a security agent. It is designed specifically for and is accessible only to SFI2 tenants, ensuring a secure and controlled environment for federated learning processes. However, we recognize the potential benefits of using blockchains and distributed coordinators for federated learning, particularly in enhancing participant security, transparency, and trust. As such, exploring these approaches represents a valuable direction for future work, where decentralized coordination mechanisms could further strengthen the system's resilience and scalability further [73].

Table 5
Dataset file description and distribution.

Day	Size (# lines)	% of Malignant	Assigned to which ML-Agent
Tuesday	445,909	3.1	#1
Wednesday	692,703	36.48	#2
Thursday	288,602	0.01	#3
	170,366	1.28	#4
Friday	286,467	55.48	#5
	191,033	1.03	#6
	225,745	56.71	#7

5.2.2. Dataset

We chose four days to train and validate the deep neural networks, encompassing tuples of network flows from different days and times. We used 90% of the time for training and 10% for testing, and each experiment was performed 10 (ten) times. Each capture or dataset acquisition day was assigned to a single federated client during the learning process, as listed in Table 5. The dataset was divided as follows: Monday featured only regular activity (normal traffic with different applications), whereas Tuesday through Friday included hybrid attacks and regular activity. The neural network structure and the layers employed for each federated client are shown in Fig. 6.

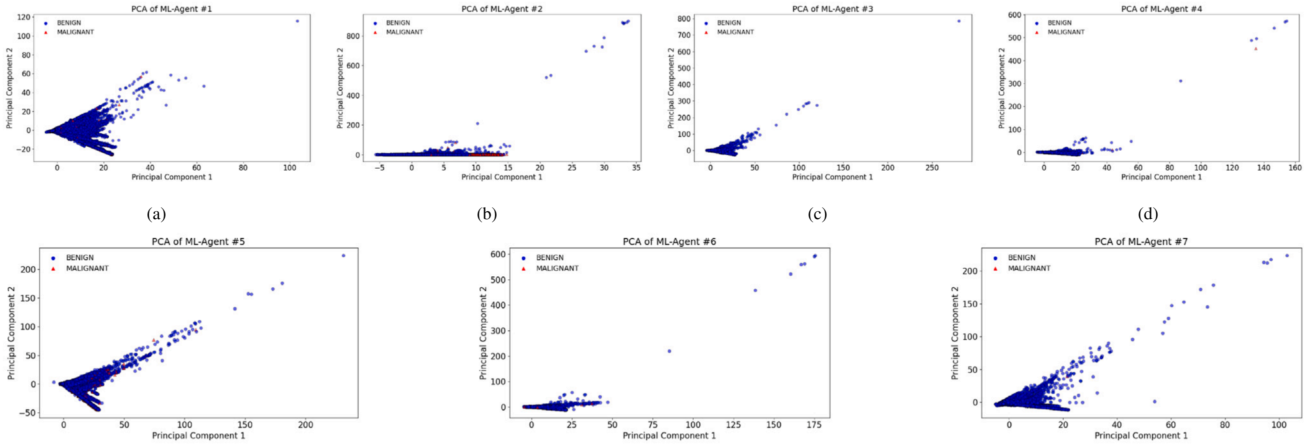
Within just four (4) days of the existing dataset [48], time divisions into morning, afternoon, and evening led to the separation of more than four (4) datasets for each ML-Agent, as shown in Table 5 (column "Assigned to which ML-Agent"). The dataset comprises a real network encompassing various devices such as firewalls, switches, and routers. We previously trained models for the SFI2 AI Management building block, allowing future microservices of the SFI2 reference architecture to import the model. Later, using the trained and operational model, we validated the performance of the security API in classifying network flows from both network slices and functional blocks of the architecture.

The chosen dataset represents the diversity of devices and user behaviors it brings. The dataset was constructed to include various intrusions and benign traffic at different times of the day to capture the unique characteristics of each time slot. The first day of the week, Monday, was excluded from the local training process of our experiment because it consisted of only regular traffic. The remaining days were considered for training, as they contained a mixture of benign and malignant traffic.

To understand the type of data ML-Agents have dealt with, we conducted a Principal Component Analysis (PCA), which is a statistical technique used for dimensionality reduction. PCA reduces dimensionality by transforming the original variables into a new set of variables, the main components. The PCA-generated scatter plots reveal that classes have considerable overlap within features, which may lead to difficulties in achieving high accuracy or convergence in terms of learning in certain AI models. Fig. 7 shows that the malignant and benign classes are mixed, indicating an intrinsic classification challenge for ML-Agents. To migrate some of this overlap, we conducted a hyperparameter optimization.

5.2.3. Analysis method

We present the results of optimizing the hyperparameters of each ML-Agent using its local dataset, where the hyperparameters were refined using the Tree-Structured Parzen Estimator (TPE) algorithm [74] to maximize the accuracy of each model coupled to the local ML-Agent. Table 6 lists the search space and hyperparameters. It is worth noting that the *non*-IID format of the dataset managed by each ML-Agent resulted in the discovery of diverse hyperparameters, even when using the same neural network for every ML-Agent. In our experiments, it was necessary to optimize the hyperparameters, as we



(a) (b) (c) (d)

(e) (f) (g)

Fig. 7. PCA of each ML-Agent dataset.

Table 6
Dataset description and Hyperparameters for each ML-Agent.

ML-Agent	Dataset	Attacks	Learning rate (LR)	Optimizer	Epochs
#1	Tuesday	Brute Force, FTP-Patador, and SSH Patator	0.0003074258400864182	Adam	10
#2	Wednesday	DoS/DDoS: Slowloris, Slowhttptest, Hulk, and GoldenEye	0.0005025961155459187	RMSprop	10
#3	Thursday	Infiltration: Dropbox, Meta exploit Win Vista, Cool disk – MAC, Dropbox download, and Win Vista	0.00010603472201401003	RMSpro	10
#4	Thursday	Web Attack: Brute Force, XSS, and SQL Injection	0.00013936442920558617	Adam	10
#5	Friday	Firewall Rules: On and Off	0.000587441102433820	RMSprop	10
#6	Friday	Botnet Ares	0.0006052967400865347	SGD	10
#7	Friday	DDoS LOIT	0.00012091571705782663	Adam	10

found that the hegemonic hyperparameters for each ML-Agent and its dataset in the *non*-IID scenario prevented the global model from converging its learning rate.

Through the process of optimizing the hyperparameters, we were able to ensure that each ML-Agent could train effectively on its respective local dataset. Subsequently, we evaluated the training capacity and performance of each ML-Agent using their local dataset. The ML-Agent column indicates the training agent employed during the experiment. In contrast, the "Dataset" column provides details on the data assigned to the ML-Agent, and the "Attacks" column specifies the types of attacks the ML-Agent was trained to classify/identify.

