2025-05-01

Enhancing Security and Resiliency in Operational Technology Environments Through Network Slicing and Federated Learning

Brian Giovanni Rodiles Delgado
University of Texas at El Paso

ENHANCING SECURITY AND RESILIENCY IN OPERATIONAL TECHNOLOGY ENVIRONMENTS THROUGH NETWORK SLICING AND FEDERATED LEARNING

BRIAN G. RODILES DELGADO

Master's Program in Computer Science

APPROVED:

Deepak K. Tosh, Ph.D., Chair

Christian Servin, Ph.D.

Jaime Acosta, Ph.D.

Stephen Crites, Ph.D.
Dean of the Graduate School

to my

FAMILY

with love

ENHANCING SECURITY AND RESILIENCY IN OPERATIONAL TECHNOLOGY ENVIRONMENTS THROUGH NETWORK SLICING AND FEDERATED LEARNING

by

BRIAN G. RODILES DELGADO, B.S.

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Department of Computer Science

THE UNIVERSITY OF TEXAS AT EL PASO

May 2025

Acknowledgements

I wish to begin by expressing my deepest gratitude to my family for their unwavering support. To my abuela (mamita), Margarita Delgado Díaz, thank you for your boundless love and the comforting meals you prepared every day. To my madre, Margarita Rodiles Delgado, your tireless work ethic and constant drive to give back to our family have been a guiding example. To my hermana, Sayumi Margarita Rodiles Delgado, your strength and early lessons—even as a toddler—inspire me daily. To my tíos, Juan Oscar Rodiles Delgado, thank you for your wise counsel and steadfast encouragement, and Juan Carlos Rodiles Delgado, for awakening my passion for engineering and creative problem-solving. And finally, to my abuelo, Juan Oscar Rodiles Landa, I am grateful for your hard work and unwavering support.

I would also like to express my heartfelt thanks to my advisor, Dr. Deepak Tosh of the Computer Science Department at The University of Texas at El Paso, for his expert advice, insightful feedback, and constant encouragement. Research in critical infrastructure and federated learning is no easy undertaking, but his clear explanations and passion for the field fueled my determination to ask questions and seek answers.

I extend my gratitude to the other members of my committee: Dr. Jaime Acosta of the Computer Science Department at The University of Texas at El Paso, whose enthusiasm for cybersecurity inspired my own aspirations; and Dr. Christian Servin of the Computer Science program at El Paso Community College, who first recognized my potential, guided me toward academic excellence through the peer-leader program, and encouraged me to lead my first student organization. The shared dedication to education, rooted in my mother's twenty-plus years as a middle school teacher and my uncle's innovation leadership at Tecnológico de Monterrey, finds reflection in Dr. Servin's mentorship and in Chapter 5's discussion of bringing this framework into the classroom.

I would also like to acknowledge the dedication and expertise of the professors and staff in the Computer Science Department at The University of Texas at El Paso. The following individuals represent just a portion of those whose support has made this journey possible, though many more have contributed in ways both large and small:

Dr. Marcelo Frias

As Director of the Master of Science in Computer Science program, his advising ensured I completed all coursework for the master's degree.

Dr. Shirley Moore

Her teachings in computer architecture and parallel computing improved my problem-solving for technical issues related to this thesis' tasks, and I am grateful to her for co-advising the 5G DOE project, which forms much of this work.

Dr. Luc Longpré, Dr. Martine Ceberio, and Dr. Vladik Kreinovich

Their courses in automata, advanced algorithms, and theory of computation revealed the beauty of theoretical computer science and showed me how discrete mathematics can solve problems in unexpected ways.

Ms. Jennifer Green, M.A.

As Director of the Office of Student Fellowship and Awards (OSFA) at UTEP, her encouragement and guidance helped me recognize my potential and successfully apply to national and international competitive programs. Her support improved my storytelling and honed my writing and communication skills.

NOTE: This thesis was submitted to my Supervising Committee on the May 6, 2025.

Abstract

The growing convergence of Information Technology (IT) and Operational Technology (OT) within Industry 4.0 environments has introduced new demands on industrial network infrastructure. As cyber-physical systems become increasingly interconnected, ensuring the secure, timely, and efficient exchange of critical data is essential. This thesis explores how network slicing, a method of creating isolated virtual network segments, can be applied within OT environments to address challenges such as latency, security, and resource allocation.

The first research question addressed in this thesis is: How can OT networks take advantage of NFV and SDN technology to become cyber resilient? This study examines the operational, security, and architectural implications of introducing network slicing into traditionally static OT infrastructures such as Industrial Control Systems (ICS) and SCADA. Through simulated deployments and case studies, the research demonstrates how slicing enables better isolation between critical and non-critical services, thereby improving response time, throughput, and security in sensitive environments.

The second question considers: How to dynamically implement network slicing and take advantage of network resources towards integrating decentralized machine learning? In response, this thesis proposes a framework that combines Software-Defined Networking (SDN), Network Function Virtualization (NFV), and Federated Learning (FL) to enable real-time analytics while maintaining data locality. The proposed approach reduces the burden on centralized infrastructure and minimizes privacy risks by supporting on-site training of models across distributed OT nodes, coordinated through dynamically allocated network slices.

The third focus explores: How slicing helps to increase the resiliency of OT networks through the orchestration of a dynamic DMZ? To answer this, the thesis presents a method for creating and managing Dynamic Demilitarized Zones (DMZs) using network slicing. This enables flexible and automated isolation of sensitive subsystems during threat scenarios or high-risk operations. Coupled with intelligent orchestration and containerized security services, the dynamic DMZ significantly enhances the system's ability to respond to cyber incidents without halting production.

Ultimately, this thesis contributes a comprehensive architecture that blends network slicing with machine learning, secure segmentation, and automation, paving the way for resilient, adaptive, and intelligent OT environments. Performance evaluations across multiple scenarios show improvements in system reliability, threat response time, model accuracy, and resource utilization, providing a strong foundation for future industrial automation systems.

Table of Contents

List of Figures

Chapter 1

Introduction

Industrial environments are undergoing a fundamental transformation, transitioning from rigid, hardware-centric infrastructures to dynamic, software-defined ecosystems. Traditionally, industrial networks such as Supervisory Control and Data Acquisition (SCADA) systems and Industrial Control Systems (ICS) were designed for reliability, stability, and long-term operational continuity. These systems typically operated in isolated, closed environments, and were tailored for deterministic performance with minimal changes over time. However, the rapid adoption of Industry 4.0 technologies, including the Industrial Internet of Things (IIoT), cyber-physical systems, and digital twin architectures, has significantly increased the demands placed on industrial networks [1]. Modern industrial systems now require support for low-latency communication, high-throughput data exchange, real-time analytics, and enhanced cybersecurity. These new requirements expose the limitations of legacy Operational Technology (OT) infrastructures, which were not originally designed to accommodate such advanced and dynamic capabilities.

1.1 Overview of ICS and SCADA Networks

ICS and SCADA systems are foundational to the operation of critical infrastructure sectors, including manufacturing, transportation, power generation and distribution, oil and gas, and water treatment. SCADA systems enable centralized supervisory control by collecting, analyzing, and visualizing data from field devices such as sensors, actuators, and controllers, which are often dispersed across vast geographic areas. ICS, as a broader category, encompasses SCADA along with other control system configurations such as Distributed Control Systems (DCS) and Programmable Logic Controllers (PLCs).

These systems have traditionally prioritized high reliability, deterministic response times,

and availability. Communication protocols used in ICS environments, such as Modbus, DNP3, and PROFIBUS, were developed with simplicity and deterministic behavior in mind, often at the expense of modern security and scalability. As such, they are vulnerable to a range of attacks and performance issues when exposed to interconnected or cloud-enabled networks.

1.1.1 Operational Technology and Its Unique Networking Requirements

Operational Technology (OT) refers to hardware and software systems that monitor and control physical devices, processes, and infrastructure. Unlike Information Technology (IT) systems, which are built around data storage, processing, and user applications, OT systems are purpose-built for safety, efficiency, and process reliability. OT networks often operate in harsh environments, use specialized hardware, and must function with minimal tolerance for downtime or failure.

Because OT networks require deterministic communication, minimal downtime, and purpose-built reliability in contrast to traditional IT systems, the Purdue Model, as defined in ANSI/ISA-62443-2-1 [2] and portrayed in Figure 1.1, uses a hierarchical zone-and-conduit architecture tailored to industrial control environments. It segments systems into discrete layers, with field devices and controllers at the bottom and enterprise and Internet-facing services at the top, and enforces clear trust boundaries through firewalls, intrusion-detection systems, and carefully regulated conduits. This structure enables defense in depth and gives operators the flexibility to apply security controls that meet each layer's specific performance and reliability requirements.

At Level 0, the Physical Process layer, sensors, and actuators interact directly with pumps, valves, motor,s and other industrial equipment. Data here travels over hard-wired or field-bus links, with minimal buffering or processing. Level 1, the Controller LAN, houses programmable logic controllers (PLCs), distributed control system (DCS) controllers, and safety-instrumented system (SIS) devices. These controllers execute real-time control logic (often with millisecond-level deterministic requirements) and communicate with sensors and

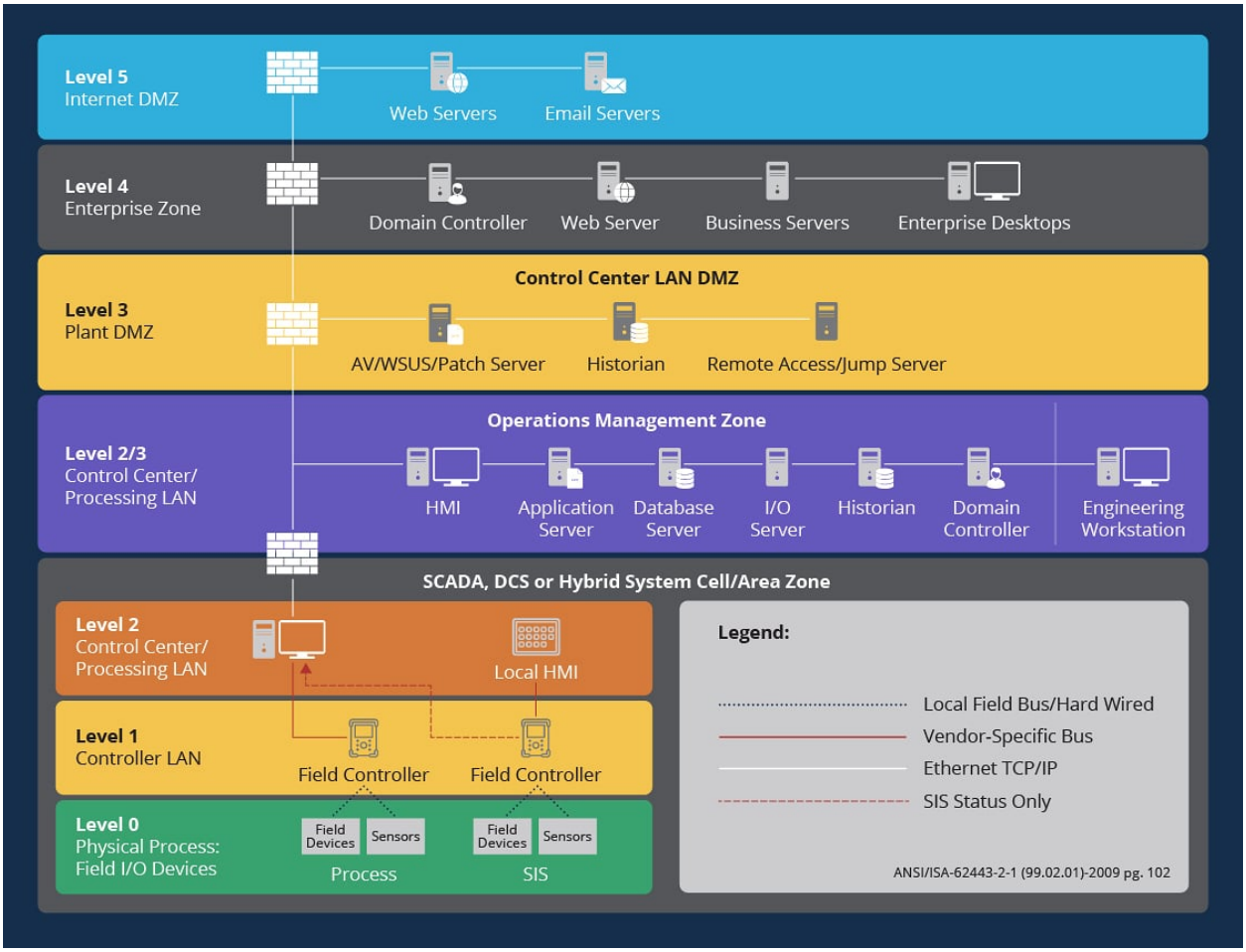

Figure 1.1: ANSI/ISA-62443-2-1 Purdue Model [2]

actuators over vendor-specific or real-time protocols.

Level 2, the Control Center or Processing LAN, provides supervisory control and data acquisition functions. Human-machine interfaces (HMIs), SCADA servers, local historians, and engineering workstations aggregate real-time data from the controllers, allow operators to monitor and adjust process setpoints, and log critical events. This layer typically uses Ethernet/IP or other TCP/IP-based networks, balancing the need for performance with more conventional IT management tools.

Sitting between control and enterprise, Level 3 is the Plant DMZ (also called the Control Center LAN DMZ). It hosts patch and antivirus servers (for example, WSUS), central historians, application servers, and jump servers for secure remote access. By funneling cross-domain traffic through hardened jump hosts and data-diode-style conduits, Level 3

minimizes attack surfaces and enforces strict one-way or bi-directional flows as needed for data exchange.

Level 4, the Enterprise Zone, integrates OT insights into corporate IT systems. In this zone, you'll find domain controllers, ERP/MES servers, business intelligence tools, and standard desktops. Access between Level 4 and Level 3 is governed by enterprise firewalls and user-authentication systems, ensuring that only sanctioned, logged transactions move from the plant floor to back-office applications.

At the top, Level 5 is the Internet DMZ, hosting publicly reachable services such as web servers, email gateways, and VPN concentrators. This outermost barrier faces the untrusted Internet and is protected by the strictest perimeter defenses, including next-generation firewalls, intrusion-prevention systems, and behavioral detectors. By isolating public services in their own DMZ, organizations prevent direct exposure of critical OT and IT layers to external threats.

Another critical difference between IT and OT lies in the performance expectations. OT systems require deterministic and time-sensitive communication with latencies often measured in milliseconds or microseconds. Network behavior must remain predictable, especially in safety-critical environments such as chemical plants or power grids. In contrast, IT systems can accommodate variable latencies and are more tolerant of temporary service disruptions. This disparity complicates the integration of IT and OT systems, as traditional IT networking approaches may not satisfy the real-time and reliability needs of industrial systems.

To address these stringent demands, Quality of Service (QoS) mechanisms are essential in OT networks. QoS ensures that critical data traffic, such as real-time sensor readings or emergency control commands, receives priority handling over less time-sensitive communications. By classifying and prioritizing traffic flows, QoS helps maintain predictable network behavior and consistent performance. In industrial settings, implementing QoS is fundamental for supporting use cases that require Ultra-Reliable Low Latency Communication (URLLC) [3], ensuring that even under heavy load, critical operations maintain their required timing constraints and reliability guarantees.

1.1.2 Current Network Orchestration in OT Environments

At present, network orchestration in OT settings is mostly static and manually configured. Devices are provisioned individually, and their communication routes are defined using fixed rules, resulting in limited scalability and poor adaptability. Network changes, such as adding a new sensor or reconfiguring data flows, often require system downtime or manual intervention by engineers. These manual configurations make OT networks rigid and slow to respond to evolving demands or threat landscapes.

Some current implementations attempt to achieve segmentation using Virtual Local Area Networks (VLANs). While VLANs offer a degree of logical separation, they come with notable limitations. Implementing VLANs typically requires dedicated hardware switches that support VLAN tagging, increasing infrastructure costs [4]. Additionally, VLANs operate solely at Layer 2 of the OSI model, restricting flexibility in cross-layer visibility and centralized control. These constraints hinder the scalability and adaptability needed for modern OT networks.

A related advancement is data plane programmability, which allows network operators to define how packets are processed directly within forwarding devices. This enables fine-grained control of traffic flows, supporting advanced functions such as packet inspection, filtering, and custom forwarding behaviors [5]. While data plane programmability can enhance visibility and control across the network, its practical use in OT environments faces limitations similar to those of VLANs. Most programmable data plane solutions operate predominantly at lower OSI layers and often require specialized hardware or protocols such as P4. This can reintroduce infrastructure complexity and vendor lock-in, making it harder to achieve seamless integration across diverse OT components. Additionally, programming at the data plane level often lacks holistic awareness of application-layer behavior, reducing its effectiveness for dynamic, policy-based orchestration.

As OT systems expand in scale and complexity, especially with the integration of IIoT devices, the demand for more flexible and automated network orchestration becomes increasingly urgent. However, the lack of interoperability between different vendor devices and the proprietary nature of many control systems create additional barriers. Moreover, most OT

networks lack the abstraction layers that are common in IT systems, making centralized and policy-driven orchestration difficult to implement without extensive system redesigns.

1.2 Cybersecurity Challenges in Modern OT Networks

OT engineers today face an evolving and increasingly complex set of cybersecurity challenges. Unlike traditional IT environments, where patching and security updates are routine, many OT systems are designed to run continuously for years or even decades without interruption. As a result, security vulnerabilities often go unaddressed, especially in older systems that are difficult to update or replace. The increased convergence of IT and OT networks further exacerbates the risk by creating new attack vectors.

High-profile cyber incidents such as the Stuxnet worm, Triton malware, and recent ransomware attacks targeting industrial control networks have demonstrated the catastrophic consequences of compromised OT systems. These attacks have led to physical damage, production downtime, environmental harm, and even threats to human safety. They have also underscored the inadequacy of traditional perimeter-based security approaches and the urgent need for real-time threat detection, isolation, and response capabilities within OT environments.

1.2.1 Standard Attacks Against ICS

Understanding the range of cyberthreats against Industrial Control Systems (ICS) is essential for developing effective defenses. The NIST SP 800-82 Revision 2 guide outlines the specialized security controls and risk profiles unique to ICS [6], emphasizing that conventional IT safeguards must be adapted to meet the deterministic and availability requirements of OT environments.

Adversaries often initiate attacks with reconnaissance, using techniques like Remote System Information Discovery (T0888) to fingerprint device models, roles, and configurations [7]. By querying system APIs or vendor management protocols, attackers map out potential targets and craft subsequent exploits tailored to specific controllers or sensors.

Supply-chain compromise (T0862) represents a stealthy initial-access vector in which

hardware, firmware, or software components are tampered with before deployment [8]. Once introduced into the control network, these backdoored elements provide attackers with persistent access, bypassing perimeter defenses and enabling long-term espionage or sabotage.

Replay and spoofing attacks involve capturing legitimate control or sensor messages and retransmitting or forging them to mask malicious activities. By replaying old sensor readings or injecting false telemetry, attackers can induce unsafe operations, such as opening valves or silencing alarms, without triggering conventional anomaly detectors [9].

Denial-of-Service (DoS) attacks against PLC communications can deliberately delay or drop network packets, degrading control-loop performance and potentially forcing emergency shutdowns. Research has demonstrated that targeting fieldbus protocols or exploiting insider access can halt process control and inflict significant downtime [10].

Unauthorized command injection (T0855) occurs when adversaries send forged instructions to controllers or actuators, causing them to operate outside safe parameters [11]. High-impact incidents, including false alarm activations in public warning systems, illustrate how illicit commands can lead to process disruptions or safety hazards.

Ransomware and destructive malware tailored for ICS environments can encrypt historian databases, lock operator interfaces, or plant "logic bombs" in PLC firmware [12]. These attacks exploit IT/OT convergence and legacy-system vulnerabilities to inflict physical damage, extort organizations, or sabotage critical infrastructure.

A robust defense against this spectrum of threats requires dynamic network orchestration. Software Defined Networking (SDN) and Network Function Virtualization (NFV) enable policy-driven micro-segmentation and automated enforcement of least-privilege flows, quickly isolating suspect traffic [13]. Complementing this, deep reinforcement-learning-based orchestration platforms can correlate distributed threat signals and autonomously coordinate containment actions across control-system layers, reducing response times and preserving system performance [14].

1.3 Introduction to NFV and SDN in Network Engineering

Network Function Virtualization (NFV) and Software-Defined Networking (SDN) represent foundational shifts in network architecture. NFV enables the decoupling of network functions from dedicated hardware appliances by implementing them as software running on general-purpose computing platforms [15]. Examples include firewalls, load balancers, and intrusion detection systems, which can be deployed and scaled as needed without purchasing new hardware.

NFV can be implemented using two primary deployment models: Virtualized Network Functions (VNFs) and Cloud-Native Network Functions (CNFs). VNFs are typically deployed as virtual machines on hypervisor-based infrastructure. They benefit from mature virtualization ecosystems and are often used in environments requiring compatibility with legacy systems. However, VNFs tend to be resource-intensive, have longer startup times, and introduce more operational overhead due to the complexity of managing full guest operating systems.

In contrast, CNFs, which are often deployed as Kubernetes Network Functions (KNFs), leverage container-based architectures orchestrated by platforms like Kubernetes. These functions are lightweight, start faster, and offer better scalability than VNFs. Their microservices-based design allows for modularity and granular updates, aligning well with DevOps and CI/CD practices. Despite these advantages, CNFs are relatively newer and may face adoption challenges in certain industrial or regulated environments. Additionally, they require a mature container orchestration platform and skilled personnel, which can introduce a learning curve.

SDN, on the other hand, separates the control plane from the data plane, allowing centralized controllers to define network behavior programmatically. This separation facilitates automated configuration, global network visibility, and real-time traffic optimization [16]. SDN and NFV together provide dynamic programmability, resource efficiency, and adaptability, which are particularly valuable in environments with diverse, rapidly changing requirements.

1.3.1 The Role of NFV and SDN in Industrial Networks

While NFV and SDN were initially developed for data centers and cloud environments, their benefits are now being explored in OT contexts. By integrating SDN into industrial networks, organizations can gain centralized visibility and control over traffic flows, enabling dynamic prioritization of safety-critical traffic or rapid isolation of compromised devices. NFV, in parallel, facilitates the deployment of network services such as virtual gateways, protocol translators, and firewalls without the need for physical infrastructure changes.

