Intelligent Anomaly Detection for 5G & Beyond: Securing Service-Based Architecture Against HTTP/2-Driven Attacks

Nathalie Wehbe

A Thesis

in

The Concordia Institute

for

Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Information and Systems Engineering) at

Concordia University

Montréal, Québec, Canada

March 2025

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Mrs. Nathalie Wehbe**

Entitled: **Intelligent Anomaly Detection for 5G & Beyond: Securing Service-Based Architecture Against HTTP/2-Driven Attacks**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Claudio Contardo

_____ External Examiner
Dr. Ashraf Matrawy

_____ Examiner
Dr. Ferhat Khendek

_____ Examiner
Dr. Lingyu Wang

_____ Examiner
Dr. Jun Yan

_____ Supervisor
Dr. Chadi Assi

_____ Co-supervisor
Dr. Hyame Assem Alameddine

Approved by _____
Dr. Farnoosh Naderkhani, Graduate Program Director

_____ 2025

_____
Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Intelligent Anomaly Detection for 5G & Beyond: Securing Service-Based Architecture Against HTTP/2-Driven Attacks

Nathalie Wehbe, Ph.D.

Concordia University, 2025

The Fifth Generation (5G) networks power diverse applications, from autonomous vehicles to smart cities, by enabling ultra-reliable low-latency communications, massive IoT connectivity, and enhanced mobile broadband. At the core of this advancement is the 5G Service-Based Architecture (SBA), which ensures scalability and flexibility through cloud-native deployment and virtualized Network Functions (NFs). The adoption of the Hypertext Transfer Protocol version 2 (HTTP/2) in the 5G SBA has become essential for enabling efficient communication between NFs. However, the adoption of HTTP/2 for NF communication introduces security risks, including stream multiplexing, slow-rate, and rapid-reset attacks, which can lead to Denial of Service (DoS) and disrupt critical services. Addressing these vulnerabilities is essential to maintaining the stability and security of 5G networks.

This thesis explores the impact of HTTP/2 vulnerabilities on the 5G SBA, identifying attack vectors that compromise the Quality of Service (QoS) of critical services. While prior studies largely assessed these threats theoretically, this research demonstrates the practical vulnerabilities of 5G networks to HTTP/2 attacks, such as stream multiplexing attacks (SMAs). To address these challenges, the thesis introduces 5GShield, an application-layer anomaly detection solution using autoencoder-based Machine Learning (ML). By profiling normal NF behavior with application-layer features, 5GShield effectively detects

deviations indicative of SMAs. Building on this, 5GGuardian is proposed as a more advanced solution to detect nuanced variations of SMAs. Leveraging 5G-Stream features and a time-series transformer, 5GGuardian captures fine-grained NF behaviors and complex patterns in HTTP/2 streams, achieving superior accuracy for both stealthy and non-stealthy anomalies. Recognizing the limitations of single detection approaches, the research introduces an ensemble learning-based solution that leverages and combines the strengths of multiple ML models trained on different feature sets in order to provide superior detection performance of HTTP/2 attacks, including slow-rate and rapid-reset attacks. By providing scalable and advanced anomaly detection, this thesis strengthens 5G SBA security, ensuring reliable service delivery and supporting the secure growth of future communication networks.

Acknowledgments

Special appreciation goes to my Ph.D. supervisor, Dr. Chadi Assi, and my co-supervisor, Dr. Hyame Assem Alameddine for their continuous support and help throughout this journey. I would like to thank Dr. Chadi Assi for his support and encouragement which has left an enduring mark on my academic and personal growth.

A special note of appreciation goes to Ericsson team members Dr. Makan Pourzandi, Dr. Amine Boukhtouta, Dr. Luis Suárez, and Dr. Boubakr Nour for their support and encouragement during challenging moments. Your camaraderie made the journey more enjoyable and memorable. I would also like to extend my thanks to Dr. Mohammad Ali Sayed and Khaled Sarieddine for their friendship and support.

My heartfelt gratitude goes to my family for their unconditional love and steadfast support throughout this journey. To my mother and my dear siblings, your endless encouragement and belief in me have been my greatest strength. This achievement would not have been possible without your unwavering presence and boundless support. Thank you for always being my foundation.

I would like to express my heartfelt gratitude to my husband, Salim Awad, for his unwavering support and encouragement throughout this journey. His understanding, patience, and steadfast belief in my abilities have been a cornerstone of my success. Thank you for being my rock, my cheerleader, and my inspiration in every step of the way.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Problem Statement

The rise of new industries, such as automotive, manufacturing, healthcare, and energy, has driven demand for stringent Quality of Service (QoS) requirements, including ultra-low latency and high reliability Christine Jost (2020); ENISA (2021); *Security Considerations for the 5G ERA* (2020). In response, mobile network operators have undertaken significant transformations of their telecommunications networks Ahmad et al. (2019); TS.23.501 (2024). This evolution led to the development of Fifth Generation (5G) networks, largely driven by the rapid expansion of Internet of Things (IoT) devices and dynamic changes in the telecommunications landscape. The mobile network is broadly divided into the Radio Access Network (RAN) and the core network. The RAN handles wireless communication between user devices and the network, while the core network manages data routing, connectivity, and overall network functionality. To support these advancements, the 5G Core (5GC) network has adopted a Service-Based Architecture (SBA) and integrated cloud-native applications 3GPP TS.29.500 (2024); Ahmad et al. (2019); TS.23.501 (2024). The 5G SBA follows a cloud-native deployment and leverages virtualization technologies

for the implementation of its Network Functions (NFs) that provide access to network resources and capabilities via Service Based Interfaces (SBIs) 3GPP TS.29.500 (2024), thus enabling better scalability, flexibility and service management TS.23.501 (2024). 5G relies on a standardized set of REpresentational State Transfer (RESTful) Application Programming Interfaces (APIs) combined with web-based technologies including the Transport Control Protocol (TCP) /Transport Layer Security (TLS) /Hypertext Transfer Protocol version 2 (HTTP/2) /JavaScript Object Notation (JSON) protocol suit for the communication between its NFs Hu, Liu, Liu, You, and Zhao (2018).

With the large-scale deployment of 5G networks and the introduction of 5G Standalone (SA) networks in late 2020 Report (2022), several incidents in recent years have underscored the growing security concerns in the 5G landscape. In Canada, leading mobile operators have reported multiple outages, such as Rogers, Bell Canada, and TELUS Canadian Radio-television and Telecommunications Commission (2024). A notable example occurred in July 2022 Xona Partners Inc (2024), Rogers Communications' 5G network suffered a major outage, affecting millions of users. The outage, which is linked to a software update failure, revealed vulnerabilities related to network software management. Similarly, in February 2023 CNN (2024), a software update glitch in AT&T's 5G network in the United States caused at least 70,000 phones to lose connectivity, demonstrating the potential consequences of improper update management in 5G systems. AT&T attributed this disruption to an internal error, marking one of several incidents the company has faced in recent years. On 4 June 2024, another issue prevented customers from completing calls between carriers. Furthermore, critical 911 services have faced disruptions, including an April 2024 outage in Nevada, South Dakota, and Nebraska caused by installing a light pole and a June 2024 incident in Massachusetts resulting from a computer firewall failure. Moreover, in July 2024, a global tech outage further affected 911 services in Alaska and Arizona, highlighting the fragility of essential communication systems. These incidents

underscore the urgent need for hightened security measures, comprehensive testing solutions, and robust system architectures to maintain stability and safeguard the security of critical communications infrastructure.

The adoption of the HTTP/2 protocol, which facilitates communication among NFs within the 5G SBA, is instrumental in enabling efficient signaling. With features like stream multiplexing, header compression, and prioritization IETF (2015), HTTP/2 plays a crucial role in enabling efficient signaling between NFs, improving resource utilization and service responsiveness 3GPP TS.29.500 (2024). While HTTP/2 provides performance advantages, design vulnerabilities can expose critical 5G infrastructure to potential risks, jeopardizing the network's security and availability. One of the most alarming incidents was highlighted in the February 2023 Cloudflare report Cloudflare (2023) on hyper-volumetric HTTP/2 DDoS attacks in the web, where request rates peaked at 70 million per second. These attacks leveraged HTTP/2's multiplexing feature to overwhelm servers, causing widespread service disruptions.

Despite its inherent security-by-design principles, the 5G SBA and its reliance on protocols like HTTP/2, require enhanced security mechanisms to address emerging vulnerabilities TS.33.501 (2025a). Existing literature has touched on some HTTP/2 attacks that exploit its features in the web Ahmad et al. (2019); Hu et al. (2018), yet the implications of these attacks on 5G SBA require further exploration. Furthermore, other works have primarily focused on vulnerabilities arising from virtualization technologies Ahmad et al. (2019); Christine Jost (2020); ENISA (2021). By addressing these challenges, operators can ensure that 5G networks remain resilient and reliable, unlocking their full potential to support transformative applications while maintaining the trust and confidence of users and industries alike.

This thesis is motivated by the realization that HTTP/2 and APIs are well-known targets for potential attackers. Recent studies Imperva (2016); National Vulnerability Database

(NVD) (2023); Praseed and Thilagam (2019); Tripathi and Hubballi (2018); Tripathi and Shaji (2022) have shown that HTTP/2 is vulnerable to Denial of Service (DoS) attacks, including slow-rate, stream multiplexing, and rapid reset attacks, which exploit some of its features. Following these findings, some Machine Learning (ML) based anomaly detection solutions leveraging flow-based and event-based features have been developed for web-based HTTP/2 security Praseed and Thilagam (2020, 2021); Tripathi (2022); Tripathi and Shaji (2022). Therefore, the cybersecurity implications of web-based technologies on 5G SBA need more attention. In fact, Hu et al. (2018) presented some HTTP/2 attacks exploiting its features without a thorough discussion on their implications on 5G SBA.

As a result, our work begins with a thorough analysis of the security implications of web-based technologies, particularly HTTP/2, within the 5G SBA. We examine the security controls of 5G SBA, the role of HTTP/2, and its potential as an attack surface. This includes investigating vulnerabilities in HTTP/2 features and their exploitability in 5G networks. Furthermore, we extend our examination to address the security challenges of HTTP/2 to 5G networks and propose anomaly detection solutions within the 5G SBA. In the context of 5G networks, anomaly detection plays a vital role in identifying unusual activities that could compromise the network's performance or security. Given the dynamic and distributed nature of 5G SBA, the challenge lies in detecting anomalies across diverse and high-dimensional data sources, such as HTTP/2 protocol traffic and interactions among NFs Karim, Mubasshir, Rahman, and Bertino (2023). ML-based approaches, particularly deep learning models, have emerged as powerful tools for detecting anomalies by leveraging their ability to learn intricate patterns and dependencies in data Anderson et al. (2023); Xu, Wu, Wang, and Long (2021). In this thesis, we explore how anomaly detection solutions can be tailored to address the unique security challenges posed by HTTP/2 vulnerabilities within the 5G SBA. These solutions are designed to detect sophisticated attacks that exploit the HTTP/2 protocol while ensuring adaptability and scalability for their

deployment across 5G networks.

1.2 Contributions

This dissertation contributes to advancing the state-of-the-art in securing 5G SBA by addressing critical cybersecurity challenges associated with the HTTP/2 protocol. The focus is on assessing HTTP/2-driven attacks in 5G SBA, analyzing the HTTP/2 attack's impact on 5G SBA, and developing anomaly detection solutions to enhance the security of 5G networks. These contributions are closely related and build upon each other, forming a cohesive framework for improving 5G SBA security. The research begins with an in-depth security assessment of HTTP/2 vulnerabilities in the 5G SBA, establishing the groundwork for understanding attack vectors and their potential impact. This is followed by the development of multiple anomaly detection solutions, each progressively refining detection granularity, scalability, and effectiveness. The contributions transition from application-layer detection (5GShield) to sequence-based analysis of NF behavior (5GGuardian) and finally to a robust ensemble learning-based approach (Kraken), which unifies insights across different feature types to provide a comprehensive defense against HTTP/2 attacks. An overview of the contributions is provided below, with each contribution detailed in its respective subsection:

1.2.1 A Security Assessment of HTTP/2 Usage in 5G Service Based Architecture

In Chapter 3, we delve into the security implications of adopting the HTTP/2 protocol within the 5G SBA, focusing on its features, as well as its standard and custom headers. HTTP/2 message headers consist of multiple fields, with standard headers employed in both requests and responses. Requests to the HTTP/2 server include a structured set of

header fields that identify the client and facilitate communication. Additionally, 3GPP has introduced HTTP/2 custom headers specifically tailored for 5G SBA, some of which are critical for load and overload control by enabling the sharing of NFs load information 3GPP TS.29.500 (2024). Motivated by the fact that HTTP/2, APIs, and JSON are well-known to attackers, we make the following contributions in this chapter:

- We provide a detailed examination of the HTTP/2 standard and custom headers and their roles in enhancing or potentially undermining the security of 5G SBA.
- We analyze the applicability of known HTTP/2 attack vectors in the context of standardized APIs within 5G SBA. This analysis highlights not only the vulnerabilities introduced by HTTP/2 but also the security opportunities it presents. We identify promising research directions to address these challenges and explore the potential of related technologies to fortify the 5G SBA ecosystem.

1.2.2 5GShield: HTTP/2 Anomaly Detection in 5G Service-Based Architecture

While many works Praseed and Thilagam (2021); Tripathi and Shaji (2022) developed anomaly detection solutions to secure the web against HTTP/2 attacks using ML techniques, HTTP/2 attacks on 5G SBA were only assessed theoretically in Hu et al. (2018); Wehbe, Alameddine, Pourzandi, Bou-Harb, and Assi (2022). To the best of our knowledge, no practical implementation of these attacks in a 5G environment exists. Further, an evaluation of existing HTTP/2 anomaly detection solutions in a 5G network remains absent. We argue that 5G networks are vulnerable to HTTP/2 attacks and demonstrate that HTTP/2 Stream Multiplexing Attacks (SMA) can occur between two 5G NFs. Furthermore, most of the existing anomaly detection solutions rely on flow-based features

collected at the network layer. We contend that application-layer attacks (e.g., HTTP/2 attacks) that exploit vulnerabilities in application-layer protocols may not appear malicious when observed from the network or transport layers Xie and Zhang (2012). As a result, existing anomaly detection methods that rely on flow-based features fail to efficiently detect such application-layer attacks. Our contributions can be summarized in Chapter 4 as follows:

- We generate a 5G SBA HTTP/2 dataset that captures both normal and abnormal 5G SBA network behavior under the HTTP/2 SMA in both stealthy and non-stealthy modes, using the open-source Free5GC Free5GC (2021a) testbed and UERANSIM aligungr (2021), a User Equipment (UE)/RAN emulator.

- We develop 5GShield, an application-layer anomaly detection solution based on Autoencoder (AE) Mirsky, Doitshman, Elovici, and Shabtai (2018). 5GShield acts as a shield for 5G NFs that provides intelligent attack detection capabilities for increased security. As the rate and statistics of 5G API calls between 5G NFs vary under an HTTP/2 SMA in comparison to a normal network state, 5Gshield extracts application-layer features (e.g., numberofAttemptedNetworkInitiatedServiceRequest, numberofSuccessfulNetworkInitiatedServiceRequest, etc.) to capture these statistics. It then uses them to profile normal NFs behavior. Thus, deviation from the captured normal profile can then be detected by 5GShield as an attack. Further, we show that 5GShield can detect HTTP/2 SMA, outperforming a flow-based anomaly detection solution.

1.2.3 Empowering 5G SBA Security: Time Series Transformer for HTTP/2 Anomaly Detection

After a thorough review of the existing literature, existing works are not fine-grained enough to capture 5G API calls dependencies and sequences for fulfilling 5G procedures Praseed and Thilagam (2018, 2019, 2020, 2021). The latter can be exploited to perform HTTP/2 attacks. Many detection solutions Praseed and Thilagam (2019, 2021); Wehbe, Alameddine, Pourzandi, and Assi (2023) are computationally intensive, which can pose challenges for real-time applications, particularly in a 5G environment. However, the practicality of these solutions depends on the specific needs of the operator, as they take advantage of the features of the application layer to address privacy concerns. In Chapter 5, we make the following contributions.

- We emulate five variations of HTTP/2 SMA that use different 5G procedures and examine the impact on 5G network performance. Using the pcaps generated by the aforementioned emulation, we develop a method to extract 5G-specific NF behavior from HTTP/2 streams (i.e., a stream represents an HTTP/2 request and response), referred to as 5G-Stream features. These features capture fine-grained details of an NF behavior through its 3GPP APIs, which allow the detection of any anomalous behavior that might be hidden by flow-based features or application-layer features. The used 5G-Stream features are general enough to make our anomaly detection solution adaptable to any NF of the 5GC.

- We develop an anomaly detection solution, namely 5GGuardian, which leverages a time series transformer Wen et al. (2022) that introduces an attention-based transformer encoder. Transformers have been shown to be highly effective for anomaly detection, given their ability to capture long-range dependencies, process sequential

information, and adapt to unique patterns in data which makes them a good candidate for HTTP/2 anomaly detection in 5G networks. Furthermore, we demonstrate the effectiveness of utilizing 5G-Stream features in our anomaly detection model in identifying HTTP/2 SMA. 5GGuardian showed superior performance when compared to its counterpart, 5Gshield Wehbe et al. (2023), demonstrating the efficiency of transformers and stream features in comparison to autoencoder and application-layer features. The results highlight the superiority of applying transformers, even in the presence of contaminated data, surpassing the performance of previous solutions.

1.2.4 HTTP/2 DoS Attacks in 5G Networks: Impact Analysis and Anomaly Detection

This work addresses the lack of practical studies and analyses on the impact of HTTP/2 attacks on 5G networks, especially given the absence of a 5G-compliant dataset for anomaly detection. Notably, Wehbe et al. (2023) studied the impact of HTTP/2 stream multiplexing attacks on 5G NFs without publishing the data, while Caccavale, Nguyen, Cavalli, Montes De Oca, and Mallouli (2023) proposed methodologies using 5Greplay without emphasizing their impact on 5G networks as they did not use a 5G testbed. Motivated by existing studies Caccavale et al. (2023); Hu et al. (2018); VIII (2022); Wehbe et al. (2022), we make the following contributions in Chapter 6:

- Using 5G Testbed, we create a 5G SBA HTTP/2 dataset, capturing both normal and malicious network behavior including a total of six different variations of stream multiplexing, rapid reset, and slow rate attacks National Vulnerability Database (NVD) (2023); Praseed and Thilagam (2019); Tripathi and Hubballi (2018); Wehbe et al. (2023). We show that these attacks cannot only cause a DoS on the targeted NF but also affect differently the availability of the whole network. To the best of our knowledge, our 5G SBA HTTP/2 dataset is among the first to capture different 5G

procedures.

- We pre-process our 5G HTTP/2 dataset to extract flow-based features that are widely known in the literature for their ability to distinguish between normal and malicious behaviors. To the best of our knowledge, this dataset will be the first publicly available resource. The extracted features are used in the training of three renowned machine learning techniques, mainly; AE, Long Short Term Memory Autoencoder(LSTM-AE), and Isolation Forest for HTTP/2 anomaly detection. Our results demonstrate good detection performance, confirming that flow-based features are effective for 5G traffic characterization by extracting the feature from 5G SBA instead of per NF.

1.2.5 Kraken: Multi-Layer Ensemble Learning Detection of HTTP/2 Attacks in 5G and Beyond

5GShield Wehbe et al. (2023) is an Autoencoder-based anomaly detection solution, that was proposed for HTTP/2 SMA stealthy and non-stealthy detection in 5G SBA. It achieved an F1-score of 0.992 when trained on application-layer features, but its F1-score dropped to 0.78 with flow-based features, indicating the shortcomings of flow-based anomaly detection solution, usually used in the web for detecting HTTP/2 attacks. Building on this, Wehbe, Alameddine, Pourzandi, and Assi (2025) proposed 5GGuardian, a model leveraging 5G-stream features to train a time-series transformer per NF, reaching an average F1-score of 0.98 after showing the shortcomings of 5GShield in detecting variations of HTTP/2 SMA. Wehbe et al. (2023) and Wehbe et al. (2025) demonstrate that the choice of features and ML model lead to variable detection performance across different variations of SMA. While both solutions addressed SMA, other HTTP/2 attacks, like slow-rate and rapid reset attacks, remain unexplored in 5G Cloudflare (2023); Tripathi and Hubballi (2018). Motivated by the limitations of single-type feature detection, we make the following contributions in Chapter 7:

- We develop Kraken, a multi-layer ensemble learning solution designed for 5G SBA anomaly detection. Kraken leverages ensemble learning at each NF and across the 5G SBA NFs to detect sophisticated attacks exploiting the SBA interconnected nature. We explore the benefits brought by three distinct feature sets combined with unsupervised ML models in capturing the unique aspects of each NF behavior to improve HTTP/2 attack detection. Thus, we consider flow-based, 5G-stream, and HTTP/2 event-frame features to respectively train a time-series transformer, an AE, and an LSTM-AE.

- We evaluate Kraken across six emulated attack scenarios, and a network surge scenario that stresses the 5GC under high-traffic conditions. We show that Kraken consistently achieves a high F1-score, outperforming a flow-based LSTM-AE, a 5G-stream time-series transformer, and HTTP/2 event-frame AE attack detection solution. Kraken also demonstrates accuracy and reliability in differentiating HTTP/2 DoS attacks from a network surge.

1.3 Thesis Organization

The road map of this thesis is as follows. In Chapter 2, we provide an overview of the 5G SBA signaling, security mechanisms, and the adoption of the HTTP/2 protocol, along with related work on HTTP/2 anomaly detection and 5G security solutions to secure the network against HTTP/2 attacks. Chapter 3 explores the security features of the 5G SBA, highlighting HTTP/2 vulnerabilities, attacks, and potential research directions. Chapter 4 introduces 5GShield, an application-layer anomaly detection solution using neural networks, evaluated through emulated HTTP/2 attacks. Chapter 5 presents 5GGuardian, a time-series transformer-based detection solution for HTTP/2 Stream Multiplexing Attacks, showcasing its robustness against contaminated data and superior performance compared

to 5GShield. In Chapter 6, we emulate six HTTP/2 attacks to analyze their cascading effects on NFs and train ML models using extracted flow-based features. Chapter 7 proposes Kraken, a multi-layer ensemble learning system integrating diverse feature sets across NFs to detect multi-stage attacks. Finally, Chapter 8 concludes this thesis, summarizes its contributions, and identifies some research gaps for future exploration.

Chapter 2

Background and Literature Review

The emergence of 5G technologies represents a transformative shift in the telecommunications landscape. 5G brings new capabilities that will reshape how we connect and communicate in the digital age Tang, Ermis, Nguyen, De Oliveira, and Hirtzig (2022). At the core of 5G lies its exceptional capacity to deliver significantly higher data transfer speeds, reduced latency, extensive device connectivity, and heightened network reliability. These innovations aim to offer a diverse spectrum of applications, ranging from ultra-high-definition streaming and immersive augmented reality experiences to the expansive ecosystem of the IoT and the essential foundation of mission-critical industrial automation. The proliferation of 5G technology holds the promise of introducing a new era of connectivity, one that is set to reshape entire industries, transform consumer experiences, and ignite innovation on a global scale Dutta and Hammad (2020); TS.23.501 (2024). This chapter explores 5G networks, with an emphasis on the security of 5G SBA in the context of web technologies like HTTP/2.

2.1 Background

2.1.1 Overview of 5G Network

The rise of new vertical industries, such as automotive and manufacturing, with stringent QoS requirements like ultra-low latency and high reliability, has driven mobile network operators to revolutionize their telecommunications networks to support these emerging use cases Ahmad et al. (2019); TS.23.501 (2024). Consequently, 5G networks have been developed, introducing a new radio access technology and transitioning from traditional hardware-based systems to fully virtualized SBA 3GPP TS.29.500 (2024).

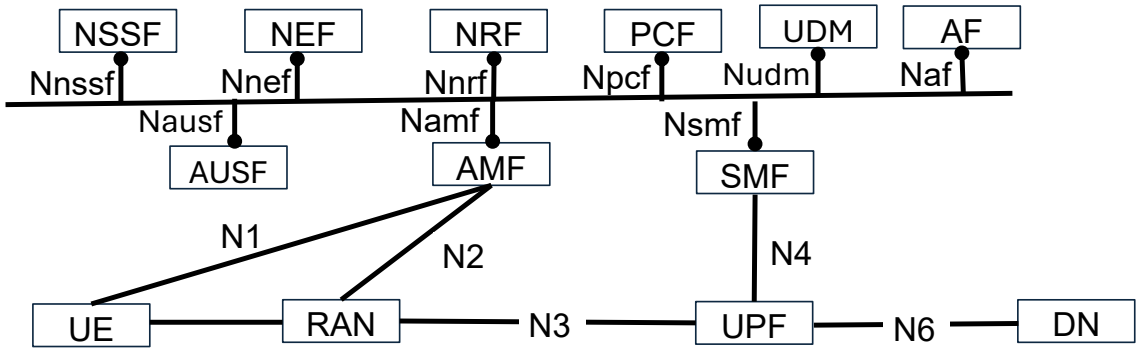


Figure 2.1: 5G Service-Based Architecture TS.23.501 (2024)

The 5GC network is a cloud-native design that adopts a SBA as depicted in Figure 2.1, based on Network Function Virtualization (NFV) and Software Defined Networking (SDN) principles TS.23.501 (2024). Virtual Network Functions (VNFs) is used in 5GC to decouple network functions from dedicated and proprietary hardware, thus allowing their instantiation as VNFs on commercial off-the-shelf servers Madi, Alameddine, Pourzandi, and Boukhtouta (2021). SDN enables the separation of control and user plane functions. SDN and NFV are designed to support the unique requirements of 5G networks, such as low latency, high throughput, and a large number of connected devices. The 5G SBA enables a granular design and delivery of 5G network functionality through a decoupling of the User

Plane (UP) and Control Plane (CP), as depicted in Figure 2.1. The CP manages the connection between the UE (i.e., 5G smartphones, 5G cellular devices) and the network, including tasks such as authentication, policy control, and mobility management. The UP handles the actual data transmission between the UE and the network using a Packet Data Unit (PDU) session. 5G SBA is composed of a set of interconnected NFs that allow extended exposure of network capabilities and resources by offering a multitude of NF services through an SBI 3GPP TS.29.500 (2024). These services are offered to other NFs using well-defined RESTful APIs over HTTP/2 3GPP TS.29.500 (2024), thus, enabling a secure, reliable, efficient, and bidirectional TS.33.501 (2025a). NFs in the control plane communicate with each other using either a *Request-Response* or *Subscribe-Notify* interactions between an NF consumer (NFc) and an NF producer (NFp) 3GPP TS.29.500 (2024) as depicted in Figure 2.2. *Request-Response* is used when an NF consumer requests a service and an NF NFp responds, while *Subscribe-Notify* is employed when an NF consumer subscribes to an NFp event that causes an NFc to be called back when the event occurs 3GPP TS.29.500 (2024); TS.23.501 (2024). Be it a *Request-Response* or *Subscribe-Notify*, interactions between 5G NFs are based on service exchange through 3GPP standardized Restful APIs TS.123.502 (2025).

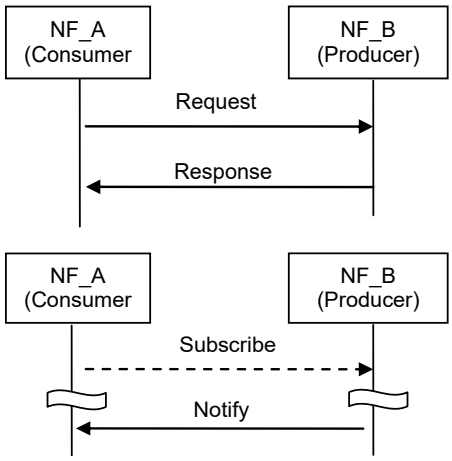


Figure 2.2: Request-Response and Subscribe-Notify TS.23.501 (2024)

5G SBA is a new approach for designing and deploying 5G networks based on SDN and NFV. The use of NFs abstracted from the underlying hardware and implemented as software services that can be deployed and orchestrated in a virtualized environment is intrinsic to 5G SBA. This enables a more flexible and efficient network management, as well as faster deployment of new services. Each NF has a list of services, for example, the SMF is the central management entity responsible for creating, managing, and terminating sessions between users and their desired services in 5G SBA, as depicted in Table 2.1.