5.2.4. *Training behavior*

We evaluated the training performance of a neural network using binary classification of malignant or benign traffic. Our native AI and security architecture are flexible enough to support different types of approaches for handling threats defense and training AI models to handle security. Initially, we validated two behaviors of SFI2 AI Management, triggering centralized training or distributed training across ML-Agents coupled as microservices in the architecture. For centralized, we grouped the seven datasets in this experimental scenario, leading to a centralized training approach. Thus, after ten (10) different runs, we obtained an average test accuracy of 90.01%. We ensured that learning was consistent by presenting the loss function and training accuracy graphs in Fig. 8, and Fig. 9.

Although visually, the Accuracy and Loss graphs in Fig. 8 and Fig. 9 appear to fluctuate slightly; in our experiments, the model converged at epoch 10 when there were no more significant gains in accuracy, and we activated the early stopping of the learning process. It should also be noted that the aggregation of the dataset culminated in a dataset with higher CPU and memory consumption, and training took an average of 1072 s.

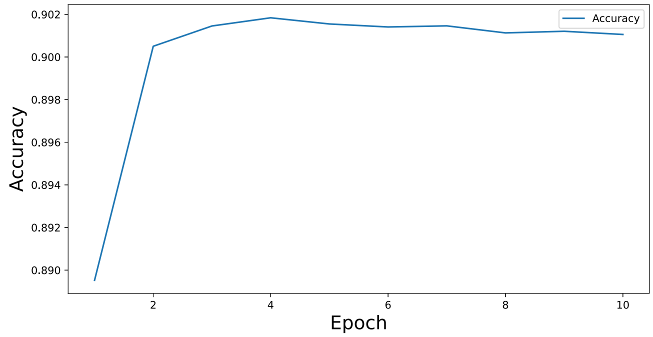


Fig. 8. Accuracy behavior for a joint dataset using a centralized training approach.

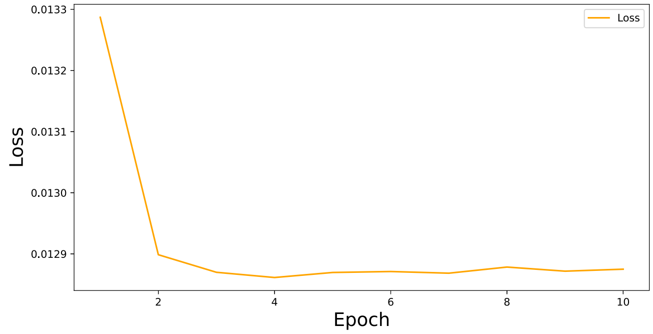
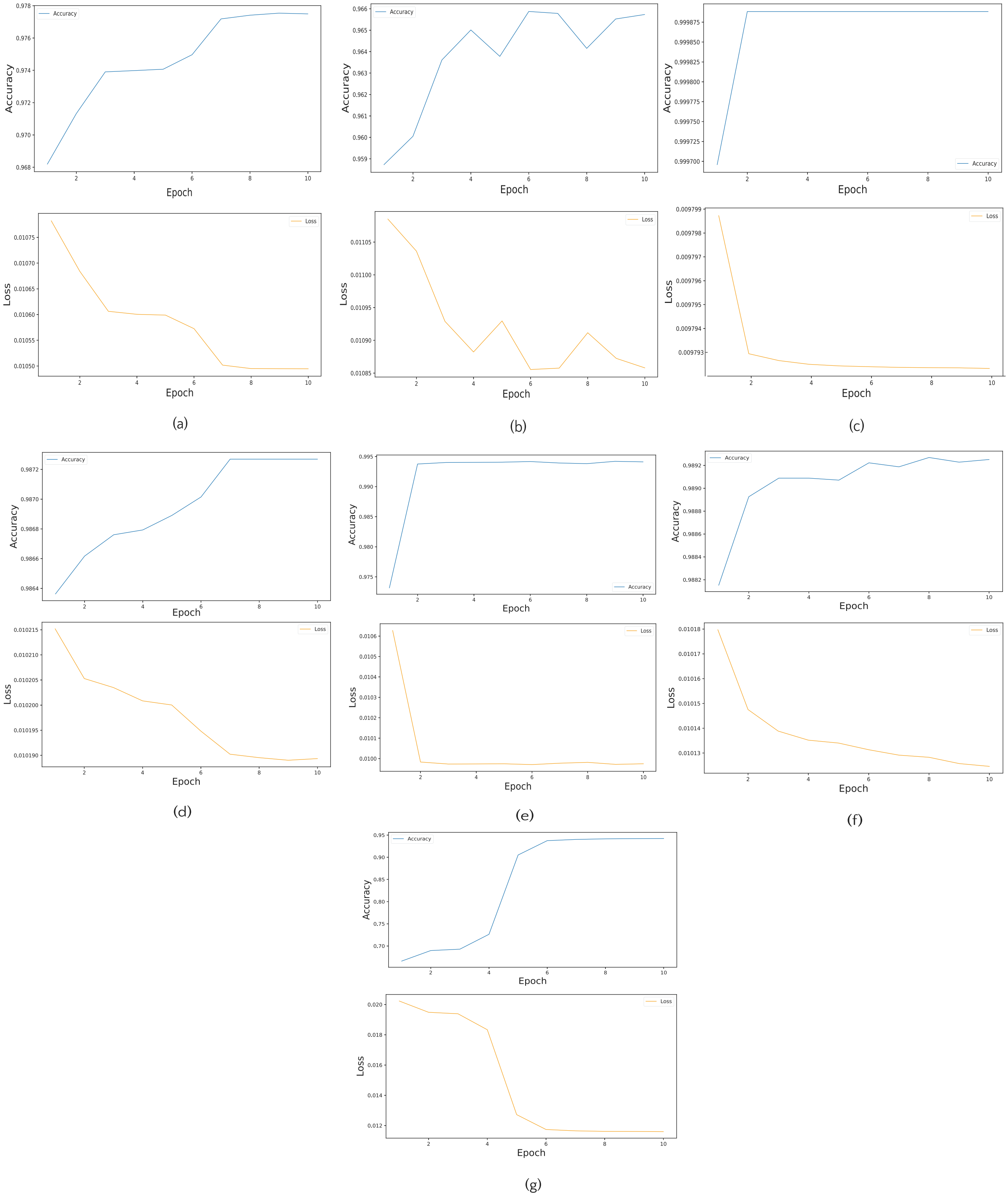


Fig. 9. Loss behavior for a joint dataset using a centralized training approach.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Fig. 10. Accuracies for each ML-Agent. The graphs *(a)* to *(g)* refer to the accuracy and loss behaviors of each ML-Agent with its local dataset according to Table 6.

Subsequently, we analyzed in Fig. 10 the behavior of the accuracy and loss curves for the federated learning scenario involving two training rounds. This distributed scenario aligns with our contribution because it enables each Security Agent to deal with potential security threats as a specific entity in the network slicing architecture. Biases in weight aggregation averages can compromise centralized AI models.