When compared with VLANs or DPP, SDN combined with NFV enables virtualization across multiple layers of the network stack. This approach allows for dynamic traffic control, centralized policy enforcement, and service deployment without the need for specialized hardware, offering a more scalable and cost-effective alternative to traditional segmentation in industrial networks.

1.3.2 Challenges of Integrating Network Slicing, SDN, and NFV

From a throughput perspective, integrating these technologies allows the network to maintain high data rates across diverse applications. Network slicing ensures that bandwidth can be allocated according to the needs of each slice, while SDN optimizes traffic routing and avoids congestion. NFV enables services to be scaled out when throughput demands increase, preventing performance bottlenecks and ensuring smooth operation even under high load conditions.

In terms of latency, the architecture allows the deployment of low-latency paths for time-sensitive applications. SDN can define optimized routes that reduce hop count and delay, and NFV allows services to be placed closer to the data source, such as deploying virtual firewalls at the edge. Network slicing prevents interference from non-critical traffic, contributing to consistent and predictable latency essential for real-time OT processes.

Regarding response time, this integrated framework supports dynamic reconfiguration and rapid deployment of services. For instance, if a slice experiences a spike in traffic or detects an anomaly, SDN can immediately redirect flows, and NFV can launch remediation services, such as a virtual security function, within seconds. This supports OT environments

where immediate reaction to changes and threats is vital.

A key advantage of this integration is the ability to orchestrate and manage slices to handle computationally intensive workloads such as Federated Learning (FL) and cybersecurity monitoring. FL tasks can be isolated within dedicated slices, allowing edge devices to train models locally without performance interference [17]. Similarly, cybersecurity operations can be managed through a dynamically provisioned demilitarized zone (DMZ), where virtualized security tools like intrusion detection systems and analyzers are deployed on demand and scaled based on the current risk.

Another promising use case enabled by this integration is the orchestration of digital twins within industrial environments, which are virtual representations of physical assets used for real-time monitoring, predictive maintenance, and simulation [18]. With network slicing, each digital twin can operate within an isolated and prioritized communication channel tailored to its performance needs. SDN enables precise routing of data streams between the physical asset and its virtual counterpart ensuring timely synchronization. NFV supports the deployment of twin-related services, such as analytics engines or simulation environments, close to the edge, reducing latency and bandwidth usage. As with FL, digital twins benefit from the resource isolation, dynamic scaling, and policy-driven control made possible by this architecture.

Holistically, combining network slicing, SDN, and NFV creates an adaptive, programmable, and secure infrastructure for OT environments. This architecture ensures that stringent requirements for throughput, latency, and rapid response are consistently met while enabling advanced capabilities that traditional networks cannot support. The result is a future-ready industrial network that can evolve with technological advancements and operational demands, offering resilience, scalability, and operational intelligence within a unified framework.

1.4 Research Scope and Thesis Questions

To explore this convergence of technologies, this thesis presents a modular and adaptive framework that integrates network slicing, Software-Defined Networking (SDN), Network

Function Virtualization (NFV), Federated Learning (FL), and dynamic security mechanisms into OT environments. This architecture aims to improve scalability, adaptability, security, and performance in line with Industry 4.0 objectives.

This thesis is guided by the following research questions:

1. How can OT networks take advantage of NFV and SDN technology to become cyber resilient?

   Industrial control networks typically rely on static, hardware-bound functions that cannot be easily updated or scaled. By virtualizing network and security services through NFV and centralizing control with SDN, operators could dynamically reconfigure traffic paths and enforce granular policies. However, most OT environments lack the abstraction layers and performance headroom to support real-time control-plane updates and virtualized services without interrupting critical operations. Reconciling the need for low-latency, deterministic communication with the agility NFV and SDN promise remains an open challenge.

2. How can dynamic network slicing be implemented to optimize network resource usage and support decentralized machine learning?

   Network slicing organizes physical infrastructure into multiple virtual networks tailored to specific requirements, but OT networks currently lack the orchestration frameworks to create and manage slices on the fly. Ensuring each slice meets strict latency and reliability constraints while sharing underlying links is complex given the heterogeneity of IIoT devices and control protocols. Decentralized machine learning workloads at the edge demand predictable isolation of compute and network resources to avoid interference with control traffic. Developing slice lifecycle management, resource allocation and monitoring mechanisms that minimize overhead and align with OT service-level objectives is essential yet understudied.

3. In what ways can slicing improve the resiliency of OT networks through the orchestration of a dynamic DMZ?

   Traditional OT demilitarized zones are statically configured and often fail to adapt to

evolving threat landscapes or changing process topologies. Applying network slicing to orchestrate dynamic DMZs would allow on-demand deployment of isolation barriers around suspicious traffic flows or critical assets. Virtual security functions such as intrusion detection, data diodes, and protocol translators could be instantiated within dedicated slices without reconfiguring the entire network. The challenge lies in coordinating slice-based DMZ deployment with existing safety and performance policies to ensure that containment actions do not disrupt real-time operations.

Thesis Organization: Chapter 2 provides a detailed review of background concepts and related work. Chapter 3 presents the implementation of network slicing in OT environments. Chapter 4 examines the integration of Federated Learning into the slicing framework. Chapter 5 introduces the dynamic DMZ orchestration and security architecture. Chapter 6 discusses the potential for educational applications of the framework. Chapter 7 concludes with findings and outlines future research directions.

Chapter 2

Background and Related Works

In this chapter, we examine the foundational technologies and previous research that inform the development of this thesis. The core technologies considered include Software-Defined Networking (SDN), Network Function Virtualization (NFV), Federated Learning (FL), Digital Twins, and dynamically orchestrated Demilitarized Zones (DMZs). Each of these contributes to building secure, flexible, and resilient architectures for Operational Technology (OT) networks under Industry 4.0 conditions. The first four sections provide an in-depth technical explanation of each of these technologies and their relevance to OT environments, while the final section synthesizes related academic and industrial works to highlight current research directions and gaps.

2.1 Software-Defined Networking and Network Function Virtualization

SCADA and ICS networks have traditionally relied on static, hardware-centric architectures, making them less adaptable to evolving performance and security needs. SDN addresses this limitation by decoupling the control and data planes, enabling centralized management and dynamic reconfiguration of traffic flows [19]. NFV complements SDN by allowing network functions such as firewalls, load balancers, and intrusion detection systems to be virtualized and executed on general-purpose hardware.

In Figure 2.1, the architecture illustrates the integration of SDN and NFV in a typical industrial setting, structured according to the Purdue model, which is a hierarchical framework for organizing industrial control systems into levels that separate business, supervisory, control, and field layers. This structure improves security and functional clarity across sys-

tems. In the figure, the supervisory layer includes Virtual Network Functions (VNFs) like network configuration tools, supervisory logic, security functions, and Quality of Service (QoS) APIs. These VNFs are deployed and managed through an orchestrator, which can be one of many available platforms. One of the most prominent orchestrators used in practice is Open Source MANO (OSM), where MANO stands for Management and Orchestration [20].

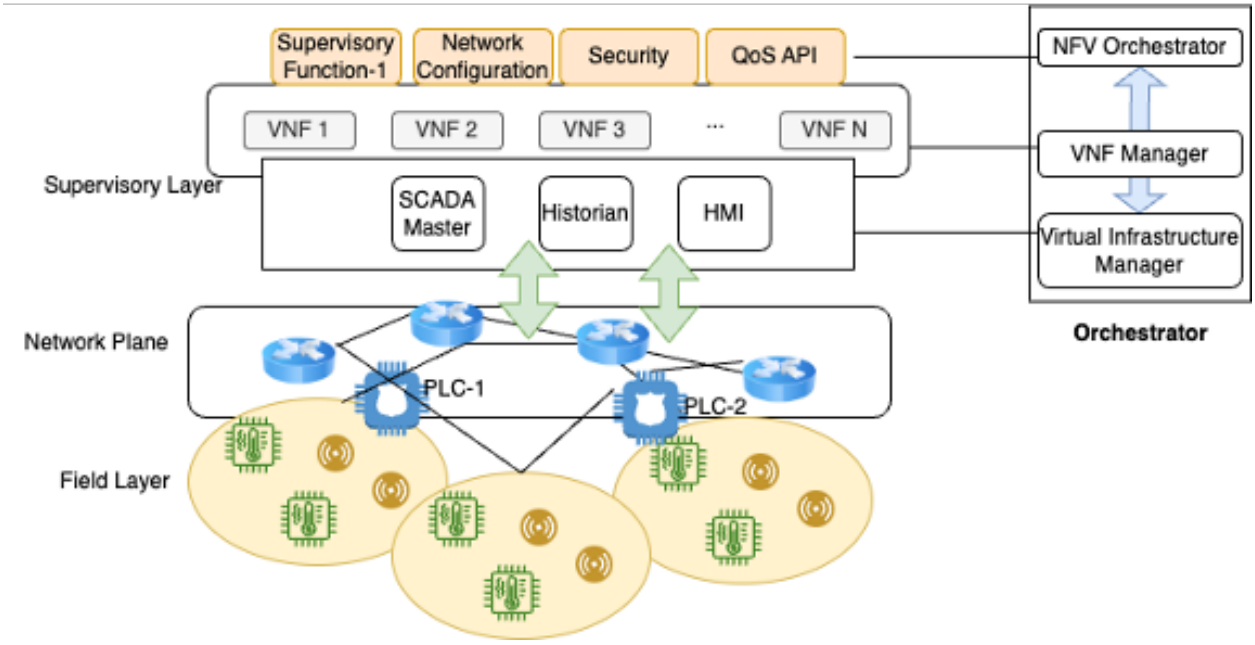

Figure 2.1: NFV-Enabled ICS Architecture [21]

An SDN controller is a critical component of the architecture that maintains a global view of the network and dynamically programs data plane devices such as switches and routers. It operates through a northbound interface to communicate with applications (such as orchestrators) and a southbound interface to interact with infrastructure. In many implementations, the Virtual Infrastructure Manager (VIM) also includes an integrated SDN controller. A VIM is responsible for managing the computing, storage, and networking resources of the virtualized infrastructure. For example, OpenStack is an open-source VIM that supports private cloud deployment and includes Neutron as its integrated SDN component [22].

Public cloud platforms such as Google Cloud Platform (GCP) implement the concept

of Virtual Private Cloud (VPC), which acts as an SDN by abstracting physical networks and enabling users to programmatically manage their networking topologies, subnets, and routes. These abstractions ensure scalable and flexible resource provisioning while maintaining security and performance constraints.

To deploy VNFs or KNFs (Cloud-Native Network Functions, also called CNFs), appropriate descriptors are required. VNFs typically use Virtual Network Function Descriptors (VNFDs) and Network Service Descriptors (NSDs), which specify deployment requirements, configuration, and interconnections. CNFs, which are container-based, additionally require a Kubernetes cluster to be provisioned within the VIM and leverage Helm charts or other container orchestration templates for deployment.

Overall, SDN and NFV technologies together provide dynamic programmability, scalable service deployment, and multi-layered network abstraction. These capabilities are crucial for OT environments where centralized control, real-time adaptability, and network security must be maintained alongside mission-critical operations.

2.1.1 Related Works on Software-Defined Networking and Network Function Virtualization

In Leonardi et al., the authors leverage SDN to introduce runtime reconfigurability into Time-Sensitive Networking (TSN) environments. Their architecture overlays an SDN controller atop TSN-enabled switches, allowing dynamic reassignment of scheduled traffic classes without violating the strict timing guarantees required by industrial automation. Through real-time monitoring of network load and application demands, the controller can rebalance bandwidth allocation and adjust gate control lists on the fly, demonstrating minimal impact on end-to-end latency in their prototype implementation [23].

Ray and Kumar present a systematic review of combined SDN/NFV architectures aimed at optimizing edge-cloud interactions for IoT deployments. They categorize existing solutions by their approaches to virtualized IoT service chaining, latency reduction techniques, and scalability mechanisms. The survey highlights orchestration frameworks that dynamically place VNFs, such as protocol translators or security gateways, close to edge devices, thereby

reducing round-trip times and enabling on-demand service provisioning across heterogeneous IoT ecosystems [24].

Szatkowski et al. explore the application of SDN in naval SCADA networks, introducing a programmable topology generator that automates the creation and modification of network paths. Their design integrates an SDN controller with a custom plugin that interprets naval mission profiles to instantiate resilient communication flows among shipboard PLCs and control stations. Experimental results show that the system can reconfigure critical links, ensuring continuous data delivery even under changing operational scenarios [25].

Jakaria et al. propose SDNSynth, a formal synthesis model for deploying hybrid SDN-enabled smart grid SCADA systems with resilience guarantees. By encoding reliability and latency constraints into a synthesis engine, SDNSynth generates network configurations that optimally place forwarding rules and redundant links to mitigate node or link failures. Simulation studies on a representative power-distribution testbed reveal a significant reduction in packet-loss probability compared to static deployments while satisfying strict SCADA timing bounds [26].

Varadharajan et al. introduce a software-centric security framework tailored for Internet-connected ICS environments. Their approach combines NFV-based deployment of virtualized firewalls and intrusion-detection functions with context-aware orchestration driven by real-time cyber-physical risk assessments. The framework's situational-awareness module analyzes sensor and network telemetry to trigger on-demand instantiation of security functions, achieving rapid containment of detected threats without manual reconfiguration [27].

2.2 Fault Tolerance Through Network Function Virtualization

NFV decouples network functions from dedicated hardware by running them as software instances on commodity servers. In industrial control networks this allows rapid instantiation, scaling, and migration of critical services such as firewalls, protocol translators, and intrusion–detection systems. Orchestration platforms monitor the health of each VNF and

trigger failover or relocation in case of faults if implemented, ensuring continuity of both control-plane and data-plane operations [28].

Container orchestration with Docker Swarm provides a lightweight fault-tolerance layer for NFV deployments. Swarm maintains a desired state by replicating service tasks across manager and worker nodes, detecting container or node failures, and redistributing workloads automatically [29]. Leader election among Swarm managers and built-in service discovery enables uninterrupted connectivity even when individual containers crash or hosts go offline, making it well suited for edge deployments in OT environments.

Redundancy and automatic recovery in Docker Swarm extend to traffic flows as well. Swarm's overlay network reroutes service requests away from failed containers, while recovery scripts can trigger on-the-fly redeployment of VNFs [29]. Combined with real-time anomaly-detection models, this approach minimizes downtime in SCADA and ICS applications by keeping NFV services continuously available.

Kubernetes offers native fault-tolerance mechanisms that address both crash faults and leader election faults. Liveness and readiness probes allow the kubelet to detect and restart unhealthy pods automatically [33]. At the control plane level, Kubernetes components such as the API server and controller manager participate in leader election to guarantee high availability; if the current leader fails, another takes over without manual intervention.

ReplicaSets and StatefulSets enforce desired replica counts and maintain stable network identities for critical VNFs. Kubernetes also supports pod disruption budgets and automated node health checks, which coordinate rolling updates and graceful evictions. This robust crash-fault recovery and leader election process is key to preserving the deterministic performance and high availability required by OT networks [33].

Although Docker Swarm offers simplicity and rapid deployment with its built-in service replication and leader election, Kubernetes delivers a far more comprehensive set of native fault-tolerance features. Automated liveness and readiness probes allow the system to detect and restart unhealthy pods without manual intervention. Rolling updates with automatic rollbacks and pod disruption budgets ensure that VNFs maintain minimum availability even during upgrades or node failures. Kubernetes' multi-master control-plane high availability handles leader-election faults seamlessly, while its extensible API and custom-controller

framework support advanced health monitoring and automated scaling of network functions based on performance or security alerts.

2.2.1 Related Works on Fault Tolerance Through Network Function Virtualization

A. Sajid, H. Abbas and K. Saleem review cloud-assisted IoT-based SCADA security architectures and show how NFV combined with machine-learning can dynamically deploy virtual security functions at cloud or edge locations to detect and mitigate legacy vulnerabilities [28].

A. Mikhail, I. A. Kamil and H. Mahajan propose redundancy and failover strategies for SCADA systems by integrating NFV with real-time anomaly detection; their design automatically reroutes traffic away from faulty controllers and triggers failover of virtualized control functions to maintain uninterrupted operation [30].

K. Olorunnife et al. present a container-based framework for IoT edge applications in which NFV orchestration detects container failures, spins up replacement services, and reroutes data streams, achieving automatic failure recovery with minimal disruption to industrial processes [31].

R. Botez, A.-G. Pasca and V. Dobrota demonstrate Kubernetes-based orchestration of containerized network functions for 5G core networks using Open Source MANO, showing seamless redeployment of VNFs to preserve service availability under dynamic load and fault conditions in industrial scenarios [32].

S. Novanana, A. Kliks, A. S. Arifin and G. Wibisono explore the provisioning of coexisting enhanced Mobile Broadband and Ultra-Reliable Low-Latency Communication slices in 5G networks through Kubernetes-based MANO, illustrating how NFV-enabled network slicing can support both high-throughput and latency-sensitive workloads while ensuring rapid recovery from component failures [33].

2.3 Federated Learning and Distributed Intelligence

With the proliferation of IIoT devices, OT environments face increased challenges related to data privacy, bandwidth limitations, and centralized processing bottlenecks. Federated Learning (FL) offers a decentralized solution by enabling local training of machine learning models on edge devices, transmitting only model updates to a central aggregator [34]. This minimizes data movement, preserves confidentiality, and reduces reliance on cloud infrastructures.

In practical deployments, both the aggregator and the edge worker components in FL architectures are typically containerized, allowing them to be deployed flexibly across distributed infrastructures. Containerization facilitates portability, scalability, and orchestration using platforms such as Kubernetes. Several frameworks support the implementation of FL systems, including OpenFL, TensorFlow Federated, and PySyft. Among these, the Flower framework has demonstrated superior adaptability across varied environments. Flower also benefits from an active developer community and comprehensive documentation, making it accessible for rapid prototyping and production-scale deployments [35].

Technically, FL workflows in OT environments can be orchestrated through network slices supported by SDN and NFV. This enables edge devices to participate in collaborative model training within isolated communication environments that prioritize latency and bandwidth requirements. By using SDN controllers to route FL traffic through low-latency paths and NFV to dynamically deploy aggregator services at the edge, system performance is optimized while maintaining data privacy and security.

FL also introduces an additional layer of adaptability, allowing OT systems to evolve in their decision-making capacity without extensive infrastructure changes. Distributed intelligence allows multiple edge devices or industrial assets to train on localized data sources in parallel, each contributing to models that are either task-specific or part of a unified global intelligence system [37]. This supports parallel learning for different model types (e.g., predictive maintenance, anomaly detection, fault classification) in separate network slices. As a result, OT systems benefit from real-time insights, improved system awareness, and faster response to process deviations or cybersecurity threats.

By integrating FL with SDN and NFV in OT networks, intelligent analytics can be deployed closer to the source of data generation, enabling not only real-time decision-making but also long-term strategic improvements through decentralized and iterative learning.

2.3.1 Related Works on Federated Learning and Distributed Intelligence

D. Upadhyay et al. develop a comprehensive SCADA/IoT test bench that emulates industrial control workflows on a hardware simulator, against which they evaluate lightweight cipher algorithms for secure communications. Beyond cipher benchmarking, they integrate a federated learning layer that enables distributed nodes to collaboratively train anomaly-detection models without sharing raw process data. Their architecture demonstrates how combining realistic testbed emulation with privacy-preserving learning can enhance both data security and attack detection in resource-constrained OT environments [34].

Z. Du et al. apply federated learning to the problem of channel state information (CSI) feedback in massive MIMO systems. Rather than centralizing all CSI measurements, their scheme allows multiple base stations or user-equipment nodes to locally train sub-models on channel observations and then securely aggregate updates. This approach preserves user privacy, reduces the need for raw data transfer, and achieves model accuracy comparable to a fully centralized solution [36].

W. Marfo et al. propose a federated anomaly-detection framework specifically tailored for ICS networks. Each control node trains a lightweight detection model on its own telemetry, and only model gradients are shared with a coordinating server. This collaborative setup protects sensitive operational data while steadily improving global detection capabilities against ICS-specific anomalies [37].

A. Santorsola et al. introduce a reinforcement-learning agent designed to discover safety-critical states within smart grid environments. By virtualizing grid components and running simulations under varying load scenarios, their agent learns to identify conditions that precede hazardous events. This distributed intelligence approach enables proactive defense measures by signaling controllers before critical thresholds are crossed [38].

A. N. Jahromi et al. explore a two-level ensemble strategy for cyber-attack detection and attribution in IoT-enabled cyber-physical systems. Local detection modules first flag suspicious behavior at the device level, then their outputs feed into a higher-order classifier that attributes threats to specific attack types. This layered architecture improves both the speed of detection and the precision of attribution without centralizing raw sensor data [39].

2.4 Digital Twins for Predictive Monitoring

Digital twin technology enables the creation of virtual representations of physical systems that are continuously synchronized with real-world operations. These models offer benefits in ICS by allowing predictive maintenance, fault simulation, and real-time monitoring.

From a technical perspective, digital twins require consistent and low-latency data synchronization, which can be facilitated by network slicing and edge service deployment [18]. SDN ensures optimal data routing between physical systems and their digital counterparts, while NFV supports scalable deployment of analytics services close to the source of data. This tight coupling between network control and function placement allows digital twins to operate with minimal latency and maximum accuracy.

The software-based nature of digital twins allows for rapid instantiation of virtual environments without the need for physical infrastructure modifications. This makes them highly cost-effective for developing testbeds, especially when conducting training programs, auditing exercises, or cybersecurity incident response simulations [42]. Organizations can simulate various failure modes, threat scenarios, or maintenance routines in a safe and controlled environment without disrupting live systems.

In this context, the Purdue model becomes particularly relevant. Its hierarchical structure, which segments industrial control systems into discrete layers, aligns well with digital twin deployments. Each layer of the Purdue model, from field devices to enterprise systems, can be independently modeled and monitored within a digital twin environment. This layered approach enhances security, visibility, and operational clarity, allowing targeted simulations and diagnostics at each level.

Digital twins also play a critical role in predictive monitoring, which involves continu-

ously analyzing data from physical systems to forecast future states, detect anomalies, and anticipate failures before they occur [43]. By correlating live data with historical patterns and simulation models, predictive monitoring enables maintenance teams to intervene proactively, reducing downtime and optimizing resource usage.

Furthermore, the orchestrated deployment of digital twin services in OT environments enables modular updates, seamless scaling, and fault isolation. These characteristics are essential for maintaining uninterrupted industrial processes and improving decision-making across critical infrastructure.