Table 2.1: 5GC components Sree Lekshmi (2022)

5G Network Elements	**Functionalities**
Access and Mobility Management Function (AMF)	Provides access and mobility control, UE Registration. AMF also ends Non-Access Stratum (NAS) signaling.
Session Management Function (SMF)	Performs session management and controls user plane traffic.
User Plane Function (UPF)	Handles packet forwarding and routing. It performs packet inspection and QoS implementation.
Network Slice Selection Function (NSSF)	Supports network slice selection. Selects network slice instances to serve a UE.
Network Repository Function (NRF)	Maintains NF profile and their functions. Discover connections between NFs.
Network Exposure Function (NEF)	Exposes securely network capabilities and events.
Unified Data Management (UDM)	Stores the subscriber information to support identification, access authorization, and billing.
Authentication Server Function (AUSF)	Stores authentication keys to authenticate UEs.
Policy Control Function (PCF)	Maintains network policies to manage network behavior.
Application Function (AF)	Fulfills the role of application server. Interacts with 5GC to provide services.

5G SBA also provides a higher level of network automation, programmability, and orchestration, allowing operators to more easily manage and optimize the network, and quickly launch new services and applications TS.33.501 (2025a). It is also designed to be highly programmable and to support network automation, which enables faster deployment

of new services and more efficient management of the network. Table 2.1 shows the key NFs of a 5GC network, in which each NF plays a critical role and has its own services.

The security risks associated with inadequately protected virtualized deployments are acknowledged, and network equipment vendors must address them through implementation-specific measures. 5GC security is a complex and evolving field, and new threats and vulnerabilities are constantly being discovered due to the adoption of technologies such as HTTP/2 ENISA (2021); *Security Considerations for the 5G ERA* (2020).

2.1.2 HTTP/2 as the 5G Signalling Protocol

HTTP/2 is an application-layer protocol used for signaling between 5G NFs in the form of a *request-response* or *subscribe-notify* to enable communications between an NFc and an NFp 3GPP TS.29.500 (2024), where the former is a 5G NF that can access a service of the latter. HTTP/2 introduces the notion of a *stream*, which corresponds to an HTTP request-response exchange. An HTTP/2 *message* is represented by either a request or a response. HTTP/2 messages are composed of HTTP/2 frames. Thus, a stream can be defined as a bidirectional flow of frames IETF (2015). An HTTP/2 *frame* represents the basic HTTP/2 data unit (i.e., smallest unit of communication within an HTTP/2 connection) with binary encoding. A frame can be of different types from which we mention: (1) *HEADERS* frame which is used to open a stream and carries different header fields in the form of key-value pairs; (2) *DATA* frame carries HTTP request or response payload; (3) *SETTINGS* frame is used by both client and server to convey configuration parameters that affect their communication IETF (2015). Extending HTTP/2 is possible through the addition of new frame types, settings, and error-codes IETF (2015). HTTP/2 protocol provides multiple features such as stream multiplexing, flow control, stream dependency and prioritization, header compression, and server push.

A. HTTP/2 Features

Due to the adoption of HTTP/2 in 5G SBA between NFs 3GPP TS.29.500 (2024), in this section, we describe the HTTP/2 features and their advantage. HTTP/2 is an updated version of the HTTP that was designed to address some of the limitations of HTTP/1.1. HTTP/2 features IETF (2015) include:

- Binary protocol: HTTP/2 uses a binary format for its messages, as opposed to the text-based format used by HTTP/1.1. This makes it more efficient and less error-prone.

- Stream Multiplexing: HTTP/2 allows multiple requests and responses to be sent and received over a single TCP connection, eliminating the need for multiple connections to be established for each resource.

- Prioritization: HTTP/2 allows an HTTP/2 client to assign priorities to different requests so that the HTTP/2 server can prioritize the delivery of resources that are most important for the current page.

- Flow control: HTTP/2 enables flow control to determine the size of data that the sender is allowed to send to the receiver by utilizing many parameters such as the `WINDOW_UPDATE` frame and the `SETTINGS` frame (e.g., `SETTINGS_MAX_CONCURRENT_STREAMS`) (Table 2.2) IETF (2015). The receiver uses the `WINDOW_UPDATE` frame to inform the sender how much data it is willing to receive on each stream IETF (2015).

- Header compression: HTTP/2 uses a compression algorithm called HPACK to compress the `HEADER` frame (Table 2.2) of HTTP requests and responses, reducing the amount of data that needs to be sent over the network.

- Server push: HTTP/2 allows a server to push resources to a client before they are requested thus, reducing the number of round trips needed to load a page/service.

Table 2.2: Frame type IETF (2015)

Frame Type	**Role**
DATA	Carries HTTP request or response.
HEADERS	Opens a stream and carries a header block.
PRIORITY	Specifies the sender-advised priority of a stream.
SETTINGS	Contains configuration parameters that affect how the endpoint communicates.
RST_STREAM	Allows for immediate termination for a stream.
PUSH_PROMISE	Notifies the peer endpoint in advance of stream the sender intends to push.
PING	Is a mechanism for measuring a minimal round trip time from the sender.
GOAWAY	Initiates countdown of a TCP connection or to signal a serious error condition.
WINDOW_UPDATE	Implements a flow-control.
CONTINUATION	Continues a sequence of header block fragments.

In addition, HTTP/2 has added security features to defend against typical attacks. Similar to HTTP/1.1, a secure connection is established between client and server by utilizing Secure Sockets Layer (SSL)/TLS encryption, ensuring all information exchanged is confidential and protected from interference or alteration IETF (2015).

2.2 Literature Review

This section lays the groundwork for addressing the security challenges in the 5G SBA. We begin with a review of HTTP/2, its integration into the 5G SBA, and its associated security implications, including existing HTTP/2 attacks in web environments and 5G, as well as related datasets. The discussion then shifts to anomaly detection techniques, since the thesis focuses on anomaly detection as a solution to HTTP/2-driven attacks, we review existing works on anomaly detection techniques, emphasizing their application to 5G and web-based technologies.

2.2.1 HTTP/2 and 5G SBA Security

The security of HTTP/2 was discussed in Imperva (2016) in which the authors showed that all web servers are vulnerable to at least one attack vector such as slow-read attack, stream dependency DoS, and stream abuse attacks. Work in Tripathi and Hubballi (2018) presented five versions of slow-rate DoS attacks that exploit different frame types of an HTTP/2 stream. They showed the impact of these attacks on lab-based HTTP/2 web servers. The work in Praseed and Thilagam (2019) discussed Application Layer DDoS (AL-DDoS) attacks against web servers, such as the multiplexed asymmetric attack that results in heavy computational overhead on the server. Only few works Hu et al. (2018); VIII (2022) assessed HTTP/2 security in 5G SBA. Authors of Hu et al. (2018) investigated 5G signaling security vulnerabilities exposed by the use of the HTTP/2 protocol. The authors focused on the features that can be misused to launch DoS attacks in 5G SBA. The work in VIII (2022) presented a report on security vulnerabilities in HTTP/2 and their impact on 5G networks. The discussion on HTTP/2 attacks in the literature is limited to a qualitative assessment of their applicability to 5G SBA without any practical demonstration.

2.2.2 HTTP/2 Attacks

Different protocols define the communication between the 5GC and UE/RAN over network communication. However, these standard protocols are not immune to attacks. Focusing on HTTP/2 as it is the only protocol used in the CP signaling, HTTP/2 is designed to improve the performance and efficiency of web/NF communications, but like any technology, it can also introduce new vulnerabilities Hu et al. (2018). In the context of 5G SBA, the following are some of the potential attacks exploiting vulnerabilities in the HTTP/2 protocol. HTTP/2 SMA Praseed and Thilagam (2018) is one of the attacks that is caused by the stream multiplexing feature. An attacker who can establish a TCP connection to a server and then open multiple streams at the same time can launch this type of attack. The

attacker can then use these streams to flood the server with requests, causing it to become overwhelmed and unavailable to legitimate clients. This is a type of DoS attack because it can cause the targeted server to become unavailable and disrupt the service. An HTTP/2 SMA can also be used to deplete server resources like memory or CPU. Another group of researchers Praseed and Thilagam (2019) focused on the misuse of HTTP/2 stream multiplexing feature to launch AL-DDoS attacks against web servers such as the multiplexed asymmetric attacks. Such attacks are applicable when an attacker sends a high workload of requests that results in heavy computational overhead on the server. HTTP/2 SMA may not be detected by any firewall Praseed and Thilagam (2021) and require an efficient anomaly detection mechanism Praseed and Thilagam (2020).

Another attack, such as a Slow-Read attack, can be launched by an attacker who establishes a connection to a server and then opens multiple streams simultaneously. The attacker can then send a large number of requests over these streams, the attacker does not allow the server to send the response as a whole by setting a small limit to the `WINDOW_SIZE`. Furthermore, the server needs to wait for `WINDOW_SIZE` update to send the remainder of the response, which keeps it busy for a long time and causes the attack. There is one thread per stream that remains open at the server side, which overwhelms the server as all its threads will be consumed, and hence, it cannot process other incoming requests. This can overload the server and make it unavailable for legitimate clients. This type of attack can be particularly effective in HTTP/2 because of the multiplexing feature, which allows multiple requests and responses to be sent and received over a single connection simultaneously. This can make it difficult for the server to detect and respond to the attack. HTTP/2 security was discussed in Imperva (2016) where the authors showed that all web servers are vulnerable to at least one attack vector such as Slow Read attack, HPACK (compression) bomb, stream dependency DoS, and stream abuse attacks. Work in Tripathi and Hubballi (2018) presented five versions of Slow-Rate DoS attack that exploit the `SETTINGS` of an HTTP/2

stream such as slow-rate DoS Attack using Complete POST Header and slow-rate DoS Attack using connection preface. Moreover, authors in Tripathi and Shaji (2022) showed the impact of such attacks on HTTP/2 web servers in a lab setup only. In subsequent work, the same authors demonstrated in Tripathi (2022) the impact of a slow-rate DoS attack on a real web server and developed a real-time detection strategy that is based on an event sequence analysis to detect it in real-time.

One of the latest attacks discovered, called HTTP/2 Rapid Reset Attack, identified as CVE-2023-44487 National Vulnerability Database (NVD) (2023), exploits the stream multiplexing feature of HTTP/2. It employs the `RST_STREAM` frame to terminate streams that are currently processing requests IETF (2015). In this case, the number of streams that were reset by the `RST_STREAM` frame do not count towards `SETTINGS_MAX_CONCURRENT_STREAMS` IETF (2015). The mitigation for this attack considers counting any request reaching the server, even if it is a `RST_STREAM` frame, as part of the defined maximum stream limit. It involves limiting the number of simultaneously executing handler routines (`SETTINGS_MAX_CONCURRENT_STREAMS`= 200) and prevents server overload by queuing incoming requests until a current request is completed. If the queue becomes excessively long, the server terminates the connection as a safeguard. However, increasing the `SETTINGS_MAX_CONCURRENT_STREAMS` slightly could significantly impact network performance. Cloudflare highlighted its challenges in mitigating the rapid reset attack as it overwhelms server resources and disrupts HTTP/2 operations globally Cloudflare (2023).

2.2.3 HTTP/2 Dataset

Previous studies, as highlighted in Hussain, Du, Sun, and Han (2020); Pourahmadi, Alameddine, Salahuddin, and Boutaba (2022); Praseed and Thilagam (2020, 2021); Salahuddin, Pourahmadi, Alameddine, Bari, and Boutaba (2021), have extensively utilized various datasets, including CICIDS2018, CICDDoS2019, 4G-LTE, modified HTTP/1.1 to HTTP/2

dataset, and HTTP/2 web server dataset to assess HTTP/2 attacks in the web. These studies have demonstrated the importance of flow-based features in detecting anomalies and potential threats. These features are crucial for understanding network traffic behavior and patterns, by providing valuable data on traffic volume and duration, which can help identify malicious activities such as DoS attacks or unauthorized access attempts. However, these HTTP/2 datasets are often private and not specifically collected within a 5G context, while the publicly available datasets are not ideally suited for cutting-edge research within 5G networks.

Several works on 5G Amponis et al. (2023); Karim et al. (2023); Samarakoon et al. (2022) that leverage a 5G testbed focus on emulating non HTTP/2 attacks and do not provide any HTTP/2 based dataset from 5G. Moreover, the work of Caccavale et al. (2023), proposed testing methodologies using an open-source solution called 5Greplay, allowing network operators to defend against flooding and fuzzing attacks. However, the authors did not focus on the impact of these attacks on 5G networks, as their study did not employ a 5G testbed. Additionally, the HTTP/2 dataset in their research was created using MMT-DPI, a tool developed to parse and mutate HTTP/2 packets, which is not 5G specific.

2.2.4 Anomaly Detection

In the following, we discuss the literature review on HTTP/2 anomaly detection and the application of ML as a solution for anomaly detection.

A. HTTP/2 Anomaly Detection

Few works in the literature presenting anomaly detection solutions Hussain et al. (2020); Lam and Abbas (2020); Praseed and Thilagam (2020); Salahuddin et al. (2021); Xie and Zhang (2012) focused on DDoS attacks including HTTP/1.1 flooding attack rather than HTTP/2 attacks. Authors of Praseed and Thilagam (2020) employed statistical methods

to detect HTTP AL-DDoS attacks. Xie and Zhang (2012) proposed an application-layer anomaly detection method that utilizes keywords from application-layer protocols such as HTTP and SMTP, like GET, PUT, and POST to create a hidden semi-Markov model to detect anomalies. Except the work in Tripathi and Shaji (2022) that focused HTTP/2 slow-read attack, none of these works used an HTTP/2 dataset. Further, to the best of our knowledge, the work in Praseed and Thilagam (2021) is the only work that addressed HTTP/2 SMA detection. However, the authors used an HTTP/1.1 dataset just like the remaining works which focused on HTTP/1.1 attacks.

The use of web-based technologies such as HTTP/2, JSON, and RESTful API extends the 5G SBA attack surface. Only a few work Hu et al. (2018); VIII (2022) assessed HTTP/2 security in 5G SBA. Authors of Hu et al. (2018) investigated 5G signaling security vulnerabilities exposed by the use of HTTP/2 protocol. The authors examined four DoS attacks and two privacy attacks such as MITM and interconnect attacks. They focused on stream reuse, flow control, dependency and priority attacks along with header compression attacks which are DoS attacks that are relevant in 5G SBA. Numerous studies examine threats exploiting the HTTP/2 protocol in web environments, such as HTTP/2 SMA Praseed and Thilagam (2019), HTTP/2 rapid reset attacks National Vulnerability Database (NVD) (2023); (NVD) (2023), and HTTP/2 slow-rate attacks Tripathi (2022); Tripathi and Hubballi (2018). These studies often suggest anomaly detection methods that are generally less effective against HTTP/2 threats targeting web environments. For example, Praseed and Thilagam (2019) shows that the HTTP/2 stream multiplexing feature can be exploited to launch multiple streams over the same connection, overwhelming the server or causing DoS. In HTTP/2 slow-rate DoS attacks, attackers send multiple specially crafted incomplete requests that occupy the server's connection queue space, preventing it from processing other requests Tripathi and Hubballi (2018). They can still exhaust server resources, leading to performance degradation and DoS. To detect such attacks, Tripathi and Hubballi (2018) proposes

using a Chi-square test to identify abnormal intervals of HTTP/2 traffic. However, its effectiveness varies with the attack rate and the chosen detection interval. The same authors later developed an event sequence analysis method, achieving high accuracy with minimal computational demands Tripathi (2022), only for this attack. Another method by Praseed and Thilagam (2021) focuses on identifying HTTP/2 multiplexed asymmetric DDoS attacks by contrasting the behavior of legitimate users and attackers. While effective for DDoS attacks, this approach fails to detect HTTP/2 slow-rate DoS attacks due to the minimal computational overhead and pattern mimicking by attackers. Both slow-rate and SMAs exploit various HTTP/2 parameters, leading to a DoS. One of the latest attacks related to HTTP/2 is the HTTP/2 rapid reset attack, identified as CVE-2023-44487 National Vulnerability Database (NVD) (2023). The mitigation for this attack involved bounding the number of simultaneously executing handler routines to a defined limit (SETTINGS_MAX_CONCURRENT_STREAMS=200), preventing server overload by queuing incoming requests until a current request is completed. If the queue becomes excessively long, the server terminates the connection. Hence, the discussion on HTTP/2 attacks in the aforementioned work was limited to a qualitative study of their applicability in 5G SBA without any implementation or test of their impact on 5G SBA.

Flow-based anomaly detection techniques are widely used in the literature. Autoencoder with flow-based features Salahuddin et al. (2021) and Convolutional Neural Network Hussain et al. (2020) were employed to detect DDoS attacks including HTTP attacks. These works do not consider a 5G environment nor account for an HTTP/2 dataset. Similar to Xie and Zhang (2012), we argue that flow-based features and application-layer features are inefficient in detecting HTTP/2 attacks as HTTP/2 may not exhibit malicious activities when their network traffic is observed from the network or transport layers.

B. Transformer-based Anomaly Detection

Anomaly detection plays a vital role in diverse domains such as image processing, time series analysis, and network security. In recent years, transformer-based models have gained high attention due to their remarkable ability to capture intricate dependencies and extract meaningful representations from data Xu et al. (2021). Several papers have introduced pioneering anomaly detection methods that leverage transformer architectures to tackle this challenge Wen et al. (2022).

One notable method is HaloAE Mathian et al. (2022), which proposed an autoencoder architecture combined with transformers for anomaly detection. By reconstructing features, HaloAE delivered competitive results, effectively capturing intricate relationships within the data and enabling accurate identification and localization of anomalies. However, TransAnomaly Zhang, Xia, Yan, and Liu (2021) took a different approach by combining a Variational AutoEncoder with a transformer for unsupervised anomaly detection in multivariate time series data. Leveraging the transformer's ability to capture temporal dependencies at various scales, TransAnomaly achieved superior performance in anomaly detection. This approach addressed the limitations of traditional autoregression methods and enhanced the accuracy of anomaly detection.

In the domain of time series data, AnomalyTrans Xu et al. (2021) emerged as a method to enhance anomaly detection. This model simultaneously models prior and series associations by integrating transformers and Gaussian prior-association, resulting in a more distinguishable association discrepancy. The developed model was used to detect anomalies on data associated with service monitoring, space and earth exploration, and water treatment. Another work, Dilated Convolutional Transformer-based Generative Adversarial Networks Li, Peng, Zhang, Li, and Wen (2021) presented a novel approach for time series anomaly detection by fusing transformers with dilated convolutional neural networks. Through a GAN-based model, DCT-GAN simultaneously accomplished reconstruction and anomaly

detection, capturing temporal information at different scales. This fusion of transformer and CNN architectures provided a comprehensive framework for accurate anomaly detection in time series data. The authors of Adformer Zeng et al. (2023), introduced as a two-stage adversarial transformer model, focus on detecting anomalies in multidimensional time series data within the IoT context. By amplifying reconstruction error, capturing short-term trends, and leveraging prior knowledge, Adformer significantly enhanced anomaly detection.

Transformer-based anomaly detection techniques have revolutionized anomaly detection techniques by capturing complex dependencies, enabling accurate detection, and precise localization of anomalies. The aforementioned works Xu et al. (2021); Zeng et al. (2023) presented innovative approaches to tackle anomaly detection challenges across various domains using transformers. Nonetheless, their efficiency for HTTP/2 attack detection in a 5G environment where time and dependencies between HTTP/2 messages reflect 5G services and procedures, was not explored.

C. Ensemble Learning

Ensemble learning Rincy and Gupta (2020) is a powerful machine learning technique that combines multiple models to improve prediction accuracy and robustness. It is particularly suited for network anomaly detection's complex and dynamic nature. The core principle behind ensemble learning is to leverage the complementary strengths of diverse models to achieve better overall performance. Standard ensemble methods include bagging, where models are trained independently on bootstrapped datasets, and their predictions are aggregated to reduce variance and enhance stability; boosting, which involves sequentially training models to focus on correcting the errors of their predecessors, as seen in AdaBoost and Gradient Boosting; and stacking, where predictions from multiple base models are

combined through a meta-model that learns the optimal way to integrate individual outputs for superior accuracy Sagi and Rokach (2018). These techniques have effectively addressed the complexity and variety of security challenges, particularly in evolving 5G network environments.

In the context of 5G security, Tian, Patil, Gurusamy, and McCloud (2023) employed ensemble learning by leveraging bidirectional LSTM networks to model NF sequences, identifying control plane threats such as reconnaissance and flooding. However, the sensitivity of this method to sequence length variations limited its robustness in the dynamic 5G environment. Complementary approaches such as Haider, Waqas, Hanif, Alasmary, and Qaisar (2023); Saha, Priyoti, Sharma, and Haque (2022) used ensemble learning with classifiers like decision trees, random forests, and support vector machines for anomaly detection and network load prediction, achieving notable accuracy. While effective, these solutions relied on supervised learning, limiting their ability to detect new and zero-day attacks. In broader network contexts, studies like Liao, Teo, Kundu, and Truong-Huu (2021) and Aljebreen et al. (2023) applied ensemble frameworks to respectively detect anomalies launched from IoT devices and others targeting SDN. While these unsupervised methods exhibited good detection performance, they were less suited to HTTP/2-5G-specific, as they relied on non-5G datasets. To the best of our knowledge, ensemble learning has not been explored for HTTP/2 anomaly detection in 5G.

2.3 5G Testbed

In order to emulate normal or attack behavior in the 5G network, we need a 5G testbed that supports HTTP/2 and allows us to connect multiple UEs to 5GC. To this end, we use free5GC Free5GC (2021a), an open-source 5G testbed, and UERANSIM aligungr (2021), which provides a UE/RAN emulator, as shown in Figure 2.3.

We created two versions of the 5G testbed as follows:

- Version 1 : We deploy free5GC and UERANSIM on a Virtual Machine (VM) running on top of OpenStack OpenStack (2021) where the VM runs Ubuntu 20.04-Focal with 4 vCPUs and 4GB RAM. We install the docker-compose version of the free5GC called free5GC-compose, version 3.0.5 Free5GC (2021b), in which the 5G NFs are deployed on different containers in the same VM.

- Version 2 : We deploy free5GC and UERANSIM on two separate VMs running on OpenStack OpenStack (2021). The VMs are equipped with Ubuntu 20.04-Focal, 8 virtual Central Processing Units (CPU), and 64 GB of RAM. We use free5GC docker-compose version 3.4.0 Free5GC (2021b), which runs NFs in separate containers on the same VM.

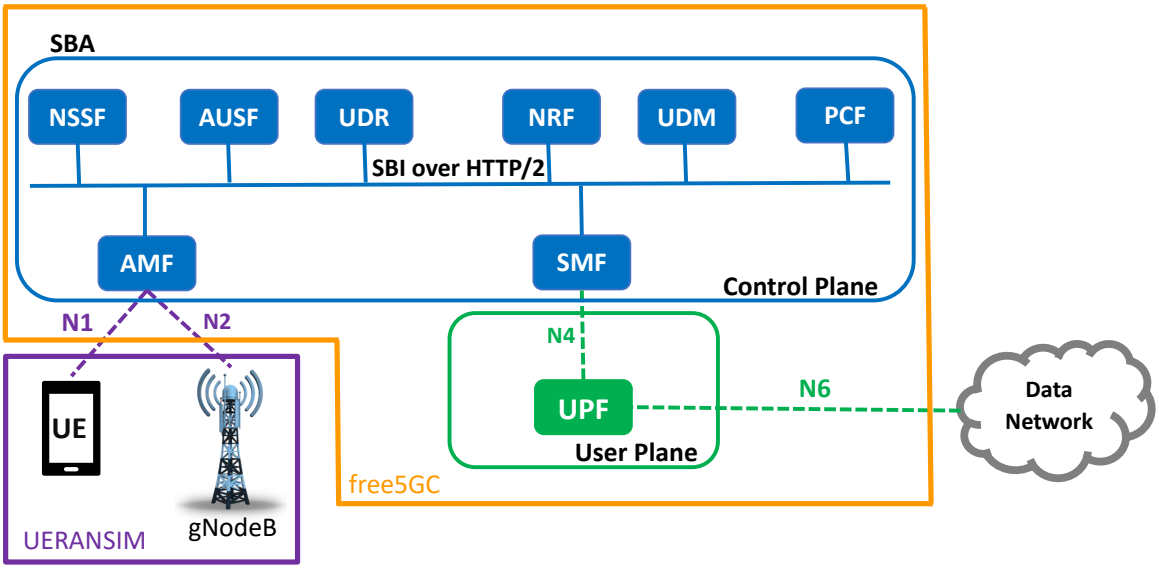


Figure 2.3: Our 5G testbed based free5GC and UERANSIM aligungr (2021); Free5GC (2021a); TS.23.501 (2024)

The 5G testbed allows us to emulate different 5G procedures such as registering an UE, deregistering an UE, releasing a PDU session, and service request procedures. To be able to emulate normal network behavior with a relatively significant number of UEs and collect HTTP/2 data from the 5G testbed, we performed the following changes to the

default free5GC implementation:

- HTTP/2 protocol: free5GC initially supported the H2c protocol, the first version of the HTTP/2 protocol that does not support the HTTP/2 stream multiplexing feature. We develop an HTTP/2 server code that we deploy over each NF to support the HTTP/2 stream multiplexing feature.

- Total number of UEs supported by free5GC: We attempted to connect more than 10 UEs to the 5G testbed. However, due to an issue with the free5GC code, we were unable to register more than 10 UEs. To overcome the free5GC limitation that allows a emulation of a maximum of 10 simultaneous UE connections, we modify the free5GC UPF buffer size. This enables us to extend the number of UE running simultaneously.

We built our 5G testbed to be able to emulate normal and HTTP/2 attacks behavior and show their impact in a 5G environment.

Chapter 3

A Security Assessment of HTTP/2 Usage in 5G Service Based Architecture

In this chapter, we discuss different security features introduced by 5G SBA and explore these security challenges and their solutions in this new architecture. We carefully examine HTTP/2 features, standard and custom headers and discuss their security implications in 5G SBA. We comment on the applicability of some known HTTP/2 attacks in 5G SBA in light of the standardized APIs and discuss the security opportunities and research directions brought by this protocol and its related technologies.

3.1 5G Service Based Architecture (SBA)

3.1.1 Overview

5G networks revolutionized the telecommunication architecture by adopting a cloud-native, service-driven deployment promoting enhanced network operational efficiencies. The 5G SBA (Figure 3.1) enables a granular design and delivery of 5G network functionality through a decoupling of UP and CP, hence, providing independent scalability and

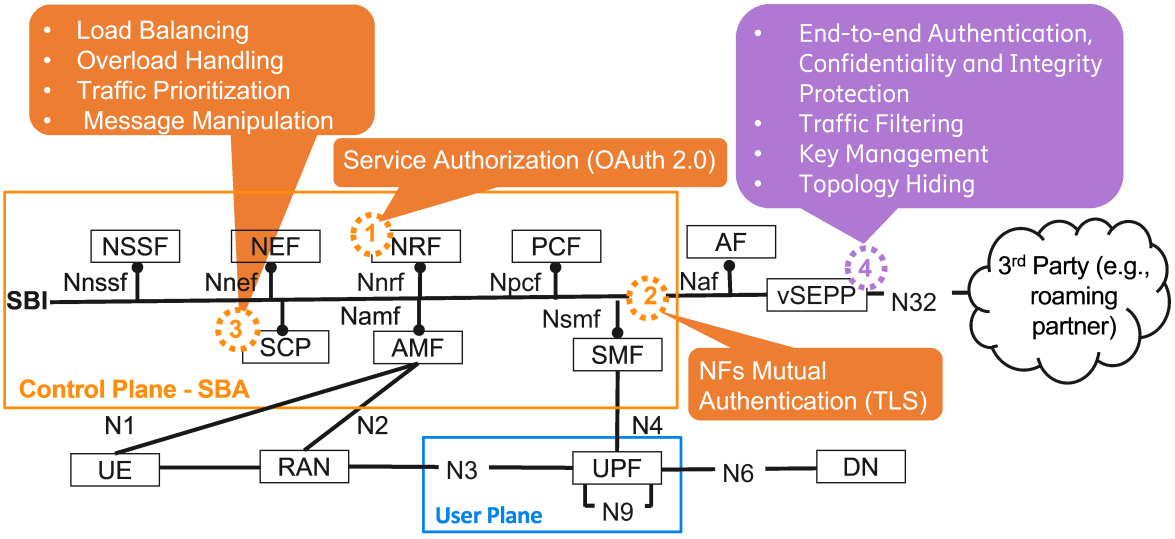


Figure 3.1: Security features implemented in 5G SBA GSMA (2021); TS.23.501 (2024)

flexible deployments Alliance (2018); TS.23.501 (2024). The UP and CP consist of multiple interconnected NFs, each providing a set of "*services*". Examples of such services include service registration, authorization and discovery Alliance (2018). The 5G CP is defined by a SBA. The interactions between the CP NFs are enabled by a service-based representation in which the SBIs can be easily extended without the need to introduce new reference points.