As Fig. 10 shows, each model exhibited different behaviors during the training process. However, it is worth noting that the ML-Agents showed appropriate behavior in the curves, indicating that they could converge in learning. Consequently, the server model achieved an average accuracy of 90.8% using two training rounds. The variation in accuracy and loss levels between the different ML-Agents, as shown in Fig. 10, is an intrinsic characteristic of federated learning in a *non*-IID

Table 7
Comparison of training time considering two different approaches.

Approach	Round	Count	Mean Training Time	Standard Error Mean	Standard Deviation	Minimum	Q1	Median	Q3	Maximum
Federated	Two	280	362.37	6.43	107.57	236.66	257.7	344.06	452.29	619.3
	Four	420	358.55	5.22	107	235.57	255.5	338.67	446.07	622.77
	Eight	980	358.9	3.4	106.34	235.57	256.8	339.78	446.51	622.77
	Sixteen	1832	378.64	2.65	113.24	235.57	277.29	344.95	453.35	628.73
Centralized			**Mean Training Time**	**SE Mean**	**StDev**	**Minimum**	**Q1**	**Median**	**Q3**	**Maximum**
		10	1072.3	0.249	0.789	1070.5	1072.3	1072.5	1072.5	1073.5

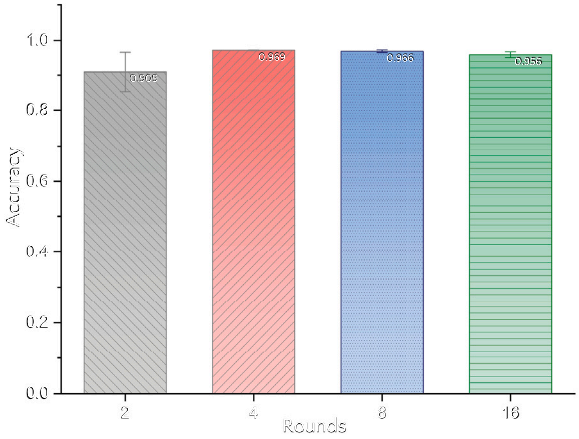


Fig. 11. Accuracy for different Federated Rounds.

Table 8
Descriptive Statistics of cosine divergence for each training round.

	N Analysis	Mean	Standard deviation	SE of Mean
2	7	0.00446	0.02164	0.00818
4	7	−0.00045	0.02043	0.00772
8	7	0.01440	0.01721	0.00650
16	7	0.01746	0.02376	0.00898

Table 9
Analysis of variance test.

	DF	Sum of squares	Mean square	F value	Prob>F
Model	3	0.00149	0.00050	1.13519	0.35479
Error	24	0.01048	0.00044		
Total	27	0.01196			

Table 10
Fit statistics.

R-Square	Coeff Var	Root MSE	Data mean
0.12427	2.33603	0.02089	0.00894

scenario, where each ML-Agent can have a unique learning behavior over the time.

In addition, as shown in Fig. 10 graphs represent how challenging it was for ML-Agents to deal with *non*-IID data while creating a global AI model to empower network slicing architectures, dealing with different types of DDoS and intrusion attacks. Although challenging, the models of each ML-Agent converged in learning because the stabilization of the loss and accuracy after the end of the epochs was noted. With this, we have that each ML-Agent dealt with different types of DDoS attacks and, at the same time, contributed to adjusting a generic and robust AI model for the architecture of networking slicing.

After conducting a thorough analysis of the capabilities of the ML-Agents to train federatively with *non*-IID datasets and reporting the weights to the central model in the SFI2 Architecture, we examined the effect of federated learning rounds on accuracy. Fig. 11 shows the variability of the aggregate accuracy of the model concerning different interaction rounds. Our findings indicate that, with an error of less than 5%, increasing the number of training rounds had no significant influence on the accuracy of the central model. It can, therefore, be deduced that the *non*-IID datasets may not benefit from long federated training rounds.

We present a comparison of the training times for the centralized and federated approaches in Table 7. Subsequently, we analyzed the models resulting from federated clients with the aggregated server model by utilizing cosine divergence to assess the significance of any differences between the server and client models (vector of weights) across various training rounds. A substantial difference suggests that a particular customer may be overlooked because of its minimal impact on the convergence of the model. The average cosine divergence between client models and the server is presented in Table 8.

5.2.5. Analysis of models

Regarding the training paradigm analyses, we compared the models resulting from federated clients with the aggregated server model by utilizing cosine divergence to assess the significance of any differences between the client and server models across various training rounds. Upon obtaining the samples and the average cosine divergence between the client models and server, we analyzed the variance (ANOVA) to evaluate the statistical equivalence of these samples. We employed an ANOVA with four levels, each representing a sample of the cosine difference for the different training rounds. We formulate the following hypotheses for our analysis:

- *Null Hypothesis*: The means of all levels are equal.
- *Alternative Hypothesis*: The means of one or more levels are different.

The ANOVA test results presented in Table 9, especially the *p*-value, indicate no statistically significant difference between the means of the four variables; namely, increasing the number of training rounds in federated learning does not affect the accuracy achieved. Specifically, the value of *p* is 0.35479, indicating that we should accept the null hypothesis that the means are equal at a significance level of 5%.

According to Table 10, the R-squared value is 0.12427, which indicates that the model accounts for only 12.43% of the variation in the data. Hence, it can be inferred that other factors, such as the size of the dataset, the type of algorithm employed, and the quality of communication between nodes, significantly influence the model's accuracy beyond the number of training rounds.

Fig. 12 shows the statistical equivalence between the cosine differences and Standard Error (SE) of the mean according to the results of the ANOVA test. This implies that the model can learn and improve, albeit without statistically significant improvements. Additionally, the model seems to converge towards a solution as it progresses towards distributed learning.

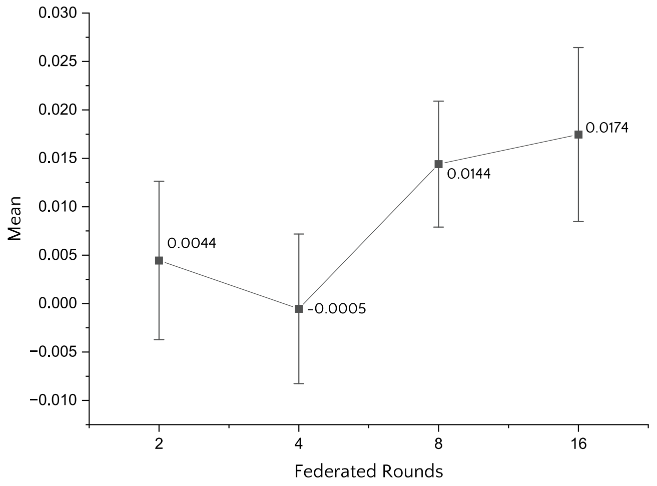


Fig. 12. Cosine Variation through different training Rounds.