2.4.1 Related Works on Digital Twins for Predictive Monitoring

V. Beliakova et al. develop a digital twin of a continuous casting machine mold that fuses real-time sensor data with a physics-based model of heat transfer and wear. By continuously updating the twin's parameters, their system forecasts the remaining useful life of the mold, enabling maintenance teams to schedule interventions just in time to avoid unplanned downtime. This predictive monitoring approach demonstrates how integrating live operational feedback into a virtual replica supports data-driven asset management in heavy-industry contexts [40].

M. Sanz Rodrigo et al. propose a comprehensive methodology for modeling and deploying digital twins within 5G network slices. Their framework formalizes the twin creation lifecycle, from abstract asset representations through deployment on edge-cloud platforms, and shows how real-time telemetry from network functions can feed predictive analytics engines. By mapping network KPIs to twin state parameters, they enable proactive detection of performance degradations and automated QoS adjustments, illustrating the role of digital twins in maintaining ultra-reliable, low-latency communications [18].

S. A. Varghese et al. introduce an intrusion-detection system that leverages a digital twin of ICS network behavior to identify anomalies. The twin mirrors control loop traffic patterns and device states, against which live network flows are continuously compared, and deviations trigger alarms and can automatically isolate compromised segments. This work shows how digital twins provide a predictive monitoring layer for early threat detection and containment in critical infrastructure [41].

P. Abinaya et al. present a case study of a digital-twin-driven cyber-physical system in a smart-manufacturing environment. Their twin integrates data from CNC machines, robotic cells, and conveyor systems to simulate production workflows and detect deviations before they manifest on the shop floor. This predictive monitoring capability supports optimized scheduling and adaptive quality control, demonstrating the practical benefits of twin-based oversight in complex manufacturing chains [42].

N. Mohamed and J. Al-Jaroodi survey architectures for predictive analytics in digital-twin platforms, highlighting common design patterns for data ingestion, state estimation, and anomaly forecasting. They discuss how stream-processing pipelines fuse historical logs with current sensor streams and apply machine-learning models to predict future operational states. Their conceptual framework guides practitioners in building twin ecosystems that proactively surface potential failures and performance bottlenecks across diverse cyber-physical applications [43].

2.5 Dynamic Demilitarized Zones (DMZs) for Operational Technology Cybersecurity

Traditional DMZs are network segments that act as protective barriers between internal control systems and external networks. Their primary purpose is to filter and monitor the data that flows between operational systems and business or internet-facing systems, offering a space where security mechanisms can inspect and control traffic before it reaches critical components [44]. However, in OT environments that require flexibility and adaptability, static DMZs often fall short because they cannot scale or reconfigure rapidly to meet changing network conditions or emerging threats.

In the Purdue model, DMZs are generally placed between Level 3 (operations and control) and Level 4 (enterprise or IT systems). This strategic placement allows them to inspect and regulate traffic between the business layer and the plant-floor operational layers, ensuring that only safe and necessary data can cross into more sensitive parts of the industrial system [2]. DMZs serve as security checkpoints, enforcing strict controls and providing visibility

into inter-domain communications.

Within a DMZ, several security services are typically deployed to strengthen the perimeter of the OT network. Intrusion Detection Systems (IDS) are used to monitor traffic for malicious patterns or policy violations. These systems can be signature-based, using known attack profiles, or anomaly-based, detecting deviations from expected behavior. Security Information and Event Management (SIEM) platforms aggregate logs and security data from across the network, analyze patterns, and generate alerts or reports to assist with incident response and regulatory compliance. Firewalls are fundamental components that filter traffic based on predetermined rules, blocking unauthorized communication and enforcing policy boundaries [47]. Sandbox environments provide isolated, controlled spaces where potentially malicious files or scripts can be safely executed and analyzed without risking disruption to critical systems.

Dynamic DMZs build on this concept by using containerized services and network slicing to create agile and responsive security perimeters. Instead of relying on fixed hardware or manual configurations, these systems deploy and adjust security tools on demand. SDN enables traffic flows to be reprogrammed in real time, diverting data through firewalls or monitoring tools as needed [44]. NFV makes it possible to instantiate and scale security services like IDS or SIEM based on current network loads or threat conditions, eliminating the delays and limitations of physical provisioning.

This approach allows OT environments to dynamically adapt to cyber threats, minimize the risk of lateral movement by attackers, and ensure that communication policies are strictly enforced based on the context and criticality of networked assets [48]. By embedding security into a programmable, scalable framework, dynamic DMZs provide a powerful mechanism for maintaining resilience and regulatory compliance in increasingly complex industrial control systems.

2.5.1 Related Works Dynamic Demilitarized Zones (DMZs) for Operational Technology Cybersecurity

Miteff and Hazelhurst describe NFShunt, a Linux-based firewall architecture that integrates hardware bypass via OpenFlow switches to offload trusted flows around the software pipeline while directing suspicious traffic for deep inspection. This design enables high-performance DMZ segmentation in OT networks by combining hardware-accelerated forwarding with software-based security checks [44].

Chowdhary et al. propose the Science DMZ, an SDN-based testbed tailored for secure, high-bandwidth scientific data transfers. Their architecture uses OpenFlow controllers to segment network flows at the perimeter, isolating data-intensive workloads in a dedicated DMZ and enforcing flow-specific security policies without compromising throughput. This model offers a template for OT DMZ deployments that balance performance and security requirements [45].

Kilincer et al. present a security framework that combines IEEE 802.1X port-based authentication, DMZ segmentation, and SSL-VPN tunnels for IoT network protection. In their approach, devices authenticate before gaining access to DMZ subnets, and all IoT traffic is encrypted through SSL-VPN gateways. The layered DMZ structure restricts lateral movement and secures remote access to critical control systems [46].

Maulana et al. analyze a dual-DMZ firewall architecture implemented in Java Madura Bali Electrical Systems to enhance IT/OT cybersecurity. Their design positions two sequential DMZ layers—each enforced by dedicated firewalls—between corporate and control networks. By filtering traffic in stages, the dual DMZ arrangement reduces unauthorized cross-domain access and strengthens the security posture of industrial infrastructure [47].

Laili et al. introduce the Industrial DMZ (IDMZ) concept for secure edge computing in industrial environments. IDMZ isolates edge computing nodes handling communication-intensive offloading tasks by enforcing strict access control and secure channel establishment at the DMZ boundary. This framework ensures that edge workloads can be offloaded securely while protecting the core OT network from potential threats [48].

2.6 Summary

The reviewed literature provides a strong foundation for addressing the key challenges of flexibility, decentralization, and resilience in modern OT networks. SDN and NFV offer agile, programmable network management. Federated Learning introduces privacy-preserving intelligence at the edge. Digital Twins enhance operational awareness and proactive maintenance. Dynamic DMZs provide adaptive cybersecurity tailored to the sensitivity of each service. Despite progress in each area, there remains a gap in holistic integration. This thesis addresses that gap by proposing a slicing-enabled architecture that unifies these emerging technologies, aiming to meet the unique requirements of Industry 4.0 in SCADA and ICS systems.

Chapter 3

Reconfigurable Network Slicing Orchestration in OT Environment

The foundation of this chapter is built around the first research question: How can OT networks take advantage of NFV and SDN technology to become cyber resilient? Industrial control environments have long depended on static, hardware-bound infrastructures that prioritize reliability and deterministic performance but lack the agility to respond to today's fast-evolving cyber threats. Integrating NFV and SDN into OT networks introduces the ability to decouple network services from proprietary hardware and centralize traffic control, enabling dynamic adaptation to faults, attacks, or operational changes. By leveraging these technologies, OT networks can move beyond rigid, perimeter-based defenses toward architectures capable of automated threat mitigation, fine-grained traffic management, and scalable service deployment. However, realizing this vision requires addressing key challenges, including maintaining low-latency communication, ensuring system availability during reconfigurations, and aligning virtualized services with strict industrial safety and performance constraints.

This chapter contributes to answering this research question by presenting the architecture and experimental deployment of a dynamic network slicing orchestrator designed for ICS environments. We describe the system's components, detail the process of slice instantiation, and explain how Quality of Service (QoS) mechanisms are integrated to safeguard critical traffic flows. The chapter further reports on a series of scenario-based experiments that evaluate the orchestrator's performance across key metrics, including deployment time, throughput, latency, jitter, and packet loss. These results establish a practical foundation for applying NFV and SDN in OT networks to enhance cyber resilience, providing empirical insights into their capacity to deliver adaptable, secure, and high-performance operations

under Industry 4.0 demands.

3.1 Network Slicing Enabled ICS Architecture

Integrating network slicing into traditional Industrial Control System (ICS) environments introduces a transformative approach to improving security, enhancing operational efficiency, and fostering adaptability. By segmenting a unified network into distinct slices tailored to specific data transfer requirements, industrial sectors can address cybersecurity challenges, ensure data availability, and protect critical processes and services.

This thesis focuses on orchestrating network segmentation within integrated ICS landscapes, with particular emphasis on optimizing industrial operations and elevating network performance. The proposed architecture highlights key design considerations and strategic orchestration processes essential for the seamless deployment of network slicing in modern ICS infrastructures, especially as the number of Internet of Things (IoT) and advanced sensor devices continues to grow rapidly.

To overcome challenges posed by legacy communication protocols and to maintain the integrity and security of sensing data, the architecture incorporates advanced security mechanisms such as real-time threat detection systems. Supporting this transformation requires the adoption of next-generation components, including robust and heterogeneous devices capable of sustaining segmented and scalable industrial networks.

Our proposed architecture, illustrated in Figure 3.1, introduces the concept of slicing, where each slice corresponds to a distinct operational layer within the ICS environment.
Supervisory Layer: Slice 1 represents the supervisory layer, containing critical ICS components such as SCADA servers, historians, and Human-Machine Interfaces (HMIs). This layer acts as a centralized management hub, responsible for overseeing and coordinating decentralized slices across the system.
Control Layer: Slice 2 (and potentially sub-slices within it) constitutes the control layer, housing Programmable Logic Controllers (PLCs) and Remote Terminal Units (RTUs). This layer handles process-specific control and monitoring tasks essential for efficient management of industrial operations.

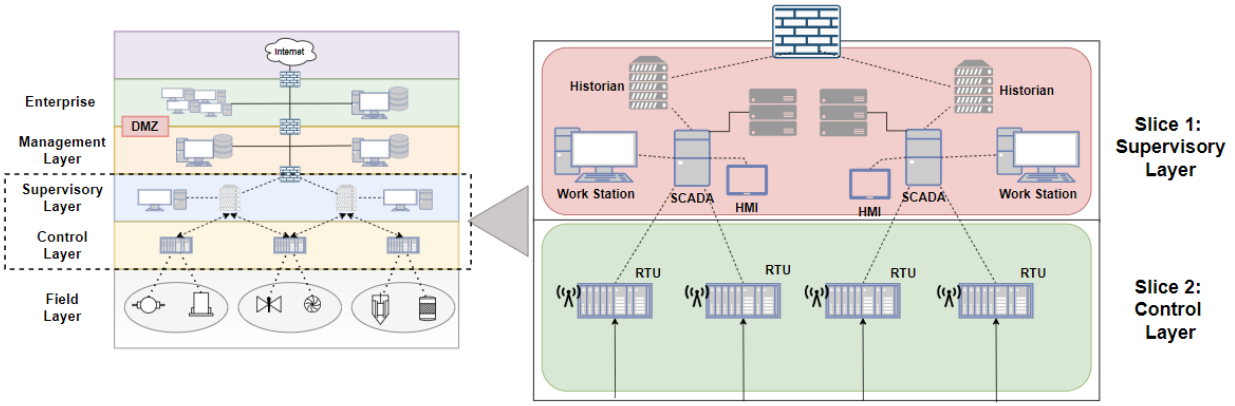

Figure 3.1: Network Slicing Integrated ICS Architecture [49]

Each slice operates autonomously, significantly reducing the risk of a single point of failure affecting the entire network. The centralized management slice (Slice 1) facilitates streamlined coordination and incident response, while decentralized slices ensure isolated execution of control functions. This segmentation model supports scalable expansion, allowing new slices to be added or modified without disrupting ongoing processes.

3.2 Slicing Orchestration Deployment Process

3.2.1 Environment Description

This section describes the experimental environment designed to validate the custom scenario-based network slicing orchestrator for Industrial Control Systems (ICS). Figure 3.2 presents the software components and platform used during implementation:

- **Open Source MANO (OSM):** OSM [20] acts as the central orchestrator, managing and coordinating network services and Virtual Network Functions (VNFs) across the SCADA environment. It handles the full lifecycle of VNFs—from instantiation to termination—ensuring that all network components operate according to predefined policies. OSM enables rapid deployment, scaling, and reconfiguration of network services, significantly simplifying network operations and enhancing system flexibility.

- **OpenStack:** OpenStack serves as the Virtual Infrastructure Manager (VIM), man-

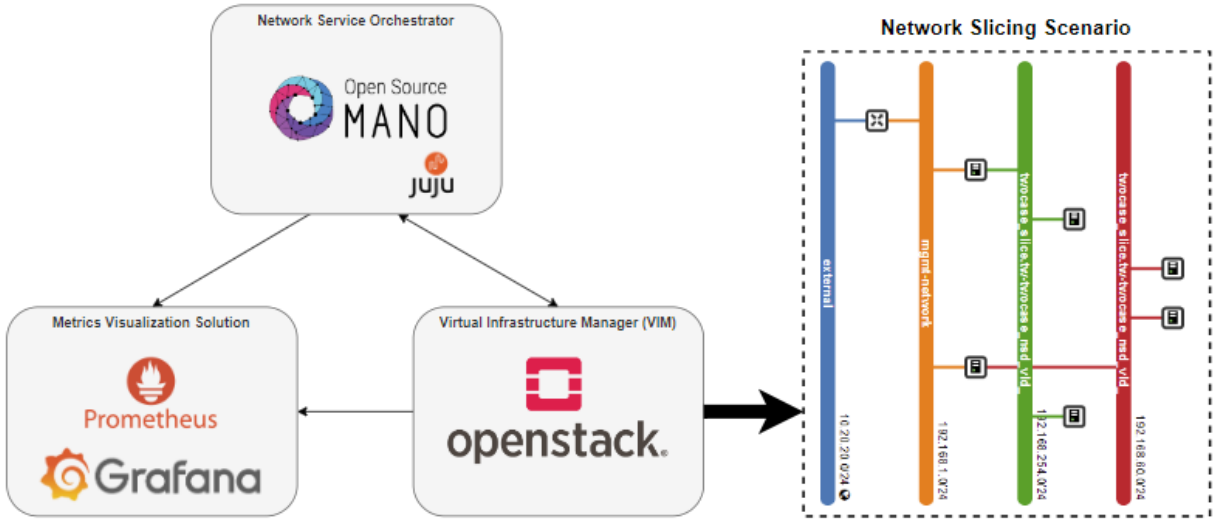

Figure 3.2: Orchestrating Network Slices with OSM [49]

aging the underlying virtualized infrastructure. It oversees resource allocation, virtual machine provisioning, and virtual networking. Through OpenStack, a scalable and flexible infrastructure was created to support the instantiation of network services necessary for robust SCADA operations.

- **Grafana and Prometheus:** Grafana and Prometheus were integrated for real-time monitoring and visualization. Prometheus collects, stores, and queries network and system metrics, while Grafana offers a flexible dashboard for visual analysis. Together, they enable proactive monitoring of network health, performance, and security, facilitating rapid anomaly detection and mitigation.

Once the VIM is successfully integrated with OSM, three key configurations are required to orchestrate the instantiation of network services:

- **Virtual Network Function Descriptor (VNFD):** Specifies the functional description, deployment requirements, connection points, configuration parameters, and lifecycle management policies of individual VNFs.
- **Network Service Descriptor (NSD):** Defines how multiple VNFs are interconnected to form complete network services. It outlines communication requirements,

service chaining policies, and relationships between VNFs.

- **Network Slice Template (NST):** Describes the composition of an end-to-end network slice, including virtualized functions and infrastructure resources. It specifies resource allocation, network topology, and slice-specific configurations required for slice instantiation and management.

3.2.2 Network Services Instantiation

To deploy a custom network topology for the scenario, it is first necessary to define a Network Slice Template (NST) diagram. As shown in Figure 3.3, the management slice for the supervisory layer is represented at the top (slice_nsd_3). In this deployment, two additional slices are instantiated for the control layer (slice_nsd_1 and slice_nsd_2), connected through virtual links (VLs) specified in the NST and connection points (CPs) declared in the Network Service Descriptor (NSD).

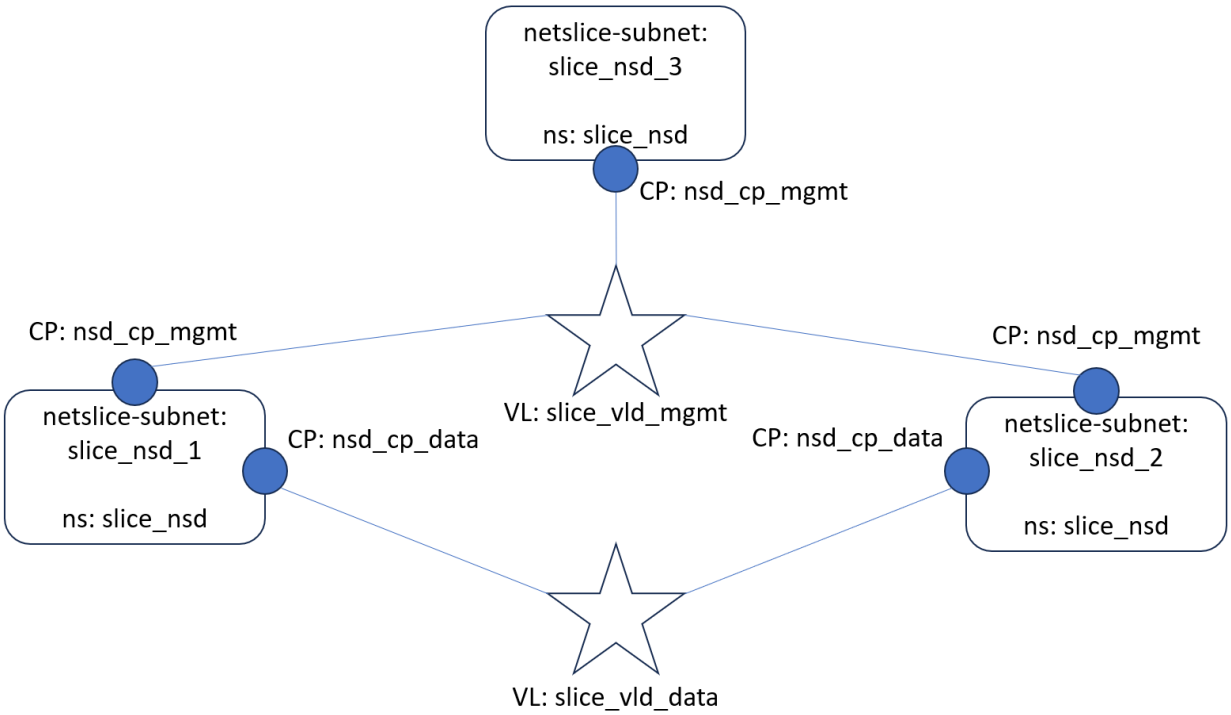

Figure 3.3: Network Slice Template (NST) Diagram [49]

In this configuration, the first control slice is designated for handling data from sen-

sors and actuators, while the second slice is dedicated to transmitting health and status information from these devices.

The elements outlined in the NST diagram are translated into YAML documents for practical implementation. Subsequently, each Virtual Network Function Descriptor (VNFD) is created as a package, populated with its required specifications, verified for correctness, and uploaded to the Open Source MANO (OSM) platform using its API. The same process is followed for the NSDs, with a specific consideration: if endpoint instances are intended to be deployed simultaneously with the network services, a secondary NSD containing a single connection point must also be created. Once the VNFDs and NSDs are in place, the NST can be verified and uploaded, enabling the network services to be instantiated through either an API call or via OSM's graphical user interface (GUI).

On the OpenStack side, the resulting topology mirrors the intended network slicing scenario described previously in Figure 3.2. The management slice is connected through router-like instances to the orchestrated control layer slices. Two endpoint instances are connected to each control slice to demonstrate the segmentation of nodes within the network. A router linked to both the management slice and the external subnet is used to provide Internet access to the management layer and the intermediary router instances through floating IPs.

3.2.3 Quality of Service (QoS) in Slices

To ensure quality of service (QoS) within the deployed network slices, OpenStack's API capabilities were utilized. At the time of writing, Open Source MANO (OSM) does not natively support QoS parameter mapping to OpenStack's Virtual Infrastructure Manager (VIM). Additionally, by default, OpenStack does not offer built-in QoS capabilities within private cloud deployments.

To address this limitation, the Neutron QoS extension, part of OpenStack's network management service, was manually installed and configured. After installation, the Neutron service was restarted to enable QoS policy definitions across the virtual network.

A QoS policy is required to define the specific network constraints. For this evaluation, a policy was created to limit bandwidth to 5 Gb/s, with a maximum burst bandwidth set at 80% of that value to stabilize throughput under varying loads. In OpenStack, policies

can either be applied automatically to all newly generated slices or selectively associated with specific slices. For this project, the latter approach was used: the QoS policy was assigned explicitly to the control layer slices, aligning with the focus on controlling intra-slice communication efficiency.

QoS rules and policies can be dynamically updated and reassigned to floating IPs, virtual ports, or slices without requiring service downtime. This flexibility enables real-time adjustments to network performance according to operational needs. However, to maintain consistency during the controlled testing phase of this thesis, the QoS settings were fixed at the predefined values throughout all evaluations.

3.3 Experimental Results and Discussion

To assess the performance of the proposed network slicing orchestrator and the deployed network segments, a series of experiments were conducted focusing on key network performance metrics:

- Deployment time
- Throughput
- Average round-trip time (RTT)
- Jitter
- Packet loss

The experiments were structured around two categories, distinguished by the number of flows and the corresponding number of slices and connected endpoints. In the single-flow scenario, intra-slice communication over TCP/IP was evaluated, recognizing that TCP/IP forms the transport foundation for many OT protocols such as Modbus and DNP3. Throughput was measured using iPerf, where one endpoint acted as a server and another as a client over a 30-second data transfer window. Average RTT was calculated using the `ping` command by transmitting 10 Internet Control Message Protocol (ICMP) packets, each of 1,500 bytes.

In the dual-flow scenario, similar TCP-based evaluations were performed, along with additional UDP testing. The average RTT was again measured using `ping`, while throughput, jitter, and packet loss were analyzed using iPerf, with a predefined bandwidth constraint. To configure the desired bandwidth limits, OpenStack's Quality of Service (QoS) functionality was employed through its API. Specifically, a maximum bandwidth of 5 Gb/s was configured, along with a maximum burst threshold of 80% (4 Gb/s), to stabilize throughput behavior during the evaluations of the associated network slices.