To enable the communication between the 5G SBA NFs (also referred to as "*5G signaling*"), the 3GPP selected the HTTP/2 protocol with JSON as the application layer serialization protocol, which runs over TCP at the transport layer 3GPP TS.29.500 (2024). For added security, the NFs shall support TLS 1.2 and TLS 1.3 3GPP TS.29.500 (2024); Christine Jost (2020). In addition, Restful API is used to invoke 5G services Alliance (2018).

Signaling through direct communication between 5G NFs is enabled by HTTP/2 while being facilitated by the NRF (Figure 3.1). Signaling allows NFs to consume services provided by their peers. In fact, a NFp will first register itself to the NRF. This enables the NRF to maintain a NF profile that includes the available NF instances and their services.

A NFc can then discover the available NF instances and services by consulting the NRF. Once discovered, a NFc can directly consume authorized services through APIs exposed by a NFp TS.23.501 (2024). These APIs are standardized by 3GPP and can be either *Request-Response* or *Subscribe-Notify* TS.23.501 (2024).

Signaling through indirect communication between the NFs consumers and producers is also possible through the *Service Communication Proxy (SCP)* NF (Figure 3.1) 3GPP TS.29.500 (2024). The SCP can route the requests and responses of service consumers and producers respectively, and offload the service registration and discovery requests to the NRF. Note that the SCP also provides load balancing, overload handling, traffic prioritization and message manipulation functionalities 3GPP TS.29.500 (2024); Shetty, Jangam, and Simlai (2021).

3.1.2 5G SBA Security

5G SBA leverages cloud-native principles where NFs are created and destroyed dynamically and communicate through an SBI message bus using different APIs. These NFs should be authenticated and their communication needs to be protected to prevent unauthorized access to their services. 3GPP identified two main security mechanisms:

A. Mutual authentication and transport security

They are enforced through TLS between SBA NFs and between NF-NRF during service discovery and registration to mitigate against message spoofing, tampering, repudiation and information disclosure Christine Jost (2020); TS.33.501 (2025b).

B. Authorization of the requests

Access authorization of NFcs to services provided by NFps prevents privilege escalation. It follows a token-based authorization through the NRF using OAuth 2.0 IETF (2022);

TS.33.501 (2025b). OAuth 2.0 is an authorization framework that enables a third-party application to obtain limited access to an HTTP service on its behalf or on behalf of a resource owner IETF (2022). In 5G SBA, an access token to a certain service is generated by the NRF (OAuth 2.0 authorization server) following a request of a NFc (i.e., OAuth 2.0 client) to access a service of a NFp (i.e., OAuth 2.0 resource server) TS.33.501 (2025b). The token is granted based on authorization rules which can be provided by the NFp during its registration at the NRF and after the mutual authentication between the NRF and the NFc (using TLS) Christine Jost (2020).

Authorization and authentication are applied in non-roaming and in roaming scenarios. Nonetheless, to better protect the 5G network from unauthorized access and attacks that can be performed by outsiders (e.g., roaming partners, etc.), a *Security Edge Protection Proxy (SEPP)* (Figure 3.1) has been introduced. SEPP acts as a security gateway on the interconnections between roaming partners. It provides application-layer security between NFs associated with roaming partners to enable their secure communication. SEPP functionalities include traffic filtering, end-to-end authentication, confidentiality and integrity protection via signatures and encryption of HTTP/2 messages. SEPP is also responsible of key management mechanisms used to perform the security capability procedures. Finally, the SEPP offers topology hiding capability along with prevention of bidding down attacks TS.33.501 (2025b).

3.2 Implications of HTTP/2 Features on 5G SBA

HTTP/2 introduces multiple features that we explore hereafter and discuss the security impact of their possible exploitation by attackers in 5G SBA.

3.2.1 Streams Multiplexing

HTTP/2 streams multiplexing feature allows carrying multiple streams over a single TCP connection IETF (2015), thus improving services' latency. In fact, an HTTP/2 client/ server can limit the maximum number of concurrent streams over a single TCP connection with its peers using the HTTP/2 `SETTINGS_MAX_CONCURRENT_STREAMS` setting. While IETF recommends a minimum value of 100 streams for this setting to benefit from the stream multiplexing feature, it does not provide any recommendations on its upper limit which can go up to 2,147,483,647 streams IETF (2015). This allows attackers to exploit the stream multiplexing feature through sending as many as 2,147,483,647 streams of computationally expensive requests (i.e., APIs) towards the NFp and replicate it over multiple TCP connections to scale the attack and cause a DoS Imperva (2016). Hence, network operators should carefully configure the `SETTINGS_MAX_CONCURRENT_STREA MS` for their 5G NFs to limit such attack.

3.2.2 Flow Control

The flow control feature is introduced to prevent streams on the same TCP connection from interfering with each others IETF (2015). Flow control determines the size of the data the sender is permitted to send to the receiver using many parameters such as the `WINDOW_UPDATE` frame, and the SETTINGS frame IETF (2015). The `WINDOW_UPDATE` frame is used by the receiver to inform the sender how much data it is willing to receive on each stream IETF (2015). The flexibility provided by this feature can be misused by a malicious receiver (i.e., NFc in 5G) to influence the streams processing at the NFp into intensive resource consumption, thus causing a slow-read DoS attack on the NFp Hu et al. (2018). In fact, in such attack, a NFc imposes very small data transmission using the `WINDOW_UPDATE` frame on the NFp, thus keeping the NFp resources busy to complete its request. However, a possible preventive measure that can be taken in 5G networks, is to set

a processing timeout limit for requests on each NFp based on the vertical industry the NFp is serving.

3.2.3 Stream Dependency and Prioritization

HTTP/2 carries a dependency-based prioritization feature that allows a client to assign a priority for each stream through a PRIORITY frame. Stream priority determines the order at which the client wants its streams to be processed IETF (2015). A client can also specify dependency between streams that will be expressed in a dependency tree at the server. It can assign weights to dependent streams to dictate to the server the relative proportion of available resources that it has to allocate them IETF (2015). The dependency-based prioritization feature was introduced with the intention of improving user experience. However, since no limit was set in RFC 7540 IETF (2015) on the size of the dependency tree, a NFp which naively trusts a NFc may be deceived to build a dependency tree that will consume its memory and CPU, thus causing a DoS on the NFp Hu et al. (2018); Imperva (2016). The exploitation of this feature can be partially limited in 5G SBA by configuring the size of dependency tree at NFp for each TCP connection.

3.2.4 Header Compression

HTTP/2 introduces header compression through the HPACK protocol to reduce the request size by eliminating redundant header fields across multiplexed streams, which leads to lower bandwidth utilization IETF (2015). HTTP/2 request and response header metadata are compressed using HPACK through: (1) encoding the transmitted header fields to reduce their individual transfer size; (2) maintaining an HPACK static table that holds a predefined static list of headers; (3) updating and maintaining an HPACK dynamic table that holds a dynamic list of headers IETF (2015). It is used as a cache for each connection direction separately. The sender can signal to the receiver what values to insert in the dynamic table,

hence, it can refer to their locations in subsequent streams. The size of the dynamic table is restricted to limit the memory requirement on the decoder side, however, the size of the header value field inside this table is not constrained IETF (2015); Imperva (2016). The lack of restriction on the size of the header value creates a vulnerability that can be exploited to launch an HPACK Bomb attack Imperva (2016). An attacker can generate a first stream with a large header (i.e., of size equal to the dynamic table of its peer), then open new streams over the same connection that reference the same header. Decompressing the large header for each subsequent stream causes memory exhaustion, and hence a DoS on the server Imperva (2016). Limiting the header value in the dynamic table can potentially prevent the HPACK Bomb attack.

3.2.5 Server Push

The server push uses the PUSH_PROMISE frame to enable the server to send inline resources to the client without an explicit request for each resource IETF (2015). This feature improves the client's experience by reducing the load time and workload, however, it places the burden on the server. The server push feature, combined with the multiplexing feature can be misused to launch a DDoS attack against an HTTP/2 server. A malicious client can force a server to serve a high number of simultaneous requests, each of which has multiple associated inline resources that the server needs to push Praseed and Thilagam (2019). This leads to a flooding attack which affects the server egress bandwidth and nearby routers, thus resulting in a DoS attack at the network layer as well Praseed and Thilagam (2019). The server push feature may not always be advantageous as it can use an excess of bandwidth to push unneeded assets. Mobile operators must carefully assess the need to enable this feature in their 5G networks, as bandwidth and connection stability are crucial to meet the QoS requirements of their services.

Table 3.1: HTTP/2 features and their security implications.

HTTP/2 Feature	Threat Model	Implications	Attack Type	Possible Countermeasures
Streams Multiplexing	Flooding a NFp by exploiting `SETTINGS_MAX_CONCURRENT_STREAMS`	NFp overloading	DoS Attack	Limiting `SETTINGS_MAX_CONCURRENT_STREAMS` Adding SCP with 3GPP custom HTTP header
Flow Control	Imposing very small data transmission using the `WINDOW_UPDATE` frame on the NFp	Server resources depletion while completing the request processing	Slow-read DoS attack	Setting a processing timeout limit for requests per HTTP/2 connection
Stream Dependency and Prioritization	Building a large dependency tree through enforcing many streams dependencies	NFp memory and CPU consumption	DoS attack	Limiting the size of the dependency tree for each HTTP/2 connection
Header Compression	Exploiting the lack of restriction on the size of the header value in the HPACK dynamic table and reference a header of large value multiple times	Memory Exhaustion	HPACK Bomb attack, DoS attack	Limiting the header value in the HPACK dynamic table
Server Push	Flooding a NFp with requests associated with inline resources that it needs to push	Egress bandwidth overconsumption	DoS attack	Carefully assess the need for this feature in 5G networks

3.2.6 Discussion

5G networks implement tighter security than the general web, which reduces the likelihood of HTTP/2 attacks (Table 3.1). Nonetheless, some of these HTTP/2 attacks are likely to apply to 5G networks as attackers can exploit them through vulnerabilities related to virtualization technologies ENISA (2021). In fact, the move of mobile network operators to the public cloud increases the attack surface through virtualization vulnerabilities (e.g., CVE-2016-5195, CVE-2019-5736). Similarly, virtualization vulnerabilities and misconfiguration can be exploited by attackers to breach the isolation between 5G network slices, for example, through a shared NF AdaptiveMobile (2021). In such a scenario, HTTP/2 attacks on the shared NF from one slice can impact the functionality of the other slice.

In addition, HTTP/2 attacks can be initiated from malicious roaming partners and remain undetected by the filtering techniques at the SEPP GSMA (2021). Although they take a new form in HTTP/2, HTTP/2 multiplexing and slow-read attacks common in the Internet may occur now in 5G networks. In contrast, we envision stream dependency and prioritization based attacks along with server push and HPACK bomb attacks are less likely to happen in 5G networks as they are highly related to the mobile operators implementation and configuration. For instance, an operator may disable server push functionality, thus preventing its related attack. To the best of our knowledge, the usage of server push has been left by 3GPP to the mobile operator choice. Finally, with the risk of misconfiguration of HTTP/2 settings and its related attacks, intelligent anomaly detection solutions that can detect HTTP/2 attacks to enable automated mitigation measures are needed.

3.3 Implications of HTTP/2 standard and custom headers on 5G SBA

HTTP/2 message header is composed of multiple standard and custom header fields that we elucidate and discuss their role in 5G SBA security.

3.3.1 Standard HTTP/2 Headers

The standard HTTP/2 header fields are used in both requests and responses. The request sent to the HTTP/2 server includes a list of header fields that identify the client. Figure 3.2 includes some of these standard headers: *accept-encoding* specifies the used data encoding; *accept* determines the content type the client is able to handle; *authority* defines the Fully Qualified Domain Name (FQDN) or IP address of the target Uniform Resource Identifier (URI) (i.e., target NF service); *path* includes the path and query parts of the target URI (i.e., API URI); *scheme* declares the version of HTTP used (e.g., http or https) 3GPP TS.29.500

(2024); IETF (2015). *User-agent* header key defines the HTTP/2 client. An HTTP/2 response carries HTTP header response fields (Figure 3.2) such as: *status* which carries the HTTP status code, *content-type* specifies the type of the content returned by the server, *content-length* determines the length of the content in bytes, and the originating date of the response presented in the *date* header 3GPP TS.29.500 (2024); IETF (2015).

```
#HTTP Request Header
accept-encoding: gzip
accept: application/json
:authority: amf.5g.org:8000
:method: POST
:path:
    /namf-comm/v1/ue-contexts/{ueId}/n1-n2-messages
:version: HTTP/2.0
:scheme: https
user-agent: SMF

#HTTP Response Header
:status: 200 OK
content-type: application/json
content-length: 5613
date: Mon, 14 March 2022 09:44:16 GMT
```

Figure 3.2: HTTP/2 request and response headers.

Furthermore, other HTTP standard header fields such as *Authorization* in the request and *WWW-Authenticate* in the response are used to mitigate multiple attacks on 5G NFs that could originate from a third party connection (e.g., roaming partner). For example, the *Authorization* header holds the OAuth 2.0 access token that the NFp should validate (i.e., validate the token, its expiration date, and access scope) before granting access to the requested resource 3GPP TS.29.500 (2024). In case the OAuth 2.0 access token is deemed invalid by the NFp (i.e., expired token, or the required scopes to invoke the requested service operation are not covered by the token); the NFp rejects the API request. The NFp will use the *WWW-Authenticate* header to determine the reason behind the rejection (i.e., invalid token, insufficient scope) in its error attribute 3GPP TS.29.500 (2024); IETF (2022).

3.3.2 Custom HTTP/2 Headers

3GPP introduced HTTP/2 custom headers dedicated to 5G SBA. Some of these custom headers are defined to enable load and overload control as they allow sharing of NFs load information 3GPP TS.29.500 (2024). Hereafter, we discuss the importance of these custom headers on 5G SBA security.

A. 3gpp-Sbi-Lci

3gpp-Sbi-Lci enables a NFp to signal its Load Control Information (LCI) to a NFc either directly or through the NRF during service discovery. This enables the NFc to decide whether or not to select a different NFp, hence, enabling a better load balancing in the network. Figure 3.3 represents a 3gpp-Sbi-Lci custom header, generated on specific date/time defined in *Timestamp*, by a NFp, to signal its load level through the *Load-Metric* to a SCP instance (i.e., SCP1 specified in *SCP-FQDN*) 3GPP TS.29.500 (2024).

```
3gpp-Sbi-Lci: Timestamp: "Tue, 04 Feb 2020 08:49:37
    GMT"; Load-Metric: 25%; SCP-FQDN:
    scp1.example.com
```

Figure 3.3: LCI for SCP 3GPP TS.29.500 (2024).

B. 3gpp-Sbi-Oci

A NFp/NFc uses the *3gpp-Sbi-Oci* custom header to signal its Overload Control Information (OCI) to its peer. Through this header, the overloaded NF instructs its peer to throttle the service/notification requests, in an attempt to reduce its signaling load 3GPP TS.29.500 (2024). Figure 3.4 depicts a *3gpp-Sbi-Oci* header sent by a NFp, identified by its instance ID (i.e., *NF-Instance*), asking a NFc to throttle 50% of its requests as determined in *Overload-Reduction-Metric*. Note that an *Overload-Reduction-Metric* of "0" indicates that the sender is not overloaded. The *3gpp-Sbi-Oci* also includes the *Timestamp* indicating

the time at which it was generated and its validity period identified by *Period-of-Validity* 3GPP TS.29.500 (2024).

```
3gpp-Sbi-Oci: Timestamp: "Tue, 29 Mar 2021 08:49:37
    GMT"; Period-of-Validity: 75s;
    Overload-Reduction-Metric: 50%; NF-Instance:
    54804518-4191-46b3-955c-ac631f953ed8
```

Figure 3.4: OCI for a NF Instance 3GPP TS.29.500 (2024).

C. 3gpp-Sbi-Message-Priority

In contrast to the PRIORITY frame used to determine stream (i.e., request and response) priority at the connection level, 3GPP introduced the *3gpp-Sbi-Message-Priority* to provide the flexibility of assigning a priority for the response that differs of the one assigned to its corresponding request 3GPP TS.29.500 (2024); IETF (2015). The primary usage of SBI Message Priority (SMP) is to assist NFp/NFc/proxies when making throttling decision related to an overload control or when routing messages through proxies 3GPP TS.29.500 (2024). For instance, a server may process higher-priority messages first, however, this may block lower-priority messages from ever being handled. In 5G SBA, this will result in the messages being retried, and in more traffic than the network usually handles without the use of the SMP mechanism.

3.3.3 Security Implications

HTTP/2 standard and custom headers play a critical role in security enforcement. HTTP/2 standard headers include API information and handle authentication and service authorization in 5G, thus preventing illegal service access. In contrast, 3GPP custom headers prevent DoS and DDoS attacks by enabling load balancing on NFs through *3gpp-Sbi-Lci*, and overload handling using *3gpp-Sbi-Oci* while staying compliant with the message priority defined in *3gpp-Sbi-Message-Priority*. However, *3gpp-Sbi-Message-Priority* can be

abused and result in starving low-priority messages. This unwanted starving needs to be correctly handled by following 3GPP recommendations on the usage of this header and by limiting the number of higher-priority messages in comparison to lower-priority ones 3GPP TS.29.500 (2024). Similarly, *3gpp-Sbi-Lci* and *3gpp-Sbi-Oci* can be abused by attackers to trick the network into assuming that a certain NF is (over)loaded by forging the *Overload-Reduction-Metric* in OCI and *Load-Metric* in LCI. This may trigger unneeded scaling of the victim NF, which may lead to over-provisioning and, hence, incur revenue losses for the operator.

3.4 Security challenges and opportunities

In the following, we discuss existing security challenges and shed light on possible security opportunities and research directions that can play a critical role in addressing them Figure 3.5.

3.4.1 Broken Service Access Control

The use of token-based authorization through OAuth 2.0 exposes the 5G network to a token tampering attack, allowing attackers to access the services of another NF within the same or different Public Land Mobile Network (PLMN). It also enables them to launch a DoS attack on the NFc by replacing the granted service (i.e., API) of the NFp in the request with an unavailable one ENISA (2021). The risk of gaining unauthorized service access through the NF-NRF interface is also possible and can result in disclosing sensitive information of a PLMN GSMA (2021). A holistic distributed attack detection and network monitoring framework is intrinsic to reveal unauthorized access and alert NFs of tampered tokens that need to be revoked and malicious requests that should be rejected. Further, with

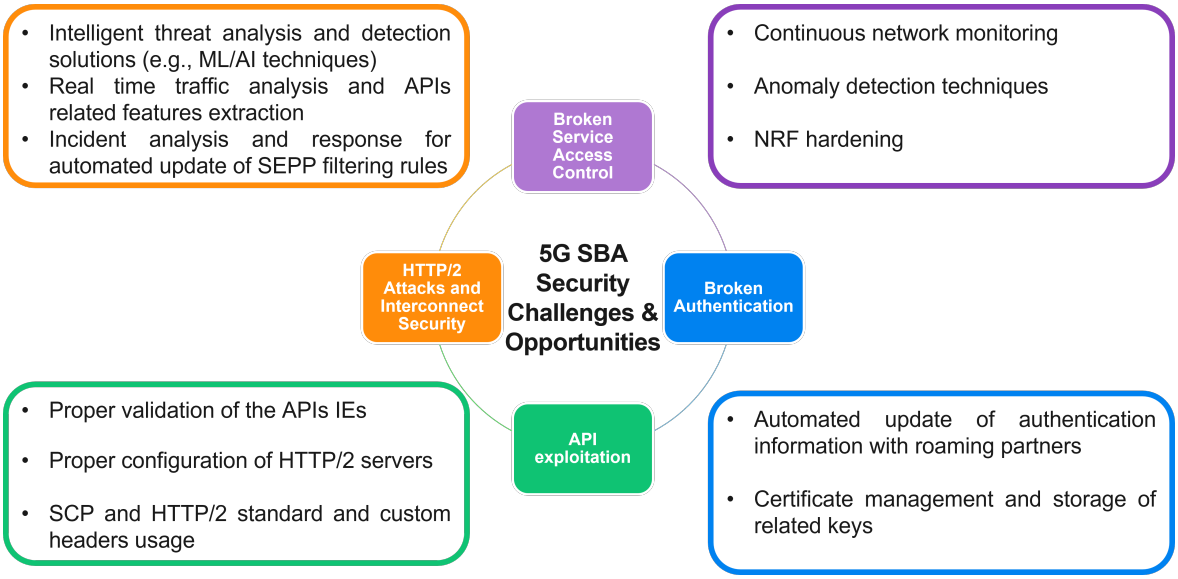


Figure 3.5: 5G SBA security challenges and opportunities.

the large number of roaming partners that an operator can have, misconfiguration of authorization rules is possible GSMA (2021). This requires standard contracts and authorization templates to lighten the configuration burden.

3.4.2 Broken Authentication

The usage of TLS for SBA protection at the network and transport layer and service authorization, respectively, rely on Public-Key Infrastructure (PKI) (i.e., X.509 certificate, public/private keys) Christine Jost (2020); TS.33.501 (2025b). In a non-roaming scenario, there is a risk of fraudulent certificates and compromise of private keys. In contrast, in a roaming scenario, the roaming database (IR.21) may contain outdated information and revoked certificates. This can result in broken authentication, which can lead to compromising JSON web token (i.e., used between SEPPs of roaming partners), hence granting illegal network access GSMA (2021). Thus, automated certificate management and storage of related keys to cope with the dynamism of a 5G cloud-native environment are research questions yet to be explored Christine Jost (2020).

3.4.3 API Exploitation

The reliance of 5G SBA on APIs extends the 5G attack surface to vulnerabilities associated with their exploitation. APIs are exposed to all endpoints within the same PLMN or with roaming partners through the SEPP.

DoS attacks can be launched by exploiting the resources an API can consume if no limits are imposed on the size or number of those resources GSMA (2021). Attackers can exploit the HTTP/2 multiplexing feature to overload the NFp with requests that exploit APIs requiring heavy server resource consumption. The attack can be further intensified by a slow-read attack during which the attacker manipulates the flow-control information to keep the NFp resources allocated for those requests for a longer period, hence facilitating the DoS. Therefore, a proper configuration of HTTP/2 settings such as SETTINGS_MAX_CONCURRENT_STREAMS to limit DoS attacks is also needed. For instance, a network operator can limit the number of maximum concurrent streams that a server allows per connection. This will make a DoS attack costly to the attacker, who will need to allocate more resources to establish multiple TCP connections with the server to exhaust it. Further, HTTP/2 with usage of SCP in a 5G network offers many opportunities for early detection and mitigation of a server overload and DoS attacks through the use of HTTP/2 custom headers standardized by 3GPP for 5G SBA (Section 3.3.2).

3.4.4 HTTP/2 Attacks and Interconnect Security

HTTP/2 attacks can be left unnoticed by the SEPP at the interconnect network on the N32 interface (Figure 3.1), if they originated from malicious roaming partners GSMA (2021); TS.33.501 (2025b). HTTP/2 filtering at the SEPP aims at blocking 5G interconnect messages based on certain criteria (i.e., URI, specific IEs, etc.) to prevent malicious roaming partners from extending their services beyond the roaming agreement. Nonetheless, filtering techniques do not prevent attacks such as HTTP/2 multiplexing attacks in which

malicious roaming partners can request legitimate services from a specific NF.

To counter the above threats, intelligent threat analysis and detection solutions that overcome the limitations of filtering mechanisms and which leverage ML and Artificial Intelligence (AI) techniques are needed. They can learn traffic patterns from data collected at filtering nodes such as the SEPP, 5G NFs, and other monitoring logs collected from the 5G SBA. Real-time or near-real-time traffic analysis and features extraction at network and application layers while accounting for API calls, IEs, HTTP/2 standards, and custom headers are yet to be explored as indicators of compromise that may enhance the detection accuracy of these ML/AI models that yet to be developed. Further, ML/AI solutions need to be complemented with effective incident analysis and response and used to automatically update filtering rules at the SEPP and the 5G SBA firewalls. The proposed security controls should be designed to complement each other in an automated holistic security orchestration and management framework designed and adapted for 5G networks.

Chapter 4

5GShield: HTTP/2 Anomaly Detection in 5G Service-Based Architecture

In this chapter, we propose 5GShield, a novel application-layer anomaly detection solution that uses neural networks, namely, Autoencoder, for anomaly detection. To evaluate our approach, we deploy a 5G testbed, emulate the HTTP/2 SMA, and collect HTTP/2 data. Our experimental results show that 5GShield can detect HTTP/2 SMA with an F1-score of 0.992, outperforming a flow-based anomaly detection solution that exhibits an F1-score of 0.78. 5GShield shows the efficiency of 5G-specific application-layer features in exposing HTTP/2 attacks that can go undetected at the network layer.

4.1 Threat Model

While accounting for the secure design of 5G SBA, we detail herein, the list of assumptions that allow launching the HTTP/2 SMA from a compromised NFc towards an NFp in a 5G network, and describe its threat model.

4.1.1 Assumptions

(1) *Attacker compromises an NFc:* Many standardization documents discuss threats brought by NFV and virtualization technologies (e.g., container, virtual machines, etc.) to telecommunication networks and 5G in particular ETSI (2020). The adoption of hyper-scale cloud by mobile operators extends the attack surface of their networks and makes their virtual NFs vulnerable ETSI (2020). An attacker can compromise 5G NFs deployed on docker containers in the cloud, by exploiting docker vulnerabilities to perform container escape (i.e., CVE-2016-5195 (NVD) (2019), and CVE-2019-5736 (NVD) (2016)) Madi et al. (2021). Attackers can take advantage of a breach of isolation between 5G network slices that share one or multiple NFs AdaptiveMobile (2021).

(2) *NFc successfully authenticates with the NFp:* We assume that if TLS is used, the compromised NFc can still authenticate with the NFp as the attacker has access to its public/private key pairs.

(3) *NFc is authorized to access NFp services:* We assume that the malicious NFc has already acquired OAuth 2.0 access tokens to the NFp services. These tokens are cached and can be reused by the attacker TS.33.501 (2025a). Alternatively, the malicious NFc can request new access tokens from the NRF given that it can successfully authenticate with it (i.e., assumption (2)). Vulnerabilities related to network slicing and service authorization, such as those mentioned in AdaptiveMobile (2021) can also be exploited to access the NFp services.

(4) *Attacker has access to UE information:* As some network services require exchanging UE information (e.g., Subscription Permanent Identifier (SUPI)) TS.123.502 (2025), we assume that the attacker can gain access to such information by monitoring NFc communications or even requesting such information from other NFs.