Table 11
Comparison of FL-based Security Defense Mechanisms for NS Architectures.

Approach	Dataset	Security Threat	Employed method	Multiple NS architecture support	Pro-active monitoring	Low-overhead monitoring	Defense efficiency
Niboucha et al. [52]	Own	DDoS	Gradient Boosting	○	○	○	Accuracy: 96.76%
Wijethilaka et al. [37]	NSL-KDD [75]	DoS, and User to Root Attack (U2R)	DL	○	●	○	Accuracy: 99.99%
Sedjelmaci and Boualouache [76]	CSE-CIC-IDS-2018 [64]	DDoS and Botnet	Mean-field Game	○	○	○	Accuracy: 97.00%
Wijethilaka et al. [20]	NSL-KDD [75]	DDoS, Man-in-the-Middle (MITM), and Botnet	DL	○	●	○	Accuracy: 98.00%
Rumesh et al. [77]	Own	DDoS	LSTM	○	●	○	Accuracy: 99.87%
Mirzaee et al. [78]	NSL-KDD [75]	DoS and U2R	DL	○	●	○	Accuracy: 99.50%
Thantharate [79]	Own	DDoS	Sequential Model	○	○	○	Precision: 94.00%
Our Approach	CIC-IDS2017 [48]	DDoS	LSTM	●	●	●	Accuracy: 96.60%

The Table 11 presents a comparative analysis of various state-of-the-art approaches in addressing different security threats using distinct datasets and employed methods. Each approach is evaluated based on its capability to support multiple network slicing (NS) architectures, provide proactive monitoring, implement low-overhead monitoring, and its overall defense efficiency. The results indicate that the accuracy or precision of these methods varies. Our proposed approach, utilizing the CIC-IDS2017 dataset with an LSTM model, demonstrates competitive accuracy at 96.60%, while also meeting the criteria for multiple NS architecture support, proactive monitoring, and low-overhead monitoring.

6. Concluding remarks

In this paper, we presented a microservice-based approach to enhance the intelligence and security of network-slicing architectures. Our research reveals that several network slicing architectures lack the necessary intelligence and security features to effectively operate and protect network slices. To address this shortcoming, we propose using `ML-Agents` and `Security Agents`, which collaborate to provide intelligent and secure management and orchestration for network slice core entities.

Our research demonstrated that federated learning, when associated with microservice architectures, can enhance network-slicing architectures, improve their resilience to various security attacks, and build robust AI models for attack prediction for architecture blocks and intra-slices. We conclude that federated learning can make an architecture more resilient to threats, mainly by providing on-demand adjustments and adapting to changing data. In addition, we evaluated the behavior of training rounds in federated learning. We determined that the number of training rounds was insignificant for constructing these AI models.

The results of this study offer new insights into the evolution of architectures and frameworks for network slicing, allowing their extension to the design of architectures adapted to security that support the requirements of new applications and business verticals.

For future work, we plan to focus on enhancing our solution by integrating new machine learning models to bolster its capacity to respond to security threats. Additionally, we aim to explore the application of reinforcement learning to ensure the robustness of network slicing architectures, even in novel attack scenarios. We also intend to investigate the impact of different security methods on service-level agreements and operator revenue.

This study reports recent advancements and highlights significant research opportunities in intelligent and security-aware resource sharing for future network architectures.

CRediT authorship contribution statement

Rodrigo Moreira: Software, Conceptualization. **Rodolfo S. Villaça:** Writing – review & editing, Data curation, Conceptualization. **Moisés R.N. Ribeiro:** Writing – original draft, Validation, Supervision. **Joberto S.B. Martins:** Writing – original draft, Validation, Supervision, Formal analysis, Data curation. **João Henrique Corrêa:** Methodology, Investigation, Data curation. **Tereza C. Carvalho:** Visualization, Project administration, Funding acquisition. **Flávio de Oliveira Silva:** Writing – review & editing, Supervision, Resources, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Tereza Cristina Melo de Brito Carvalho reports financial support was provided by State of Sao Paulo Research Foundation. Joberto Sergio Barbosa Martins reports financial support was provided by Anima Institute. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

We acknowledge the financial support of the Brazilian National Council for Scientific and Technological Development (CNPq), grant #421944/2021-8 and the FAPESP MCTIC/CGI Research project 2018/23097-3 - SFI2 - Slicing Future Internet Infrastructures. The authors also thanks CNPq, CAPES, FAPES and Instituto ANIMA.

References

[1] H.P. Phyu, D. Naboulsi, R. Stanica, Machine learning in network slicing—A survey, IEEE Access 11 (2023) 39123–39153, http://dx.doi.org/10.1109/ACCESS.2023.3267985.

[2] M.A. Habibi, B. Han, A. Fellan, W. Jiang, A.G. Sánchez, I.L. Pavon, A. Boubendir, H.D. Schotten, Toward an open, intelligent, and end-to-end architectural framework for network slicing in 6G communication systems, IEEE Open J. Commun. Soc. 4 (2023) 1615–1658, http://dx.doi.org/10.1109/OJCOMS.2023.3294445.

[3] C.D. Alwis, P. Porambage, K. Dev, T.R. Gadekallu, M. Liyanage, A survey on network slicing security: Attacks, challenges, solutions and research directions, IEEE Commun. Surv. Tutor. (2023) 1, http://dx.doi.org/10.1109/COMST.2023.3312349.

[4] A. Donatti, S.L. Correa, J.S.B. Martins, A. Abelem, C.B. Both, F. Silva, J.A. Suruagy, R. Pasquini, R. Moreira, K.V. Cardoso, T.C. Carvalho, Survey on machine learning-enabled network slicing: Covering the entire life cycle, IEEE Trans. Netw. Serv. Manag. (2023) 1, http://dx.doi.org/10.1109/TNSM.2023.3287651.

[5] R. Moreira, P.F. Rosa, R.L.A. Aguiar, F. de Oliveira Silva, Enabling multi-domain and end-to-end slice orchestration for virtualization everything functions (VxFs), in: L. Barolli, F. Amato, F. Moscato, T. Enokido, M. Takizawa (Eds.), Advanced Information Networking and Applications, Springer International Publishing, Cham, 2020, pp. 830–844.

[6] C. De Alwis, P. Porambage, K. Dev, T.R. Gadekallu, M. Liyanage, A survey on network slicing security: Attacks, challenges, solutions and research directions, IEEE Commun. Surv. Tutor. 26 (1) (2024) 534–570, http://dx.doi.org/10.1109/COMST.2023.3312349.

[7] R. Moreira, P.F. Rosa, R.L.A. Aguiar, F. de Oliveira Silva, NASOR: A network slicing approach for multiple autonomous systems, Comput. Commun. 179 (2021) 131–144, http://dx.doi.org/10.1016/j.comcom.2021.07.028, URL https://www.sciencedirect.com/science/article/pii/S0140366421002917.