3.3.1 One Flow Findings

The deployment times for each flow scenario are presented in Figure 3.4. As the number of instantiated network services per slice and associated endpoints increases, the time required to deploy the defined topology from the Network Slice Template (NST) also rises. Deployment times, measured from service instantiation to the readiness of instances and subnets, range from under 90 seconds to approximately 150 seconds.

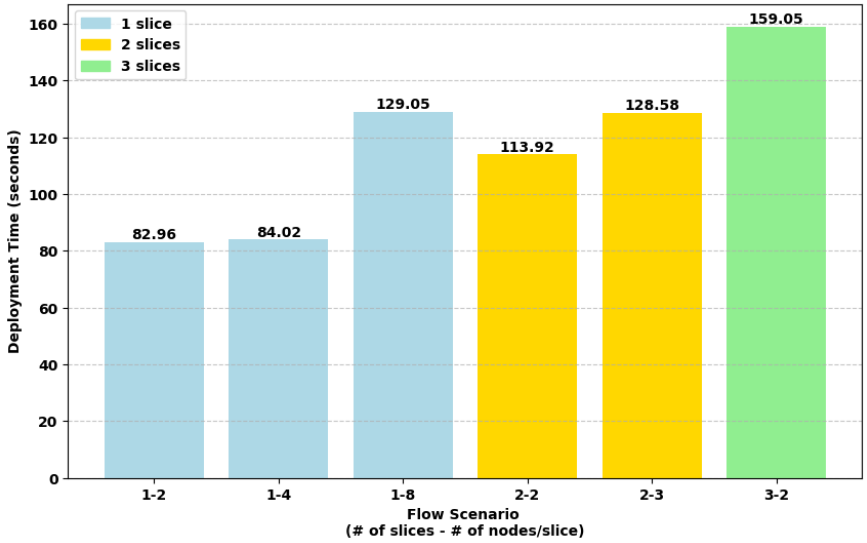

Figure 3.4: Deployment times per flow scenario [49]

As part of Industry 4.0 initiatives and to meet the networking demands of OT environments, maintaining high throughput is essential. Figure 3.5 illustrates the throughput values achieved across the evaluated scenarios, consistently measured in the gigabit-per-

second (Gb/s) range, which is suitable for production-level solutions. While throughput decreases progressively as the number of slices and connected endpoints increases, this behavior is expected due to the growing overhead in managing additional network processes, inter-slice communications, and device connections. Importantly, the reduction in throughput remains moderate, and the overall system continues to operate within the parameters of a high-throughput environment.

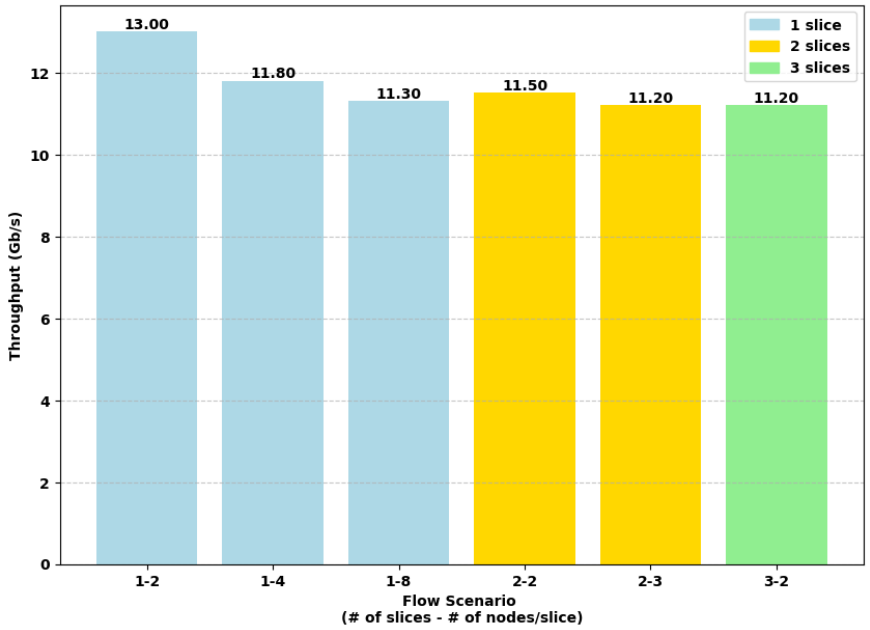

Figure 3.5: One flow intra-slice throughput [49]

Finally, the average round-trip time (RTT) across flow scenarios is shown in Figure 3.6. RTT, which serves as an indicator of network latency, is a critical metric for ultra-reliable low-latency communications (URLLC), where latencies below 20 milliseconds (ms) are typically required. Across all evaluated scenarios, the average RTT remained below 1 ms, providing substantial headroom for supporting heavier workloads.

3.3.2 Two Concurrent Flows Findings

Previously, TCP was utilized to evaluate throughput under single-flow scenarios. In this phase of testing, UDP is also introduced to assess network performance in terms of jitter and packet loss, enabling a comparison with TCP throughput measurements obtained under

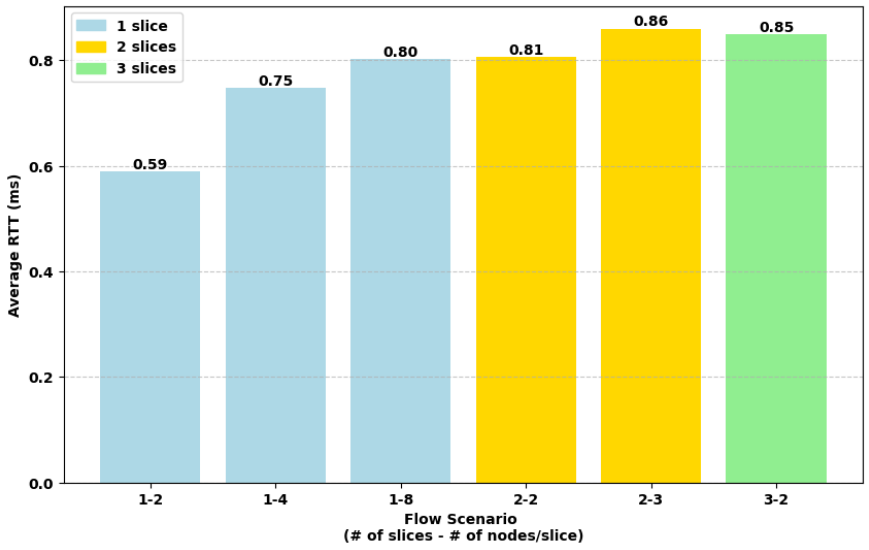

Figure 3.6: One flow average round-trip time (RTT) [49]

varying buffer and window sizes. Throughput performance across TCP and UDP flows is compared to understand differences in behavior under different transport protocols.

For all comparisons, the scenario examined corresponds to a topology with one slice and four endpoints (labeled 1–4). Additional experiments were conducted under 1-8 and 2-3 configurations (# of slices – # of endpoints per slice), as a minimum of three endpoints per slice was required to properly simulate concurrent flows. In each setup, one endpoint acted as the iPerf server or the ping target, while the remaining two endpoints served as iPerf clients and ping requesters.

For TCP throughput, Figure 3.7 illustrates how increasing the TCP window size leads to a balancing effect between the two concurrent client flows, with throughput values converging toward approximately 2.6 Gb/s. This behavior aligns with the enforced Quality of Service (QoS) configuration, where a bandwidth limit of 5 Gb/s and a maximum burst allowance of 80% were applied. The observed convergence highlights the effectiveness of QoS policies in maintaining throughput stability across concurrent flows.

In the case of UDP throughput, Figure 3.8 demonstrates a similar trend of flow balancing. However, the overall throughput values achieved using UDP were significantly lower than those observed with TCP. Specifically, UDP throughput remained in the megabit-per-second (Mb/s) range, attributable to the lack of congestion, flow control, and error correction

mechanisms inherent in the UDP protocol.

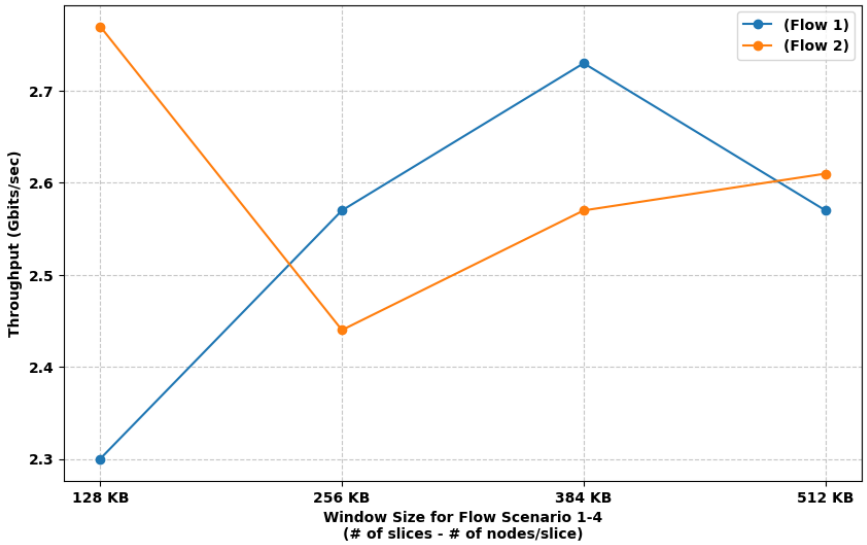

Figure 3.7: TCP throughput by flows [49]

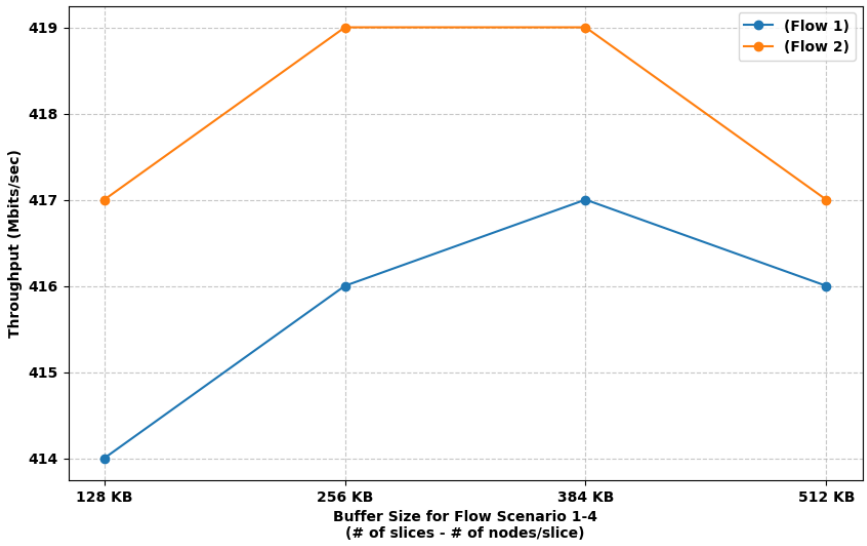

Figure 3.8: UDP throughput by flows [49]

Further analysis of UDP performance is provided in Figure 3.9, which presents packet loss measurements for single-flow and dual-flow scenarios. Results show that packet loss rates are generally lower under single-flow conditions but increase noticeably when two concurrent flows are active. Additionally, larger buffer sizes exacerbate packet loss, as multiple rapid

UDP transmissions overwhelm available buffer capacity, leading to higher rates of dropped packets.

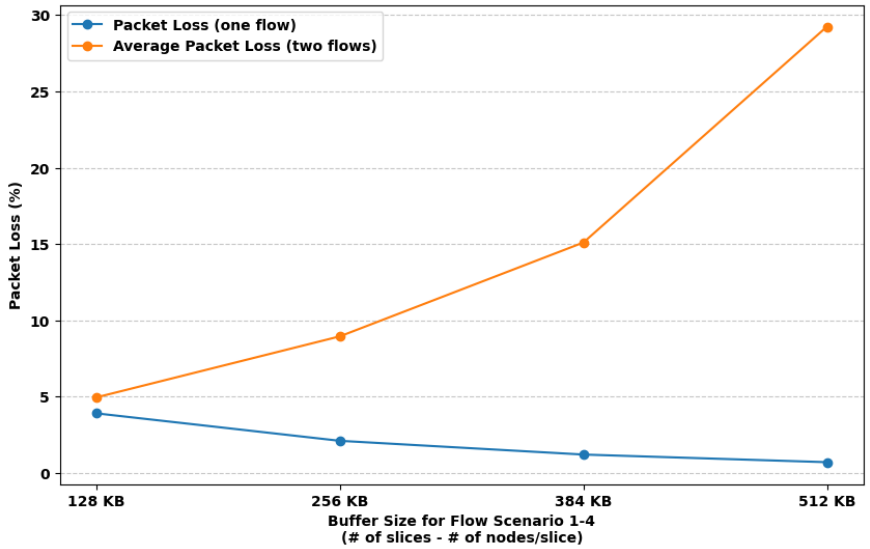

Figure 3.9: Packet loss flow comparison [49]

Figure 3.10 illustrates the jitter measurements for the evaluated scenarios. Interestingly, jitter was found to be lower for two concurrent flows compared to a single flow. This outcome is largely influenced by the active QoS policies, which regulate bandwidth and maximum burst sizes, resulting in more controlled packet delivery patterns under heavier network loads. The nearly linear behavior observed in the two-flow jitter measurements further reinforces the impact of QoS on maintaining network stability.

Finally, in Figure 3.11, the average round-trip time (RTT) results reveal distinct patterns: for single-flow scenarios, RTT exhibited a gradual linear increase, whereas in the dual-flow scenarios, RTT values remained relatively stable. This behavior can be attributed to the influence of QoS on jitter control, and consequently, on overall latency, as lower jitter tends to stabilize RTT measurements.

The metrics and behaviors observed across these experiments provide an important baseline for evaluating the scalability and reliability of the proposed network slicing solution. Maintaining high-throughput, low-latency, and reliable data transfer in ICS environments is critical for ensuring that operational demands are met while upholding stringent standards of availability, performance, and security.

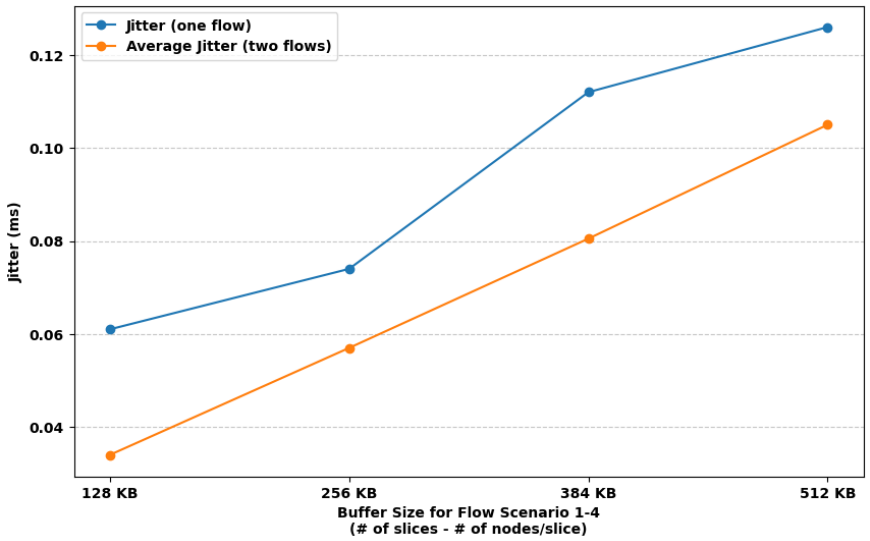

Figure 3.10: Jitter flow comparison [49]

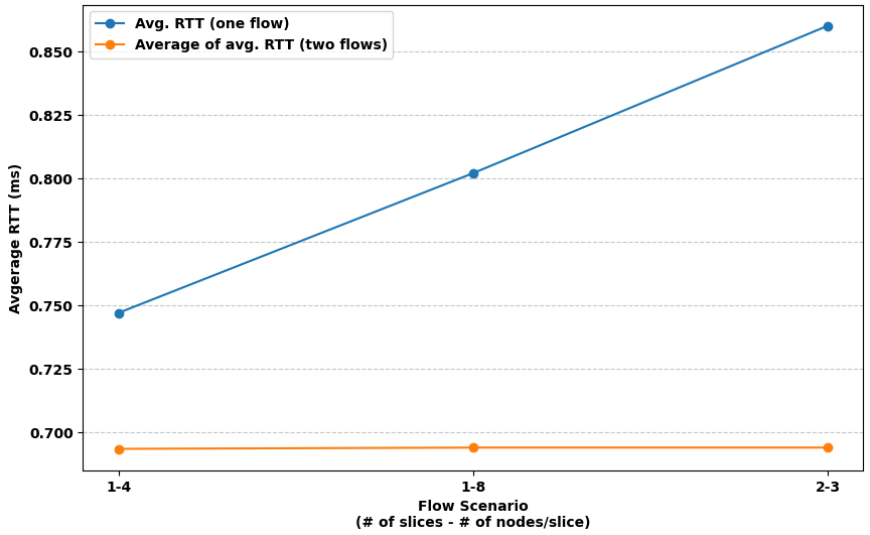

Figure 3.11: Average RTT flow comparison [49]

3.4 Summary

This chapter presented the design, deployment, and experimental evaluation of a dynamic network slicing orchestrator tailored for industrial control system (ICS) environments. By integrating Software-Defined Networking (SDN) and Network Function Virtualization (NFV), the proposed architecture enables OT networks to move beyond static, hardware-bound infrastructures toward adaptive and cyber-resilient operations. We detailed the system

components, the slicing deployment process using Open Source MANO and OpenStack, and the incorporation of Quality of Service (QoS) mechanisms to prioritize critical traffic flows. Through a series of single- and multi-flow experiments, we assessed deployment time, throughput, latency, jitter, and packet loss, demonstrating the orchestrator's capacity to maintain high performance, low latency, and reliable communications even under increasing load. These findings establish a foundational framework for applying NFV and SDN in OT networks, offering practical insights into enhancing scalability, security, and operational efficiency as industrial environments evolve toward Industry 4.0 standards.

Chapter 4

Network Slicing for Federated Learning in OT Environment

The second research question asks how dynamic network slicing can be implemented to optimize network resource usage and support decentralized machine learning. As industrial environments increasingly generate large volumes of data from distributed devices, the need to process this data efficiently and securely at the edge has become critical. Decentralized approaches like federated learning (FL) offer a promising solution by enabling local model training without transferring sensitive raw data to centralized servers. However, running FL alongside industrial control processes places competing demands on shared network and compute resources, requiring predictable isolation, robust resource allocation, and minimal interference with real-time operations. Dynamic network slicing provides a pathway to meet these needs by creating logically separate, policy-driven network partitions tailored to the distinct requirements of FL and process-centric workloads, ensuring both operational efficiency and cybersecurity.

This chapter contributes to answering this research question by extending the network slicing framework to embed federated learning within industrial control system environments. We introduce a dual-slice architecture that cleanly separates process-control functions from FL-centric monitoring and describe the orchestration workflow that enables dynamic instantiation and scaling of these slices on cloud infrastructure. We also explain how the design integrates a dynamic DMZ to secure inter-slice traffic, alongside quality-of-service and fault-management mechanisms that uphold deterministic, time-critical operations. Finally, we report on scenario-based experiments that assess both the infrastructure overhead and the anomaly-detection accuracy of the federated models. These results offer practical insights into deploying intelligent, slice-aware monitoring systems in OT environments, laying the

groundwork for resilient, privacy-preserving analytics under Industry 4.0 conditions.

4.1 Federated Learning Integrated Network Slicing

Integrating network slicing into traditional Industrial Control System (ICS) environments offers a powerful mechanism to enhance security, optimize efficiency, and increase adaptability across a wide range of industrial applications. This method involves dividing a shared physical network into isolated virtual segments, each tailored to specific data transmission needs. By doing so, it mitigates cybersecurity threats, preserves data integrity, and ensures that critical processes and services remain continuously available.

This research moves beyond simply applying segmentation in ICS environments. It investigates the role of network slicing within Industry 4.0 scenarios, particularly those that involve distributed and computationally intensive operations such as edge-based federated learning (FL). The goal is to not only improve operational performance but also enable networks to support complex requirements such as fault tolerance, self-healing behavior, and intelligent task distribution—all of which are critical in modern industrial systems.

The approach taken in this work focuses on architectural planning and orchestration strategies that allow seamless deployment and scaling of network slices. These strategies become increasingly important as ICS infrastructures evolve to incorporate a variety of Internet-of-Things (IoT) devices and advanced sensors.

To ensure the confidentiality and reliability of sensing data, it is essential to overcome limitations associated with legacy communication protocols. Advanced security mechanisms, such as real-time anomaly detection systems, must be implemented to monitor for threats and maintain operational trust. Furthermore, the adoption of scalable and modular technologies enables more resilient network deployments, allowing systems to adapt to changing industrial workloads without compromising performance.

Figure 4.1 illustrates the proposed architecture, where each slice serves a distinct purpose within the overall ICS framework:

Process Slice: This segment combines elements from both the Supervisory and Control layers of the Purdue Model to manage industrial operations securely and efficiently.