4.1.2 HTTP/2 Stream Multiplexing Attack in 5G SBA

Given the prior assumptions, we emulate the HTTP/2 SMA between an SMF acting as the malicious NFc and an AMF representing the targeted NFp. The choice of the AMF as the attacker target is related to the importance of the role it plays in providing UE authentication, authorization, and mobility management services TS.129.518 (2025). In addition, the AMF is exposed to external networks, which extend its attack surface and put it at risk Pell, Moschoyiannis, Panaousis, and Heartfield (2021). A DDoS attack against the AMF can significantly reduce the availability of 5G services and even cause network outages Pell et al. (2021). Without loss of generality, we consider the SMF as the compromised NFc by the attacker given that it is one of the major consumers of the AMF services TS.129.518 (2025). Thus, in this attack, we assume that the attacker, acting as the malicious SMF, requests the `Namf_Communication_N1N2MessageTransfer` API from an AMF. Note that this API is triggered between SMF and AMF in multiple 5G procedures such as UE registration, network-triggered service request, and UE-triggered service request, etc.TS.129.518 (2025). We leverage this API to perform the HTTP/2 SMA in two forms:

- *Stealthy HTTP/2 SMA:* consists of triggering different randomly selected 5G procedures for randomly selected UEs.
- *Non-stealthy HTTP/2 SMA:* consists of triggering the same 5G procedure simultaneously for the same subset of UEs.

In Figure 4.1, we illustrate the HTTP/2 SMA in four steps: (1) The attacker compromises an SMF via virtualization vulnerabilities. The SMF may or may not belong to a malicious roaming partner that has already been authenticated and authorized to access the AMF service(s). (2) The attacker (i.e., malicious SMF) establishes the first TCP

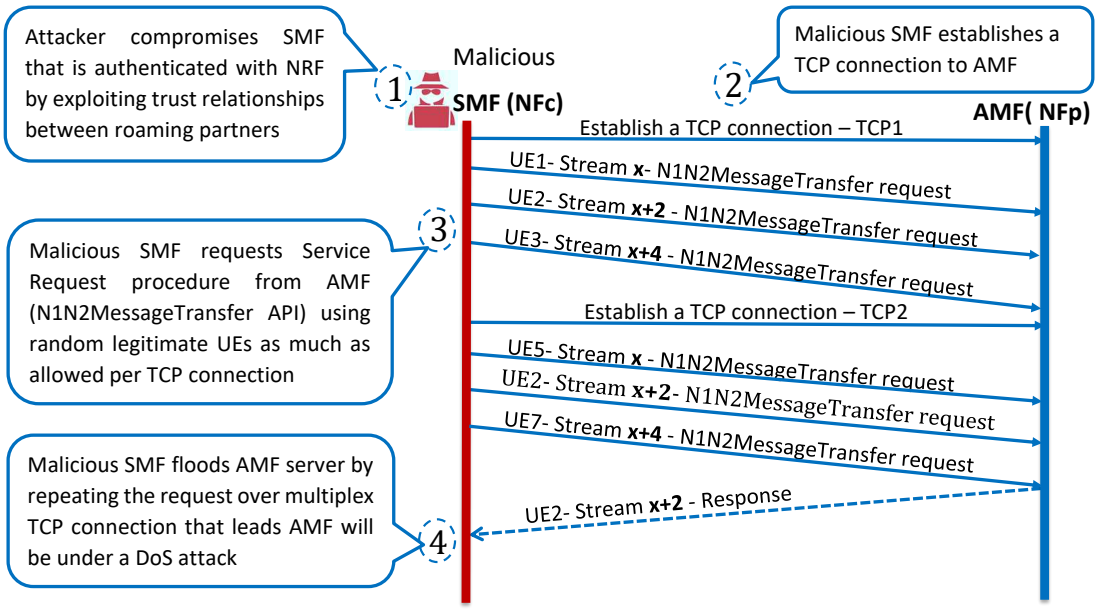


Figure 4.1: HTTP/2 stream multiplexing attack on AMF

connection with the AMF. (3) Then the malicious SMF initiates a service request procedure using `Namf_Communication_N1N2MessageTransfer` API by sending as many requests as the AMF allows per a single TCP connection using legitimate UEs information. Note that the number of streams (i.e., request-response) an endpoint (e.g., AMF) allows its peer to initiate on their established connection is specified by the HTTP/2 `SETTINGS_MAX_CONCURRENT_STREAMS` setting. (4) Given a sizable number of computationally expensive requests, the AMF becomes overloaded. The attacker can scale this attack by repeating it over multiple TCP connections, which causes a DoS attack at the AMF. Note that the default and maximum value of `SETTINGS_MAX_CONCURRENT_STREAMS` is 2 147 483 647, which makes the scaling of the attack easier IETF (2015). Finally, as the attacker used a subset of legitimate UEs information and requests, the detection of this application-layer attack becomes challenging.

4.2 Methodology - 5GShield Solution

In this section, we introduce the 5GShield solution (Figure 4.2), our novel and intelligent application-layer anomaly detection solution designed to detect HTTP/2 attacks including SMA.

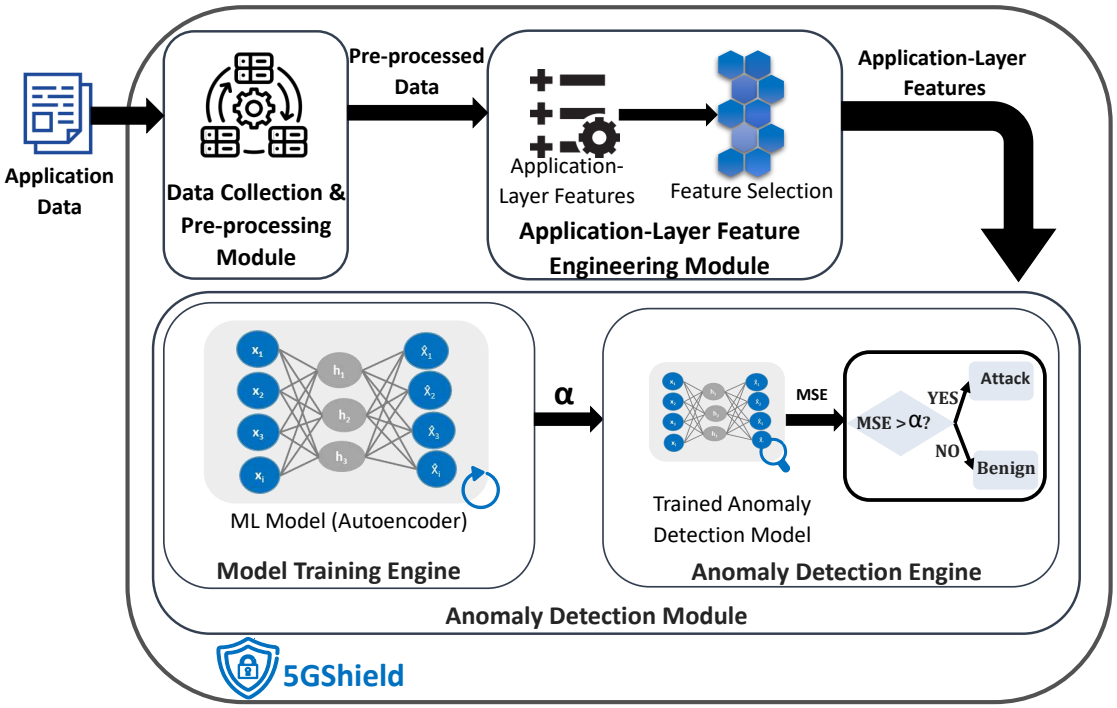


Figure 4.2: An overview of 5GShield solution and its modules

4.2.1 Data Collection and Pre-processing Module

The data collection and pre-processing module aims at collecting application-layer information and pre-process it for feature engineering and anomaly detection. This module collects the data provided by the application, that is the monitored NF that we aim at protecting (e.g., AMF). In 5G networks, application-layer information includes Performance Measurements (PM) counters that are standardized by 3GPP TS.28.552 (2024) and other counters that can be available by the NF application. PM counters convey how well an application is performing and can be used to determine system bottlenecks and fine-tune

the application performance. For example, AMF PM counters, standardized by 3GPP, present procedures related measurements such as registration, service request, UE configuration update procedures measurements among others such as mobility-related measurements TS.28.552 (2024). Thus, these counters permit profiling an NF normal behavior as they depict aggregated information pertaining to its provided services. They represent statistics of the communication patterns between the NF they represent and all the peer NFs it interacts with.

4.2.2 Feature Engineering Module

The feature engineering module performs feature extraction, normalization, and selection based on the data it receives from the data collection and pre-processing module. It extracts application-layer features belonging to two categories, mainly; 3GPP-based features depicting 3GPP PM counters for the targeted NF and HTTP/2-based features that reflect requests and responses between the targeted NF and its peer NFs. We note that 3GPP features represent, in majority, measurements related to the APIs (i.e., services) provided and received by the targeted NF. In contrast, the HTTP/2-based features are more general and can be accounted for any targeted NF while considering its peers. 3GPP-based and HTTP/2-based features capture the communication patterns between NFs through API calls statistics. This enables the successful detection of HTTP/2 attacks, including the SMA, as

Table 4.1: 3GPP and HTTP/2 application-layer features collected at the AMF

3GPP-AMF features	**HTTP/2-AMF features**
numberofAttemptedNetworkInitiatedServiceRequest	receivedRequestToAMF, sentRequestFromAMF
numberofSuccessfulNetworkInitiatedServiceRequest	receivedRequestToAMFperNRF, sentResponseFromAMFperNRF
numberofAttemptedUEInitiatedServiceRequest	receivedRequestToAMFperAUSF, sentResponseFromAMFperAUSF
numberofSuccessfulUEInitiatedServiceRequest	receivedRequestToAMFperNSSF, sentResponseFromAMFperNSSF
totalNumberofAttemptedServiceRequests	receivedRequestToAMFperPCF, sentResponseFromAMFperPCF
totalNumberofSuccessfulServiceRequests	receivedRequestToAMFperSMF, sentResponseFromAMFperSMF
	receivedRequestToAMFperUDM, sentResponseFromAMFperUDM
	receivedRequestToAMFDiscarded
	sentErrorResponseFromAMF, receivedErrorResponseToAMF
	totalSuccessfulRequest, totalUnsuccessfulReques

these attacks exhibit a deviation from the normal communication patterns between NFs.

Given that we consider securing the AMF as a proof of concept of 5GShield, we present in Table 4.1 the AMF features that we select. We distinguish the 3GPP-AMF features that are based on 3GPP guidelines TS.28.552 (2024), from which we choose the PM of AMF pertaining to the $Namf\_Communication\_N1N2MessageTransfer$ API (Section 4.1.2). Note that while other 3GPP-AMF features can be selected and relevant for the AMF profiling and HTTP/2 attack detection, we limit our selection to those related to the $Namf\_Communication\_N1N2MessageTransfer$ API that we leverage to launch the attack. Other 3GPP features were disregarded given their absence from our collected dataset. In addition to 3GPP-AMF features, we account for the HTTP/2-AMF features consisting of the number of sent/received, successful/unsuccessful requests per peer NF. Following the extracted features (Table 4.1), we perform feature normalization and then, we select the most relevant ones.

At the feature selection stage, we use the variance threshold scikit learn (2021) function to determine the most relevant variance value of the features. We choose this selection function, as it is well known for its usage in unsupervised models scikit learn (2021). The purpose of its usage is to help in removing features with minimal variations or those deemed as noise. As 5GShield is highly dependent on NF behavior patterns, the features selected to train the model must be accurately represented (i.e., have high variance) and provided to the anomaly detection module.

4.2.3 Anomaly Detection Module

The anomaly detection module consists of a model training engine and an anomaly detection engine (Figure 4.2). The model training engine trains the anomaly detection model and selects an appropriate threshold that assists in attack and benign data classification. The anomaly detection engine consists of the trained model and an attack classification

add-on that enables benign and attack data classification based on the output of the trained model and the selected threshold. Hence, for the anomaly detection model, we choose a feed-forward neural network, an AE, which is composed of one input layer, one or more hidden layers, and one output layer. In contrast to conventional methods (i.e., k-nearest neighbors), AE has been used for anomaly identification and has produced improved results Mirsky et al. (2018). Due to the limitation of data labeling, we choose unsupervised learning rather than supervised. We use the selected application-layer features as input to train an AE to learn the normal traffic behavior of the targeted NF (e.g., AMF). The AE identifies any malicious traffic that deviates from normal traffic as an attack. It learns a good lower-order mapping of the input data with the help of a reconstruction error loss function. The discovered lower-order mapping can then be employed to reconstruct the input data Salahuddin et al. (2021). Thus, when the AE is tested on data similar to that used to train it, it should provide a low reconstruction error. In contrast, if the test and training data differ significantly, the AE probably produces a high reconstruction error. As a result, we train the AE with benign data to efficiently detect any deviations as anomalies.

We choose the Mean Squared Error (MSE) to measure the model reconstruction error. MSE assesses the average squared difference between the input and the predicted values Mirsky et al. (2018). As model errors increase, the MSE values increase. The acceptable margin of difference between the input and the predicted value needs to be specified to determine if the input is benign or anomalous. Hence, to discriminate between benign and malicious data, there is a need for an efficient threshold selection α such that an $MSE \leq \alpha$ yields the data is benign while an $MSE > \alpha$ determines that the data is malicious. As such, a high threshold value would result in missing attacks (i.e., high false negatives, low recall), whereas a low threshold value can cause a lot of mis-classifications of benign data into malicious one, thus resulting in low precision. Both cases result in degraded performance of the AE. F1-score represents the harmonic mean between precision and recall and is

ideally equal to 1. Thus, given that it takes both false negatives and false positives into consideration, we select the threshold that maximizes the F1-score in this work.

4.3 Environment Setup

In this section, we present details on the 5G network emulation and discussions on the data pre-processing and the feature engineering that we perform on the data collected from our testbed.

4.3.1 Emulation Setup

Using Python 3.8, we implement our 5GShield solution, while our anomaly detection AE model leverages PyOD library 1.0.6 Zhao (2019).

4.3.2 5G Network Emulation

Given the lack of a public 5GC dataset that can be used for anomaly detection, we leverage our 5G testbed for normal and HTTP/2 SMA emulation. This requires emulating UE-initiated and network-triggered 5G procedures that can occur in a 5G network. To this end, we employ the functionalities provided by UERANSIM (Table 4.2).

Table 4.2: Procedures order

Triggered procedure	**Possible subsequent procedures**
UERegister	UEReleasePDUSession, RANReleasePDUSession, UEDeregister, Uplink, Downlink
Uplink	UEReleasePDUSession, RANReleasePDUSession, UEDeregister, Downlink
Downlink	UEReleasePDUSession, RANReleasePDUSession, UEDeregister, Uplink
UEReleasePDUSession	UEReleasePDUSession, RANReleasePDUSession, UEDeregister, Uplink, Downlink
RANReleasePDUSession	Uplink, Downlink
UEDeregister	UERegister

A. Normal network behavior - Benign dataset generation — To emulate normal network traffic behavior, we consider 50 UEs and perform multiple 5G procedures selected randomly from those provided by the UERANSIM (Table 4.2). As 5G procedures have

logical dependency and precedence constraints between them, the random choice of a procedure $p+1$ for a UE, is performed from a list containing all possible subsequent procedures that can be triggered following a procedure p. For example, a UE cannot deregister from the network unless it is already registered. In addition, each 5G procedure initiates varying communications between NFs based on the UE state (i.e., CONNECTED, IDLE, etc.) and other conditions (network, RAN resources, etc.) TS.123.502 (2025). This is reflected through the API calls and/or API information elements initiated/used by the NFs. For example, if the network-triggered service request procedure (i.e., downlink) TS.123.502 (2025) is initiated while the UE's state is CONNECTED, the API requests will not trigger the paging procedure. Note that the procedures listed in Table 4.2 are triggered at different times for the same UE to replicate 5G communications and can switch the UE to various states. For example, (1) UE registers to the network[1]; after a while, (2) RAN releases the PDU resources allocated to the UE, which switches its state to IDLE; (3) Then, a downlink procedure is triggered which switches the UE state from IDLE to CONNECTED.

B. Malicious network behavior - Attack dataset generation — In our proof of concept, we consider an attack from a malicious SMF towards an AMF. Thus, we select the procedures that trigger `Namf_Communication_N1N2MessageTransfer` API, such as UE-triggered service request (i.e., uplink), network-triggered service request (i.e., downlink), and UE release PDU session, given that this API covers most of the service operations provided by the AMF and consumed by the SMF (Section 4.1.2). Using 15 legitimate UEs, which information were compromised by the attacker, the malicious SMF requests these procedures from the AMF by establishing multiple TCP connections. Each HTTP/2 connection running on top of a TCP connection established between SMF and AMF has `SETTINGS_MAX_CONCURRENT_STREAMS=250`, which is the default value used in the

[1]UE PDU session establishment is automatically triggered in Free5GC Free5GC (2021a) after a UE registration.

5G testbed version 1 (2.3). We initiate the malicious requests while other legitimate requests are being processed concurrently in the 5G network. We emulate both stealthy and non-stealthy versions of the HTTP/2 SMA. For stealthy attack emulation, we randomly select UEs from the 15 compromised UEs that we dedicated for the malicious activities. Each of the selected UEs randomly triggers one or multiple 5G procedures TS.123.502 (2025) while respecting their precedence constraints (Table 4.2). In contrast, for the non-stealthy attack emulation, the 15 compromised UEs are used to perform the same procedure(s) simultaneously. That is a combination of (1) Uplink procedure; (2) Downlink procedure; (3) UE release PDU session procedure[2] in which UE requests to release its PDU session and switches to IDLE state. This combination of procedures is performed in any order. However, all the compromised UEs will be performing the same chosen order of (1), (2), and (3) at a time.

C. Data collection and attack impact — Using the benign and malicious network emulations described above, we collect from the 5G testbed version 1 (Subsection 2.3) the application layer information at the AMF (Section 4.2.1). Further, as we aim to compare 5GShield with flow-based anomaly detection solution, we collect the incoming and outgoing traffic flows (pcaps) to/from the AMF. We use these flows for flow-based features extraction as it will be described in Section 4.3.3. During the attack emulation, we observe an increase in the Central Processing Unit (CPU) consumption at the AMF once the attack starts at 576 seconds (Figure 4.3). Nonetheless, such an increase cannot be used for attack detection as it can also be observed during normal network conditions following a peak in network traffic (e.g., scheduled events during particular periods).

[2]UE PDU session establishment procedure is automatically triggered after the UE release PDU session procedure in Free5GC Free5GC (2021a).

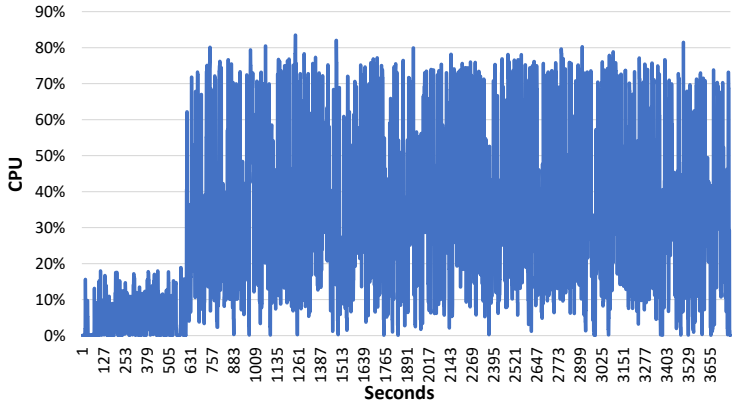


Figure 4.3: AMF CPU consumption during the attack

4.3.3 Data Pre-processing & Feature Engineering

We pre-process the collected data to extract application-layer features to use in 5GShield, and flow-based features to train a flow-based anomaly detection model that we aim to compare 5GShield against.

- **Application-layer features —** From the PM counters collected at the AMF, we retain a total of 25 3GPP-AMF and HTTP/2-AMF features (Section 4.2.2), listed in Table 4.1. From these features, we disregard low-weight features such as *receivedRequestToAMFperAUSF*, *receivedRequestToAMFperNSSF*, *receivedRequestToAMFperPCF*, *sentErrorResponseFromAMF* and retain high-weight ones based on the variance threshold ML method scikit learn (2021) (Section 4.2.2). The retained features are normalized and depict communications between the AMF and all the NFs in the network, and not only the SMF. This allows the detection of attacks originating from any NF(s) towards the AMF.

- **Flow-based features —** We extract flow-based features from the collected network flow traffic using CICFlowMeter Cybersecurity (2020). This results in 84 features listed in Appendix A. We clean and normalize the collected features using oneHotEncoder. Then using the same variance threshold ML method scikit learn (2021)

employed for application-layer features selection, we discard the flow-based features with low weight such as *Bwd IAT Mean*, *Bwd IAT Max*, *Bwd PSH Flags*, *IAT Tot* Cybersecurity (2020), etc., and retain the rest (e.g., *flow duration, total Fwd Packet, total Bwd packets*).

In summary, we end up with benign and malicious records associated with the emulated stealthy and non-stealthy attacks, with a total of 19 application-layer features and 56 flow-based features. We label our data to evaluate our anomaly detection model performance by depending on our knowledge of the compromised UEs that we used for attack emulations. We consider the attack as our positive class. However, we do not use the label as a feature in our models given that we adopt an unsupervised learning technique.

4.3.4 Dataset for Anomaly Detection

To train and evaluate our 5GShield anomaly detection solution, we divide the application-layer features dataset into three categories: (1) *Training and Validation Dataset:* Benign data used to train and validate the unsupervised model; (2) *Optimization Dataset*: Benign and malicious data used to select the threshold; (3) *Test Dataset*: Benign and malicious data used to evaluate 5GShield detection performance. These datasets are mutually exclusive and do not include redundant records. We similarly split the flow-based features dataset and use it to train and test a flow-based anomaly detection solution.

4.4 Experiments and Results

In this section, we evaluate the performance of 5GShield against a traditional flow-based anomaly detection solution and test its performance in the presence of contaminated data.

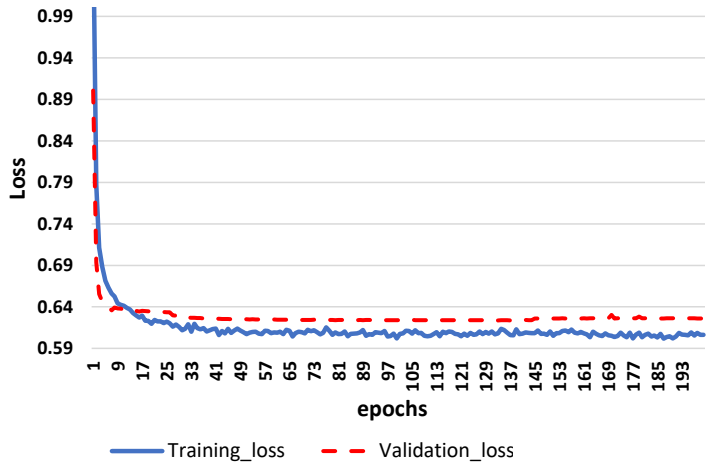


Figure 4.4: Training and validation loss of 5GShield AE model

4.4.1 5GShield Application-layer Anomaly Detection Solution

A. AE architecture selection

To determine the architecture of the AE to use in our 5GShield anomaly detection module (Section 4.2.3), and which better recognizes the HTTP/2 SMA, we train and validate the performance of multiple AE architectures. We use 20000 benign records as a training dataset to train the models and validate their performance using a validation dataset that yields 10% of the training dataset. Our tests show that a basic AE with one hidden layer is the most efficient. Thus, we train the selected model with a combination of hyperparameters for 200 epochs (Table 4.3). We observe the average reconstruction loss across the different training epochs for the training model using benign unlabelled data. As shown in Figure 4.4, the training loss and the validation loss start to converge after 30 epochs and the AE depicts a reasonable convergence within 200 epochs.

Table 4.3: Autoencoder hyperparameters

Hyperparameter	Architecture	Number of epochs	Dropout	Batch size	Loss	Optimizer	Hidden activation
AE - 5GShield	[19; 3; 19]	200	0.2	32	MSE	Adam	ReLU
AE Flow-based	[56; 8; 3; 8; 56]	200	0.2	32	MSE	Adam	ReLU

B. 5GShield performance and threshold selection

To evaluate the detection performance of the AE, we select a threshold $\alpha = 4.399$ as it maximizes the F1-score. The threshold selection was done by evaluating the AE performance using an optimization dataset of 1400 benign and 4600 malicious records. Using the selected threshold $\alpha = 4.399$ displayed as a green line in Figure 4.5, we evaluate the model performance using a test dataset of another (other than optimization dataset) 1400 benign and 4600 malicious records. Figure 4.5 shows that the test records between 0 and 4600 are related to stealthy and non-stealthy attacks and depict an anomaly score (i.e., MSE) greater than the selected threshold. In contrast, only a few of those records, i.e., belonging to the stealthy attack, are predicted as benign given that their MSE is under the threshold. This is expected as a stealthy attack is comparable to a benign behavior which makes its detection more challenging. In addition, test records starting at record #4600 are benign and are correctly classified. Their anomaly scores drop under the threshold as depicted in Figure 4.5. As a result, 5GShield with AE using application-layer features achieves good detection performance with an F1-score of 0.992.

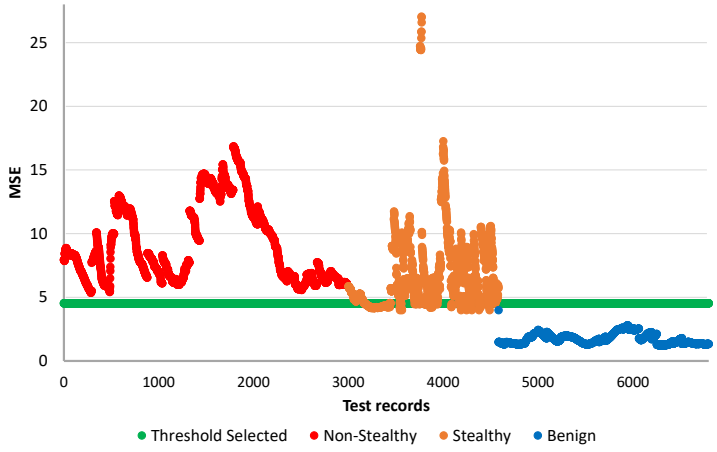


Figure 4.5: Anomaly scores for test dataset records

C. 5GShield performance with contaminated data

In real operational network settings, access to purely benign data is challenging. In contrast to the previous test in which we trained our model using only benign data, we evaluate the performance of our 5GShield AE when trained on partially contaminated data (i.e., a mix of unlabeled benign and significant malicious data) in this experiment. We consider the training dataset and contaminate it with 0.1%, 0.5%, 1%, 1.5%, and 2% of malicious data. Then we train the AE with the same hyperparameters (Table 4.3). We use the optimization and test datasets to select the threshold and test the model respectively. Figure 4.6 depicts a degradation of 5GShield model's F1-score with the increase of the contamination percentage in the training data. When contamination exceeds 1%, the F1-score falls below 0.85. In the presence of higher contamination, our model needs to be fine tuned to better detect HTTP/2 attacks. We leave this for future work.

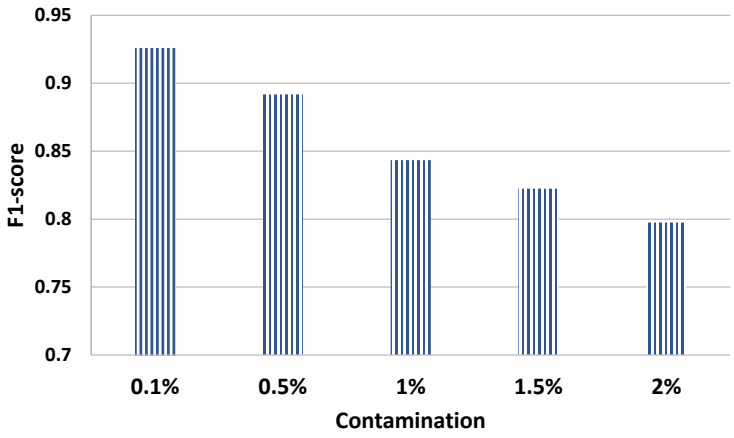


Figure 4.6: F1-score of 5GShield model with contaminated data

4.4.2 Flow-based Anomaly Detection Solution

We compare the performance of 5GShield against a traditional flow-based anomaly detection solution that is widely used in the literature. For that, we develop a flow-based

AE using the same data that we generated and employed for 5GShield AE (Section 6.4). We pre-process this data to extract flow-based features. We train our flow-based AE using a training dataset of 1500 benign records. We use an optimization dataset of 232 benign and 268 malicious records to select the threshold that maximizes the F1-score and a test dataset of 218 benign and 282 malicious records. Similar to 5GShield, we evaluate different model architectures and select the one that depicts the best performance. The selected flow-based AE architecture and hyperparameters are depicted in Table 4.3. Our results show that for a threshold $\beta = 0.2437$, the flow-based anomaly detection model achieves a detection performance with an F1-score of 0.78.