[8] S. Fdida, N. Makris, T. Korakis, R. Bruno, A. Passarella, P. Andreou, B. Belter, C. Crettaz, W. Dabbous, Y. Demchenko, R. Knopp, SLICES, a scientific instrument for the networking community, Comput. Commun. 193 (2022) 189–203, http://dx.doi.org/10.1016/j.comcom.2022.07.019, URL https://www.sciencedirect.com/science/article/pii/S0140366422002663.

[9] R. Bolla, R. Bruschi, C. Lombardo, B. Siccardi, 6G enablers for zero-carbon network slices and vertical edge services, IEEE Netw. Lett. 5 (3) (2023) 173–176, http://dx.doi.org/10.1109/LNET.2023.3262861.

[10] N.M. Yungaicela-Naula, C. Vargas-Rosales, J.A. Pérez-Díaz, SDN/NFV-based framework for autonomous defense against slow-rate DDoS attacks by using reinforcement learning, Future Gener. Comput. Syst. 149 (2023) 637–649, http://dx.doi.org/10.1016/j.future.2023.08.007, URL https://www.sciencedirect.com/science/article/pii/S0167739X23003047.

[11] L.F. Rodrigues Moreira, R. Moreira, B.A.N. Travençolo, A.R. Backes, An artificial intelligence-as-a-service architecture for deep learning model embodiment on low-cost devices: A case study of COVID-19 diagnosis, Appl. Soft Comput. 134 (2023) 110014, http://dx.doi.org/10.1016/j.asoc.2023.110014, URL https://www.sciencedirect.com/science/article/pii/S1568494623000327.

[12] D. Oliynyk, R. Mayer, A. Rauber, I know what you trained last summer: A survey on stealing machine learning models and defences, ACM Comput. Surv. 55 (14s) (2023) http://dx.doi.org/10.1145/3595292.

[13] R. Dangi, A. Jadhav, G. Choudhary, N. Dragoni, M.K. Mishra, P. Lalwani, ML-based 5G network slicing security: A comprehensive survey, Future Internet 14 (4) (2022) http://dx.doi.org/10.3390/fi14040116, URL https://www.mdpi.com/1999-5903/14/4/116.

[14] K. Abbas, Y. Cho, A. Nauman, P.W. Khan, T.A. Khan, K. Kondepu, Convergence of AI and MEC for autonomous IoT service provisioning and assurance in B5G, IEEE Open J. Commun. Soc. (2023) 1, http://dx.doi.org/10.1109/OJCOMS.2023.3329420.

[15] D.G.S. Pivoto, T.T. Rezende, M.S.P. Facina, R. Moreira, F. de Oliveira Silva, K.V. Cardoso, S.L. Correa, A.V.D. Araujo, R.S.E. Silva, H.S. Neto, G.R. de Lima Tejerina, A.M. Alberti, A detailed relevance analysis of enabling technologies for 6G architectures, IEEE Access 11 (2023) 89644–89684, http://dx.doi.org/10.1109/ACCESS.2023.3301811.

[16] H. Guo, J. Li, J. Liu, N. Tian, N. Kato, A survey on space-air-ground-sea integrated network security in 6G, IEEE Commun. Surv. Tutor. 24 (1) (2022) 53–87, http://dx.doi.org/10.1109/COMST.2021.3131332.

[17] I. Baldin, A. Nikolich, J. Griffioen, I.I.S. Monga, K.-C. Wang, T. Lehman, P. Ruth, FABRIC: A national-scale programmable experimental network infrastructure, IEEE Internet Comput. 23 (6) (2019) 38–47, http://dx.doi.org/10.1109/MIC.2019.2958545.

[18] A.P. Silva, C. Tranoris, S. Denazis, S. Sargento, J. Pereira, M. Luís, R. Moreira, F. Silva, I. Vidal, B. Nogales, R. Nejabati, D. Simeonidou, 5GinFIRE: An end-to-end open5G vertical network function ecosystem, Ad Hoc Netw. 93 (2019) 101895, http://dx.doi.org/10.1016/j.adhoc.2019.101895, URL https://www.sciencedirect.com/science/article/pii/S1570870518309387.

[19] N.F. Saraiva de Sousa, D.A. Lachos Perez, R.V. Rosa, M.A. Santos, C. Esteve Rothenberg, Network service orchestration: A survey, Comput. Commun. 142–143 (2019) 69–94, http://dx.doi.org/10.1016/j.comcom.2019.04.008, URL https://www.sciencedirect.com/science/article/pii/S0140366418309502.

[20] S. Wijethilaka, M. Liyanage, Survey on network slicing for internet of things realization in 5G networks, IEEE Commun. Surv. Tutor. 23 (2) (2021) 957–994, http://dx.doi.org/10.1109/COMST.2021.3067807.

[21] E. Pontes, M. Martinello, C. Dominicini, M. Schwarz, M. Ribeiro, E. Borges, I. Brito, J. Bezerra, M. Barcellos, FABRIC testbed from the eyes of a network researcher, in: Anais do II Workshop de Testbeds, SBC, Porto Alegre, RS, Brasil, 2023, pp. 38–49, http://dx.doi.org/10.5753/wtestbeds.2023.230665, URL https://sol.sbc.org.br/index.php/wtestbeds/article/view/24812.

[22] C. Both, R. Guimaraes, F. Slyne, J. Wickboldt, M. Martinello, C. Dominicini, R. Martins, Y. Zhang, D. Cardoso, R. Villaca, I. Ceravolo, R. Nejabati, J. Marquez-Barja, M. Ruffini, L. DaSilva, FUTEBOL control framework: Enabling experimentation in convergent optical, wireless, and cloud infrastructures, IEEE Commun. Mag. 57 (10) (2019) 56–62, http://dx.doi.org/10.1109/MCOM.001.1900270.

[23] J.S.B. Martins, T.C. Carvalho, R. Moreira, C.B. Both, A. Donatti, J.H. Corrêa, J.A. Suruagy, S.L. Corrêa, A.J.G. Abelem, M.R.N. Ribeiro, J.-m.S. Nogueira, L.C.S. Magalhães, J. Wickboldt, T.C. Ferreto, R. Mello, R. Pasquini, M. Schwarz, L.N. Sampaio, D.F. Macedo, J.F. De Rezende, K.V. Cardoso, F. De Oliveira Silva, Enhancing network slicing architectures with machine learning, security, sustainability and experimental networks integration, IEEE Access 11 (2023) 69144–69163, http://dx.doi.org/10.1109/ACCESS.2023.3292788.