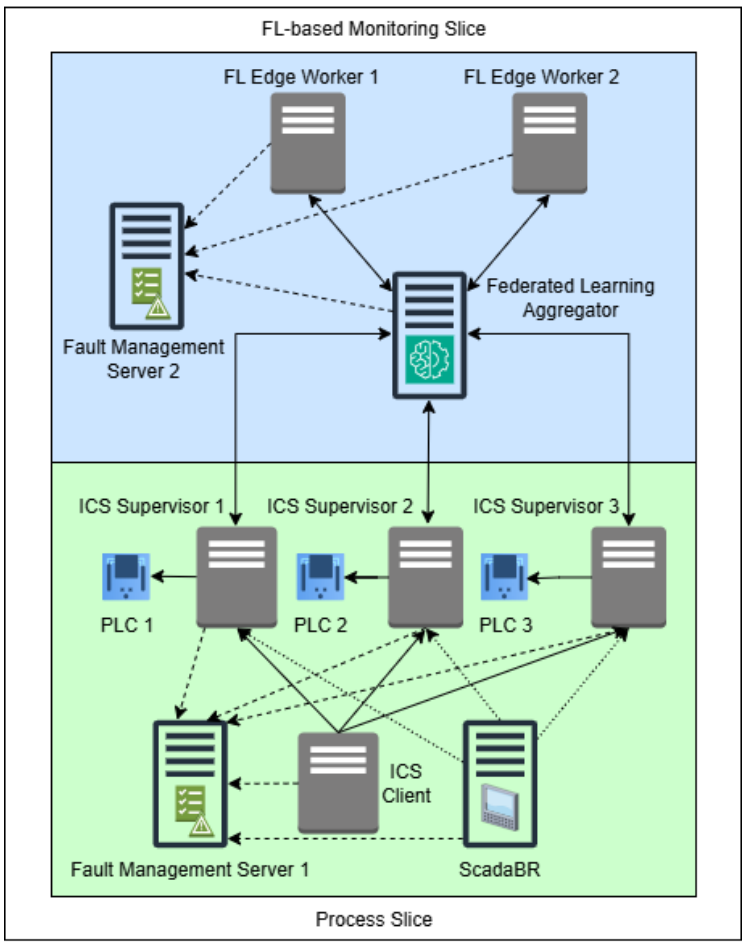

Figure 4.1: FL-based and Process Slices Architecture [50]

ScadaBR, an open-source Human-Machine Interface (HMI) developed in Brazil [51], provides centralized process visualization and monitoring. Fault Management Server 1 is responsible for overseeing all containerized services within the slice, using Docker Swarm and its API to identify and recover from service failures. The ICS Supervisors (1, 2, 3) facilitate communication between the ICS Client and Programmable Logic Controllers (PLCs 1, 2, 3), which directly control physical machinery. The ICS Client stores a dataset of Modbus packets and distributes command data to the Supervisors for execution by the PLCs.

FL-Based Monitoring Slice: Operating entirely within the Supervisory Layer, this slice enables intelligent monitoring through federated learning. The Flower framework [35] is used to deploy an FL Aggregator, which collects and processes updates from edge devices while managing model accuracy and data integrity. Edge Workers 1 and 2 locally train

machine learning models on segmented data in parallel, minimizing the need for data transfer and preserving privacy. Fault Management Server 2 ensures the reliability of this slice by supervising and redeploying containerized services via Docker Swarm as needed to maintain consistent operation.

Each slice functions as a self-contained unit, contributing to a more robust and scalable system architecture. This separation of concerns allows one slice to fail or scale without impacting the performance of others, making the network more resilient. It also enables better security enforcement, improved process isolation, and targeted optimization, which are essential for the secure and efficient functioning of ICS networks in the era of Industry 4.0.

4.2 Slicing Orchestration

4.2.1 Environment Description

This section describes the experimental environment used to evaluate the purpose-driven network slicing orchestrator designed for industrial control systems.

Open Source MANO (OSM) [20] serves as the central orchestrator in this architecture. It is responsible for managing the deployment and lifecycle of virtual network functions (VNFs), ensuring that network services operate according to defined policies. OSM supports automation from creation to decommissioning, which enables seamless adaptation to changing operational conditions. This significantly improves the responsiveness and efficiency of network management.

Google Cloud Platform (GCP) is used as the Virtual Infrastructure Manager (VIM), handling the allocation of compute, storage, and network resources. GCP provides the virtualized infrastructure required for flexible and scalable industrial operations, supporting dynamic provisioning and efficient resource utilization.

For real-time performance monitoring and visualization, Prometheus collects metrics across the system, while Grafana is used to visualize those metrics in an intuitive dashboard. These tools provide operators with insights into the status of network slices and help

detect and respond to anomalies promptly.

4.2.2 Network Services Instantiation

Once OSM is linked to the VIM, deploying a custom network topology begins with the definition of a Network Slice Template (NST). Figure 4.2 presents this architecture, showing the management slice (slice_mgmt) as part of the supervisory layer. Below it are two control layer slices: slice_process and slice_fl_based. These slices are connected using virtual links (VLs) and connection points (CPs) as defined within the NST and the corresponding Network Service Descriptors (NSDs).

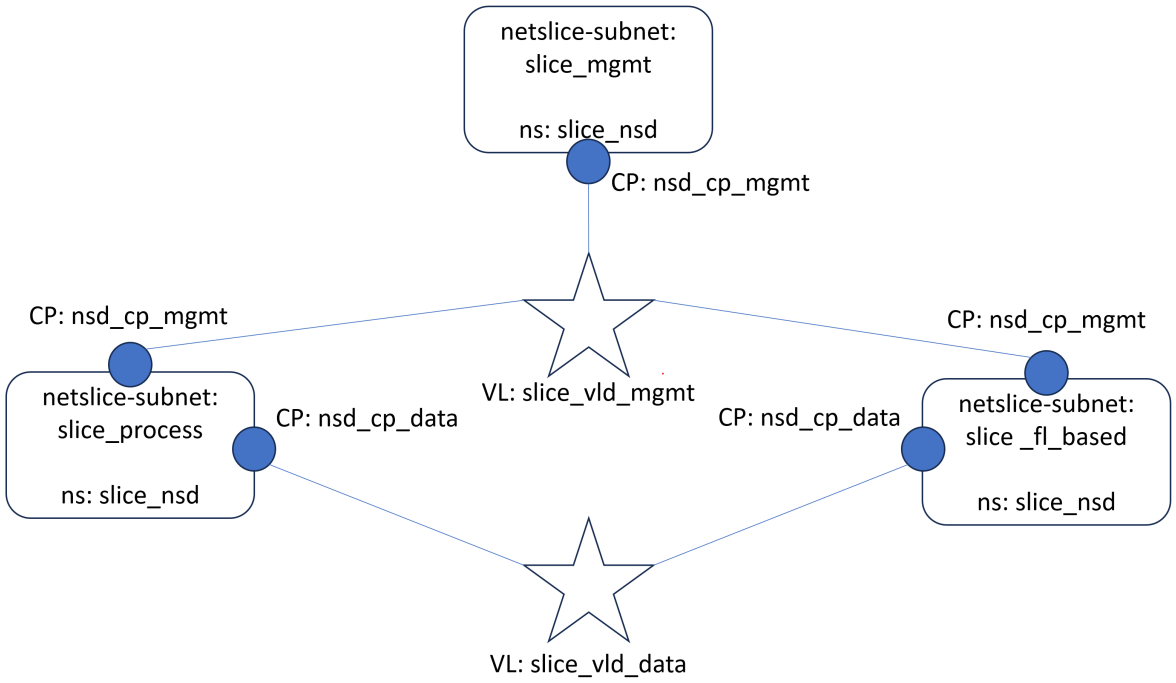

Figure 4.2: Network Slicing Template (NST) Diagram [50]

In this setup, the process slice is responsible for handling data from field-level devices such as sensors and actuators. The federated learning-based slice is focused on gathering system health data and monitoring information. The structure defined in the NST is converted into a YAML specification, from which the associated VNFDs are created, validated, and uploaded

through the OSM API. The same process applies to the NSDs. If endpoint instances must be deployed alongside the network services, an additional NSD with a single connection point is included to establish proper routing.

Once validated, these descriptors are instantiated using either the OSM graphical user interface or command-line interface. The resulting deployment on Google Cloud reflects the intended slicing configuration. The management slice connects to the control slices using router-like instances, and each slice includes two endpoint virtual machines to demonstrate intra-slice communication. Additionally, a virtual router in the management slice connects to the external subnet, providing internet access through floating IP addresses.

4.2.3 Quality of Service (QoS) in Slices

Quality of Service (QoS) is a key component of maintaining reliable network performance, particularly in industrial environments where predictable behavior is essential. Although Google Cloud does not offer direct QoS parameter control like some traditional systems, such as OpenStack, performance can still be managed effectively through network tiering and service configuration options.

For example, selecting premium networking tiers on GCP helps ensure high availability and low-latency communication, which is essential for applications requiring real-time responsiveness. Load balancing, especially at layer 4 for TCP traffic, is also used to distribute workload evenly across resources. This can be configured using GCP's command-line interface (*gcloud*), enabling dynamic response to changes in traffic patterns and service demands.

In this implementation, performance objectives are achieved by aligning network slice configurations with GCP's underlying service models. While these methods do not involve explicit QoS policy definitions, they replicate similar outcomes through careful service placement and resource allocation.

4.3 Experimental Results and Discussion

To assess the performance of the proposed network slicing orchestrator and the deployed network segments, a series of experiments were conducted. These experiments focused on

two categories: infrastructure performance and federated learning model evaluation. The NASA Bearings dataset [52] was selected due to its relevance in research on abnormality detection and its classification as an industrial control system (ICS) dataset. The evaluation metrics used include: i) throughput, ii) deployment time, iii) dataset transmission time, iv) accuracy, v) loss, and vi) F1-score.

4.3.1 Infrastructure Performance Results

The throughput measurements obtained from the infrastructure are illustrated in Figure 4.3. These values were recorded using different TCP window sizes of 128, 256, 384, and 512 KB. As the window size increased, so did the throughput, which ranged within the scale of gigabits per second (Gb/s). This trend demonstrates the system's improved efficiency in transmitting larger blocks of data, a vital feature for operational technology (OT) environments that depend on consistent, real-time data flow.

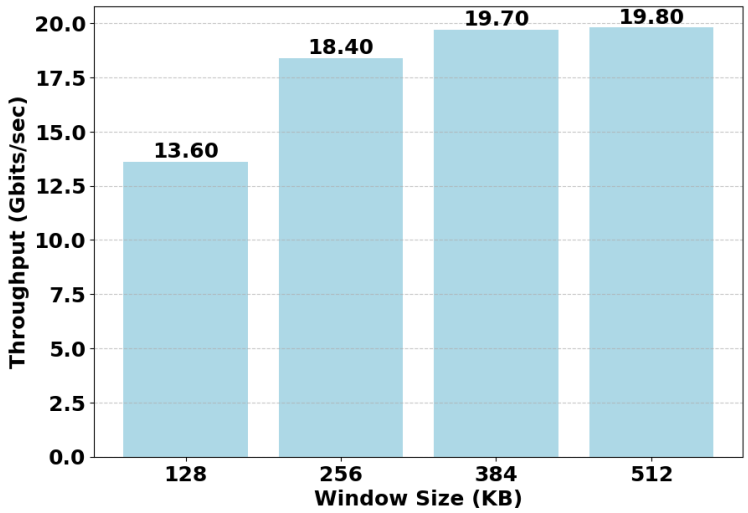

Figure 4.3: Throughput for infrastructure [50]

Deployment times for the infrastructure, presented in Figure 4.4, remained under 3.5 seconds across scenarios involving one, two, and three Programmable Logic Controllers (PLCs). These measurements were taken from the moment container initialization began to the point at which transaction 13 was completed, indicating that registers and coils were fully configured and ready for use. CPU usage during deployment was minimal. Both user-space and

kernel processing times stayed below 0.5 milliseconds for all cases. This reflects the efficiency of Modbus communication, which is typically constrained by input/output operations rather than computational processing.

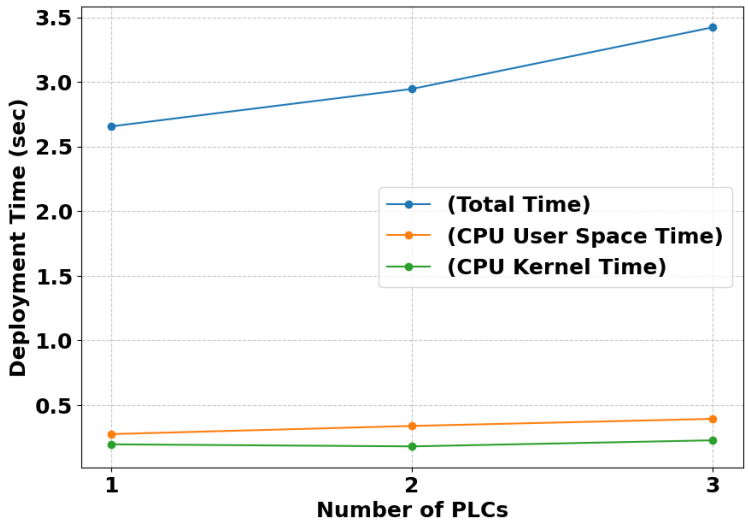

Figure 4.4: Scenario deployment times [50]

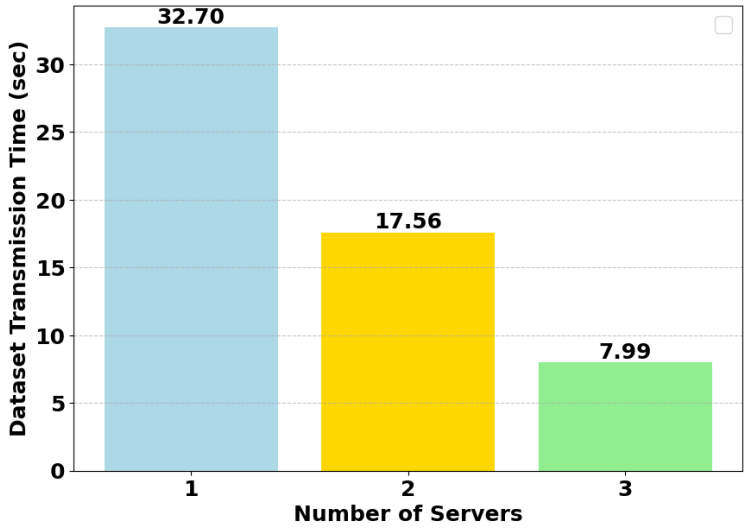

Figure 4.5: Dataset transmission times [50]

Figure 4.5 presents dataset transmission times for one, two, and three server scenarios. When using a single server, the full dataset was transmitted in 32.7 seconds. With three Modbus servers operating concurrently, this time was reduced to 7.99 seconds. These results

highlight the benefits of using distributed and parallelized systems, as they significantly reduce transmission times and maximize network resource utilization. This efficiency is crucial in environments requiring timely and dependable communication.

4.3.2 Federated Learning Model Evaluation

The anomaly detection model used in this research is based on a neural network architecture comprising four dense layers. The first layer utilizes the Exponential Linear Unit (ELU) activation function, which helps capture complex, non-linear patterns and addresses vanishing gradient issues. A latent representation layer and a reconstruction layer follow, allowing the network to learn and recreate data distributions. The final layer restores the input dimensions, and reconstruction error is measured using mean squared error (MSE) loss. This design is especially suited for anomaly detection, as it identifies deviations from established patterns with high accuracy.

A key feature of the model is its dynamic threshold mechanism, which adjusts detection criteria based on the data distribution observed at each federated node. This allows the system to adapt to local variations and maintain consistent performance across different environments. Combined with federated learning, this flexibility supports robust and scalable anomaly detection in distributed systems.

The data processing workflow incorporates Min-Max normalization and Principal Component Analysis (PCA) to standardize and reduce dimensionality, ensuring that computational costs remain low without compromising accuracy. The dataset was divided into 100 partitions to simulate a federated environment, reflecting real-world distributed processing. The federated learning process is managed using the Flower framework [35], which coordinates the aggregation of updates from multiple nodes and ensures consistent model evolution.

The accuracy performance of the federated model, shown in Figure 4.6, remains within a stable range between 0.91 and 0.96 over 10 training rounds. In comparison, a baseline model [37] displayed greater variability, with accuracy values ranging from 0.90 to 0.98. The consistency seen in the proposed model is largely attributed to the dynamic threshold mechanism, which helps maintain performance despite fluctuations in data characteristics across nodes.

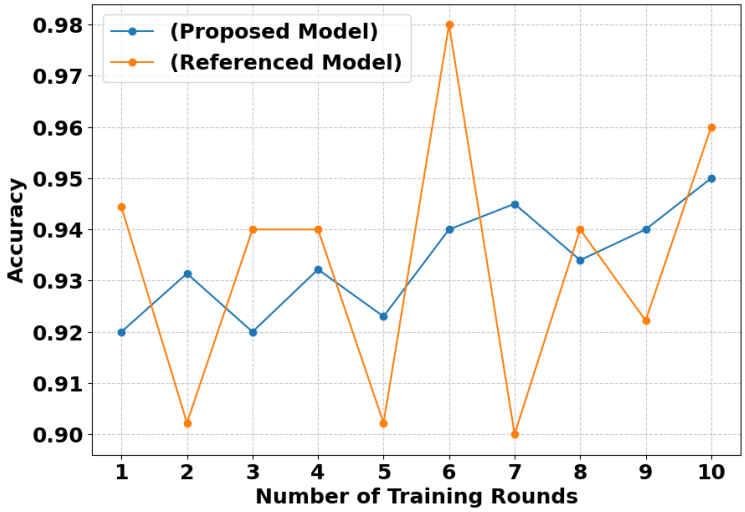

Figure 4.6: Accuracy for FL models [50]

Loss values, depicted in Figure 4.7, further validate the model's robustness. The proposed model maintains consistently lower and more stable loss levels, while the baseline model shows noticeable spikes in certain rounds. These peaks suggest instability in the learning process, likely caused by the baseline model's use of fixed thresholds. In contrast, the dynamic nature of the proposed model allows it to accommodate differences in data distribution, thus improving stability.

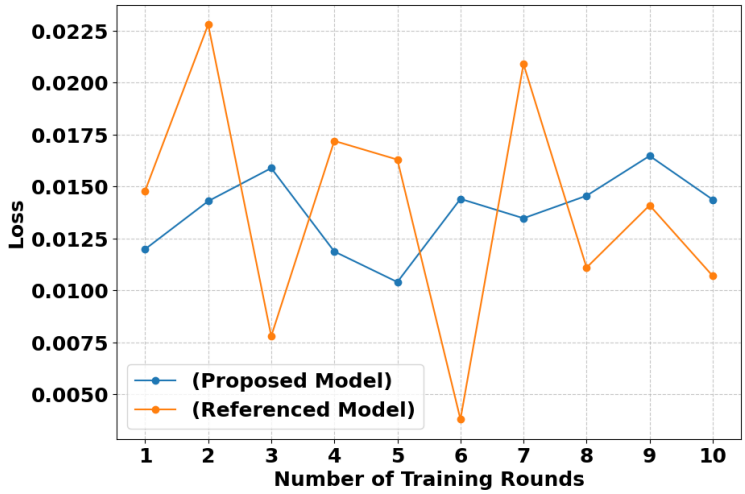

Figure 4.7: Loss for FL models [50]

F1-score results, illustrated in Figure 4.8, reinforce the model's effectiveness. The proposed approach consistently achieves F1-scores between 0.94 and 0.96, indicating strong precision and recall. Meanwhile, the baseline model experiences wider fluctuations, with F1-scores dropping as low as 0.88 in some rounds. The stability of the proposed model reflects its ability to detect anomalies accurately and reliably, even in varied distributed learning scenarios.

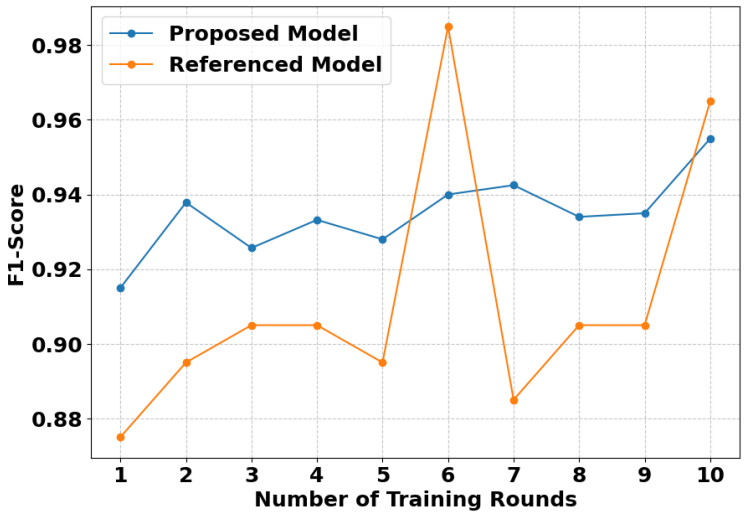

Figure 4.8: F1-Score for FL models [50]

The combination of the proposed model and the Flower framework [35] provides a solid foundation for distributed anomaly detection. By using dynamic thresholds and coordinated updates, the system ensures high performance and resilience throughout the training process.

These findings serve as a baseline for future expansion of this system to more complex scenarios. Reliable data transmission and anomaly detection are essential in ICS environments where high performance, availability, and security must be maintained consistently.

4.4 Summary

This chapter explored the integration of dynamic network slicing with federated learning (FL) in industrial control system (ICS) environments, addressing the challenge of optimizing network resource usage while supporting decentralized machine learning. We presented

a dual-slice architecture that separates process-control functions from FL-based monitoring, providing predictable isolation and robust resource allocation across slices. The chapter detailed the orchestration workflow using Open Source MANO and Google Cloud Platform, along with the integration of a dynamic DMZ and quality-of-service mechanisms to ensure secure, real-time operations. Through a series of experiments, we evaluated both the infrastructure performance and the anomaly detection capabilities of the federated models, demonstrating stable throughput, low deployment times, reduced dataset transmission times, and superior accuracy, loss, and F1-score performance compared to baseline models. These results provide a strong foundation for deploying scalable, privacy-preserving, and intelligent monitoring systems in OT networks, advancing the cyber resilience and operational intelligence of Industry 4.0 environments.

Chapter 5

Dynamic DMZ and Federated Learning in OT Environment

The third research question investigates how slicing can improve the resiliency of OT networks through the orchestration of a dynamic DMZ. Traditional industrial demilitarized zones (DMZs) are typically static and manually configured, offering limited adaptability in the face of evolving cyber threats or shifting process demands. As industrial environments become increasingly interconnected and exposed to external networks, the need for real-time, flexible defenses has grown critical. By applying dynamic network slicing to the DMZ, it becomes possible to isolate suspicious traffic, deploy virtual security services on demand, and dynamically adjust containment boundaries without disrupting time-sensitive industrial operations. Integrating federated learning further enhances this model by enabling local anomaly detection and adaptive response, strengthening both the preventive and corrective layers of OT cybersecurity.

This chapter contributes to answering this research question by presenting a DMZ-as-a-service architecture that integrates network slicing, federated learning, and cloud-native orchestration tools into a unified security framework. It details the design and cooperation of the process, control, monitoring, and DMZ slices, the cloud-based enforcement of real-time security policies, and the use of federated analytics for local anomaly detection. The chapter also describes the orchestration workflow that leverages Open Source MANO, Google Cloud Platform, and Kubernetes to automate resource provisioning and container management. Through extensive experimental evaluation on live datasets, the chapter demonstrates the framework's ability to meet the stringent security, resilience, scalability, and performance requirements of modern ICS networks.