4.4.3 5GShield and Flow-based Anomaly Detection Comparison

To better evaluate 5GShield against the flow-based anomaly detection solution, we resort to the Receiver Operating Characteristic (ROC) curves. An ROC curve summarizes the trade-off between the False Positive Rate (FPR) and the True Positive Rate (TPR) for all thresholds Dalianis (2018). The Area Under the ROC Curve (AUC) represents a metric commonly used with ROC to compare multiple ML models. It provides an aggregated measure of performance across all thresholds. An $AUC = 1$ depicts a perfect model that can reach a $TPR = 1$ and a $FPR = 0$ with a perfect threshold selection. Figure 4.7 shows the under performance of flow-based anomaly detection solution with an $AUC = 0.7365$ in comparison to 5GShield with an $AUC = 0.8673$. This highlights the advantage of profiling NFs behavior through 5G specific application-layer features in comparison to general flow-based features.

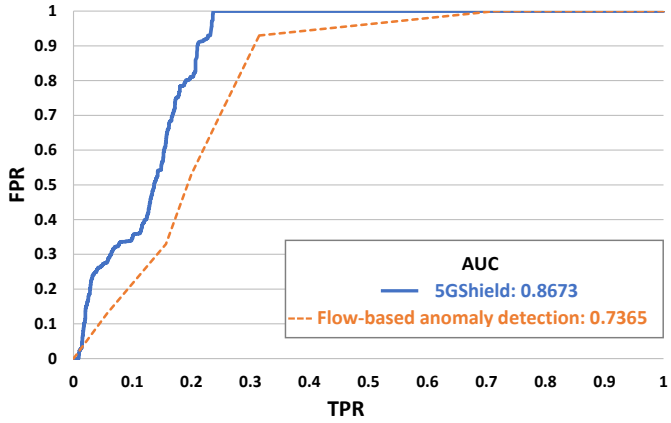


Figure 4.7: AUC-ROC of 5GShield and flow-based anomaly detection solution

4.5 5GShield Deployment Options

The 5GShield solution was designed to complement 5G NFs with additional anomaly detection capabilities in order to secure the 5G network. The novelty of 5GShield yields in its usage of standardized 5G specific application data, also known as PM counters. These PM counters are standardized and defined by 3GPP for each 5G NF. They can be used by network operators to profile NFs behavior. The use of these PM counters alleviates the need for telecom operators to deal with line-rate traffic flows that may be hardly collected and managed when their network is deployed in the cloud where they do not necessarily own the infrastructure.

3GPP PM counters collected by each NF can also be shared upon request by that NF with the Operations Administration and Maintenance (OAM) module, which in turn can share it with the Network Data Analytics Function (NWDAF) Y. Yuan, Gehrmann, Sternby, and Barriga (2022). NWDAF was introduced in 5G SBA and is responsible for 5G network data analytics generation and analysis. The generated data can also be used for closed loop control with the assistance of ML models. Thus, we envision that our 5GShield can be deployed as a built-in NWDAF at the NF, where data, insights and actions are taken by that NF. This enables an automated closed loop at the local level. 5GShield can also be

deployed at a central NWDAF in the form of a NF that collects data from other NFs and use it for a closed loop at the network level Y. Yuan et al. (2022).

4.6 Discussion

In 5G networks, ensuring security across different layers is crucial to detecting sophisticated attacks. Each layer—application, network, and transport—provides unique visibility into potential threats, making a multi-layered detection approach ideal. However, in scenarios where administrators have access only to the application layer, solutions like 5GShield remain essential in safeguarding the 5G SBA against advanced threats.

Application-layer detection plays a critical role in identifying sophisticated HTTP/2-based attacks, such as SMA, which exploit the very protocols that enable efficient communication between NFs. Unlike network-based solutions that rely on traffic flow analysis, 5GShield operates at the application level, leveraging PM counters and API behavior profiling to detect anomalies that may go unnoticed at lower layers. This makes it particularly effective in environments where security teams do not have access to network-layer packet inspection or transport-layer controls, such as in cloud-based or third-party-managed 5G deployments. Moreover, attackers often use encryption (e.g., TLS) to evade traditional network-based intrusion detection systems, making application-layer monitoring indispensable for identifying abnormal API behavior even when traffic is encrypted. Since 5GShield directly analyzes interactions between NFs and the services they consume, it provides a deeper level of detection than network-layer anomaly detection, which typically focuses on traffic volume and flow statistics.

Ultimately, 5GShield serves as a critical security layer in scenarios where administrators only have access to application-level monitoring. However, for a comprehensive security posture, it should be complemented by network and transport-layer defenses to

detect and mitigate attacks at all levels of the 5G SBA. By integrating these approaches, security teams can ensure robust protection against a wide range of threats while maintaining real-time anomaly detection capabilities in 5G networks.

Chapter 5

Empowering 5G SBA Security: Time Series Transformer for HTTP/2 Anomaly Detection

In this chapter, as our previous work Wehbe et al. (2023) on HTTP/2 attack detection in 5G SBA which presents some limitations in terms of robustness to contaminated data, existing works Praseed and Thilagam (2018, 2019, 2020, 2021) were limited to a web environment and are not fine-grained enough to capture 5G API calls dependencies and sequence for fulfilling 5G procedures. The latter can be exploited to perform HTTP/2 attacks. We propose 5GGuardian, an anomaly detection solution that leverages a time series transformer trained on 5G-Stream features. The 5G-Stream features capture fine-grained details of NFs behavior and enable robust anomaly detection of HTTP/2 SMA. Experiments on our 5GC datasets that are collected from 5G testbed version 1 (Subsection 2.3), reveal that our proposed approach achieves an average F1-score of 0.98 in identifying the HTTP/2 SMA variations. We evaluate the performance of 5GGuardian and emphasize its robustness in the presence of contaminated training data, as well as its ability to outperform

application-layer anomaly detection solutions.

5.1 Threat Model - HTTP/2 SMA Variations

In our previous work Wehbe et al. (2023), We emulate both stealthy and non-stealthy versions of the HTTP/2 SMA by leveraging `Namf_Communication_N1N2MessageTransfer` API to trigger multiple procedures in a different order. However, in this work, we leverage the `Namf_Communication_N1N2MessageTransfer` API's service operations to perform the HTTP/2 SMA in five variations exploiting the 5G procedures using the aforementioned API:

- *Random-requests-based HTTP/2 SMA (Random-SMA)*: consists of sending random requests from the SMF towards the AMF using legitimate UE information and list of procedures (network triggered service request (Downlink), UE triggered service request (Uplink), UE release PDU session (UEReleasePDUSession)) selected randomly.
- *Down HTTP/2 SMA (Down-SMA)*: consists of launching the network triggered service request procedure (Downlink procedure) by sending malicious requests from SMF to AMF using the same subset of legitimate UEs.
- *Up HTTP/2 SMA (Up-SMA)*: involves launching the UE triggered service request procedure (Uplink procedure) by sending malicious requests from SMF to AMF using the same subset of legitimate UEs information.
- *Release HTTP/2 SMA (Release-SMA)*: consists of triggering the UE release PDU session procedure (UEReleasePDUSession procedure) by sending requests from SMF to AMF using the same subset of legitimate UEs.
- *Uniform-requests-based HTTP/2 SMA (Uniform-SMA)*: consists of sending the same type of malicious requests repetitively following the same order from SMF to AMF using the same subset of legitimate UEs. All UEs will be used by the malicious SMF to trigger

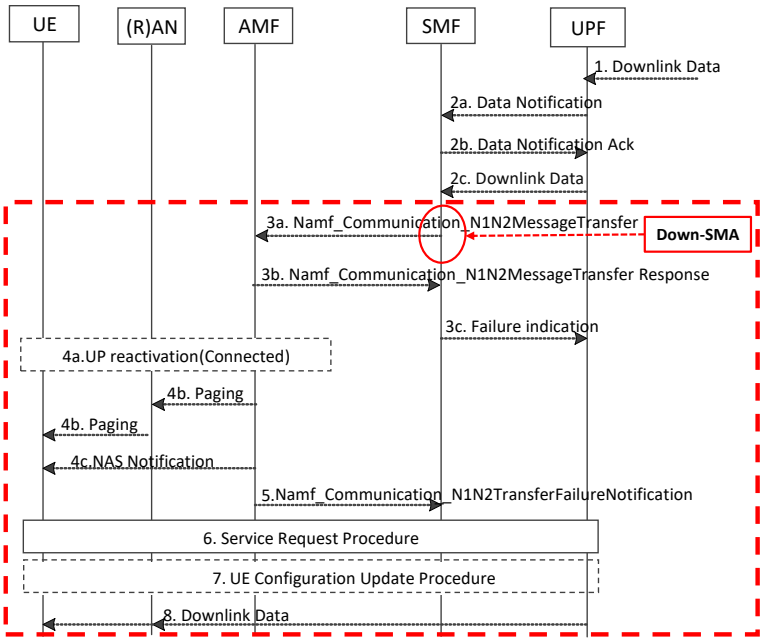


Figure 5.1: Down-SMA emulation in network triggered service request procedure TS.129.518 (2025)

Downlink, Uplink, and UEReleasePDUSession procedures in order.

A. HTTP/2 SMA Variations - An Example

The detection of HTTP/2 SMA can be challenging based on the exploited 5G procedures and the impact of the attack on 5G network. To better illustrate this, we explain in the following how the *Down-SMA* can be performed by exploiting the Downlink procedure and we theoretically highlight its impact on the AMF. Quantitative evaluation of this attack along with other HTTP/2 SMA variations (i.e., that can be emulated similarly) are presented in Section 5.4.1. Figure 5.1 illustrates the normal Downlink procedure triggered from the DN when the UE is in an IDLE state TS.129.518 (2025). The SMF sends a request to the AMF using `Namf_Communication_N1N2MessageTransfer` API (Figure 5.1 (3a)). The AMF responds to the SMF indicating that the UE is not reachable, and subsequently sends a Paging Request to the RAN/UE (Figure 5.1 (4b)). The Paging Request

triggers the Uplink procedure in order to activate the UE. In the dashed red box (Figure 5.1), we highlight a scenario where an attacker compromises the SMF and launches a malicious `Namf_Communication_N1N2MessageTran sfer` request towards the AMF, triggering a Downlink procedure for a UE in IDLE state without requiring network signaling. Although the attacker only initiates a single request, it results in a chain of other messages (Figure 5.1 (3b-8)) related to paging, service request, PDU session update, and PDU session modification in the 5G network. This depicts the high overhead that an attacker can introduce to the network with a single malicious request. Hence, the attacker can scale the attack by exploiting the stream multiplexing feature with only few requests. Note that the overhead on the AMF can also be high based on the triggered procedure. For instance, when starting the *Down-SMA*, the AMF is called twice by SMF (Figure 5.1 (3a and 6)), while in the *Up-SMA* involves the AMF three times, and the *Release-SMA* involves the AMF five times TS.129.518 (2025). Note that, in our emulations (Section 5.4.1), the UE release PDU session, once launched, automatically triggers a PDU session establishment for the UE in question as a default functionality of the used testbed. Thus, when emulating the Release-SMA, the PDU session establishment is automatically triggered.

Following the above discussion, it is clear that the attacker can utilize requests belonging to different services (e.g., APIs) and involve different NFs to launch an HTTP/2 SMA that can appear as a stealthy network overload. Hence, to address the existing shortcomings of HTTP/2 anomaly detection models, we define two fundamental criteria to which any anomaly detection mechanism targeting HTTP/2 attacks within 5G networks should adhere to:

- *Fine-grained*: A robust HTTP/2 detection model should possess a fine-grained approach, allowing it to focus on capturing nuanced features and specific aspects of the input data. The model should be able to better distinguish between different patterns and anomalies within the data that contains NFs API calls, the sequence in calls for 5G procedure,

dependencies between procedures, etc.This level of granularity empowers the model to gather and analyze particular segments that might indicate NF behaviors or anomalies, resulting in more precise and accurate identification.

- *Adaptive*: An effective detection technique must be adaptable to evolving NFs behavior. This necessitates regular updates to the underlying model including retraining on new data depicting updated 5G NFs behavior based on various network conditions.

5.2 Methodology - 5GGuardian Solution

In this section, we introduce 5GGuardian (Figure 5.2), our innovative anomaly detection solution that leverages transformer models trained on 5G-Stream features, extracted from 5G network traffic data for HTTP/2 anomaly detection. The 5GGuardian solution is composed of two main modules; (1) the 5G-Stream data collection and extraction module that extracts network traffic and performs the features engineering; and (2) the transformer module that trains a transformer model using the collected and pre-processed data, then uses the trained model for online anomaly detection. Details of this solution are discussed in the following.

5.2.1 Data Collection and Pre-processing

The process of data collection involves the utilization of a network monitoring tool, such as Wireshark The Wireshark Team. (2021). This tool enables the monitoring of network traffic and captures raw packets (e.g., HTTP/2 packets) during normal operation of the 5G network. During the training mode, the collected packets are compiled into a PCAP file, which serves as the input data for further analysis as shown in Figure 5.2. We employ TShark The Wireshark Team. tshark (2021), a network protocol analyzer, to process the

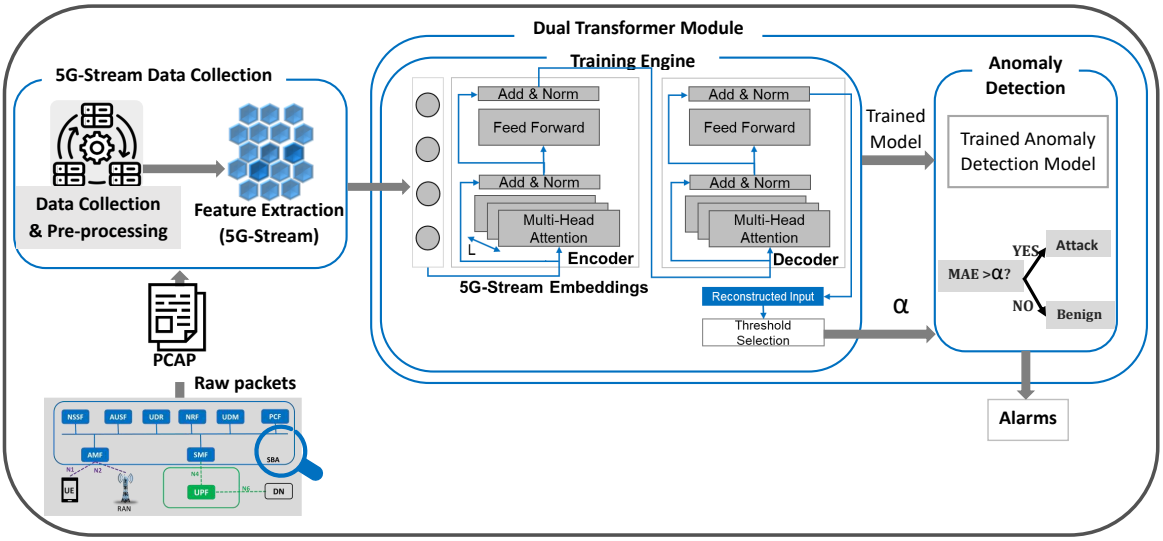


Figure 5.2: 5GGuardian solution

PCAP file. The TShark packet parser reads raw binary data, parses the packets, and extracts relevant metadata into a CSV file (e.g., source/destination IP, port numbers, frame length, path, stream Id, etc.). Our 5G-Stream data collection and feature extraction module (Figure 5.2) utilizes the metadata to derive meaningful 5G-Stream features that we detail in the following.

5.2.2 5G-Stream Features Extraction

Using the metadata obtained from the collected PCAP file, we extract features that serve as valuable indicators of typical NF behavior during normal network operations. However, during HTTP/2 attacks, these features provide insights about the malicious behavior encountered by the NF. This malicious behavior can be originated from any neighboring NF. Given that in this work, we consider an attack on the AMF as a use case (Section 5.1), the extracted features will reflect the AMF behavior. Nonetheless, as we use generic stream features to train our anomaly detection model, our approach can be applied to detect anomalies on any NF. To build a stream-level behavioral profile of the 5G NFs, we develop the 5G-Stream features algorithm (Algorithm 1) that aims at extracting stream features from

the collected PCAP network traffic file.

Algorithm 1 5G-Stream features algorithm

InputInputs OutputOutput $data$: the data collected in a CSV file $features$: the 5G-Stream features

1: $rawPacket \leftarrow$ ReadCSV(data)
2: $filteredRaw \leftarrow$ rawpacket.Filter(AMF_ip)
3: $groupedRaw \leftarrow$ filteredRaw.GroupBy(srcPort, dstPort)
4:
5: **for each** $packets \in groupedRaw$ **do**
6: $streamsRow \leftarrow packets.Unique()$
7: $requestsAndResponses.Append(streamsRow)$
8: $distinctRaw \leftarrow$ requestsAndResponses.Distinct()
9: $requestOonly \leftarrow$ distinctRaw.Select(requests)
10: $responseOnly \leftarrow$ distinctRaw.Remove(requests)
11: $features \leftarrow$ Concat(requestOnly, responseOnly)
12: **return** $features$ =0

In Algorithm 1, we read and filter the raw packet data to isolate a single request representing the header and identify its corresponding response (i.e., success or failure of the request) (line 1-10). By aggregating the request and response, we obtain a single record that represents an individual stream (line 11). Each stream presents 10 features (Table 5.1), and the collection of these streams forms the set of the 5G-Stream dataset.

Table 5.1: 5G-Stream features

5G-Stream Features	Latency, http2_protocols, http2_headers_method, http2_headers_path, Header_request_size, Header_response_size, ResponseCode, IMSIfromAPI, Http2_max_concurrent_stream, HasResponse

The 5G-Stream features provide fine-grained details and specific characteristics of the input data. For example, the extracted streams highlight essential 5G information, as described in Table 5.1, focusing on features that indicate specific aspects of HTTP/2 protocol in 5G. The http2_protocols feature captures the presence of 5G protocols such as 5G-NAS, NGAP, or a combination of both, while the IMSIfromAPI feature contains the extracted International Mobile Subscriber Identity (IMSI) number obtained from the API or JSON data.

By incorporating these detailed features, the anomaly detection model can better distinguish between different patterns, variations, or anomalies that are present in the data. This level of granularity enables more precise and accurate detection, as it allows the identification of HTTP/2 SMA and potential security threats within the 5G network.

5.2.3 Time Series Transformer Architecture

For anomaly detection, the 5GGuardian solution leverages the time series transformer ML technique, which has shown promising outcomes in network traffic analysis and intrusion detection Xu et al. (2021). Unlike traditional models that rely on recurrence, the transformer model uses self-attention mechanism to establish global input-output dependencies Vaswani et al. (2017). This mechanism enables a higher level of parallelization leading to improved efficiency and performance.

A. Time Series Transformer

As our data represents a sequence of data points taken at evenly spaced intervals, we choose the transformer model specifically designed for analyzing time series data Wen et al. (2022). Time series transformers have been used for a variety of tasks including time series forecasting, and anomaly detection, making them suitable for our case study which involves long-term dependencies in the data Lin, Wang, Liu, and Qiu (2022). The key difference between a standard transformer and a time series transformer lies in how they treat input tokens. While the former treats each token independently, the latter takes into account the order and temporal dependencies of the input sequence. In our proposal, we employ a network that utilizes the time series transformer model to detect anomalies in network traffic. The model architecture consists of two essential components; an embedding 5G-Stream layer and a standard transformer encoder/decoder. To process the normal time series data from NF streams which are structured as a 2D tensor with dimensions of

sequence length multiplied by the number of features, we initially encode them into sequences of embeddings. These embeddings then undergo a series of multi-head-attention blocks and feed-forward layers alternatively. This stacking structure (Figure 5.2) facilitates the learning of underlying associations from deep multi-level features.

B. Multi-head Attention

The transformer model consists of a sequence of L multi-head self-attention layers and point-wise fully connected layers for the encoder. In the transformer architecture, the attention employs the Query-Key-Value (QKV) model and the scaled dot-product attention technique Vaswani et al. (2017), given by Eq.(1).

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{D_k}})V \tag{1}$$

where queries Q $\in \mathbb{R}^{L*D_k}$, keys K $\in \mathbb{R}^{M*D_k}$, values V $\in \mathbb{R}^{M*D_v}$ where L, M denote the lengths of queries and keys (or values) respectively, and D_k, D_v denote the dimensions of keys (or queries) and values. A Transformer uses multi-head attention Vaswani et al. (2017) with h different sets of learned projections instead of a single attention function as in Eq.(2)

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O \tag{2}$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$; $W_i^Q \in \mathbb{R}^{L*D_k}$, $W_i^K \in \mathbb{R}^{M*D_k}$, $W_i^V \in \mathbb{R}^{M*D_v}$, and $W^O \in \mathbb{R}^{M*LD_v}$.

C. Feed-forward Network

The feed-forward network is a fully connected module that employs the Gaussian error Linear Unit (GeLU) activation function (Eq.(3)). The GeLU activation function is utilized

to control the output and gradient contribution of the deep neural network neurons. Introduced in Hendrycks and Gimpel (2016), GeLU is a deterministic activation function that combines stochastic regularization with other techniques or components. Its ability to effectively control the neuron's output and gradient contribution contributes to the improved effectiveness of our solution in anomaly detection.

$$Gelu(x) = x\Phi(x) \tag{3}$$

where $\Phi(x)$ is the cumulative distribution function of Gaussian distribution Vaswani et al. (2017).

D. Reconstruction Error

The model predicts the original values of the input sequence. In our approach, the loss function is the Mean-Absolute Error (MAE) Hodson (2022) which measures the absolute difference between the input stream and the predicted stream as shown in Eq.(4).

$$\sum_{i=1}^{D} |x_i - y_i| \tag{4}$$

The MAE measures the overall absolute difference between the actual values y_i and the predicted values x_i by the model. During the training phase, the model learns about the various features and variations observed in normal behavior, thereby the prediction error will be minimal when tested on normal network data. In contrast, the prediction error will be higher when tested on abnormal data as the model will fail to correctly predict it. In fact, the transformer model determines whether the predicted loss error exceeds a predefined threshold in order to identify abnormalities in the testing data.

E. Threshold selection

The selected threshold plays a crucial role in determining whether a given data sequence is considered normal or abnormal during the anomaly detection stage. This is accomplished by comparing the sequence's reconstruction error to the threshold value. During the testing phase, the model is provided with a specific input sequence, and its predicted error is compared to the threshold. If the anticipated error exceeds the threshold, the input sequence will be classified as malicious; otherwise, it will be classified as benign.

$$Threshold = mean(train\_loss) \tag{5}$$

Thus, the model detection performance is highly dependent on the selected threshold. The latter can be adapted to the test data, upon the need. Many approaches can be used to select a threshold such selecting the value that maximizes the F1-score as in Salahuddin et al. (2021) or computing as two standard deviations above the mean loss of the trained normal data, which serves as a measure of data variability as in Alamr and Artoli (2023). However, in our work, we set the threshold to the mean loss of the trained normal data given that our tests using a validation set showed that such threshold selection depicts good detection performance. The latter was further validated on our test data (Section 5.6).

5.2.4 Online Detection

The trained 5GGuardian transformer model is used for online anomaly detection. In fact, based on the data dependencies that the model learned during the training phase, it can successfully identify data that deviates from the normal behavior learned during the training phase, as anomalous. When inputting new data, the model provides a reconstruction error that is compared to the selected threshold as explained above to determine if the data is benign or malicious.

5.3 Environment Setup - Model Training Setup

To train our 5GGuardian transformer model, we exclusively use normal data in batches of 16 sequences. We perform all experiments in Python (v3.8) using ML libraries such as Tensorflow (v2.12.0) and Transformer (v4.27.4). To ensure computational efficiency, we perform all studies on a separate VM equipped with an NVIDIA GPU and 28GB of RAM. To fine-tune our model, we run numerous tests with varying numbers of transformer encoder blocks (consisting of a multi-head attention layer followed by a feedforward layer), hidden state sizes, and attention heads. For regularization, we only apply dropout on the first dense layer with a rate of 0.1. To optimize our model, we employ the Adam optimizer with a 10^{-12} learning rate and early stopping to prevent overfitting. Although the model has a maximum training duration of 200 epochs, all experiments converge before reaching this limit.

5.4 Data Evaluation & Analysis

In this section, we analyze the collected data to extract 5G-Stream features. In addition, we evaluate the impact of the HTTP/2 SMA on the performance of the 5GC.

5.4.1 Emulation of Normal & Malicious 5G Network Behavior

Due to the absence of publicly available datasets for anomaly detection in the 5GC, we employ our 5G testbed to emulate both normal network behavior represented by random UE activities in addition to HTTP/2 SMA. To emulate normal UEs activities in our testbed, we leverage different 5G procedures that are implemented in UERANSIM. The same procedures are used to emulate HTTP/2 SMA variations. We provide a detailed explanation of each procedure below.

- Registration procedure TS.123.502 (2025) (Register): UE registers to the network to gain

authorization for accessing 5G services such as enabling mobility tracking and becoming reachable.

- Deregistration procedure TS.123.502 (2025) (Deregister): UE initiates this procedure to unregister from the 5G network.
- PDU Session Establishment procedure TS.123.502 (2025): UE initiates a PDU session establishment procedure.
- PDU Session Resource Release procedure TS.38.413 (2024) (RANRelease): RAN releases the PDU session resources that were previously allocated to a UE previously.
- PDU Session Release procedure TS.123.502 (2025) (UEReleasePDUSession): UE requests to release one of its PDU sessions.
- UE Triggered Service Request procedure TS.123.502 (2025) (Uplink): UE sends uplink signaling messages when it is in an IDLE state.
- Network Triggered Service Request procedure TS.123.502 (2025) (Downlink): The network signals to a UE using this procedure.

By emulating these procedures, we generate a comprehensive dataset that encompasses both normal network behavior and the specific behaviors associated with HTTP/2 SMA.

A. Normal network behavior - Benign dataset generation

To emulate normal network behavior, we consider 50 UEs which arrival to the network is modeled using a Poisson process Raaijmakers, Mandelli, and Doll (2021) and engage in one or multiple 5G procedures randomly selected from those provided by the UERANSIM (Table 6.1). It is important to note that 5G procedures exhibit logical dependencies and precedence constraints. Therefore, a subsequent procedure $(p + 1)$ for a UE is randomly selected from a list that includes all possible procedures that can follow the preceding procedure (p). The list of subsequent procedures for each 5G procedure available in UERANSIM is presented in Table 6.1. For example, a UE cannot initiate the deregistration

procedure if it has not been previously registered. Additionally, each 5G procedure triggers communications between NFs, which can vary depending on the UE's state (i.e., CONNECTED, IDLE, etc.) and other conditions such as network, RAN resource TS.123.502 (2025). Consequently, the API calls and/or the corresponding information elements used may differ. For instance, if the Downlink procedure starts while the UE is in the CONNECTED state, the API requests shown in Figure 5.1 will not trigger the paging procedure (steps 4b and 6).

In our emulation, each UE starts by registering to the network (register procedure), then based on the possible subsequent procedures listed in Table 6.1, the following procedure is randomly chosen from the available options, which include UEReleasePDUSession, RANRelease, and Deregister procedures. After the registration, let us assume, for example, RANRelease procedure was selected by the UE. Following its execution, either Uplink or Downlink procedures can be initiated. It is essential to note that these procedures are triggered for the same UE at different times to replicate 5G communications and can switch the UE between different states. For example, (1) UE registers with the network; after a certain period of time, (2) RAN releases the PDU resources allocated to the UE, switching its state to IDLE; (3) Subsequently, a Downlink procedure is triggered from the network to signal to the UE which is in IDLE state. This Downlink procedure switches the UE state from IDLE to CONNECTED.