[24] R. Moreira, T. Carvalho, F. Silva, Designing and evaluating a high-reliable and security-aware identity and access management for slicing architectures, in: Anais do XIV Workshop de Pesquisa Experimental da Internet do Futuro, SBC, Porto Alegre, RS, Brasil, 2023, pp. 1–6, http://dx.doi.org/10.5753/wpeif.2023.722, URL https://sol.sbc.org.br/index.php/wpeif/article/view/24653.

[25] C. De Alwis, P. Porambage, K. Dev, T.R. Gadekallu, M. Liyanage, A survey on network slicing security: Attacks, challenges, solutions and research directions, IEEE Commun. Surv. Tutor. 26 (1) (2024) 534–570, http://dx.doi.org/10.1109/COMST.2023.3312349.

[26] V.P. Singh, M.P. Singh, S. Hegde, M. Gupta, Security in 5G network slices: Concerns and opportunities, IEEE Access 12 (2024) 52727–52743, http://dx.doi.org/10.1109/ACCESS.2024.3386632.

[27] M.J. Abood, G.H. Abdul-Majeed, Classification of network slicing threats based on slicing enablers: A survey, Int. J. Intell. Netw. 4 (2023) 103–112, http://dx.doi.org/10.1016/j.ijin.2023.04.002, URL https://www.sciencedirect.com/science/article/pii/S2666603023000076.

[28] A. Imteaj, U. Thakker, S. Wang, J. Li, M.H. Amini, A survey on federated learning for resource-constrained IoT devices, IEEE Internet Things J. 9 (1) (2022) 1–24, http://dx.doi.org/10.1109/JIOT.2021.3095077.

[29] J. Gang, V. Friderikos, Inter-tenant resource sharing and power allocation in 5G virtual networks, IEEE Trans. Veh. Technol. 68 (8) (2019) 7931–7943, http://dx.doi.org/10.1109/TVT.2019.2917426.

[30] N. Alliance, Description of network slicing concept, NGMN 5G P 1 (1) (2016) 1–11.

[31] 3GPP, Management and orchestration; concepts, use cases and requirements, in: Technical Specification TS 28.530 TS 28.530, 3rd Generation Partnership Project (3GPP), 2023, Release 15.

[32] K. Park, S. Sung, H. Kim, J. il Jung, Technology trends and challenges in SDN and service assurance for end-to-end network slicing, Comput. Netw. 234 (2023) 109908, http://dx.doi.org/10.1016/j.comnet.2023.109908, URL https://www.sciencedirect.com/science/article/pii/S1389128623003535.

[33] C. Parada, J. Bonnet, E. Fotopoulou, A. Zafeiropoulos, E. Kapassa, M. Touloupou, D. Kyriazis, R. Vilalta, R. Muñoz, R. Casellas, R. Martínez, G. Xilouris, 5Gtango: A beyond-mano service platform, in: 2018 European Conference on Networks and Communications, EuCNC, 2018, pp. 26–30, http://dx.doi.org/10.1109/EuCNC.2018.8443232.

[34] A.P. Silva, C. Tranoris, S. Denazis, S. Sargento, J. Pereira, M. Luís, R. Moreira, F. Silva, I. Vidal, B. Nogales, R. Nejabati, D. Simeonidou, 5GinFIRE: An end-to-end open5G vertical network function ecosystem, Ad Hoc Netw. 93 (2019) 101895, http://dx.doi.org/10.1016/j.adhoc.2019.101895, URL https://www.sciencedirect.com/science/article/pii/S1570870518309387.

[35] A.A. Barakabitze, A. Ahmad, R. Mijumbi, A. Hines, 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges, Comput. Netw. 167 (2020) 106984, http://dx.doi.org/10.1016/j.comnet.2019.106984, URL https://www.sciencedirect.com/science/article/pii/S1389128619304773.

[36] S. Clayman, A. Neto, F. Verdi, S. Correa, S. Sampaio, I. Sakelariou, L. Mamatas, R. Pasquini, K. Cardoso, F. Tusa, C. Rothenberg, J. Serrat, The NECOS approach to end-to-end cloud-network slicing as a service, IEEE Commun. Mag. 59 (3) (2021) 91–97, http://dx.doi.org/10.1109/MCOM.001.2000702.

[37] S. Wijethilaka, M. Liyanage, A federated learning approach for improving security in network slicing, in: GLOBECOM 2022 - 2022 IEEE Global Communications Conference, 2022, pp. 915–920, http://dx.doi.org/10.1109/GLOBECOM48099.2022.10001190.

[38] R. Moreira, T. Carvalho, F. Silva, Designing and evaluating a high-reliable and security-aware identity and access management for slicing architectures, in: Anais do XIV Workshop de Pesquisa Experimental da Internet do Futuro, SBC, Porto Alegre, RS, Brasil, 2023, pp. 1–6, http://dx.doi.org/10.5753/wpeif.2023.722, URL https://sol.sbc.org.br/index.php/wpeif/article/view/24653.

[39] J. Konečnỳ, H.B. McMahan, F.X. Yu, P. Richtárik, A.T. Suresh, D. Bacon, Federated learning: Strategies for improving communication efficiency, 2016, arXiv preprint arXiv:1610.05492.

[40] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A.y. Arcas, Communication-efficient learning of deep networks from decentralized data, in: A. Singh, J. Zhu (Eds.), Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, in: Proceedings of Machine Learning Research, vol. 54, PMLR, 2017, pp. 1273–1282, URL https://proceedings.mlr.press/v54/mcmahan17a.html.

[41] Q. Li, Y. Wang, G. Sun, L. Luo, H. Yu, Joint demand forecasting and network slice pricing for profit maximization in network slicing, IEEE Trans. Netw. Sci. Eng. 11 (2) (2024) 1496–1509, http://dx.doi.org/10.1109/TNSE.2023.3324336.

[42] S. Gao, R. Lin, Y. Fu, H. Li, J. Cao, Security threats, requirements and recommendations on creating 5g network slicing system: A survey, Electronics 13 (10) (2024) http://dx.doi.org/10.3390/electronics13101860, URL https://www.mdpi.com/2079-9292/13/10/1860.

[43] X. Ma, J. Zhu, Z. Lin, S. Chen, Y. Qin, A state-of-the-art survey on solving non-IID data in federated learning, Future Gener. Comput. Syst. 135 (2022) 244–258, http://dx.doi.org/10.1016/j.future.2022.05.003, URL https://www.sciencedirect.com/science/article/pii/S0167739X22001686.

[44] F. Tusa, S. Clayman, End-to-end slices to orchestrate resources and services in the cloud-to-edge continuum, Future Gener. Comput. Syst. 141 (2023) 473–488, http://dx.doi.org/10.1016/j.future.2022.11.026, URL https://www.sciencedirect.com/science/article/pii/S0167739X22003971.

[45] M. Babar, B. Qureshi, A. Koubaa, Review on federated learning for digital transformation in healthcare through big data analytics, Future Gener. Comput. Syst. 160 (2024) 14–28, http://dx.doi.org/10.1016/j.future.2024.05.046, URL https://www.sciencedirect.com/science/article/pii/S0167739X24002784.