5.1 Dynamic DMZ and FL Network Slicing

Applying dynamic network slicing to conventional ICS deployments offers a practical way to improve security, boost efficiency, and add flexibility across diverse industrial settings. By dividing a single physical network into isolated virtual segments, each tuned to its own traffic profile, the technique lowers cyber-security risk, sustains data availability, and protects critical services and processes.

By relocating the important OT services to the cloud, we can potentially increase security and resiliency in ICS. DMZs have mostly been a critical component of OT infrastructure which will help us in effective verification of the network traffic to/from the enterprise network. However, establishing DMZ is a costly and time-consuming process. Therefore, in this work, we propose establishing DMZ-as-a-Service layer by leveraging the network slicing approach, which advances our prior work [50] to incorporate flexible management of network services. This architecture also includes a distributed federated learning service layer with edge workers processing local industrial data, using privacy-preserving techniques to detect anomalies without transmitting sensitive data. The models are continuously updated to a central federated learning aggregator in the cloud.

The security foundation of this architecture is based on a dynamic cloud-based DMZ, which manages real-time security posture by automatically generating adaptive policies. Fault detection and handling are done natively by Kubernetes, as KNFs are utilized for efficient container management.

The architecture is coordinated by an orchestrator that dynamically allocates cloud resources for the upper-layer slices, enabling rapid deployment of virtualized security functions and automatic failover for fault tolerance. A SCADA platform then provides supervisory control, interfacing with programmable logic controllers (PLCs) in the process slice, while all inter-slice traffic is funneled through the dynamic DMZ layer for secure, real-time process control. Below these slices lies the field layer of raw sensors and actuators, which operates outside the slice manager and orchestrator's authority to preserve strict timing and determinism at the physical level.

This arrangement satisfies the high demands of modern industrial systems such as smart

grids, delivering strong security, operational efficiency, privacy-aware analytics, and the scalability and fault-tolerant behavior expected in Industry 4.0.

Figure 5.1 shows the resulting slice structure, where each slice has a distinct role within the ICS framework.

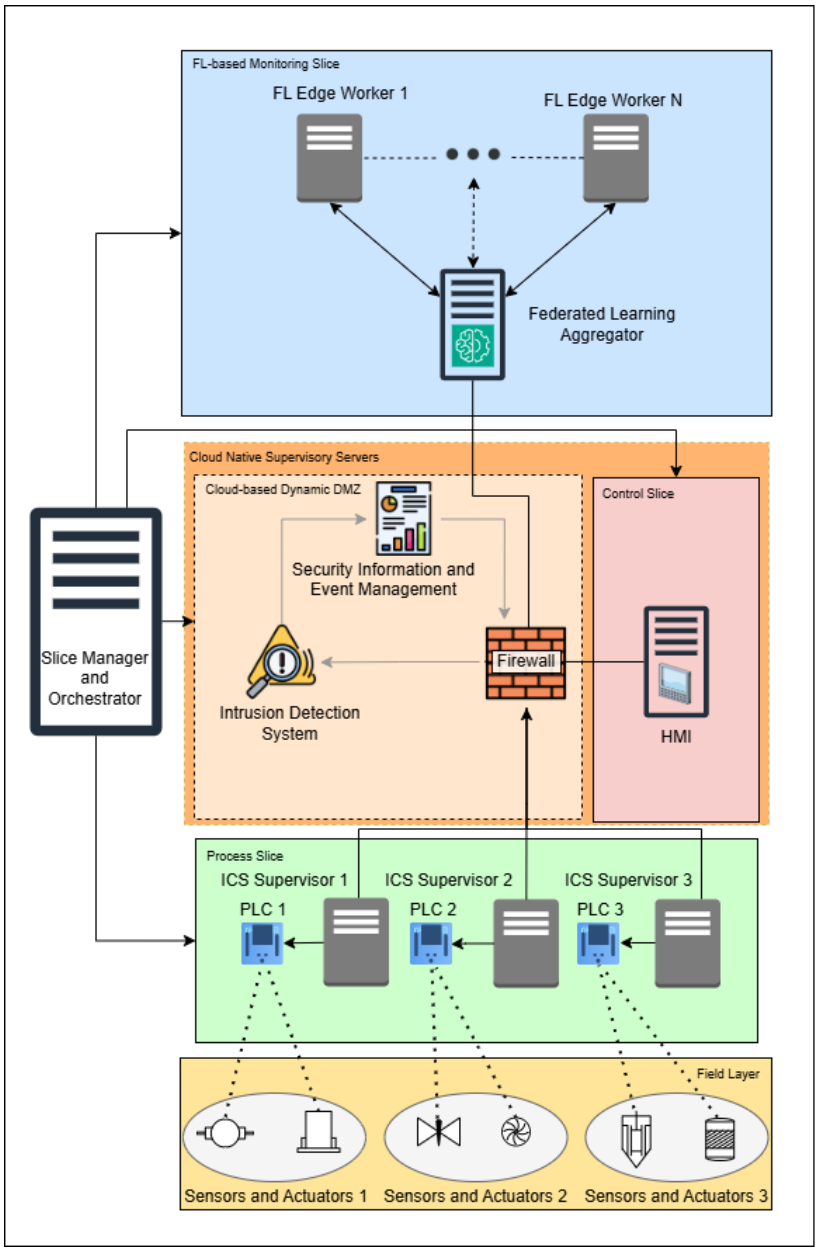

Figure 5.1: Architecture of Network Slice-enabled Dynamic DMZ in OT [53]

- *Process Slice:* This segment integrates ICS Supervisors, which are pivotal in collecting the data and command handling from the Programmable Logic Controllers to execute

operational commands to control physical processes in real time. Lastly, the ICS Supervisors forward data to the control slice, ensuring seamless process management and control.

- *FL-based Monitoring Slice:* This segment incorporates an FL Aggregator with Edge Worker 1 to N, where the number of edge workers can scale as required, depending on the capacity of the host machine. These edge workers are responsible for local data processing and federated learning model updates. The FL Aggregator serves as the central hub for aggregating the data fragments from edge workers, detecting anomalies, and improving overall model performance.

- *Control Slice:* The SCADA control functions are handled by ScadaBR, named after Brazil [51], which serves as the Human-Machine Interface (HMI), monitoring data transfers and overseeing the data moving through the ICS environment. The system maintains real-time operational requirements while securely interfacing with the PLCs, which directly control the physical processes.

- *Cloud-based Dynamic DMZ Slice:* The dynamic DMZ is implemented using OPNsense for firewall management, Suricata for intrusion detection, and Splunk for security information and event management (SIEM). This slice enables real-time monitoring, dynamic rule creation, and rapid response to security incidents, significantly enhancing the network's resilience against cyber threats.

Each slice operates independently, improving robustness and scalability. Because every layer can scale or fail in isolation, the design limits single points of failure, simplifies resource management, and ultimately strengthens the security and efficiency of industrial control networks.

5.2 Approach to Orchestrating Network Slices

This section outlines the implementation workflow for a purpose-driven network-slicing orchestrator designed for industrial control systems. Open Source MANO (OSM) [20] acts as the central coordinator, managing network services and Kubernetes Network Functions (KNFs) across the ICS environment. OSM enforces predefined policies, automates the full lifecycle of each function, from deployment to decommissioning, and rapidly adapts resources when operating conditions change, thereby improving efficiency and flexibility.

Traditional virtual network functions (VNFs), as used in previous works [49] and [50], are replaced by KNFs, described with Kubernetes Network Function Descriptors (KNFDs) instead of VNFDs. By using Kubernetes natively, KNFs gain streamlined scaling, simpler management, and higher fault tolerance in containerized deployments. A KNFD specifies both the namespace and the K3s cluster running on a Google Cloud virtual machine, simplifying orchestration and boosting scalability. Google Cloud Platform (GCP) serves as the Virtual Infrastructure Manager (VIM), allocating compute, storage, and networking resources to create a flexible foundation for ICS workloads. Prometheus collects real-time metrics, and Grafana visualizes them in an intuitive dashboard, helping operators detect anomalies quickly and maintain smooth system operation.

After connecting the VIM to the orchestrator, the network-slice topology is defined. The process starts with a Network Slice Template (NST), shown in Figure 5.2, where the supervisory slice (slice_mgmt) sits at the top. Four functional slices: slice_process, slice_control, slice_dmz, and slice_fl_based—link through virtual links and connection points defined in the Network Service Descriptor (NSD). One slice processes sensor and actuator data, whereas another monitors device health information.

NST components are converted into a YAML-based KNFD. Once completed and validated, the KNFD is uploaded via an API call. The NSD follows the same procedure, with any endpoints that must launch alongside the network services captured in a separate NSD featuring a single connection point.

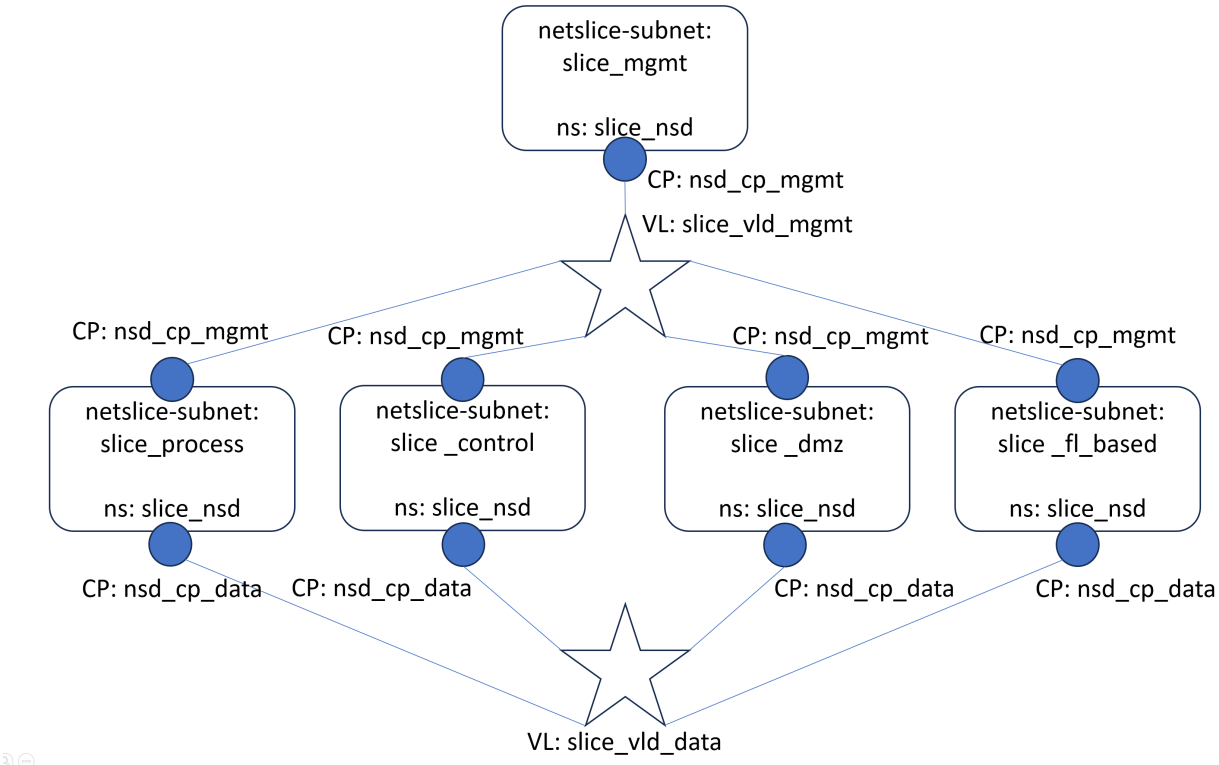

Figure 5.2: Network Slicing Template (NST) Diagram [53]

5.3 Experimental Results and Discussion

To assess the performance of the proposed dynamic DMZ and federated learning–enabled environment created with network slicing, we carried out experiments in three areas: network information capture, supervised federated learning evaluation, and unsupervised federated learning evaluation. Two datasets were used for the first two areas. The first dataset is the NASA Bearings dataset [27], widely employed for anomaly-detection research and classified as an ICS dataset.

The second dataset originates from the traffic exchanged between the supervisors and ScadaBR acting as the HMI. It contains more than three hundred thousand packets that span over four hours of continuous capture. In the third area, we reuse this capture together with the Modbus Critical Information Infrastructures Security (CRITIS) dataset [54], which was recorded in a similar PLC-and-HMI architecture. The metrics examined were: i) inter-packet interval, ii) segment length, iii) round-trip time (RTT), iv) accuracy, v) loss, vi) F1-score,

vii) average reconstruction loss, and viii) network-service rollback time.

5.3.1 Network Information Capturing Results

i) Figure 5.3 presents the inter-packet intervals measured from the captured traffic. Pronounced peaks represent long delays between consecutive packets, pointing to possible network problems or anomalies. During a sending-rate attack that reduces the interval from 100 ms to 20 ms, the network becomes saturated, leading to congestion and packet loss. The firewall drops some packets, which produces the observed spikes and disrupts timely data delivery.

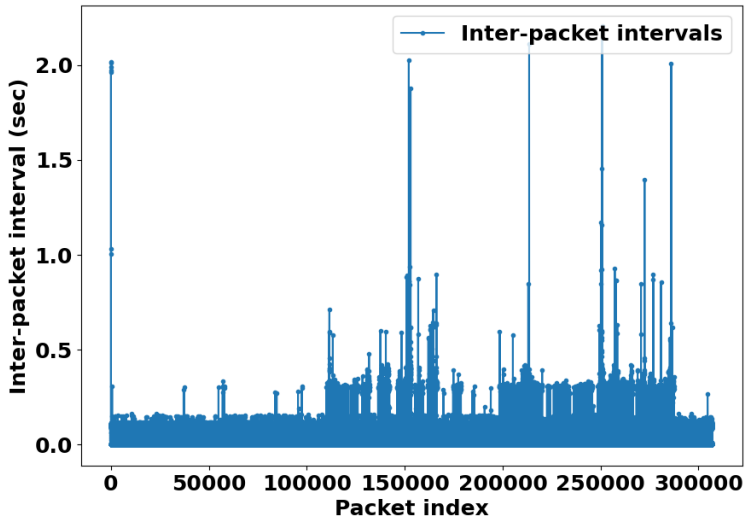

Figure 5.3: Inter-packet Interval [53]

ii) Segment-length variations appear in Figure 5.4. Significant swings in packet size signal abnormal behavior or errors in transmission. After the sending-rate attack, the network's inability to handle the intensified load yields irregular packet sizes, misconfigurations, and drops, which in turn create erratic segment-length patterns and degrade performance.

iii) Figure 5.5 illustrates RTT over the session. Spikes indicate delays in packet return paths, hinting at bottlenecks, routing problems, or faults. Post-attack congestion raises RTT because packets are delayed or discarded, and the firewall's limited throughput worsens these delays, undermining the reliability of the ICS network.

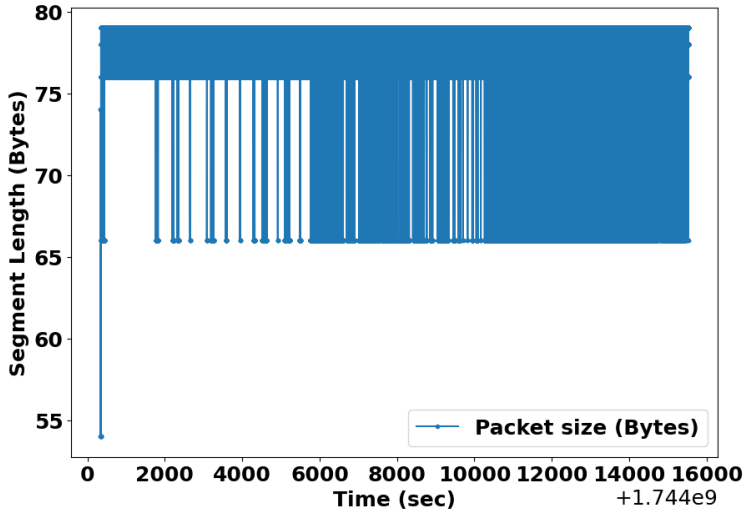

Figure 5.4: Segment Length [53]

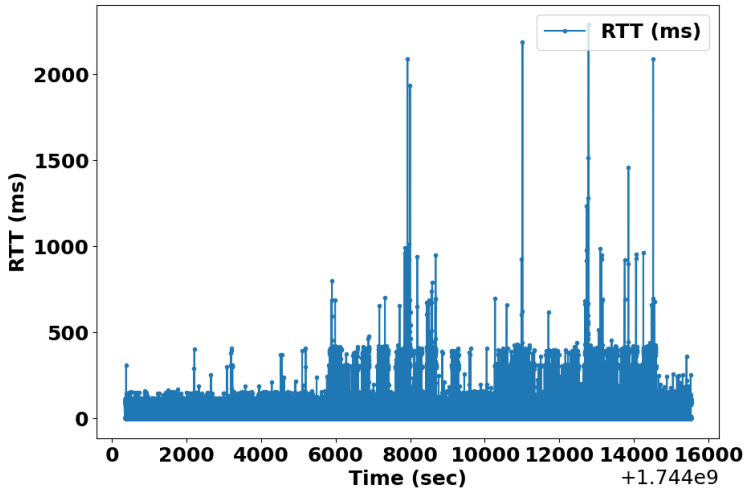

Figure 5.5: Round-trip Time (RTT) [53]

5.3.2 FL Supervised Model Evaluation Findings

The supervised anomaly-detection model employs a four-layer neural network. ELU activation in the first layer captures non-linear relationships, while a sigmoid function in the final layer outputs binary probabilities. Binary-cross-entropy loss guides optimization and a dynamic threshold adapts detection criteria to varying data distributions in federated settings. Data preprocessing applies Min-Max normalization and Principal Component Analysis (PCA) to reduce dimensionality. The data is divided into five partitions to emulate federated

nodes, each containing labeled CSV rows with network characteristics, packet indices, and anomaly markers. The Flower Framework aggregates node updates [35].

iv) Figure 5.6 shows accuracy trends for models trained on NASA Bearings anomalies (component level) and PLC-HMI traffic during a sending-rate attack (network level). Network anomaly detection achieves lower accuracy because it must consider more parameters, yet both curves rise steadily across rounds, proving consistent learning progress.

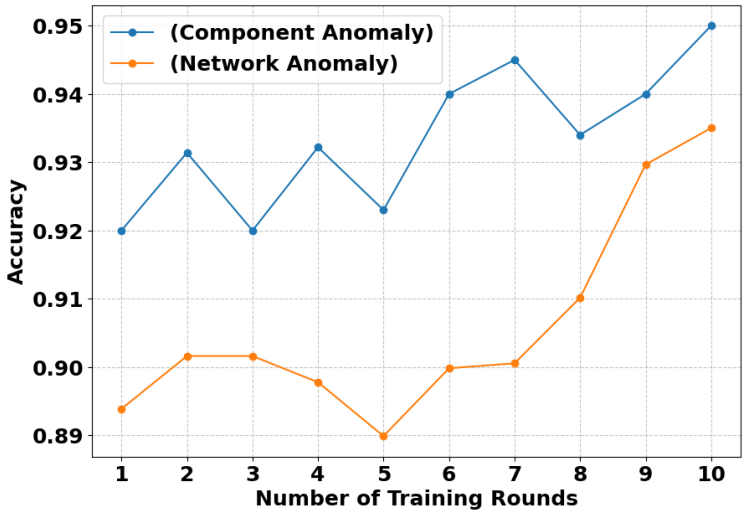

Figure 5.6: Accuracy for FL Supervised Models [53]

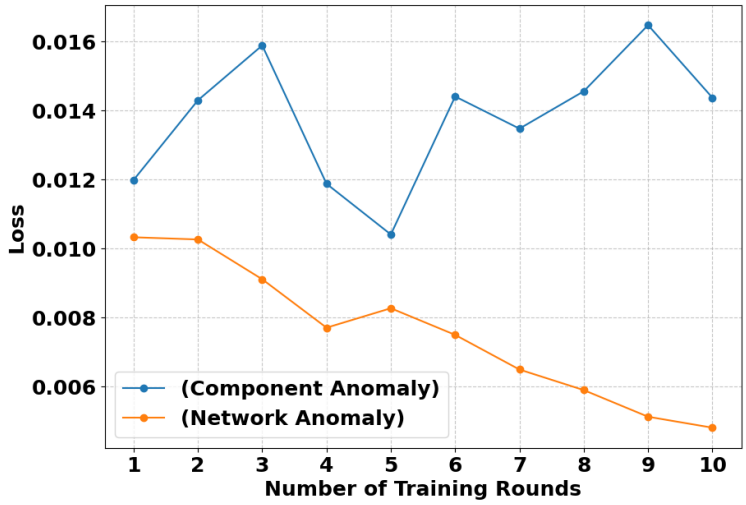

Figure 5.7: Loss for FL Supervised Models [53]

v) Loss patterns in Figure 5.7 reveal that the network anomaly model's loss falls smoothly

after several rounds, reflecting its broader parameter set and the adaptive threshold. The component-anomaly model exhibits sharper fluctuations because it learns from fewer features. Overall, dynamic thresholds promote stable convergence.

vi) F1-scores in Figure 5.8 follow an improvement path tied closely to the loss curves. The network anomaly model shows a brief dip before advancing steadily, whereas the component anomaly model climbs more quickly. Both ultimately achieve higher F1-scores, underscoring reliable detection even with distributed data.

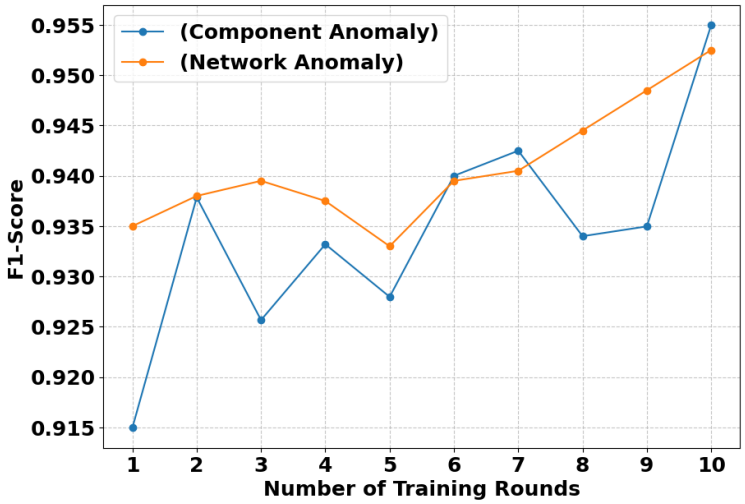

Figure 5.8: F1-Score for FL Supervised Models [53]

5.3.3 FL Unsupervised Model Evaluation Findings

vii) Average reconstruction error gauges the unsupervised autoencoder's effectiveness. Although anomalies may be harder to isolate when both normal and abnormal packets exhibit reconstruction errors, Figure 5.9 demonstrates a downward trend that stabilizes halfway for both datasets. This indicates successful learning and generalization despite the absence of labels.