B. Malicious network behavior - Attack dataset generation

To execute our HTTP/2 SMA from the SMF to the AMF, we specifically target procedures that trigger the `Namf_Communication_N1N2MessageTransfer` API, namely Uplink, Downlink, and UEReleasePDUSession, given that this API is the most used by the SMF. Out of 50 legitimate UEs, we assume that the attacker accessed the information of 15

UEs out of them, through the comprised SMF[1]. For simplicity, we will refer to these UEs as compromised UEs. Through the malicious SMF, the attacker initiates requests of the aforementioned procedures on behalf of each of the compromised UEs towards the AMF by establishing multiple TCP connections. Each of these established TCP connections is configured with `SETTINGS_MAX_CONCURRENT_STREAMS=250`, which represents the default value used in our 5G testbed.

We launch the attack requests while other legitimate requests are concurrently ongoing in the 5G network, as discussed in Section 6.2.2. We emulate five distinct forms (Section 5.1) of the HTTP/2 SMA. For *Random-SMA* emulation, we randomly select UE from the compromised UEs. The SMF randomly triggers one or multiple procedures using the selected UEs while adhering to their precedence constraints (Table 6.1). For example, using UE1 information, SMF initiates an Uplink procedure, starts a UEReleasePDUSession procedure for UE3, and after a certain period, UE1 triggers UEReleasePDUSession, which switches its state to IDLE. Subsequently, SMF uses UE2 information to signal to UE1, which is in the IDLE which launches a Downlink procedure for UE1. Similarly, we employ the compromised UEs to emulate the remaining variants of the HTTP/2 SMA.

5.4.2 Performance Metrics

We consider the following performance metrics to evaluate the impact of the different HTTP/2 SMA variants on the 5GC.

A. N1N2MessageTransfer Time

We focus on the N1N2MessageTransfer operation of the Namf_Communication service, as it is included in all the targeted procedures (Table 6.1), which allows us to compare this API in different emulations. We calculate the N1N2MessageTransfer API time, which

[1]As the SMF is responsible for session management and UE IP address allocation and management TS.29.502 (2025), it has access to UEs information such as the SUPI.

represents the total time of the N1N2MessageTransfer request originating from the SMF towards the AMF and its response from the AMF towards the SMF.

B. Procedure Completion Time (PCT)

PCT is defined as the time taken for a procedure to be completed Goshi, Jarschel, Pries, He, and Kellerer (2021). To measure the PCT, we utilize the 5g-tracer-visualizer telekom (2021) tool which calculates the PCT of any API, focusing on AMF. For example, the PCT of the Uplink procedure represents the time elapsed from the moment SMF sends the request to AMF until AMF acknowledges the completion of the procedure using the corresponding response (i.e., success or failure of the response).

C. Central Processing Unit (CPU) Utilization

We analyze the CPU profile of the AMF during both normal behavior and each HTTP/2 SMA variation emulation. As the AMF plays a central NF role in the 5GC network, ensuring availability, and it also serves as the target for HTTP/2 SMA. To compute the CPU usage of each NF in our 5G testbed, we deploy a shell script that collects data every two seconds.

5.4.3 HTTP/2 SMA Impact on 5G SBA Performance

Using the aforementioned performance metrics, we evaluate the impact of the HTTP/2 SMA variants on our 5G testbed in the following.

A. N1N2MessageTransfer Time

Using 5g-trace-visualizer telekom (2021), we calculate the average (mean), minimum (min), and maximum (max) times of N1N2MessageTransfer API for both benign and HTTP/2 SMA variations. As shown in Table 5.2, during the benign emulation, we observe

the lowest request/response time, while the mean N1N2MessageTransfer time of the rest of the attack scenario also exhibits high compared to benign emulation. However, when considering the different attack scenarios, it becomes apparent that the *Uniform-SMA* exhibits slower performance. This is indicated by the mean N1N2MessageTransfer time which is four times higher than the benign emulations, resulting in AMF becoming overloaded due to the increased number of requests originating from the SMF.

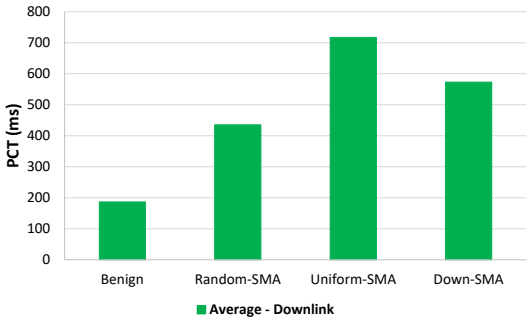
Table 5.2: N1N2MessageTransfer Time

/namf-comm/v1/ue-contexts/n1-n2-messages	**mean**(ms)	**min**(ms)	**max**(ms)
Benign Emulation	1.664	1.664	1.664
Random-SMA	2.597	0.921	9.029e+03
Uniform-SMA	8.193	1.522	1.677e+02
Down-SMA	3.734	1.432	5.160e+03
Up-SMA	2.675	1.975	3.930e+11
Release-SMA	2.19	1.522	5.16e+3

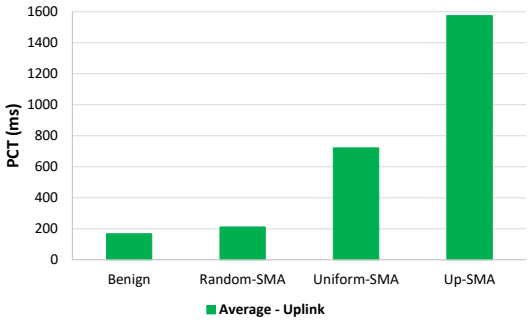
B. Procedure Completion Time

We conduct a comprehensive analysis by comparing the PCT in milliseconds (ms) of Uplink, Downlink, and UEReleasePDUSession procedures across different emulations, as these procedures are used in both benign and attack emulations. Looking to Figure 5.3, we can notice that doing an attack using Uplink (Figure 5.3b) or UEReleasePDUSession procedure (Figure 5.4) is more computationally expensive on AMF. When applying the Downlink procedure (Figure 5.3a), *Uniform-SMA* achieves greater PCT than *Down-SMA*, indicating that the Downlink procedure is not computationally expensive on its own. For example, Figure 5.3b illustrates the findings of the Uplink procedure where it appears in four distinct scenarios: benign, *Random-SMA*, *Uniform-SMA*, and *Up-SMA*. Figure 5.3b indicates that the Uplink procedure takes more time during the *Up-SMA* compared to other scenarios. This prolonged execution duration implies increased involvement of the AMF during this procedure.

Figure 5.3: Procedure completion time of downlink and uplink procedures



(a) PCT Downlink



(b) PCT Uplink

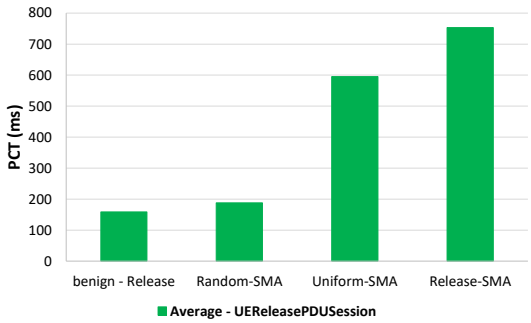


Figure 5.4: PCT UEReleasePDUSession

As discussed in Section 5.1, AMF's participation in each procedure impacts the overall performance of the 5GC. Notably, the Uplink procedure involves the AMF three API calls, contributing to its computationally expensive nature. While it is worth noting that performing *Up-SMA* solely using the Uplink procedure can lead to DoS due to the extended processing time and resource-intensive nature of the procedure. Furthermore, the release procedure involves AMF five times, yet it has lower PCT than the Uplink procedure, contributing to the Uplink procedure's computationally expensive nature. This highlights the importance of carefully managing the computational load and resource allocation within the 5GC to ensure optimal system performance and mitigate the risk of DoS attacks.

C. CPU Utilization

We conduct a comparison for HTTP/2 SMA variations and monitor the AMF CPU utilization of each.

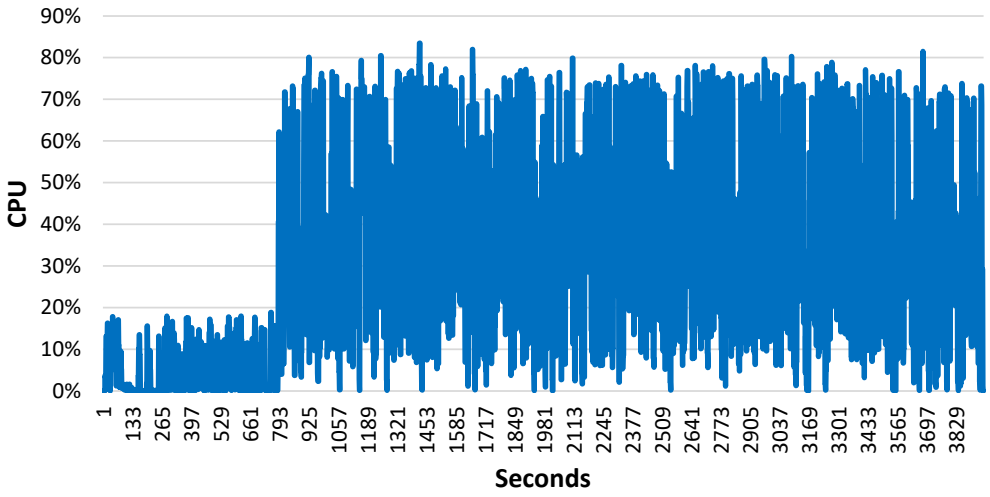
During the *Down-SMA* (Figure 5.5a), we observe an increase in the CPU consumption at the AMF once the attack starts at 576 seconds. It is worth noting that the increase in CPU utilization above 100% is not necessarily indicative of malicious activity, such as the HTTP/2 SMA. Such an increase can also be observed during normal network conditions but following a peak in network traffic (e.g., scheduled events during particular periods) which make the HTTP/2 SMA detection more challenging.

In the case of the *Up-SMA* emulation, the AMF CPU load remains consistently between 80% and 160% (Figure 5.5b). This persistent high CPU load suggests that the AMF is overwhelmed and struggling to keep up with the demands of the workload. It is evident that the *Up-SMA* emulation results in a DoS attack, as shown in Figure 5.5b, where the AMF server stops functioning after 2745 seconds. The reason behind the increased AMF CPU load in the *Up-SMA* emulation can be attributed to examining the PCT and the level of AMF involvement. The PCT time (1600 (ms)) indicates that the Uplink procedures place a heavy computational burden on the AMF (Figure 5.5b), leading to a higher CPU load.

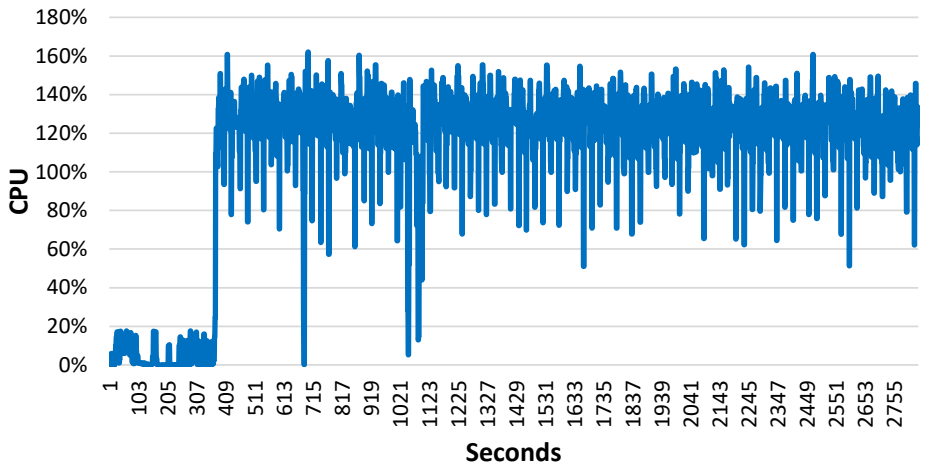
During the *Release-SMA* (Figure 5.6), we observe an increase in AMF CPU utilization, indicating the reception of unexpected workloads. This surge in CPU utilization can be attributed to the fact that when the SMF requests the UEReleasePDUsession procedure in our 5G testbed, it automatically triggers the establishment of the PDU session. As a result, the AMF is tasked with handling a chain of requests each time, leading to increased CPU load and potential performance issues.

Comparing AMF CPU utilization during *Random-SMA* (Figure 5.7a) and *Uniform-SMA* (Figure 5.7b), we notice how AMF CPU utilization during *Uniform-SMA* (Figure 5.7a) is not stable and keeps increasing and decreasing randomly, whereas, during *Random-SMA*

Figure 5.5: AMF CPU consumption during Down-SMA and Up-SMA



(a) AMF CPU consumption during Down-SMA



(b) AMF CPU consumption during Up-SMA

emulation, the AMF CPU show recurrent increase all the time, indicating that *Random-SMA* is similar to the normal behavior. While *Uniform-SMA* causes a DoS on the AMF.

We showed through different experiments that the HTTP/2 SMA variations could increase CPU consumption for 5GC or cause DoS on the AMF itself. This analysis highlights the critical role of efficiently managing AMF resources and workload demands to ensure the robustness and reliability of the 5GC system.

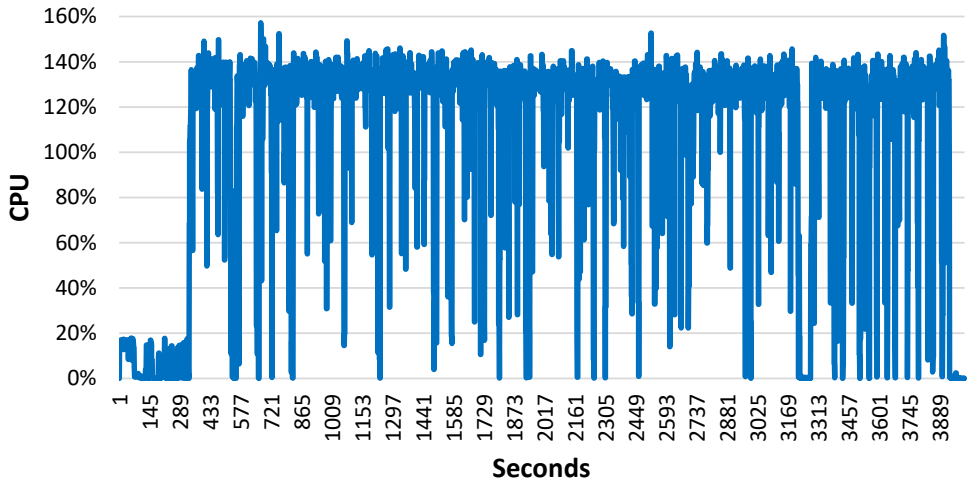


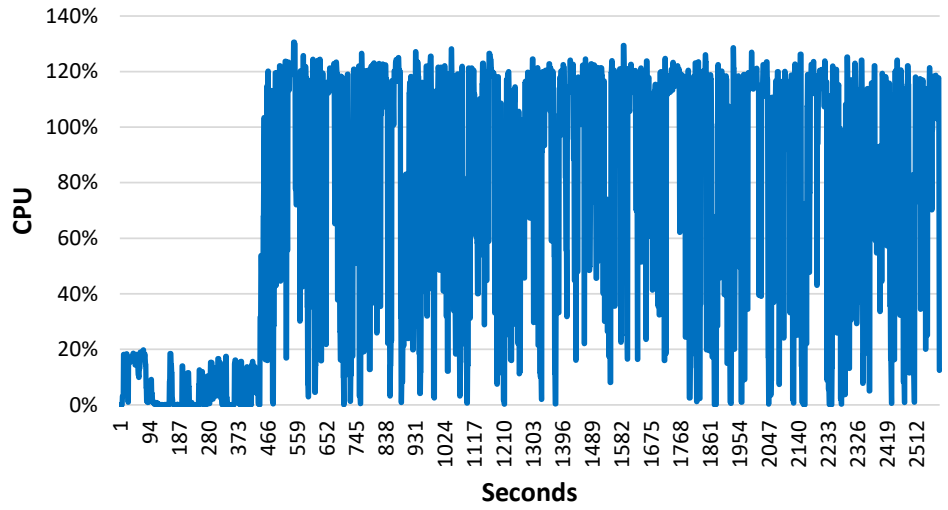
Figure 5.6: AMF CPU consumption during Release-SMA

5.4.4 Impact of HTTP/2 SMA on 5G Core Performance

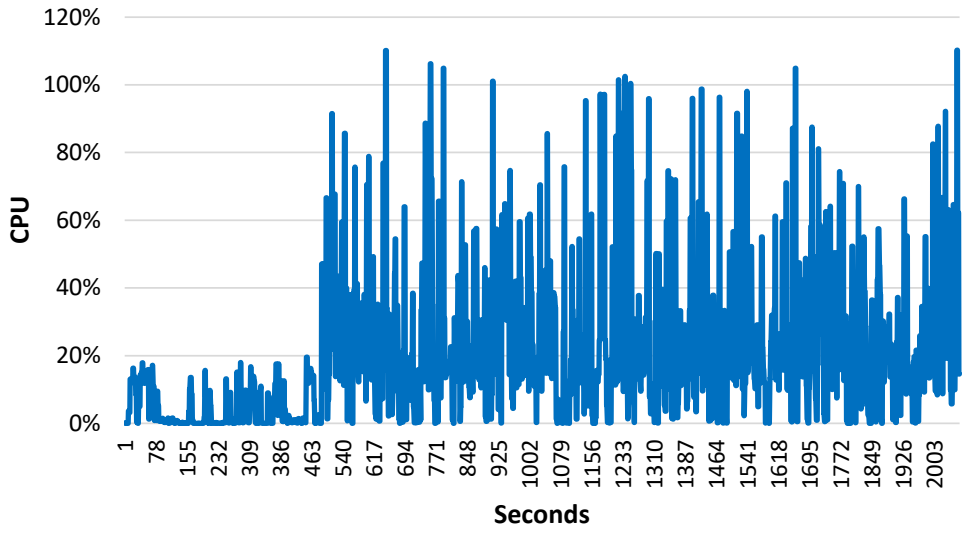
The analysis conducted using the N1N2MessageTransfer time, PCT and CPU, regarding the impact of HTTP/2 SMA on the 5G network performance, reveals the following critical findings. In emulations such as *Up-SMA*, *Release-SMA*, *Down-SMA*, and *Random-SMA*, the average time for n1n2messageTransfer API is notably high (Table 5.2). This observation indicates that HTTP/2 SMA scenarios introduce delays and inefficiencies on 5G network. Furthermore, the high CPU load on the AMF aligns with the findings derived from examining the PCT time in the *Up-SMA* emulation (Figure 5.3b) and assessing the AMF request involvement. In particular, *Up-SMA* (Figure 5.5b) and *Uniform-SMA* (Figure 5.7b) increase the CPU utilization of the AMF, potentially causing a DoS on the AMF itself.

It is worth noting that each attack scenario has a distinct impact on the performance of the 5G network. Some attacks exploit the resources of other NFs, while others specifically target and overload the AMF, leading to a DoS situation. These findings emphasize the importance of effective anomaly detection methods and robust security measures to safeguard the performance and reliability of the 5G network in the face of HTTP/2 SMA variations.

Figure 5.7: AMF CPU consumption during Random-SMA and Uniform-SMA



(a) AMF CPU consumption during Random-SMA



(b) AMF CPU consumption during Uniform-SMA

5.5 Data Collection & Pre-Processing

In this section, we delve into the data pre-processing and feature extraction that we perform on the data collected from our 5G testbed, as depicted in Figure 5.2.

5.5.1 5G Data Pre-Processing

We leverage our 5G testbed, enabling us to collect normal network traffic and HTTP/2 SMA variations, conducted over TCP through application layer protocols. The dataset contains raw packet captures in PCAP format The Wireshark Team. (2021), as shown in Figure 5.2. The benign data is captured over two periods, while the HTTP/2 SMA variations are performed at different times. To filter the dataset, we specifically focus on packets related to the HTTP/2 protocol and filter them based on the source/destination IP address, specifically targeting the AMF IP address as it represents the attack target. We process the raw packets using TShark The Wireshark Team. tshark (2021). The extracted information includes details such as source/destination NF IP addresses, port numbers, frame length, HTTP/2 protocol, frame time epoch, HTTP/2 type, HTTP/2 stream ID, HTTP/2 header method, HTTP/2 header path, HTTP/2 header status, and more. This information is then stored in a CSV file. The metadata is supplied into our 5G-Stream feature extraction module, which extracts 5G-Stream features.

5.5.2 Feature Extraction

A. 5G-Stream Features

We extract 5G-Stream features (Table 5.1) from the aforementioned metadata using Algorithm 1 as explained in Section V.B. The feature extraction process takes approximately $2.3ms$ per stream, which allow fine-grained modeling of the NFs' behavior, thus enabling better anomaly detection performance while satisfying the fine-grained criterion.

B. Application-layer features

To compare our 5GGuardian solution with our previous work 5GShieldWehbe et al. (2023), we extract a total of 20 application-layer features from the Performance Measurements (PM) counters collected at the AMF known as 3GPP-AMF and from the HTTP/2-AMF features, listed in Table 5.3. The extracted features are normalized and depict communications between AMF and all the NFs within the network and are not limited to the SMF alone.

Table 5.3: 3GPP and HTTP/2 application-layer features collected at the AMF

Type	Features
3GPP-AMF	numberofAttemptedNetworkInitiatedServiceRequest
	numberofSuccessfulNetworkInitiatedServiceRequest
	numberofAttemptedUEInitiatedServiceRequest
	numberofSuccessfulUEInitiatedServiceRequest
	totalNumberofAttemptedServiceRequests
	totalNumberofSuccessfulServiceRequests
HTTP/2-AMF	receivedRequestToAMF, sentRequestFromAMF
	receivedRequestToAMFperNRF, sentResponseFromAMFperNRF
	receivedRequestToAMFperAUSF
	receivedRequestToAMFperNSSF, sentResponseFromAMFperNSSF
	receivedRequestToAMFperSMF, sentResponseFromAMFperSMF
	sentErrorResponseFromAMF, receivedErrorResponseToAMF
	totalSuccessfulRequest, totalUnsuccessfulRequest
	receivedRequestToAMFDiscarded

Finally, we obtain the features dataset comprising both benign and malicious records associated with the emulated HTTP/2 attacks. This dataset consists of 10 5G-Stream features (Table 5.1) and 20 application-layer features (Table 5.3). To evaluate the performance of our anomaly detection model, we label our data as benign and attack data based on our knowledge of the compromised UEs used for the attack emulations. We consider the attack as the positive class in our evaluation. Note that we adopt an unsupervised learning technique in which the model is trained on data assumed to be benign in majority. However, the labels are only used to evaluate the model performance.

5.5.3 Dataset for Anomaly Detection

To train and evaluate our 5GGuardian anomaly detection solution, we divide the 5G-Stream features dataset (Table 5.4) into two distinct categories: (1) Training and Validation Dataset: consists of benign data specifically used to train and validate the unsupervised model. It serves as the foundation for the model to learn normal behavior patterns and establish baselines; (2) Test Dataset: comprises both benign and malicious datasets, which are utilized to evaluate the performance of the 5GGuardian detection system.

Importantly, these datasets are mutually exclusive, thus, ensuring that there are no overlapping or redundant records between them. Furthermore, we employ a similar approach to split the application-layer features dataset. This dataset is used to train and test our previous 5Gshield anomaly detection solution. For these datasets, we set the window size as 100, and each segment is compressed and reconstructed by 5GGuardian.

Table 5.4: Train and test dataset

Dataset	**Attack Type**	**Benign Records**		**Attack Records**	
		5G-Stream	**Application-layer**	**5G-Stream**	**Application-layer**
Training		100 000	50 000	-	-
Testing	**Random-SMA**	15 000	5 000	8 000	2 000
	Uniform-SMA	15 000	5 000	8 000	2 000
	Down-SMA	15 000	5 000	8 000	2 000
	Up-SMA	15 000	5 000	8 000	2 000
	Release-SMA	15 000	5 000	8 000	2 000

5.6 Experiments and Results

We evaluate the performance of the 5GGuardian solution, considering various attack scenarios and employing different evaluation metrics. Our experiments focus on multiple aspects, including selecting the 5GGuardian architecture, comparing its performance in the presence of 5G-Stream and application-layer features, and assessing the detection performance when dealing with contaminated data. Throughout our evaluation, we utilize the

training and test datasets depicted in Table 5.4 for model training and testing, respectively. We evaluate the performance of our model using the F1-score which serves as an effective evaluation metric to assess the model's precision and recall capabilities.

5.6.1 Time Series Transformer Architecture Selection

To determine the optimal architecture for the time series transformer that effectively recognizes the HTTP/2 SMA variations, we train and validate the performance of multiple time series transformer architectures and closely examine their performance. To accomplish this, we allocate 20% of the training dataset as a validation dataset, and we train the model using the remaining portion of the training dataset (Table 5.4). We conduct multiple tests on various time series transformer architectures, and after careful evaluation, we select the architecture with the best performance and convergence and train it using a particular combination of hyperparameters (outlined in Table 5.5) for a total of 200 epochs. Our evaluation includes a one-layer encoder with 12 attention heads and a hidden state size of 12 to capture complex patterns and dependencies in the data, with a decoder that mirrors this structure. We choose the GeLU activation function for its performance benefits. The batch size is set to 16 for optimal gradient estimates, and we employ the BERT model type for its robustness in sequence modeling tasks. To prevent overfitting, we use a dropout rate of 0.1. Finally, the Adam optimizer with a learning rate of 10^{-12} is chosen for its adaptive capabilities, ensuring stable and efficient convergence. This approach allows us to comprehensively evaluate the effectiveness of each architecture in detecting HTTP/2 SMA variations.

We perform a series of tests by varying the number of transformer encoder blocks, hidden state sizes, and attention heads. To prevent overfitting, we implement dropout exclusively on the first dense layer with a rate of 0.1. We utilize an Adam optimizer with a 10^{-12} learning rate, early stopping, and train our model using just normal data in batches

Table 5.5: Time series transformer hyperparameters

Hyperparameter	**Value**
Transformer Encoder Blocks	1
Transformer Decoder Blocks	1
Hidden Activation Function	GeLU
Hidden State Size	12
Attention Head	12
Dropout	0.1
Adam	10^{-12}
Batch	16
Model Type	Bert

of 16 sequences. During the training process, we observe the average reconstruction loss across different epochs for the model on benign, unlabelled data. As shown in Figure 5.8, the training loss and the validation loss start to converge after approximately 90 epochs, indicating a reasonable convergence of the time series transformer model within 200 epochs. Furthermore, by learning the NFs behavior, our training model meets the *adaptive* criterion, demonstrating the efficiency and effectiveness of our approach, as the model training completes within a time of 4.7 seconds for the 200 epochs.

Figure 5.8: Training and validation loss for time series transformer using 5G-Stream features

5.6.2 5GGuardian Performance & Threshold Selection

To assess the detection performance of the time series transformer, we select a threshold $\alpha = 0.0235$ which represents the mean loss of the trained normal data (Section 5.2.3) and which resulted in good detection performance on the validation set.

Using the selected threshold $\alpha = 0.0235$ (Section 5.2.3), we evaluate the model's performance using a test dataset for each attack scenario (Table 5.4). The 5GGuardian detection solution demonstrates high efficacy in detecting various HTTP/2 SMA, including *Random-SMA*, *Uniform-SMA*, *Down-SMA*, *Up-SMA*, and *Release-SMA*. As a result, the 5G-Stream-based anomaly detection model achieves exceptional detection performance, with an average F1-score of 0.98 across HTTP/2 SMA variations (Figure 5.9), underscoring its robustness.