[46] K. Kaur, V. Mangat, K. Kumar, A review on virtualized infrastructure managers with management and orchestration features in NFV architecture, Comput. Netw. 217 (2022) 109281, http://dx.doi.org/10.1016/j.comnet.2022.109281, URL https://www.sciencedirect.com/science/article/pii/S1389128622003395.

[47] A. Boualouache, T. Engel, Federated learning-based inter-slice attack detection for 5G-v2x sliced networks, in: 2022 IEEE 96th Vehicular Technology Conference, VTC2022-Fall, 2022, pp. 1–6, http://dx.doi.org/10.1109/VTC2022-Fall57202.2022.10012736.

[48] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, ICISSp 1 (2018) 108–116.

[49] S. Ben Saad, B. Brik, A. Ksentini, Toward securing federated learning against poisoning attacks in zero touch B5G networks, IEEE Trans. Netw. Serv. Manag. 20 (2) (2023) 1612–1624, http://dx.doi.org/10.1109/TNSM.2023.3278838.

[50] T. Wichary, J. Mongay Batalla, C.X. Mavromoustakis, J. Żurek, G. Mastorakis, Network slicing security controls and assurance for verticals, Electronics 11 (2) (2022) http://dx.doi.org/10.3390/electronics11020222, URL https://www.mdpi.com/2079-9292/11/2/222.

[51] M.S. Khan, B. Farzaneh, N. Shahriar, N. Saha, R. Boutaba, Slicesecure: Impact and detection of dos/ddos attacks on 5G network slices, in: 2022 IEEE Future Networks World Forum, FNWF, 2022, pp. 639–642, http://dx.doi.org/10.1109/FNWF55208.2022.00117.

[52] R. Niboucha, S.B. Saad, A. Ksentini, Y. Challal, Zero-touch security management for mMTC network slices: DDoS attack detection and mitigation, IEEE Internet Things J. 10 (9) (2023) 7800–7812, http://dx.doi.org/10.1109/JIOT.2022.3230875.

[53] Z. Wen, H.S. Pacherkar, G. Yan, VET5G: A virtual end-to-end testbed for 5G network security experimentation, in: Proceedings of the 15th Workshop on Cyber Security Experimentation and Test, CSET '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 19–29, http://dx.doi.org/10.1145/3546096.3546111.

[54] R.S. Silva, C.C. Meixner, R.S. Guimarães, T. Diallo, B.O. Garcia, L.F.M. de Moraes, M. Martinello, REPEL: A strategic approach for defending 5G control plane from DDoS signalling attacks, IEEE Trans. Netw. Serv. Manag. 18 (3) (2021) 3231–3243, http://dx.doi.org/10.1109/TNSM.2020.3035342.

[55] A. Chilukuri, S. Vittal, A.A. Franklin, SENTINEL: Self protecting 5G core control plane from DDoS attacks for high availability service, in: 2023 15th International Conference on COMmunication Systems & NETworkS, COMSNETS, 2023, pp. 554–562, http://dx.doi.org/10.1109/COMSNETS56262.2023.10041318.

[56] W. Jiang, Y. Zhan, G. Zeng, J. Lu, Probabilistic-forecasting-based admission control for network slicing in software-defined networks, IEEE Internet Things J. 9 (15) (2022) 14030–14047, http://dx.doi.org/10.1109/JIOT.2022.3145475.

[57] R. Moreira, J.S.B. Martins, T.C.M.B. Carvalho, F.d. Silva, On enhancing network slicing life-cycle through an AI-native orchestration architecture, in: L. Barolli (Ed.), Advanced Information Networking and Applications, Springer International Publishing, Cham, 2023, pp. 124–136.

[58] T. Salmito, L. Ciuffo, I. Machado, M. Salvador, M. Stanton, N. Rodriguez, A. Abelem, L. Bergesio, S. Sallent, L. Baron, Fibre-an international testbed for future internet experimentation, in: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, SBRC 2014, 2014, pp. p–969.

[59] A. Banerjee, S.S. Mwanje, G. Carle, Trust and performance in future AI-enabled, open, multi-vendor network management automation, IEEE Trans. Netw. Serv. Manag. 20 (2) (2023) 995–1007, http://dx.doi.org/10.1109/TNSM.2022.3214296.

[60] J.H. Correa, P.M. Ciarelli, M.R. Ribeiro, R.S. Villaça, Ml-based ddos detection and identification using native cloud telemetry macroscopic monitoring, J. Netw. Syst. Manage. 29 (2021) 1–28.

[61] A. Turner, Contributors, Tcpreplay: Pcap editing and replay tools for *nix, 2024, Version 4.4.1, URL https://tcpreplay.appneta.com/.

[62] C. Coldwell, D. Conger, E. Goodell, B. Jacobson, B. Petersen, D. Spencer, M. Anderson, M. Sgambati, Machine learning 5G attack detection in programmable logic, in: 2022 IEEE Globecom Workshops, GC Wkshps, 2022, pp. 1365–1370, http://dx.doi.org/10.1109/GCWkshps56602.2022.10008647.

[63] Free5G project, 2024, Website, URL https://www.free5g.org/. (Accessed March 2024).

[64] I. Sharafaldin, A.H. Lashkari, A.A. Ghorbani, Toward generating a new intrusion detection dataset and intrusion traffic characterization, in: International Conference on Information Systems Security and Privacy, 2018, URL https://api.semanticscholar.org/CorpusID:4707749.

[65] S. Hossain, A. Boualouache, B. Brik, S.-M. Senouci, A lightweight 5G-v2x intra-slice intrusion detection system using knowledge distillation, in: ICC 2023 - IEEE International Conference on Communications, 2023, pp. 1112–1117, http://dx.doi.org/10.1109/ICC45041.2023.10279212.

[66] F.K. Rens W. van der Heijden, Veremi: A dataset for comparable evaluation of misbehavior detection in VANETs, 2018, arXiv preprint arXiv:1804.06701.

[67] A. Majeed, A.M. Alnajim, A. Waseem, A. Khaliq, A. Naveed, S. Habib, M. Islam, S. Khan, Deep learning-based symptomizing cyber threats using adaptive 5G shared slice security approaches, Future Internet 15 (6) (2023) http://dx.doi.org/10.3390/fi15060193, URL https://www.mdpi.com/1999-5903/15/6/193.

[68] S. Garcia, M. Grill, J. Stiborek, A. Zunino, An empirical comparison of botnet detection methods, Comput. Secur. 45 (2014) 100–123, http://dx.doi.org/10.1016/j.cose.2014.05.011.

[69] A. Boualouache, A.A. Jolfaei, T. Engel, Multi-process federated learning with stacking for securing 6G-v2x network slicing at cross-borders, IEEE Trans. Intell. Transp. Syst. 25 (9) (2024) 10941–10952, http://dx.doi.org/10.1109/TITS.2024.3367388.