5.3.4 Network Service Rollback Time Findings

viii) Rollback times differ markedly: 43.35 seconds for a VNF managed by Docker Swarm compared with 3.67 seconds for a KNF. VNFs incur extra overhead due to more complex

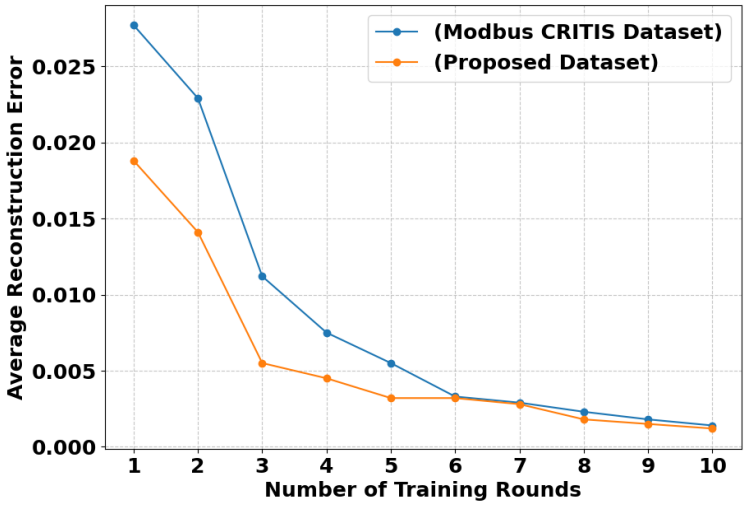

Figure 5.9: Average Reconstruction Error for FL Unsupervised Model [53]

orchestration. KNFs, managed by Kubernetes, benefit from lightweight container handling. When an anomaly such as a sending-rate attack triggers a rollback, the system reverts to a baseline configuration. In parallel, the DMZ applies mitigation actions once indicators of compromise are observed, limiting the impact of potential attacks.

These results showcase the exploration of the integration of FL, KNFs, and network slicing to create a scalable and resilient architecture for ICS networks. The framework demonstrates its capability to perform simultaneous network and component information capturing and FL-based training for anomalous detection in modern industrial environments. By leveraging KNFs in place of traditional NFV, the solution provides enhanced fault tolerance and scalability for ICS environments, ensuring optimized performance even in complex, decentralized scenarios.

5.4 Summary

This chapter introduced a dynamic DMZ-as-a-service architecture that combines network slicing, federated learning, and Kubernetes Network Functions (KNFs) to strengthen the security, resilience, and scalability of industrial control system (ICS) environments. We described how the process, control, monitoring, and DMZ slices work together, and we detailed the orchestration workflow using Open Source MANO, Google Cloud Platform, and

Kubernetes for automated resource management and container handling. The chapter also reported experimental results that evaluated the system's performance in capturing live network traffic, training supervised and unsupervised federated learning models, and measuring rollback times during security incidents. The findings showed consistent accuracy, stable loss and F1-score trends, effective anomaly detection, and fast recovery from disruptions. Overall, this work provides a strong foundation for implementing adaptive, slice-aware security architectures in operational technology networks, improving their ability to counter evolving cyber threats while maintaining reliable operations.

Chapter 6

Securing OT Environments From the Classroom

Building on the architectures and experiments of Chapters 3–5, this chapter presents a hands-on educational framework for training the next generation of OT cybersecurity professionals. By integrating SDN, NFV, network slicing, digital twins, federated learning, and dynamic DMZs, we propose a layered curriculum that spans higher education, in particular two-year colleges and four-year institutions. This framework uses realistic ICS scenarios and remote lab infrastructure to give students authentic research experiences, bridge the industry-academic gap, and satisfy Centers of Academic Excellence (CAE) and National Initiative for Cybersecurity Education (NICE) competencies.

6.1 Hands-On Lab Infrastructure

To support experiential learning, institutions can provide remote access to:

- **Slice Manager and Orchestrator Server:** A dedicated instance running an orchestrator, such as Open Source MANO (OSM) [20], with the recommended CPU, memory, and storage requirements. This server can be hosted locally, on a private cloud, or in a public cloud environment. Students use it to onboard VNFs or KNFs, define Network Slice Templates, and control the lifecycle of each slice.

- **Virtual Infrastructure Manager (VIM):** A pool of compute and storage resources (either in a public cloud or on an on-premises server) managed by OpenStack. This layer provides the virtual machines, networks, and volumes on which slices and their components (VNFs or KNFs, VMs, containers) are instantiated.

- **Security Lab Interface:** A lightweight client (any institution-provided desktop or laptop) that connects to both the orchestrator server and the VIM. Through this interface, learners launch attack scenarios, apply DMZ policies, and monitor slice behavior using tools such as Splunk or Grafana. This setup ensures reproducibility on standard hardware.

Using this infrastructure, capstone projects can integrate:

- **Digital Twin Frameworks:** As an open-source starting point, MiniCPS can be used for simulating field, control, and supervisory layers [55]. Advanced courses can incorporate pyModbus, OpenPLC, and ScadaBR along with their dependencies to enable fine-grained modeling and customization of digital-twin components.
- **Dynamic DMZ Slices:** Containerized NFV-based firewalls, intrusion detection systems, and SIEM that students deploy and update in real-time to enforce adaptive security policies.
- **Federated-Learning Slices:** Edge-worker containers that train local anomaly detection models and share only parameter updates with a central aggregator to preserve data privacy while demonstrating distributed analytics.

6.2 Framework and Competency Alignment

To ensure that graduates possess the skills and knowledge required for OT cybersecurity roles, our curriculum is grounded in three complementary competency frameworks: the NICE Workforce Framework [56], the CAE Cyber Operations knowledge units along with the Essential Elements (ABCDE) Model [57], and the evidence-based computing-education competency models [58-61]. By weaving together these frameworks, we establish clear learning outcomes, align instructional content with industry needs, and provide transparent assessment criteria.

6.2.1 NICE Workforce Framework Foundations

The NICE Workforce Framework organizes cybersecurity work into seven high-level categories; such as protect and defend, analyze, and securely provision; and over fifty defined work roles, each associated with specific tasks and knowledge areas [56]. For OT security, key categories include:

- **Protect and Defend (PR):** Tasks such as implementing firewalls and intrusion detection, corresponding to CAE knowledge unit SEC-FW ("Network Security and Defense"), e.g., see [57].
- **Analyze (AN):** Performing traffic analysis and anomaly detection, aligned with the AN-CYBER analytics tasks and anomaly-detection competencies.
- **Securely Provision (SP):** Designing and deploying secure network slices, mapping to VNF/KNF provisioning skills.

Each course module targets specific NICE tasks (e.g., PR-IPPD-133: "Implement security policy changes"), ensuring that students develop work-ready capabilities.

6.2.2 Cyber Operations Knowledge Units and ABCDE Model

The Centers of Academic Excellence (CAE) Cyber Operations knowledge units define the minimal subject-matter coverage for CAE-designated programs [57]. To translate these units into measurable outcomes, we use the ABCDE competency model:

- **Actor:** The learner.
- **Behavior:** The task (for example, "configure an adaptive DMZ policy").
- **Context:** The setting or environment (such as a simulated ICS network).
- **Degree:** The level of performance (for instance, "with fewer than 5% dropped packets").
- **Employment:** Professional dispositions (e.g., ethical judgment, teamwork).

A sample competency statement reads: *"As a graduate (Actor), configure and deploy a DMZ policy in a simulated ICS network (Behavior, Context) that blocks unauthorized SCADA commands with 95% accuracy (Degree), while demonstrating professional communication with team members (Employment)."*

By aligning each lab and assessment to ABCDE statements, we ensure coverage of core CAE knowledge units such as "Network Monitoring and Defense" and "Industrial Control System Security."

6.2.3 Computing-Education Competency Models

Research in computing education emphasizes three dimensions of competency: Knowledge, Skills, and Professional Disposition [58–61].

- **Knowledge:** Understanding SDN architectures, NFV concepts, and the Purdue ICS model as outlined in Computing Curricula 2020 [58] and CS2023 [61].
- **Skills:** Hands-on abilities such as writing OpenFlow rules, scripting KNF deployments in Kubernetes, and performing federated-learning experiments, as advocated in the ITiCSE 2018 Companion [59].
- **Professional Disposition:** Critical thinking, ethical reasoning, and collaborative teamwork, modeled in the CKC approach [60].

These dimensions inform our lab design, ensuring that students not only learn technical content but also develop the dispositions required for professional practice.

6.2.4 Three-Tier Curriculum Structure

Our curriculum is delivered in three progressive tiers, each mapped to the frameworks above:

1. **Foundations of OT Networking:**
    - SDN and OpenFlow programming labs map to NICE SP-NET-120 ("Configure network devices") and CAE unit "Network Architecture", see [57].

- NFV and KNF modules align with NICE SP-NET-122 ("Deploy virtual network functions") and CKC "Infrastructure" knowledge, see [60].
- MiniCPS-based digital twin simulations support CAE unit "ICS Simulation" and reinforce Purdue Model concepts from Computing Curricula 2020, see [58].

2. **Advanced Slicing and Security:**

   - Network slicing and QoS design map to NICE SP-NET-123 ("Manage network resources") and CAE "Quality of Service" unit.
   - Dynamic DMZ policy creation and SIEM integration labs address PR-IPPD-135 ("Implement intrusion detection measures") and CKC "Security Operations" skills, see [59].
   - Federated-learning experiments align with AN-CYBER-222 ("Perform anomaly detection analysis") and CAE "Machine Learning in Cyber Defense."

3. **Research Practicum on Remote Platforms:**

   - Slice deployment on cloud and network research nodes maps to SP-NET-124 ("Operate secure network services") and the ACM reproducibility guidelines, see [61].
   - Reproducible security experiments via sandbox portals fulfill CAE "Reproducible Experimentation" and NICE IN-DEV-128 ("Conduct research experiments").
   - Publishing findings prepare students for professional communication competencies under CKC "Professional Practice," see [60].

6.2.5 Assessment and Competency Validation

Each lab includes formative and summative assessments tied directly to ABCDE statements, e.g., [57] and NICE task performance metrics [62]. For example, configuring an SDN rule is assessed on correctness (Degree), context adherence (Context), and collaborative debugging (Employment). By triangulating across the three frameworks, we ensure that graduates can design, implement, and evaluate secure OT networks under real-world constraints.

6.3 Two-Year and Four-Year Partnership Model

In collaboration with El Paso Community College (EPCC) and the University of Texas at El Paso (UTEP), research experiences are embedded in the years of academic learning, supported by CAHSI and Google's Institutional Research Program (IRP). Community college students paired with university peers, gaining early exposure to:

- Designing and deploying digital twins of ICS networks
- Implementing fault-tolerant VNFs and KNFs with automated failover
- Developing self-healing policies and monitoring via Prometheus/Grafana
- Analyzing federated-learning results and refining DMZ rules

This structured approach builds technical depth and confidence, smoothing the transition to Associate and Bachelor's level research and reducing the time-to-competency gap.

Chapter 7

Concluding Remarks

7.1 Significance of the Results

This thesis demonstrates how the integration of network slicing, Software-Defined Networking (SDN), and Network Function Virtualization (NFV) improves the performance, scalability, and security of Operational Technology (OT) networks. The results are presented in alignment with the three guiding research questions, showing how the proposed architecture addresses core limitations in traditional industrial systems.

7.1.1 Impact of Network Slicing on OT Environments

Research Question 1: *How will network slicing affect an OT environment?*

The results show that network slicing introduces an effective method for isolating and managing diverse traffic flows within an OT network. Each slice can be configured with specific performance requirements, such as bandwidth, latency, and reliability, which helps maintain the predictability required by industrial applications. In experimental evaluations, latency-sensitive applications such as real-time sensor telemetry and emergency control signaling experienced lower delay and jitter within dedicated slices.

Additionally, the implementation of network slicing improved system observability and operational clarity. Network administrators were able to monitor traffic and performance metrics independently for each slice, making it easier to detect anomalies and diagnose problems. For instance, a slice dedicated to critical infrastructure functions maintained consistent communication quality even during periods of high network load. These findings indicate that network slicing enhances both operational stability and system manageability in OT environments.

7.1.2 Dynamic Slicing for Resource Optimization and Federated Learning Support

Research Question 2: *How can dynamic network slicing be implemented to optimize network resource usage and support decentralized machine learning?*

The integration of dynamic slicing with FL revealed that the architecture can adaptively allocate resources based on real-time workload demands. FL tasks, which often involve periodic bursts of data transfer and computation, were successfully isolated in a dedicated slice that could scale in response to traffic intensity. This approach ensured that other OT functions operating in parallel were not disrupted by bandwidth competition or increased latency.

Dynamic slicing enabled more efficient use of network and computing resources. For example, when multiple edge devices submitted model updates, the system temporarily increased the bandwidth of the FL slice and then scaled it down once the transmission was completed. This behavior allowed the network to maintain overall efficiency without overprovisioning. The use of containers and NFV also allowed FL services to be deployed closer to edge devices, improving latency and data privacy. These results show that dynamic slicing not only supports decentralized machine learning but also optimizes resource use in environments with variable workloads.

7.1.3 Enhancing Resiliency Through Orchestrated Dynamic DMZs

Research Question 3: *In what ways can slicing improve the resiliency of OT networks through the orchestration of a dynamic DMZ?*

This thesis introduces a dynamic DMZ framework that uses NFV and SDN to enhance security responsiveness and system resiliency. Unlike traditional static DMZs that require fixed hardware and manual configuration, the proposed solution can deploy virtual security functions (VSFs) such as intrusion detection systems and deep packet inspection engines on demand. These VSFs operate within a dedicated network slice and are managed by the SDN controller based on predefined security policies and real-time monitoring.

The evaluation showed that upon detecting suspicious behavior or traffic anomalies, the

system could rapidly reroute traffic through additional security layers without interrupting other critical operations. For instance, when a simulated threat was introduced, the SDN controller directed the affected flows into an isolated slice containing enhanced inspection tools. This process occurred in real-time, demonstrating the ability to respond quickly to evolving threats. The results confirm that network slicing enables flexible and scalable security zones that contribute to a more resilient OT network.

7.1.4 Summary

The results across all three research questions confirm the feasibility and value of integrating network slicing, SDN, and NFV within OT environments. The proposed architecture enables improved performance for critical processes, more efficient use of shared resources, and dynamic security enforcement. Together, these findings support the development of adaptive and intelligent industrial networks capable of meeting the evolving demands of Industry 4.0.

7.2 Future Work

Building on the results of this thesis, future research will pursue two primary directions: (1) the enhancement of fault tolerance and deployment models in industrial control systems using KNFs and (2) the expansion of the educational framework through integration with national and international testbeds.

7.2.1 Byzantine Fault Tolerance in KNF-Orchestrated Environments

Current Kubernetes orchestration models primarily address crash failures and leader election availability through basic redundancy mechanisms. However, they are not inherently resilient against Byzantine faults, which involve nodes behaving arbitrarily or maliciously. These faults are especially critical in OT environments where misbehaving components can endanger physical systems.

Future iterations of this work will explore the implementation of Byzantine Fault Tolerant

(BFT) consensus algorithms within KNF-based OT architectures. By integrating protocols such as Practical Byzantine Fault Tolerance (PBFT) [63] or more scalable variants like HotStuff [64] or SBFT [65], containerized network functions can maintain operational correctness even when facing compromised components. This would significantly increase the trustworthiness of slice orchestration, especially in security-critical use cases like dynamic DMZ reconfiguration or federated learning coordination.

7.2.2 Advanced KNF Architectures for 5G-Enabled ICS

As ICS adopts 5G for ultra-reliable low-latency communication (URLLC), the role of KNFs becomes more prominent. Future work will investigate the deployment of lightweight KNFs optimized for 5G edge environments. These will include enhanced traffic shapers, application-aware firewalls, and protocol gateways capable of real-time packet inspection and anomaly detection at the network edge.

The research will focus on extending the KNF model with enhanced fault management capabilities, integrating observability tools such as OpenTelemetry and runtime health checks. The goal is to enable predictive fault detection, efficient failover, and rapid self-healing responses across distributed ICS environments. This will help ensure service continuity and safety in mission-critical industrial processes, even during transient faults or cyber disruptions.

7.2.3 Internationalization of the Educational Framework Through Testbed Integration

To scale the educational impact of this research, future work will incorporate policy, such as the Association of Computing Machinery (ACM) policy in cybersecurity, and research testbeds to enable scalable, reproducible, and globally accessible learning environments.

Leveraging National Testbeds for Education

National platforms such as Chameleon [66], FABRIC [67], and SPHERE [68] offer ideal environments for deploying and evaluating the slicing-based architecture proposed in this

thesis. Each platform contributes unique capabilities:

- **Chameleon:** Provides bare-metal servers, SDN-enabled switches, and GPU resources for deploying digital twins, federated learning nodes, and Kubernetes clusters.
- **FABRIC:** Offers high-performance optical networking and edge compute capacity to simulate wide-area ICS environments across multiple regions.
- **SPHERE:** Facilitates security and privacy experimentation with collaborative portals, enabling reproducible attacks, defenses, and educational content sharing.

By mapping components of the educational framework to these platforms, students will be able to participate in remote-access labs involving real industrial hardware, virtualized network functions, and cross-site orchestration scenarios. Capstone teams will work collaboratively across institutions, designing and deploying secure OT architectures in federated environments.

Exploring International Curriculum Alignment

In addition to United States national platforms, the educational framework will seek alignment with international cybersecurity and engineering education standards. Partnerships will be explored with global frameworks such as:

- **European Cybersecurity Skills Framework (ECSF)** [**69**] for aligning security learning outcomes with EU-wide digital competency models.
- **CDIO Initiative (Conceive, Design, Implement, Operate)** [**70**] for project-based engineering education with a strong emphasis on systems thinking.
- **Asia-Pacific Advanced Network (APAN)** [**71**] testbed infrastructure for cross-border research collaborations in OT security.

Integrating these standards will allow the curriculum to support mutual recognition of skills, foster cross-cultural collaboration, and prepare students to operate in multinational industrial environments.

7.2.4 Summary

Future research will address both the technical and educational scalability of the framework. Technically, it will strengthen OT network resilience through Byzantine fault tolerance and optimized KNF deployments for emerging 5G-enabled ICS applications. Educationally, it will expand the hands-on lab infrastructure using testbeds and incorporate international standards to enhance global readiness. These efforts will continue to advance secure, intelligent, and adaptable OT networks, while also shaping the next generation of engineers and researchers.

References

[1] Y. Liu, N. Chen, L. Zhu and L. Zhang, "Development of a Grid Adaptability Evaluation Method for Systems with Renewable Energy Connected to Weakly-Synchronized Sending-End DC Power Grid," *2023 5th Asia Energy and Electrical Engineering Symposium (AEEES)*, Chengdu, China, 2023, pp. 207-211, doi: 10.1109/AEEES56888.2023.10114145.

[2] ANSI/ISA-62443-2-1 (99.02.01)-2009, "Security for Industrial Automation and Control Systems: Establishing an Industrial Automation and Control Systems Security Program," *International Society of Automation*, 2009.

[3] S. Bera, "Availability-Aware VNF Placement for uRLLC Applications in MEC-Enabled 5G Networks," *2023 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Jaipur, India, 2023, pp. 300-305, doi: 10.1109/ANTS59832.2023.10469344.

[4] S. A. Obaidli et al., "IEC 61850 beyond compliance: A case study of modernizing automation systems in transmission power substations in Emirate of Dubai towards smart grid," *2017 Saudi Arabia Smart Grid (SASG)*, Jeddah, Saudi Arabia, 2017, pp. 1-9, doi: 10.1109/SASG.2017.8356501.

[5] G. F. Pittalà, L. Rinieri, A. Al Sadi, G. Davoli, A. Melis, M. Prandini, and W. Cerroni, "Leveraging Data Plane Programmability to enhance service orchestration at the edge: A focus on industrial security," *Computer Networks*, vol. 246, p. 110397, 2024, doi: https://doi.org/10.1016/j.comnet.2024.110397

[6] National Institute of Standards and Technology, "Guide to Industrial Control Systems (ICS) Security," *NIST Special Publication 800-82 Revision 2*, Gaithersburg, MD, USA, 2015, doi:10.6028/NIST.SP.800-82r2.

[7] The MITRE Corporation, "Remote System Information Discovery (T0888)," *MITRE ATT&CK® for Industrial Control Systems*, Arlington, VA, USA, Mar. 2020.

[8] The MITRE Corporation, "Supply Chain Compromise (T0862)," *MITRE ATT&CK® for Industrial Control Systems*, Arlington, VA, USA, Mar. 2020.

[9] A. Erba, R. Taormina, S. Galelli, M. Pogliani, M. Carminati, S. Zanero and N. O. Tippenhauer, "Constrained Concealment Attacks against Reconstruction-based Anomaly Detectors in Industrial Control Systems," *Annual Computer Security Applications Conference (ACSAC)*, Austin, TX, USA, 2020, pp. 1–16, doi:10.1145/3427228.3427660.

[10] E. N. Ylmaz, B. Ciylan, S. Gönen, E. Sindiren and G. Karacayılmaz, "Cyber security in industrial control systems: Analysis of DoS attacks against PLCs and the insider effect," *2018 6th International Istanbul Smart Grids and Cities Congress and Fair (ICSG)*, Istanbul, Turkey, 2018, pp. 81–85, doi:10.1109/SGCF.2018.8408947.

[11] The MITRE Corporation, "Unauthorized Command Message (T0855)," *MITRE ATT&CK® for Industrial Control Systems*, Arlington, VA, USA, Mar. 2020.

[12] G. V. Santangelo, V. G. Colacino and M. Marchetti, "Analysis, prevention and detection of ransomware attacks on Industrial Control Systems," *20th IEEE International Symposium on Network Computing and Applications (NCA)*, Boston, MA, USA, 2021, pp. 1–5, doi:10.1109/NCA53618.2021.9685713.

[13] U. K. Tupakula, V. Varadharajan and K. K. Karmakar, "Software Enabled Security Architecture for Counteracting Attacks in Control Systems," *CoRR*, vol. abs/2006.15272, 2020.