5.6.3 5GGuardian App-Layer Vs. 5GGuardian 5G-Stream

We compare the performance of 5GGuardian against an application-layer-based anomaly detection solution. For that, we develop an application-layer-based time series transformer using the same data employed for 5GGuardian (Section 5.5.3) and extract application-layer features (depicted in Table 5.3 and similar to those used in 5GShield Wehbe et al. (2023)) through data pre-processing. The training dataset consists of 50000 benign records, while the test dataset is 8000 benign records with 2000 malicious records (Table 5.4). Similar to 5GGuardian, we evaluate multiple model architectures and select the one that depicts the best performance. The selected application-layer-based time series transformer architecture aligns with the hyperparameter utilized by 5GGuardian (Table 5.5). As shown in Figure 5.9, 5GGuardian demonstrates superior detection performance in the presence of 5G-Stream features. In particular, 5GGuardian 5G-Stream achieves the highest F1-score of 0.99 for *Random-SMA*, *Down-SMA*, and *Release-SMA* outperforming the application-layer-based features model (5GGuardian App-layer) with an average F1-score of 0.91. This

highlights the effectiveness of 5GGuardian in detecting variations of HTTP/2 SMA when using 5G-Stream features.

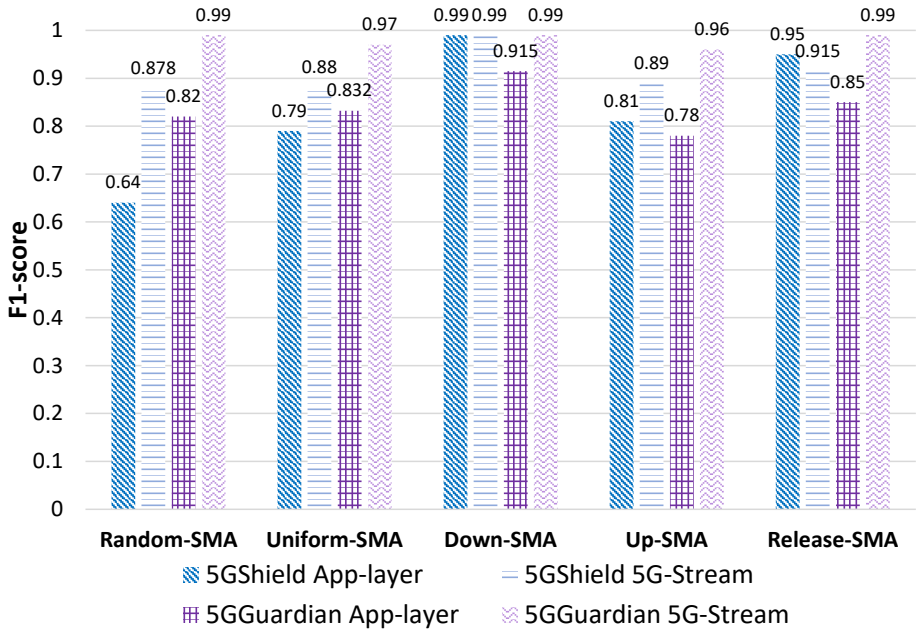


Figure 5.9: F1-score 5GGuardian vs. 5GShield

5.6.4 5GGuardian & 5GShield Comparison

By comparing 5GGuardian and our previous 5GShield model Wehbe et al. (2023), which is based on Autoencoder using application-layer features, we observe distinct detection performances for different variations of HTTP/2 SMA. As shown in Figure 5.9, 5GShield is robust against *Down-SMA* and *Release-SMA*, achieving an F1-score of over 0.95. However, this performance degrades for *Up-SMA*, with an F1-score of 0.81, primarily due to a higher number of false alarms. Notably, 5GShield underperforms for *Random-SMA* and *Uniform-SMA*, with an F1-score below 0.8. Using the 5G-Stream features dataset, we train and test the 5GShield solution, as shown in Figure 5.9. The inclusion of 5G-Stream features enhances the F1-score for HTTP/2 SMA variations, although it still under-performs 5GGuardian 5G-Stream.

With an average F1-score of 0.98, 5GGuardian demonstrates high precision and recall, effectively identifying HTTP/2 SMA variations with minimal false negatives. While an F1-score above 0.95 is generally acceptable for security applications, periodic threshold tuning, adaptive learning, and real-time monitoring are essential to optimize the trade-off between false positives and false negatives. Additionally, low processing overhead (2.3ms per stream) reinforces 5GGuardian's practicality for real-time 5G security, making its detection rates highly acceptable in its intended application.

5.6.5 Training on Contaminated Data

In contrast to our previous test, where we trained our model exclusively on benign data, we now evaluate the performance of our 5GGuardian when trained on partially contaminated data (i.e., a mix of unlabeled benign and malicious data). In this experiment, we mix unlabeled benign data with varying percentages of malicious data (0.1%, 0.5%, 1%, 1.5%, and 2%) to compare and assess the effectiveness of 5GGuardian. We train the time series transformer using the same hyperparameters (Table 5.5) and utilize the test datasets to evaluate the model.

Figure 5.10 illustrates a slight degradation in the F1-score of the 5GGuardian model as the contamination percentage in the training data increases. Nonetheless, contamination exceeds 1%; the F1-score remains consistently above 0.9, which shows its robustness. In contrast, the F1-score of the 5GShield model, as observed in Wehbe et al. (2023), experiences a degradation with the increase of the contamination percentage in the training data. Once the contamination exceeds 1%, the F1-score falls below 0.85 (Figure 5.10). Notably, 5GGuardian outperforms 5GShield Wehbe et al. (2023) in the detection of HTTP/2 SMA variations, particularly in scenarios with higher contaminated data.

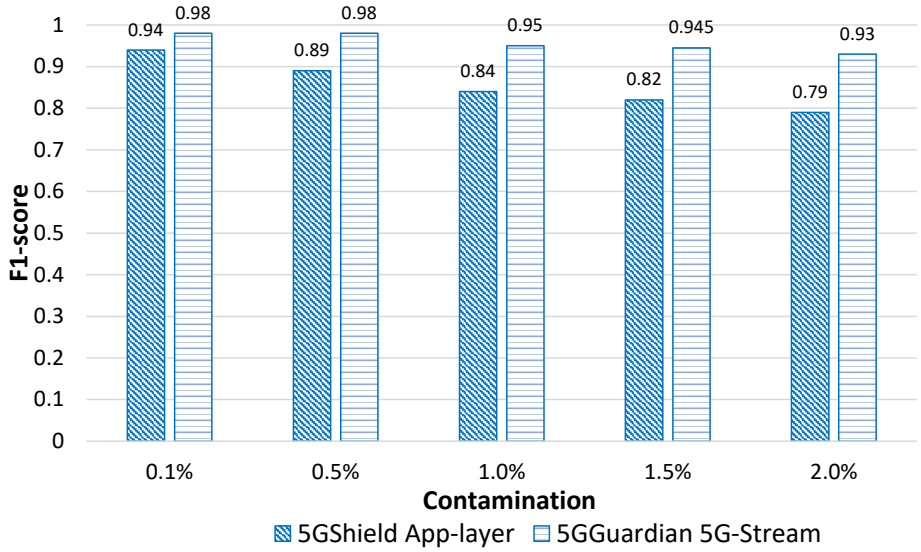


Figure 5.10: F1-score in the presence of contaminated data

5.7 5GGuardian Deployment

5GGuardian leverages stream data collected from NFs to secure the latter against HTTP/2 attacks. To ensure seamless integration of 5GGuardian within the 5G network, we envision its deployment as part of the NWDAF TS.23.288 (2024). NWDAF is a 5G NF, devised by 3GPP to collect data from 5G NFs to train and provision ML models, provide analytics and generate insights to enhance user experience and network functionality. Thus, 5GGuardian can benefit from the data collected by NWDAF and use it for anomaly detection, thus preventing any additional overhead that can result from extra monitoring and data collection that might be devised especially for its operation. When trained and provisioned as part of NWDAF, 5GGuardian can be easily monitored, maintained, retrained, and tested as part of NWDAF maintenance and update procedures. It can also benefit from NWDAF's support of accuracy information and accuracy degradation of ML models to decide on appropriate maintenance actions (e.g., 5GGuardian re-training) TS.23.288 (2024).

5.8 Discussion

While 5GGuardian demonstrates strong detection capabilities, it has certain limitations. Security-wise, it relies on training data quality, and its effectiveness may degrade if adversaries manipulate or poison training datasets. Attackers could also attempt evasion techniques, such as crafting HTTP/2 attack traffic to mimic benign behavior, potentially reducing detection accuracy. Performance-wise, the framework achieves real-time detection with a lightweight time series transformer. Additionally, analyzing 5G-Stream features at scale may incur memory overhead as data volume grows with network expansion. Operationally, integrating 5GGuardian into live 5G networks requires close coordination with NWDAF, demanding engineering efforts for seamless deployment. Model retraining needs ongoing maintenance, as new 5G protocols, configurations, and attack variations emerge. Furthermore, while the model is robust against contaminated data, organizations must ensure continuous validation to prevent false positives impacting legitimate traffic. Cost-wise, deploying 5GGuardian at scale requires investment in hardware acceleration (e.g., GPUs) for high-speed inference and storage solutions for historical data retention.

Chapter 6

HTTP/2 DoS Attacks in 5G Networks: Impact Analysis and Anomaly Detection

In this chapter, we address the lack of practical studies and analyses on the impact of HTTP/2 attacks on 5G networks, especially given the absence of a 5G-compliant dataset for anomaly detection. Utilizing version 2 of the 5G testbed (Subsection 2.3), we emulate six different HTTP/2 attacks on various NFs within the 5G SBA. We analyze their impact on the network and demonstrate that many of them cause cascading effects on other NFs involved in related jeopardized 5G procedures. Our emulations include both malicious and normal network behavior, resulting in the first 5G anomaly detection dataset that we are aware of. Using CICFlowmeter, we extract flow-based features known for their anomaly detection capabilities and train multiple machine learning models. These models can serve as benchmarks for detecting HTTP/2 attacks in 5G networks.

6.1 Threat Model

Although secure by design, the 5G SBA can still experience some attacks resulting from virtualization exploits, misconfigurations, and its HTTP/2 signaling protocol vulnerabilities. In the following, we shed light on some HTTP/2 attacks in 5G networks while detailing the vulnerabilities they exploit and their related threat models.

6.1.1 Assumptions

HTTP/2 attacks in 5G networks can be performed through misconfigured or compromised NFs. We consider the following assumptions for the HTTP/2 attacks, assuming that attackers compromise the NFc and use it to attack the NFp or vice versa.

(1) *Attacker compromises an NFc:* Many standardization documents discuss threats brought by NFV and virtualization technologies (e.g., container, virtual machines, etc.) to telecommunication networks and 5G ETSI (2020). The adoption of hyper-scale cloud by mobile operators extends the attack surface of their network and makes their NFs vulnerable ETSI (2020). An attacker can compromise 5G NFs deployed on docker containers in the cloud, by exploiting docker vulnerabilities to perform container escape (i.e., CVE-2016-5195 (NVD) (2019), CVE-2019-5736 (NVD) (2016), and CVE-2023-20864 National Vulnerability Database (2023)) Madi et al. (2021). Breach of isolation between network slices sharing the NF can also be exploited by attackers AdaptiveMobile (2021); Sattar, Vasoukolaei, Crysdale, and Matrawy (2021). In such a scenario, we assume that the malicious actor belonging to a roaming partner launches the HTTP/2 attack towards the home network AdaptiveMobile (2021).

(2) *NFc can successfully authenticate with the NFp:* We assume that if TLS is used, the malicious NFc can still authenticate with the NFp as the attacker has access to its

public/private key pairs.

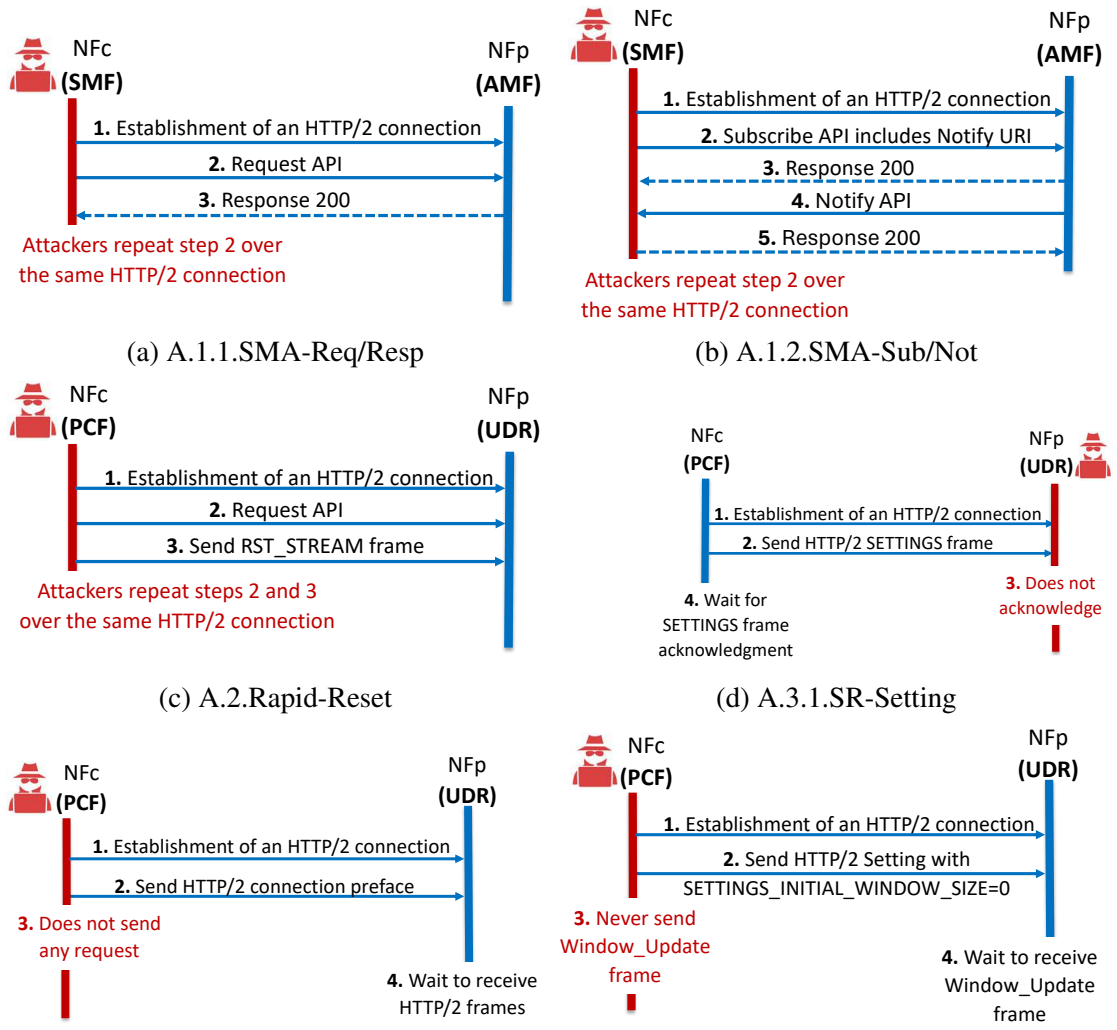
(3) *NFc is authorized to access NFp services:* We assume that the malicious NFc has already acquired OAuth2.0 access tokens to the NFp services. These tokens are cached and can be reused by the attacker TS.33.501 (2025a); TSG-SA3 (2022). Alternatively, the malicious NFc can request new access tokens from the NRF given that it can successfully authenticate with it (i.e., assumption (2)). An attacker can exploit vulnerabilities in network slicing and service authorization, as noted in AdaptiveMobile (2021), to access NFp services.

(4) *Attacker has access to UE information:* As some network services require exchanging UE information (e.g., SUPI) TS.123.502 (2025), we assume that the attacker can gain access to such information by monitoring NFc communications or even by requesting such information from other NFs.

These attacks are not new or novel. Although exploited in the web, exploiting these attacks in a 5G environment requires attacking a 5G-specific API, making these attacks more challenging to perform than on the web, where APIs are not necessarily used. Furthermore, their impact on a 5G network can be more disruptive than in a web environment, given the dependencies and interactions existing between the different 5G NFs, as we show in Section 6.3.

6.1.2 Attack 1: HTTP/2 Stream Multiplexing Attack (SMA)

To perform an HTTP/2 stream multiplexing attack, attackers send multiple requests, as much as the NFp allows in the HTTP/2 SETTINGS_MAX_CONCURRENT_STREAMS, into a single HTTP/2 connection. By default, the NFc can send up to $2,147,483,647$ (default value of SETTINGS_MAX_CONCURRENT_STREAMS) streams per HTTP/2 connection IETF (2015). Attackers can trigger HTTP/2 SMA in two ways within 5G SBA, either by employing the *Request/Response* or the *Subscribe/Notify*.



(a) A.1.1.SMA-Req/Resp

(b) A.1.2.SMA-Sub/Not

(c) A.2.Rapid-Reset

(d) A.3.1.SR-Setting

(e) A.3.2.SR-Con-Pref

(f) A.3.3.SR-Win-Size

Figure 6.1: HTTP/2 attacks in 5G SBA

A. Attack 1.1: SMA Request/Response (A.1.1.SMA-Req/Resp)

In an HTTP/2 SMA using *Request/Response* (Figure 6.1a), attackers compromise NFc and send multiple requests over a single HTTP/2 connection towards NFp. Attackers repeat this behavior over multiple HTTP/2 connections which results in a DoS on the NFp.

B. Attack 1.2: SMA Subscribe/Notify (A.1.2.SMA-Sub/Not)

According to 3GPP 3GPP TS.29.500 (2024), the *Subscribe/Notify* service operations in HTTP/2 involve two HTTP/2 connections, each handling one direction of traffic. NFc acts as an HTTP/2 client when subscribing to notifications, while NFp functions as an HTTP/2 server. Conversely, the roles are reversed when NFp sends notifications to NFc. As depicted in Figure 6.1b, a compromised NFc establishes an HTTP/2 connection with NFp and sends a subscription request containing a notify URI to signal to the NFp to notify it when the occurrence of the API-related event (e.g., N1N2TransferFailureNotification TS.129.518 (2025) is triggered). Attackers exploit the event conditions (i.e., UE state is DISCONNECTED TS.129.518 (2025)) to initiate the notification. Attackers repeat the request with the notification URI to cause an SMA and overwhelm both NFp and NFc. The NFc will be receiving an excessive number of notifications causing a DoS, while the NFp struggles with the high number of requests and from managing and forwarding the notifications to the NFc, eventually leading to resource exhaustion and DoS on the NFp.

6.1.3 Attack 2: HTTP/2 Rapid Reset Attack (A.2.Rapid-Reset)

A.2.Rapid-Reset, identified as CVE-2023-44487 National Vulnerability Database (NVD) (2023), exploits the stream multiplexing feature of HTTP/2. It employs the `RST_STREAM` frame to terminate streams that are currently processing requests IETF (2015). In this case, the number of streams that were reset by the `RST_STREAM` frame do not count towards `SETTINGS_MAX_CONCURRENT_STREAMS`. The mitigation for this attack considers counting any request reaching the server, even if it is a `RST_STREAM` frame, as part of the defined maximum stream limit. It involves limiting the number of simultaneously executing handler routines (`SETTINGS_MAX_CONCURRENT_STREAMS`= 200) and prevents server overload by queuing incoming requests until a current request is completed. If the queue becomes excessively long, the server terminates the connection as a safeguard.

However, increasing the SETTINGS_MAX_CONCURRENT_STREAMS slightly could significantly impact network performance.

In this attack (Figure 6.1c), attackers compromise the NFc and establish an HTTP/2 connection with the NFp. In this work, we assume that the reset attack is patched, however, we assume that the NFp is misconfigured to allow an unusually high number of concurrent streams (e.g., SETTINGS_MAX_CONCURRENT_STREAMS = 1000 instead of the default 200 set in the golang library). The malicious NFc then rapidly generates requests and immediately issues RST_STREAM frames for each request across multiple HTTP/2 connections, forcing the NFp to terminate the requests. This flood of reset stream requests can lead to resource exhaustion at the NFp.

6.1.4 Attack 3: HTTP/2 Slow Rate Attacks

Another type of HTTP/2 attack is an HTTP/2 slow rate DoS attack which involves attackers sending HTTP/2 frames at a deliberately slow rate to exhaust NFp resources Chatzoglou, Kouliaridis, Kambourakis, Karopoulos, and Gritzalis (2023); Tripathi (2022); Tripathi and Hubballi (2018). HTTP/2 slow rate attacks require low bandwidth and are difficult to detect. Attackers exploit the HTTP/2 frame between the NFc and NFp, such as the exchange of *SETTINGS* frame, capitalizing on the design of NFp which waits for certain responses. In this work, we target three variations of HTTP/2 slow rate attacks.

A. Attack 3.1: Slow Rate Setting (A.3.1.SR-Setting)

HTTP/2 slow Rate-Setting is a slow rate attack that is based on un-acknowledging a SETTINGS frame. In a normal HTTP/2 communication scenario, both endpoints must exchange *SETTINGS* frames at the start of a connection and may send them at any other time during the connection. *SETTINGS* frame allows each endpoint to acknowledge the parameters of the connection. When an endpoint receives a *SETTINGS* frame, it should

send an acknowledgment response that tells the sender that the *SETTINGS* frame was received and processed. Thus, the slow Rate-Setting attack mainly takes advantage of the *SETTINGS* frame to let the endpoint wait. As depicted in Figure 6.1d, attackers compromise the NFp that has already been authenticated and authorized to access NFc services. The NFc initiates the first HTTP/2 connection with the compromised NFp, followed by sending a *SETTINGS* frame. However, the malicious NFp does not acknowledge the received *SETTINGS* frame. NFc continues to send numerous requests of *SETTINGS* frame to NFp. Since the malicious NFp consistently fails to acknowledge the HTTP/2 *SETTINGS* frame for all messages received, it can exhaust the available connection pool. This not only blocks other NFs from communicating with the victim NFc but also keeps the connection from NFc open for an extended period.

B. Attack 3.1: Slow Rate Connection Preface (A.3.2.SR-Con-Pref)

The connection preface is sent from the NFc to inform the NFp that HTTP/2 will be used for further communications. In this attack (Figure 6.1e), after establishing an HTTP/2 connection, a compromised NFc sends the connection preface to the NFp, prompting it to wait for a GET/POST HTTP/2 request. However, the malicious NFc intentionally withholds any HTTP/2 requests, forcing the NFp to wait until the NFp drops the connection, thus wasting its resources and denying its service to other NFs.

C. Attack 3.1: Slow Rate Window Size (A.3.3.SR-Win-Size)

In a standard HTTP/2 connection, both endpoints are required to send an HTTP/2 payload that includes a *SETTINGS* frame with the *SETTINGS_INITIAL_WINDOW_SIZE* field, along with a complete GET request. The *SETTINGS_INITIAL_WINDOW_SIZE* field specifies the sender's capacity to receive data in bytes from its peer. Upon receiving this, the NFp expects that the NFc can receive data of the indicated size. However, attackers exploit

this mechanism for a slow rate attack by compromising the NFc (Figure 6.1f). After establishing the HTTP/2 connection, the malicious NFc sends a *SETTINGS* frame with the *SETTINGS_INITIAL_WINDOW_SIZE* set to zero, falsely indicating no available window space for data reception. The NFp, in turn, holds the data until it receives a *WINDOW_UPDATE* frame that increases the window size. Nonetheless, the malicious NFc intentionally never sends a *WINDOW_UPDATE*, thus causing the NFp to wait till the connection pool is full, resulting in dropping the connection. Thus, this attack exhausts the available connection pool at the NFp and blocks its service for other legitimate UEs.

6.2 Environment Setup

In this section, we present the environment that we use to emulate normal and malicious network traffic.

6.2.1 Emulation Configuration

Given the lack of publicly accessible datasets for anomaly detection in the 5G SBA, we employ our emulated 5G testbed (Section 2.3) to emulate normal and malicious network behaviors. For normal network behavior, we replicate the standard activities of UEs within our 5G testbed by leveraging different 5G procedures implemented in UERANSIM (Table 6.1). These same procedures are also employed to model HTTP/2 attacks and generate malicious traffic.

6.2.2 Normal Network Behavior Emulation

To emulate normal network behavior, we consider the arrival of 100 UEs to the network using a Poisson process Navarro-Ortiz et al. (2020); Raaijmakers et al. (2021) over two hours. The Poisson process is widely recognized as an effective method for modeling

Table 6.1: Logical dependency between 5G procedures

Triggered procedure	**Possible subsequent procedures**
Registration	Uplink, Downlink, UE release PDU session, gNodeB release PDU session, Deregistration
Uplink	Downlink, UE release PDU session, gNodeB release PDU session, Deregistration
Downlink	Uplink, UE release PDU session, gNodeB release PDU session, Deregistration
UE release PDU session	Downlink, Uplink, gNodeB release PDU session, Deregistration
gNodeB release PDU session	Uplink, Downlink, Deregistration
Deregistration	Registration

arrival times of events in network traffic due to its ability to capture the randomness of user behavior and service requests over time. For our implementation, we defined the load (number of requests) per 10-minute intervals as [1, 2, 3, 5, 6, 7, 8, 9, 7, 5, 3, 0.5], to reflect the dynamic nature of 5G network traffic. We use 100 UEs in each 10-minute interval, representing one predefined network load. For each load value, we emulate a number of 5G procedures calculated based on the Poisson process for each of these 100 UEs. This approach follows the principles outlined in Mehmeti and La Porta (2022), where Poisson processes are used to model the arrival of UE requests in realistic network scenarios. Each UE engages in one or more 5G procedures selected from a set provided by UERANSIM (Table 6.1). To ensure that our emulation of 5G normal network behavior is realistic, we follow the 3GPP standard definition of the different 5G procedures and their logical dependencies 3GPP 5G Standard (2025); TS.123.502 (2025). We limit these procedures to those available in UERANSIM and which we can emulate. Table 6.1 defines the possible subsequent procedure for each triggered procedure by the UE following the 3GPP standard. Given that 5G procedures have logical dependencies and specific order requirements defined by the 3GPP standard, in our emulation, we choose a subsequent procedure $(p+1)$ that follows a procedure p for a UE by randomly selecting it from a predefined list of procedures that logically follow p. The list of the possible subsequent procedures for each 5G

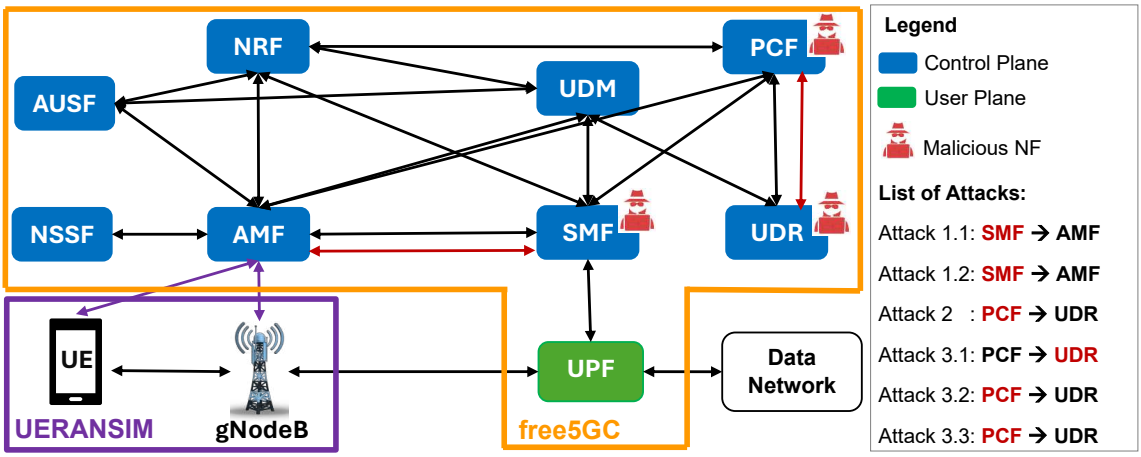


Figure 6.2: 5G Testbed with normal and malicious network behaviors

operation in UERANSIM, is outlined in Table 6.1. For instance, a UE cannot proceed to deregistration if it has not completed its registration. Moreover, each 5G procedure triggers various communication between NFs. These communications may vary based on the UE state (e.g., CONNECTED, IDLE, DISCONNECTED) and other factors such as network conditions and RAN resources TS.123.502 (2025).