[70] S. Samarakoon, Y. Siriwardhana, P. Porambage, M. Liyanage, S.-Y. Chang, J. Kim, J. Kim, M. Ylianttila, 5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network, IEEE Dataport, 2022, http://dx.doi.org/10.21227/xtep-hv36.

[71] A.H. Lashkari, A. Seo, G.D. Gil, A. Ghorbani, CIC-AB: Online ad blocker for browsers, in: 2017 International Carnahan Conference on Security Technology, ICCST, IEEE, 2017, pp. 1–7.

[72] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 2623–2631, http://dx.doi.org/10.1145/3292500.3330701.

[73] J. Zhu, J. Cao, D. Saxena, S. Jiang, H. Ferradi, Blockchain-empowered federated learning: Challenges, solutions, and future directions, ACM Comput. Surv. 55 (11) (2023) http://dx.doi.org/10.1145/3570953.

[74] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for hyper-parameter optimization, in: J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Weinberger (Eds.), in: Advances in Neural Information Processing Systems, vol. 24, Curran Associates, Inc., 2011, pp. 2546–2554, URL https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf.

[75] G.M. ud din, NSL-kdd dataset, 2009, URL https://www.unb.ca/cic/datasets/nsl.html. (Accessed 03 September 2024).

[76] H. Sedjelmaci, A. Boualouache, When two-layer federated learning and mean-field game meet 5G and beyond security: Cooperative defense systems for 5G and beyond network slicing, IEEE Trans. Netw. Serv. Manag. 21 (1) (2024) 1178–1189, http://dx.doi.org/10.1109/TNSM.2023.3294568.

[77] Y. Rumesh, D. Attanayaka, P. Porambage, J. Pinola, J. Groen, K. Chowdhury, Federated learning for anomaly detection in open RAN: Security architecture within a digital twin, in: 2024 Joint European Conference on Networks and Communications & 6G Summit, EuCNC/6G Summit, 2024, pp. 877–882, http://dx.doi.org/10.1109/EuCNC/6GSummit60053.2024.10597083.

[78] P.H. Mirzaee, M. Shojafar, Z. Pooranian, P. Asefy, H. Cruickshank, R. Tafazolli, FIDS: A federated intrusion detection system for 5G smart metering network, in: 2021 17th International Conference on Mobility, Sensing and Networking, MSN, 2021, pp. 215–222, http://dx.doi.org/10.1109/MSN53354.2021.00044.

[79] A. Thantharate, FED6g: Federated chameleon learning for network slice management in beyond 5G systems, in: 2022 IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON, 2022, pp. 0019–0025, http://dx.doi.org/10.1109/IEMCON56893.2022.9946488.

Rodrigo Moreira received his Ph.D. from the University of Uberlândia (UFU) in 2021. He is currently a Professor at the Federal University of Viçosa, where he also obtained his B.S. degree in 2014. He earned his M.Sc. degree from the Federal University of Uberlândia, Brazil, in 2017. He has several papers published and has presented at conferences. His research interests include the future of the internet, quality of service, cloud computing, network function virtualization, software-defined networking, computational intelligence, and edge computing

Rodolfo S. Villaça received his Ph.D. in Electrical Engineering from the University of Campinas (Unicamp) in 2013. He is currently an Associate Professor at the Federal University of Espírito Santo (Ufes), where he also obtained his B.S. degree in Computer Engineering in 2000 and his M.Sc. degree in 2004. His research interests include Computer Networks, Distributed Systems, Cybersecurity, and Artificial Intelligence.

Moisés R.N. Ribeiro received the B.Sc. degree in electrical engineering from the Instituto Nacional de Telecomunicações, Brazil, in 1992, a M.Sc.degree in telecommunications from the Universidade Estadual de Campinas, Brazil, in 1996, and a Ph.D. degree from the University of Essex, U.K., in 2002. In 1995, he joined the Department of Electrical Engineering at the Federal University of Espírito Santo. He was a Visiting Professor at the Photonics and Networking Research Laboratory, at Stanford University from 2010 to 2011. His research interests include fiber optic communication and sensor devices, systems, and networks.

Joberto S.B. Martins (Life Senior Member, IEEE) – Ph.D. in Computer Science at Université Pierre et Marie Curie (UPMC), Paris (1986), PosDoc at ICSI - Berkeley University (1995), and PosDoc Senior Researcher at Paris Saclay University - France (2016). International Professor at Hochschule für Technik und Wirtschaft des Saarlandes (HTW), Germany (since 2004). Full Professor at Salvador University (UNIFACS) on Computer Science, Director of NUPERC research group with research interests in Network Slicing, Resource Orchestration, Machine Learning, and Smart City. He is a key speaker, teacher, and invited lecturer at various international congresses and companies in Brazil and Europe.

João H. Corrêa is currently an assistant professor at the Federal University of Ceará (UFC). He received his Ph.D. in Computer Science in 2021 at the Federal University of Espírito Santo (UFES), his M.Sc. degree in Computer Science from the Federal University of Paraíba (UFPB) in 2017, and his B.Sc. degree in Computer Networks from the Federal Institute of Paraíba (IFPB) in 2015. His research interests include Computer Networks, network softwarization (SDN and NFV), Network Security, and Cloud Computing.

Tereza Cristina Melo de Brito Carvalho – Associate Professor of Escola Politécnica – University of São Paulo (USP) and visiting professor at Université Paris 1 Panthéon-Sorbonne. She has been the founder and general coordinator of LASSU (Laboratory of Sustainability on ITC) since 2010 and CEDIR-USP (Center for Reuse and Discard of Informatics Residuals) since 2009. She is a former assessor of CTI – USP (Information Technology Coordination) from 2010–2013 and CCE-USP (Electronic Computing Center) director from 2006–2010. She is a Sloan Fellow in 2002 from MIT (Massachusetts Institute of Technology). She has coordinated international and national R&D projects since 2000 in: Green Computing, Cloud Computing, IT Energy Efficiency, IT Governance, Digital Technologies applied to the Amazon Production Chains, WEEE (Waste Electrical and Electronic Equipment), Future Internet, Scientific DMZ, and Security. She holds several international patents.

Flávio de Oliveira Silva (Member, IEEE) received the Ph.D. from the University of São Paulo (USP) in 2013. He is a Professor at the Faculty of Computing (FACOM), Federal University of Uberlândia (UFU). He is also a professor in the Department of Informatics (DI), School of Engineering of the University of Minho in Braga, Portugal, and a researcher with the ALGORITMI Centre. He has published and presented several papers at conferences worldwide. His research interests include future networks, the IoT, network softwarization (SDN and NFV), future intelligent applications and systems, cloud computing, and software-based innovation. He is a member of IEEE, ACM, and SBC. He is a reviewer of several journals and a member of TCP at several IEEE conferences.