[14] J. Mern, K. Hatch, R. Silva, J. Brush and M. J. Kochenderfer, "Reinforcement Learning for Industrial Control Network Cyber Security Orchestration," arXiv preprint arXiv:2106.05332, Jun. 2021.

[15] S. Azadiabad, F. Khendek and M. Toeroe, "Runtime Availability and Service Continuity of Containerized VNF Instances," *2023 IEEE Latin-American Conference on Communications (LATINCOM)*, Panama City, Panama, 2023, pp. 1-6, doi: 10.1109/LATINCOM59467.2023.10361896.

[16] M. Alsaeedi, M. M. Mohamad, and A. A. Al-Roubaiey, "Toward adaptive and scalable openflow-sdn flow control: A survey," *IEEE Access*, vol. 7, pp. 107 346–107 379, 2019.

[17] D. Gahlawat, S. Suhag, U. Rani and S. Madavi, "Hybrid deep learning model for IT-OT integration in Industry 4.0," *2023 Second International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, Singapore, Singapore, 2023, pp. 1025-1030, doi: 10.1109/SmartTechCon57526.2023.10391501.

[18] M. Sanz Rodrigo, D. Rivera, J. I. Moreno, M. Alvarez Campana, and D. R. López, "Digital twins for 5g networks: A modeling and deployment methodology," *IEEE Access*, vol. 11, pp. 38 112–38 126, 2023.

[19] M. Alsaeedi, M. M. Mohamad, and A. A. Al-Roubaiey, "Toward adaptive and scalable openflow-sdn flow control: A survey," *IEEE Access*, vol. 7, pp. 107 346–107 379, 2019.

[20] R. Kołakowski, L. Tomaszewski and S. Kukliński, "Performance evaluation of the OSM orchestrator," *2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Heraklion, Greece, 2021, pp. 15-20, doi: 10.1109/NFV-SDN53031.2021.9665052.

[21] B. G. Rodiles Delgado, A. Solano, C. Servin, and D. Tosh, "Fostering transparent research collaborations between two- and four-year institutions in industrial cybersecurity education," in *Proc. 25th Annu. Conf. Information Technology Education (SIGITE '24)*, El Paso, TX, USA, 2024, pp. 1–6, doi: 10.1145/3686852.3687074.

[22] OpenStack Foundation, "OpenStack Documentation," 2024. [Online]. Available: https://docs.openstack.org/. [Accessed: May 6, 2025].

[23] L. Leonardi, L. L. Bello and G. Patti, "Exploiting Software-Defined Networking to improve runtime reconfigurability of TSN-based networks," *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, Stuttgart, Germany, 2022, pp. 1-4, doi: 10.1109/ETFA52439.2022.9921723.

[24] P. P. Ray and N. Kumar, "SDS/NFV architectures for edge- cloud oriented iot: A

systematic review," *Computer Commu- nications*, vol. 169, pp. 129–153, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366421000396

[25] J. M. Szatkowski, Y. Li, and L. Du, "Enabling reconfigurable naval scada network through software-defined networking," pp. 214–218, 2022.

[26] A. H. M. Jakaria, M. A. Rahman, and A. Gokhale, "Resiliency-aware deployment of sdn in smart grid scada: A formal synthesis model," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1430–1444, 2021.

[27] V. Varadharajan, U. Tupakula, and K. K. Karmakar, "Techniques for enhancing security in industrial control systems," *ACM Trans. Cyber-Phys. Syst.*, vol. 8, no. 1, jan 2024. [Online]. Available: https://doi.org/10.1145/3630103

[28] A. Sajid, H. Abbas and K. Saleem, "Cloud-Assisted IoT-Based SCADA Systems Security: A Review of the State of the Art and Future Challenges," in *IEEE Access*, vol. 4, pp. 1375-1384, 2016, doi: 10.1109/ACCESS.2016.2549047.

[29] H. -C. Jang and S. -Y. Luo, "Enhancing Node Fault Tolerance through High-Availability Clusters in Kubernetes," *2023 IEEE 3rd International Conference on Electronic Communications, Internet of Things and Big Data (ICEIB)*, Taichung, Taiwan, 2023, pp. 30-35, doi: 10.1109/ICEIB57887.2023.10170110.

[30] A. Mikhail, I. A. Kamil and H. Mahajan, "Increasing SCADA System Availability by Fault Tolerance Techniques," *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, Pune, India, 2017, pp. 1-5, doi: 10.1109/ICCUBEA.2017.8463911.

[31] K. Olorunnife, et al. (2021). *Automatic Failure Recovery for Container-Based IoT Edge Applications*. Electronics, 10(23), 3047. https://doi.org/10.3390/electronics10233047

[32] R. Botez, A. -G. Pasca and V. Dobrota, "Kubernetes-Based Network Functions Orchestration for 5G Core Networks with Open Source MANO," *2022 International Symposium on Electronics and Telecommunications (ISETC)*, Timisoara, Romania, 2022, pp. 1-4, doi: 10.1109/ISETC56213.2022.10010246.

[33] S. Novanana, A. Kliks, A. S. Arifin and G. Wibisono, "Provisioning of Coexisting eMBB and URLLC services in 5G Network Slicing with Kubernetes-based MANO," *2024 IEEE International Conference on Communication, Networks and Satellite (COMNETSAT)*, Mataram, Indonesia, 2024, pp. 634-640, doi: 10.1109/COMNETSAT63286.2024.10862824.

[34] D. Upadhyay, S. Ghosh, H. Ohno, M. Zaman, and S. Sampalli, "Securing industrial control systems: Developing a SCADA/IoT test bench and evaluating lightweight cipher performance on hardware simulator," *International Journal of Critical Infrastructure Protection*, vol. 47, 2024, Art. no. 100705. [Online]. Available: https://doi.org/10.1016/j.ijcip.2024.100705.

[35] D. Beutel et al., "Flower: A Friendly Federated Learning Research Framework," 2020, doi: 10.48550/arXiv.2007.14390.

[36] Z. Du, H. Li, L. Li, B. Zhang, Z. Liu and X. Gu, "Training CSI Feedback Model with Federated Learning in Massive Mimo Systems," 2023 *8th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC)*, Beijing, China, 2023, pp. 446-450, doi: 10.1109/IC-NIDC59918.2023.10390772.

[37] W. Marfo, D. K. Tosh and S. V. Moore, "Network Anomaly Detection Using Federated Learning," MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM), Rockville, MD, USA, 2022, pp. 484-489, doi: 10.1109/MILCOM55135.2022.10017793.

[38] A. Santorsola, A. Maci, P. Delvecchio and A. Coscia, "A Reinforcement-Learning-based Agent to discover Safety-Critical States in Smart Grid Environments," *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, Tenerife, Canary Islands, Spain, 2023, pp. 1-7, doi: 10.1109/ICECCME57830.2023.10252540.

[39] A. N. Jahromi, H. Karimipour, A. Dehghantanha and K. -K. R. Choo, "Toward Detection and Attribution of Cyber-Attacks in IoT-Enabled Cyber–Physical Systems," in

IEEE Internet of Things Journal, vol. 8, no. 17, pp. 13712-13722, 1 Sept.1, 2021, doi: 10.1109/JIOT.2021.3067667.

[40] V. Beliakova, A. Beliakov and D. Topolsky, "Predictive Analysis of the Remaining Useful Life of the Continuous Casting Machine Mold through the Introduction of Digital Twin Technology," *2024 International Ural Conference on Electrical Power Engineering (UralCon)*, Magnitogorsk, Russian Federation, 2024, pp. 368-372, doi: 10.1109/UralCon62137.2024.10718987.

[41] S. A. Varghese, A. Dehlaghi Ghadim, A. Balador, Z. Alimadadi and P. Papadimitratos, "Digital Twin-based Intrusion Detection for Industrial Control Systems," *2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, Pisa, Italy, 2022, pp. 611-617, doi: 10.1109/PerComWorkshops53856.2022.9767492.

[42] P. Abinaya; S. Prem Kumar; P. Sivaprakash; K. Arun Kumar; B. Shuriya; Surbhi Bhatia Khan, "Smart Manufacturing with a Digital Twin–Driven Cyber-physical System: Case Study and Application Scenario," in *Digital Twins in Industrial Production and Smart Manufacturing: An Understanding of Principles, Enhancers, and Obstacles*, IEEE, 2024, pp.247-276, doi: 10.1002/9781394195336.ch11.

[43] N. Mohamed and J. Al-Jaroodi, "Predictive Analytics for Digital Twins: The Concept and Systems Applications," *2024 7th International Conference on Information and Computer Technologies (ICICT)*, Honolulu, HI, USA, 2024, pp. 389-393, doi: 10.1109/ICICT62343.2024.00069.

[44] S. Miteff and S. Hazelhurst, "NFShunt: A Linux firewall with OpenFlow-enabled hardware bypass," 2015 IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN), San Francisco, CA, USA, 2015, pp. 100-106, doi: 10.1109/NFV-SDN.2015.7387413.

[45] A. Chowdhary, V. H. Dixit, N. Tiwari, S. Kyung, D. Huang and G. -J. Ahn, "Science DMZ: SDN based secured cloud testbed," *2017 IEEE Conference on Network Function*

Virtualization and Software Defined Networks (NFV-SDN), Berlin, Germany, 2017, pp. 1-2, doi: 10.1109/NFV-SDN.2017.8169868.

[46] I. F. Kilincer, F. Ertam, O. Yaman, and A. Sengur, "An effective security method based on combining 802.1x, DMZ and SSL-VPN for IoT network security," Acta Infologica, vol. 4, no. 2, pp. 70–76, 2020, doi: 10.26650/acin.779547.

[47] A. H. Maulana, I. G. P. Ari Suyasa and E. Kurniawan, "Analysis of the Demilitarized Zone Implementation in Java Madura Bali Electrical Systems to Increase the Level of IT/OT Cyber Security With the Dual DMZ Firewall Architecture Method," *2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*, Istanbul, Turkiye, 2023, pp. 1-6, doi: 10.1109/SmartNets58706.2023.10215960.

[48] Y. Laili, J. Gong, Y. Kong, F. Wang, L. Ren and L. Zhang, "Communication Intensive Task Offloading with IDMZ for Secure Industrial Edge Computing," in *IEEE Transactions on Cloud Computing*, doi: 10.1109/TCC.2025.3548043.

[49] B. G. Rodiles Delgado, J. M. Aguayo, A. O. Gomez Rivera and D. K. Tosh, "Reconfigurable Network Slicing Orchestration in Network Function Virtualization Compatible Operational Technology Environment," *2024 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Oslo, Norway, 2024, pp. 207-212, doi: 10.1109/SmartGridComm60555.2024.10738063.

[50] B. G. Rodiles Delgado, L.D. Estrada Aguirre, W. Marfo, C. Servin, and D. K. Tosh, "Network Slicing for Federated Learning in Operational Technology Environment," *2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, Barcelona, Spain, 2025, doi: *in press*.

[51] ScadaBR, "ScadaBR - Open source SCADA," github.com. https://github.com/ScadaBR/ScadaBR (accessed Feb. 24, 2025).

[52] J. Lee, H. Qiu, G. Yu, J. Lin, and Rexnord Technical Services, Bearing Data Set, IMS, University of Cincinnati. NASA Ames Prognostics Data Repository. 2007.

[53] B. G. Rodiles Delgado, B. E. Casio Iracheta, A. X. Solano, D. K. Tosh, C. Servin, "Network Slicing for Dynamic DMZ and Federated Learning in Operational Technology Environment," *2025 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Toronto, Canada, 2025, doi: *submitted*.

[54] I. Frazão, P. H. Abreu, T. Cruz, H. Araújo, and P. Simões, "Denial of Service attacks: Detecting the frailties of machine learning algorithms in the classification process," in *Critical Information Infrastructures Security*, Springer, 2019, pp. 230–235.

[55] D. Antonioli and N. O. Tippenhauer. 2015. MiniCPS: A toolkit for security research on CPS networks. *In Proceedings of the First ACM Workshop on Cyber-Physical Systems Security and/or Privacy*, 91–100.

[56] R. Petersen, D. Santos, M. Smith, and G. Witte. 2020. Workforce framework for cybersecurity (NICE framework). *Technical Report. National Institute of Standards and Technology.*

[57] CAE. [n.d.]. CAE Cyber Operations (CAE-CO) Knowledge Units. https://public.cyber.mil/ncae-c/documents-library/. Last Accessed: May 5, 2025.

[58] CC2020 Task Force. 2020. *Computing Curricula 2020: Paradigms for Global Computing Education.* Association for Computing Machinery, New York, NY, USA.

[59] S. Frezza, M. Daniels, A. Pears, Å. Cajander, V. Kann, A. Kapoor, R. McDermott, A.-K. Peters, M. Sabin, and C. Wallace. 2018. Modelling Competencies for Computing Education beyond 2020: A Research Based Approach to Defining Competencies in the Computing Disciplines (ITiCSE 2018 Companion). *ACM Inroads 9*, 4 (2018), 148–174. https://doi.org/10.1145/3293881.3295782

[60] A. N. Kumar, B. A. Becker, M. Pias, M. Oudshoorn, P. Jalote, C. Servin, S. G. Aly, R. L. Blumenthal, S. L. Epstein, and M. D. Anderson. 2023. A Combined Knowledge and Competency (CKC) Model for Computer Science Curricula. *ACM Inroads 14*, 3 (Aug. 2023), 22–29. https://doi.org/10.1145/3605215

[61] A. N. Kumar, R. K. Raj, S. G. Aly, M. D. Anderson, B. A. Becker, R. L. Blumenthal, E. Eaton, S. L. Epstein, M. Goldweber, P. Jalote, D. Lea, M. Oudshoorn, M. Pias, S. Reiser, C. Servin, R. Simha, T. Winters, and Q. Xiang. 2024. *Computer Science Curricula 2023*. Association for Computing Machinery, New York, NY, USA.

[62] R. Petersen, D. Santos, M. Smith, and G. Witte. 2020. Workforce framework for cybersecurity (NICE framework). *Technical Report*. National Institute of Standards and Technology.

[63] J. Zhang, Y. Liu, and X. Wang, "5G-PBFT: An Improved Practical Byzantine Fault Tolerant Consensus Algorithm for 5G Networks," *Information*, vol. 16, no. 3, p. 202, Mar. 2025, doi: 10.3390/info16030202

[64] Y. Yin, S. Malkhi, M. K. Reiter, G. Gueta, and L. Abraham, "HotStuff: BFT Consensus with Linearity and Responsiveness," in Proceedings of the *2019 ACM Symposium on Principles of Distributed Computing (PODC)*, Toronto, ON, Canada, 2019, pp. 347–356. doi: 10.1145/3293611.3331594.

[65] A. Abraham, M. Malkhi, K. Nayak, L. Ren, and S. Seth, "SBFT: A Scalable and Decentralized Trust Infrastructure," in Proceedings of the *2019 ACM Symposium on Operating Systems Principles (SOSP)*, Huntsville, ON, Canada, 2019, pp. 122–136. doi: 10.1145/3341301.3359647.

[66] J. Mambretti, J. Chen, and F. Yeh, "Next Generation Clouds, the Chameleon Cloud Testbed, and Software Defined Networking (SDN)," *2015 ICCCRI*, pp. 73–79.

[67] I. Baldin et al., "FABRIC: A National-Scale Programmable Experimental Network Infrastructure," *IEEE Internet Computing*, vol. 23, no. 6, pp. 38–47, Nov–Dec 2019.

[68] J. Mirkovic et al., "SPHERE: Security and Privacy Heterogeneous Environment for Reproducible Experimentation," *ACM CCS '24 Poster*, pp. 5051–5053, 2024.

[69] J. Hajny, M. Sikora, A. V. Grammatopoulos, and F. Di Franco, "Adding European Cybersecurity Skills Framework into Curricula Designer," *IEEE Access*, vol. 12, pp. 61741-61766, 2024, doi: 10.1109/ACCESS.2024.3392970.

[70] CDIO Initiative, "Welcome to CDIO!," *Worldwide CDIO Initiative*, Oct. 8, 2024. [Online]. Available: https://www.cdio.org. Accessed: May 5, 2025.

[71] Asia-Pacific Advanced Network (APAN), "Asia-Pacific Advanced Network: Enabling Research and Education Collaboration Across the Asia-Pacific Region," *APAN*, Kuala Lumpur, Malaysia, 2024. [Online]. Available: https://apan.net. Accessed: May 5, 2025.

Appendix I

Peer-Reviewed Publications

Out This Thesis Research

Papers

- B. G. Rodiles Delgado, J. M. Aguayo, A. O. Gomez Rivera, and D. K. Tosh, "Reconfigurable Network Slicing Orchestration in Network Function Virtualization Compatible Operational Technology Environment," *2024 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Oslo, Norway, 2024, pp. 207-212, doi: 10.1109/SmartGridComm60555.2024.10738063.

- B. G. Rodiles Delgado, A. Solano, C. Servin, and D. Tosh, "Fostering transparent research collaborations between two- and four-year institutions in industrial cybersecurity education," in *Proc. 25th Annu. Conf. Information Technology Education (SIGITE '24)*, El Paso, TX, USA, 2024, pp. 1–6, doi: 10.1145/3686852.3687074.

- B. G. Rodiles Delgado, L.D. Estrada Aguirre, W. Marfo, C. Servin, and D. K. Tosh, "Network Slicing for Federated Learning in Operational Technology Environment," *2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*, Barcelona, Spain, 2025, doi: *in press.*

- B. G. Rodiles Delgado, B. E. Casio Iracheta, A. X. Solano, D. K. Tosh, C. Servin, "Network Slicing for Dynamic DMZ and Federated Learning in Operational Technology Environment," *2025 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Toronto, Canada, 2025, doi: *submitted*.

Posters

- B. G. Rodiles Delgado, D. K. Tosh, "Orchestrating 5G Network Slicing Through Software-Defined Network Architecture and Network Function Virtualization," Presented at *UTEP COURI Symposium Summer 2023* in El Paso, Texas; *Great Minds in STEM (GMiS) 2023* in Pasadena California; and *Carnegie Mellon University (CMU) and Computer Security Incident Response Teams (CERT) Division at the Software Engineering Institute's FloCon 2024* in Mobile, Alabama.

- B. G. Rodiles Delgado, D. K. Tosh, "Custom Scenario-Based Network Slicing Orchestration in Software-Defined Networking and Network Function Virtualization Environment," Presented at *UTEP COURI Symposium Spring 2024* in El Paso, Texas.

- B. G. Rodiles Delgado, A. X. Solano, B. E. Casio Iracheta, D. K. Tosh, C. Servin, "Establishing Trust and Resiliency in Industrial Cyber-Physical Systems," Presented at *Great Minds in STEM (GMiS) 2024* in Fort Worth, Texas.

- B. G. Rodiles Delgado, L.D. Estrada Aguirre, W. Marfo, C. Servin, and D. K. Tosh, "Network Slicing for Federated Learning in Operational Technology Environment," To Be Presented at *2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)* in Barcelona, Spain.

Appendix II

Repository for Network Slicing Supporting This Thesis Research

The software artifacts and implementation scripts for this thesis work are available in the public repository titled FL-Network-Slicing, which can be accessed at the following link: `https://gitlab.com/trucyber_gitlab/FL-Network-Slicing.git`

The repository contains the complete deployment stack for orchestrating network slicing integrated with federated learning in industrial control system environments.

The repository is organized into three main folders, each corresponding to the chapters and experiments in this thesis:

1-Reconfigurable-Network-Slicing → contains the code and scripts used for Chapter 3, focused on reconfigurable network slicing in OT environments.

2-Network-Slicing-for-FL → contains the implementation used in Chapter 4, where federated learning is integrated into the network slicing framework.

3-Dynamic-DMZ-and-FL → includes the code for Chapter 5, where the dynamic DMZ and federated learning-based anomaly detection are combined to improve OT resilience.

A key advantage of the FL-Network-Slicing architecture is its deployment flexibility: although it was originally evaluated using Open Source MANO (OSM) as the orchestrator, OSM is not strictly required to run the framework for 2-Network-Slicing-for-FL. When OSM is installed, the recommended approach is to deploy the Docker Compose stack inside a VM so that OSM manages the underlying infrastructure while Docker handles the containerized services. Furthermore, the framework can be deployed on private clouds like OpenStack or on public platforms such as Google Cloud Platform (GCP), which served as the Virtual Infrastructure Manager (VIM) for the project's experimental deployments of 2-Network-Slicing-for-FL and 3-Dynamic-DMZ-and-FL, enabling researchers and practitioners to adapt and scale the system to diverse industrial and academic environments.

Curriculum Vitae

Brian Giovanni Rodiles Delgado was born on July 9, 2001, in Ecatepec, State of Mexico. After completing elementary school, he moved with his family to Mexico City, where he attended ESANS. With certifications in English and French, he pursued a technical degree in Digital Systems at IPN's CECyT No. 9 "Juan de Dios Bátiz." In 2019, his family relocated to El Paso, Texas.

Brian earned an Associate of Science in Computer Science from El Paso Community College (EPCC), guided by Dr. Christian Servin. He served as a peer leader and participated in his first NSF Computing Alliance of Hispanic-Serving Institutions (CAHSI) Local Research Experience for Undergraduates (REU), contributing to cybersecurity advancements in operational technology (OT). In 2022, he transferred to The University of Texas at El Paso (UTEP), where he engaged in research, first applying Bayesian statistics in chemistry and later shifting focus to OT cybersecurity under Dr. Deepak Tosh.

Brian accumulated numerous honors throughout his undergraduate years, including First-Place Undergraduate Research Poster at GMiS 2022, selection as one of UTEP's first Barry Goldwater Scholars, Black Hat, Google Generation, and Top Ten Senior. Graduating Summa Cum Laude, he served as undergraduate student marshal and interned at Capital One while continuing cybersecurity research through CAHSI-Google IRP. He then entered UTEP's master's program, presenting research at conferences in Oslo, Norway, and El Paso, Texas.

In his final master's year, Brian took strategic management coursework at Yonsei University in Seoul, South Korea, he co-sponsored the U.S. Army Combat Capabilities Development Command (DEVCOM) and Army Research Laboratory (ARL) hackathon with Dr. Jaime Acosta to identify and recruit students to the cybersecurity conference DEF CON 2025, led UTEP teams to top positions in cybersecurity competitions, such as the MITRE eCTF for embedded systems and Splunk and Cisco's Boss of the SOC (BOTS), and was honored as graduate student marshal and Phi Kappa Phi (PKP) inductee. After graduating with his M.S. in Computer Science, Brian plans to present at IEEE ICMLCN 2025 in Barcelona, Spain, before launching his cybersecurity career at Capital One in August 2025.