In our emulation, each UE starts by first registering to the network and then selects a subsequent procedure as detailed in Table 6.1. Hence, the following procedure is randomly chosen from the appropriate options, which include Uplink, Downlink, UE release PDU session, gNodeB release PDU session, and Deregistration procedures. After the registration, let us assume, for example, that the gNodeB release PDU session procedure was selected by the UE. Following its execution, either Uplink, Downlink, or Deregistration procedures can be initiated. Note that these procedures are triggered for the same UE at different times to replicate 5G communications and can switch the UE between different states. For example, (1) UE registers to the network; after a certain period of time, (2) RAN releases the PDU resources allocated to the UE, switching its state to IDLE; (3) Subsequently, a Downlink procedure is triggered from the network to signal to the UE which is in IDLE state, hence, switching its state to CONNECTED.

Figure 6.2 highlights the interactions between pairs of NFs observed within our 5G testbed (i.e., control plane). We extract the total number of messages reflecting these interactions between pairs of NFs during 20 minutes of emulations of different 5G procedures and present them in Figure 6.4. The latter shows that interactions involving the AMF, SMF, UDR, and PCF are more frequent, reflecting the intensive activity associated with Uplink, Downlink, and UDR management procedures during our normal network traffic. This data is crucial as it represents the peak demands for each interaction, offering insights into network load during typical operations.

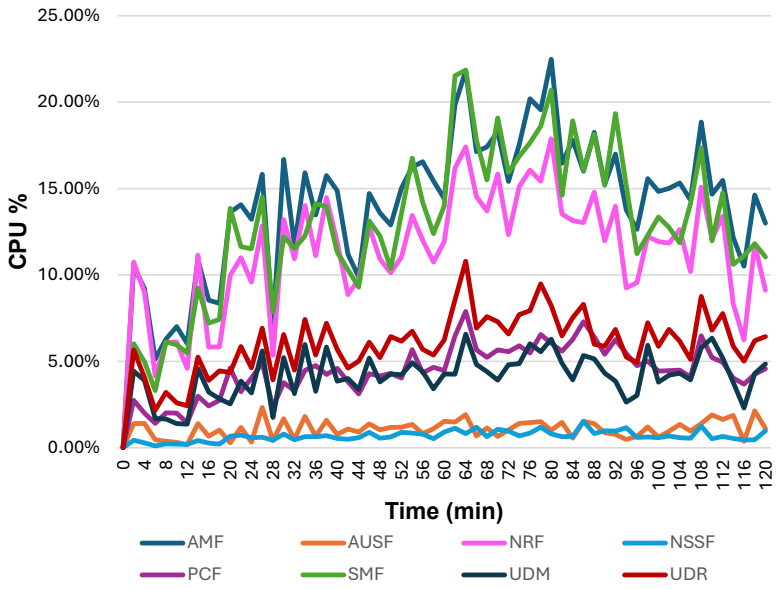


Figure 6.3: Benign network traffic - 5G SBA NFs CPU consumption

Along with observing NFs interactions, we monitor resource utilization during normal network traffic by tracking CPU consumption across various NFs over two hours, as shown in Figure 6.3. We observe that although the CPU load of the different NFs remains under 25%, AMF, SMF, and UDR exhibit higher CPU consumption than other NFs which can be explained by the high number of requests they manage (Figure 6.4).

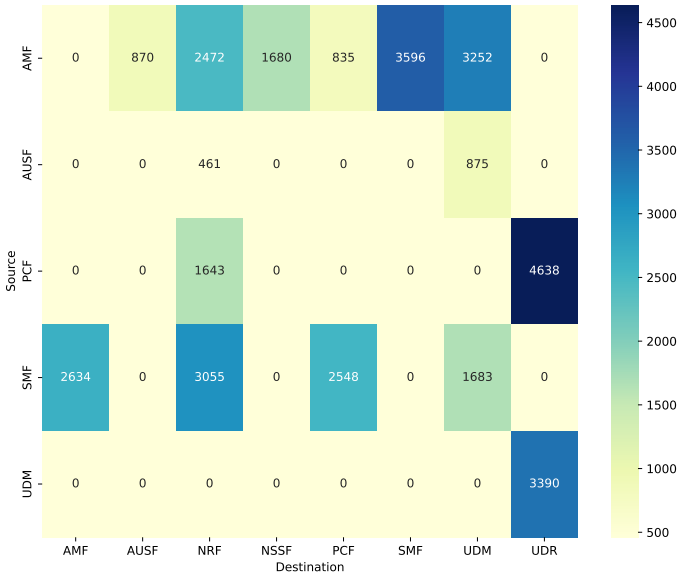


Figure 6.4: Benign network traffic - Total number of requests between pairs of NFs

6.2.3 Malicious Network Behavior Emulation

In our 5G testbed, we emulate various HTTP/2 attacks to expose potential vulnerabilities within the 5G SBA. Specifically, we target procedures such as Uplink, Downlink, UE release PDU session, and UDR Management, frequently used in our 5G testbed. We emulate HTTP/2 attacks (Section 6.1) where attackers compromise NFc/NFp, such as a PCF, UDR, or SMF, as shown in Figure 6.2, in addition to UE information (i.e., SUPI). We assume that attackers exploit 30 legitimate UEs out of 100. Through the compromised NFc, each attack is launched using the IMSI of the 30 legitimate UEs, where multiple HTTP/2 connections are established toward the NFp. These connections are configured with a default *SETTINGS_MAX_CONCURRENT_STREAMS=200* in our 5G testbed. Thus, in our attack emulations, the network operates normally for the first 60 minutes, after which the attack is initiated when the load on the network is designed to be around its peak ($load = 8$).

A. Attack 1: HTTP/2 Stream Multiplexing Attack

For instance, Figure 6.1a involves a malicious SMF that randomly triggers various procedures toward the AMF, adhering to HTTP/2 protocol precedence constraints IETF (2015); TS.129.518 (2025). For *A.1.1.SMA-Req/Resp*, we use three different procedures, that are triggered from the malicious SMF towards the AMF using the same *Namf_Communication_N1N2MessageTransfer* API, such as Uplink, downlink, and UE release PDU session. Note that this API covers most of the service operations provided by the AMF and consumed by the SMF TS.129.518 (2025). As attackers, we repeat this attack over $55,954$ HTTP/2 connections, each handling up to 907 requests, resulting in an NFp overload and a DoS. The second attack scenario (Figure 6.1b) considers an SMA that involves a malicious SMF exploiting 30 UEs by triggering only the Downlink procedure using *Namf_Communication_N1N2MessageTransfer* API, however, we include a notify URI for DISCONNECTED UEs. According to 3GPP specifications TS.129.518 (2025), when the Downlink procedure is initiated while the UE state is DISCONNECTED, the *N1N2TransferFailureNotification* API is triggered to notify SMF that the UE is unreachable TS.129.518 (2025). Consequently, the AMF sends a notification back to the malicious SMF. We emulated this attack using $54,188$ HTTP/2 connections, each handling up to 841 requests over 40 minutes before the AMF goes down. This attack effectively exploits the signaling mechanisms of the network, leading to a DoS on the AMF. The continuous failure notifications overwhelm the AMF, making it unresponsive and crippling NFs, degrading the Quality of Service (QoS) for legitimate UEs.

To better illustrate how we perform *A.1.2.SMA-Sub/Not*, we illustrate in Figure 6.5 the normal Downlink procedure triggered from the DN when the UE is in the DISCONNECTED state TS.129.518 (2025), and highlight in red how an attacker can perform the attack as in our emulation assuming that the SMF was compromised. In a normal scenario,

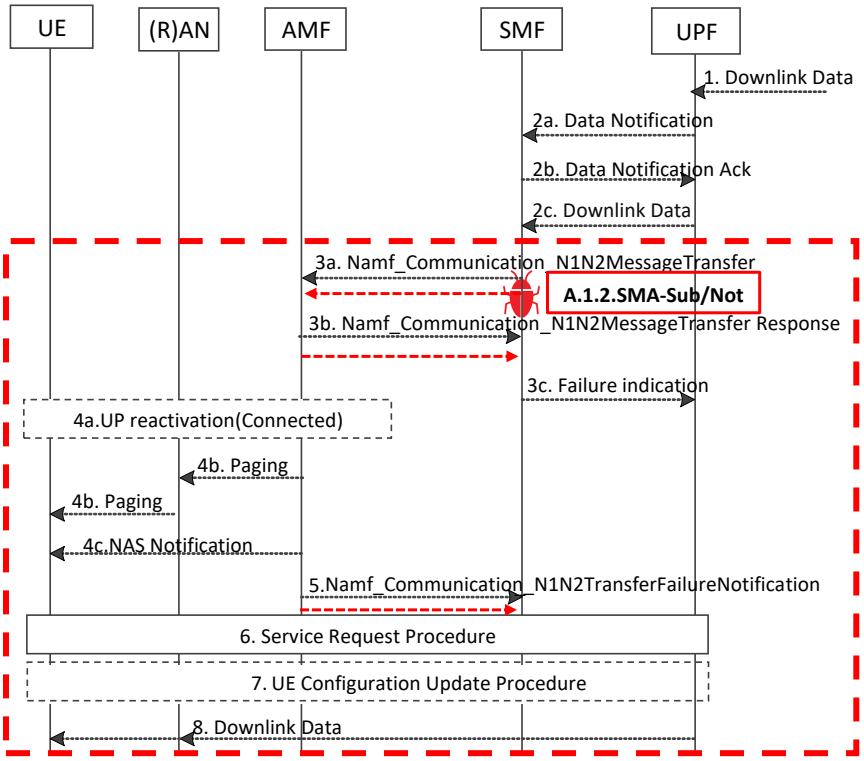


Figure 6.5: A.1.2.SMA-Sub/Not emulation in network triggered service request procedure TS.129.518 (2025)

the SMF sends a request to the AMF using *Namf_Communication_N1N2Mess ageTransfer* API (Figure 6.5 (3a)). The AMF responds to the SMF indicating that the UE is not reachable, and subsequently sends a Paging Request to the UE/RAN (Figure 6.5 (4b)). The Paging Request triggers the Uplink procedure to activate the UE. In the dashed red box, we highlight a scenario where an attacker compromises the SMF and launches a malicious Namf_Communication_N1N2MessageTransfer (Figure 6.5 (3a)) subscription request towards the AMF containing a notify URI to signal to the NF producer to notify it when the occurrence of the API-related event (e.g., Namf_Communication_N1N2TransferFailureNotification (Figure 6.5 (5)) is triggered). Although the attacker only initiates a single request, it results in a chain of other messages (Figure 6.5 (3b-8)) related to paging, service request, PDU session update, and PDU session modification in the 5G network. This depicts the high overhead that an attacker can introduce to the network with a single malicious request.

Hence, the attacker can repeat the request with the notification URI to cause an SMA and overwhelm both AMF and NF SMF. The SMF will be receiving an excessive number of notifications, while the AMF struggles with the high number of requests and from managing and forwarding the notifications to the SMF, eventually leading to a resource exhaustion and a DoS on the AMF.

B. Attack 2: HTTP/2 Rapid Reset Attack

In our emulation of *A.2.Rapid-Reset*, a compromised PCF targets the UDR using a UDR Management procedure TS.123.502 (2025). We assume that the UDR sets its HTTP/2 connection with *SETTINGS_MAX_CONCURRENT_STREAMS= 1000*. As depicted in Figure 6.1c, the PCF sends to the UDR a request using *Nudr_DataManagement* API followed by a `RST_STREAM` frame to stop the sent request. The malicious PCF establishes around $263,251$ HTTP/2 connection with the UDR over 2 hours, with up to 2306 requests and `RST_STREAM` frame per connection. This action aims to create a DoS situation, effectively disrupting the network's operations and impacting its ability to process legitimate requests.

C. Attack 3: HTTP/2 Slow Rate Attack

We emulate three variations of HTTP/2 slow rate attack from PCF to UDR. As depicted in Figure 6.1d, to emulate the *A.3.1.SR-Setting*, the PCF establishes around $3,947$ HTTP/2 connections with the UDR and sends to it on each of them a *SETTINGS* frame. As the UDR is malicious, it does not acknowledge the SETTINGS frames sent by the PCF, leading to a backlog of unacknowledged frames, hence, causing a drop of these connections after a certain timeout time. We emulate the *A.3.2.SR-Con-Pref* (Figure 6.1e) by accounting for a malicious PCF that sends a connection preface to the UDR without following it by any HTTP/2 GET/POST. This makes the UDR wait endlessly for an HTTP/2 request that never

arrives. This scenario is repeated over $5,733$ HTTP/2 connections and exhibits similar behavior to normal network traffic. During *A.3.3.SR-Win-Size* (Figure 6.1f), the malicious PCF establishes the first HTTP/2 connection and sends a manipulated HTTP/2 SETTINGS with *SETTINGS_INITIAL_WINDOW_SIZE* equal to zero, signaling that the PCF can no longer receive data. This manipulation forces the UDR to halt all data transmissions until it receives a *WINDOW_UPDATE* frame, effectively freezing the data flow. The malicious PCF repeats this attack over $3,815$ HTTP/2 connections severely impacting the 5G network availability.

6.3 Attacks Impact & Prevention

Using the data collected during the malicious network traffic, we detail the impacts of observed malicious behaviors on the 5G SBA, as summarized in Table 6.2.

6.3.1 HTTP/2 Attacks Impact

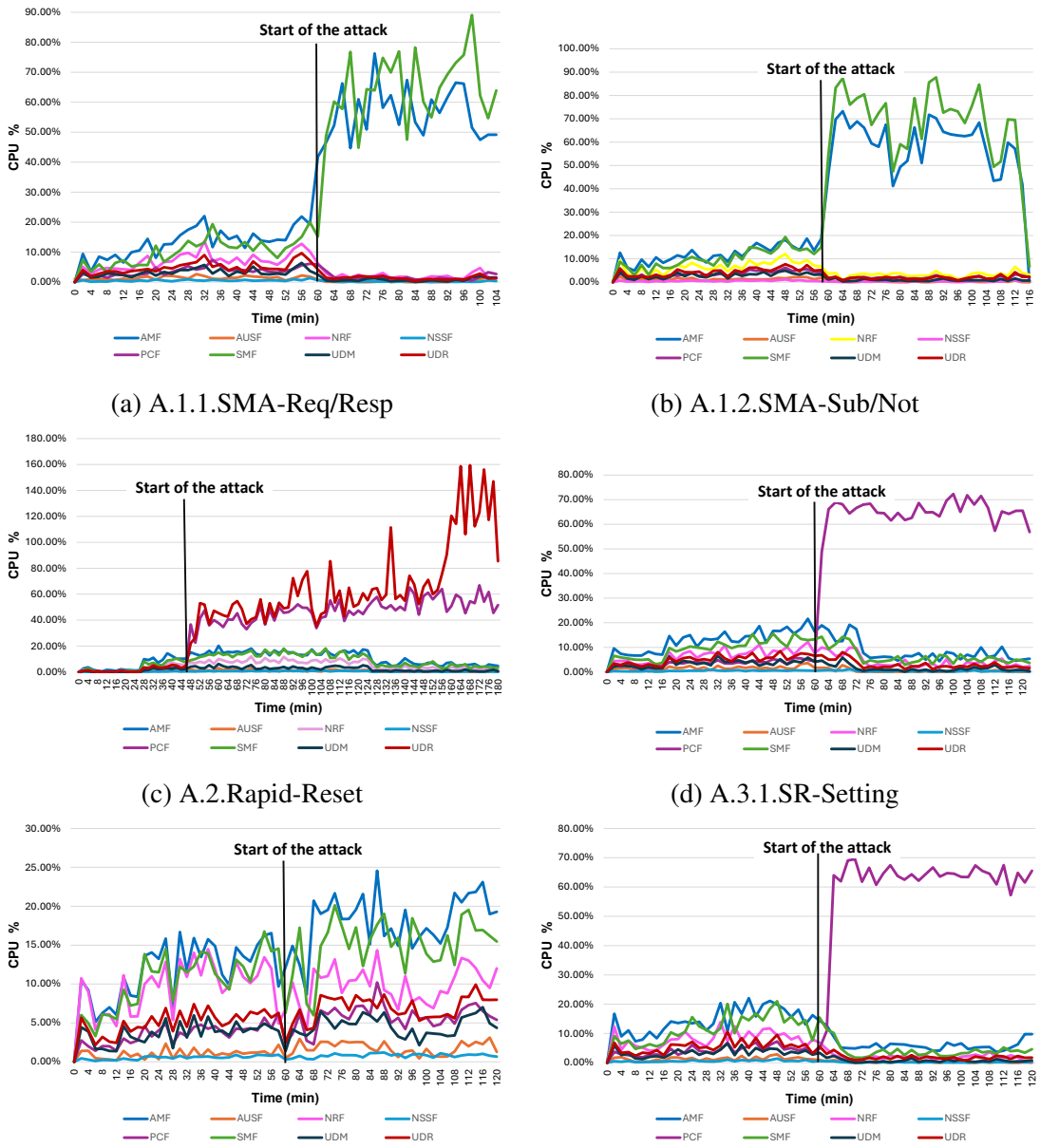
Our emulations of HTTP/2 attacks on 5G SBA present various impacts on NFs resource utilization that we measure through observing the CPU consumption of the different NFs in Figure 6.6. Additionally, we measure the total number of messages (i.e., requests and notifications only) exchanged per each pair of NFs within the 5G SBA during 20 minutes of the different attacks as shown in Figure 6.7. This metric reflects the volume of control signaling traffic impacted by the HTTP/2 attacks especially when compared to the benign traffic captured for the same period during the benign emulation. We analyze in the following the impact of the different attacks on 5G networks, suggest prevention and mitigation measures, and summarize them in Table 6.2.

Table 6.2: HTTP/2 attacks, impact and protection measures in 5G SBA

Attack Type	Description	Impact on the network	Protection measures
Attack 1.1-SMA-Request /Response	**SMF -> AMF:** A malicious SMF initiates multiple 5G requests towards an AMF by exploiting the stream multiplexing feature over many HTTP/2 connections.	Overload of attack NFs (i.e., SMF, AMF), complete DoS on the network.	Anomaly detection systems, limiting `SETTINGS_MAX_CONCURRENT_STREAMS` on NFp.
A.1.2.SMA-Sub/Not	**SMF -> AMF:** A malicious SMF exploits the stream multiplexing feature by triggering the Downlink procedure for DISCONNECTED UEs towards the AMF with a notification subscription. AMF then informs the SMF of UE's unreachability using the N1N2TransferFailureNotification API.	Overload of attack NFs (i.e., SMF, AMF), complete DoS on the network.	Intelligent adjustment of `SETTINGS_MAX_CONCURRENT_STREAMS` value, anomaly detection systems.
A.2.Rapid-Reset	**PCF -> UDR:** A malicious PCF initiates the UDR Management procedure towards the UDR to trigger a registration procedure and immediately sends a `RST_STREAM` frame.	Overload of attack NFs (i.e., PCF, UDR), degradation of other NFs QoS.	Limiting the number of `RST_STREAM` frames received on NFp, Anomaly detection systems.
A.3.1.SR-Setting	**PCF -> UDR:** The PCF sends multiple requests to the malicious UDR, starting with a *SETTING* frame, but the malicious UDR does not acknowledge any received messages.	Resource exhaustion at targeted NF, degradation of other NFs QoS.	Anomaly detection systems, intelligent monitoring systems, and timer values to drop or close malicious connections.
A.3.2.SR-Con-Pref	**PCF -> UDR:** A malicious PCF sends a connection preface to the UDR but never sends any HTTP/2 requests, causing the UDR to wait until its connection pool is full, eventually dropping the connection.	Similarities to normal network traffic, a longer period of unnoticed resource depletion impact.	Anomaly detection systems.
A.3.3.SR-Win-Size	**PCF -> UDR:** A malicious PCF sends an HTTP/2 SETTINGS frame with *SETTINGS_INITIAL_WINDOW_SIZE=0* to the UDR, halting all data transmission. The UDR waits indefinitely for a *WINDOW_UPDATE* frame, which the malicious PCF intentionally never sends.	Resource and connection pool exhaustion at targeted NF, degradation of other NFs QoS.	Anomaly detection systems, intelligent monitoring systems and timer values to drop or close malicious connections.

A. Attack 1: HTTP/2 Stream Multiplexing Attack

Upon the start of *A.1.1.SMA-Req/Resp* (Figure 6.6a) and *A.1.2.SMA-Sub/Not* (Figure 6.6b) at time 60 (i.e., after around an hour of emulations), the CPU usage of the AMF and SMF increases sharply, while the CPU usage for the rest of the NFs decreases. This is mainly



(a) A.1.1.SMA-Req/Resp

(b) A.1.2.SMA-Sub/Not

(c) A.2.Rapid-Reset

(d) A.3.1.SR-Setting

(e) A.3.2.SR-Con-Pref

(f) A.3.3.SR-Win-Size

Figure 6.6: 5G SBA NFs CPU consumption during malicious network behavior

attributed to the overload at the AMF and SMF, delaying and potentially blocking the completion of 5G procedures that are stuck at the SMF-AMF interactions. As shown in Figure 6.6a, at time 104 (i.e., after 44 minutes of the start of the attack), the AMF fails, resulting in a DoS. Further, the regular CPU spikes during *A.1.2.SMA-Sub/Not* (Figure 6.6b) highlight intense activity periods that stress the SMF and potentially degrade services of

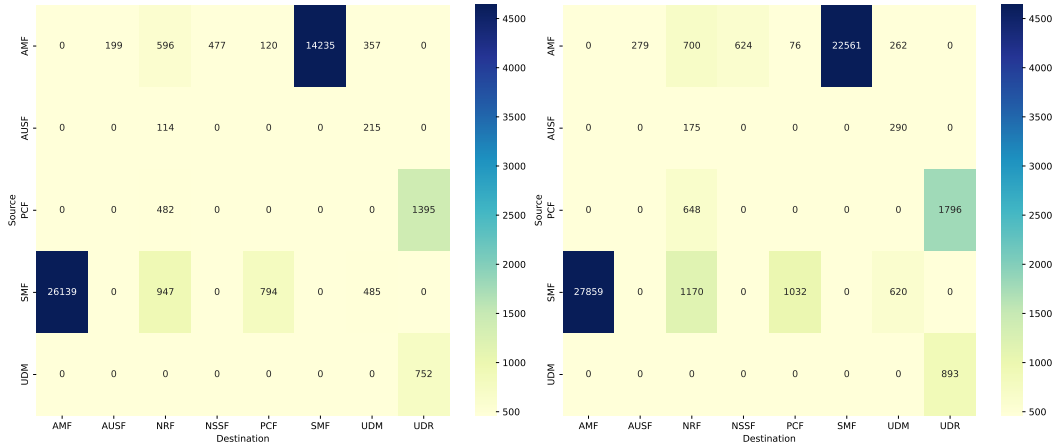
other NFs. Notably, we observe the highest number of messages during SMF-AMF interaction due to frequent attack requests from the SMF towards the AMF as illustrated in Figure 6.7a and Figure 6.7b. Additionally, the number of messages in the AMF-SMF interactions during *A.1.2.SMA-Sub/Not* (Figure 6.7b) is higher than during the *A.1.1.SMA-Req/Resp* (Figure 6.6a) due to the notifications sent from the AMF to the SMF. In summary, SMA attacks significantly impact network availability by exhausting its resources, and causing a DoS on the targeted NF and potentially on the whole 5G network.

B. Attack 2: HTTP/2 Rapid Reset Attack

Although the *A.2.Rapid-Reset* was emulated while the official patch was deployed in our network, we notice that the increase in the SETTINGS_MAX_CONCURRENT_STREAMS (Section 6.1.3) can still disrupt the provided QoS, not only by overloading the attack target (i.e., UDR) but also the attack source (i.e., PCF). Figure 6.6c shows high CPU consumption at the UDR and PCF that often reaches 80% for an hour. However, after 2 hours of running the attack, we observe CPU spikes reaching 160% between times 120 and 180, indicating moments of intense load on the UDR, always accompanied by a high load on the PCF. This is also reflected by the high number of messages exchanged between PCF and UDR in Figure 6.7c. However, a lower CPU consumption is observed at the remainder NFs between times 120 and 180 reflecting a DoS attack on the network and a degradation of the QoS of those NFs (Figure 6.6c).

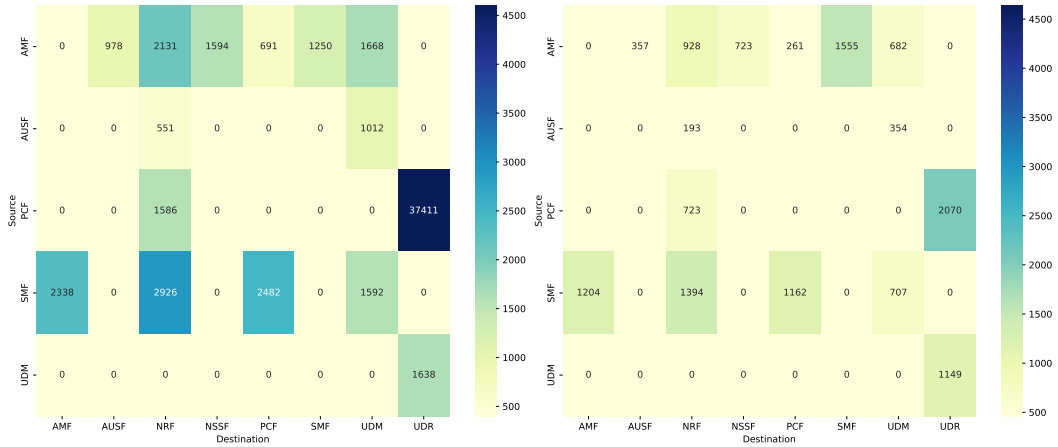
C. Attack 3: HTTP/2 Slow Rate Attack

When examining the variations of HTTP/2 slow rate attacks, we notice their distinct impacts on the NFs within the 5G SBA, starting at attack time 60. Figure 6.6d and Figure 6.6f reflect a high CPU consumption on the targeted NF, the PCF, with a degradation



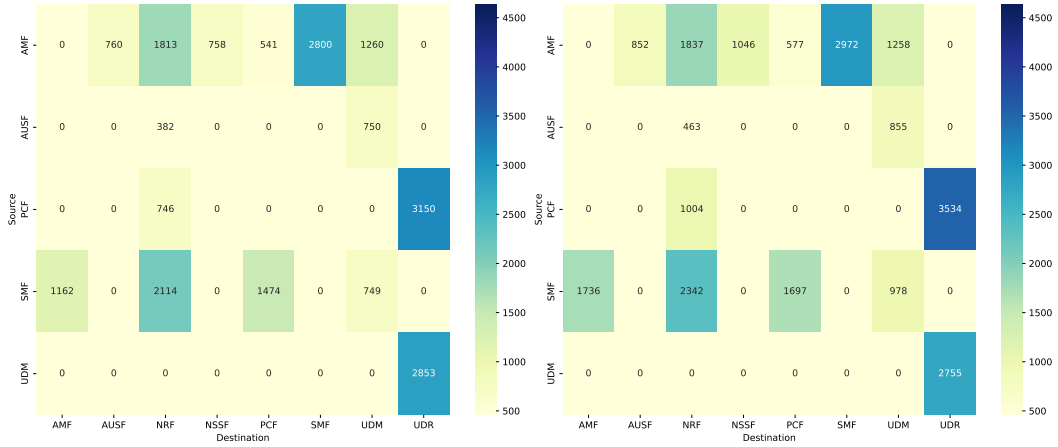
(a) A.1.1.SMA-Req/Resp

(b) A.1.2.SMA-Sub/Not



(c) A.2.Rapid-Reset

(d) A.3.1.SR-Setting



(e) A.3.2.SR-Con-Pref

(f) A.3.3.SR-Win-Size

Figure 6.7: Total number of messages between pairs of NFs in 5G SBA during malicious network behavior

of the CPU consumption on the other NFs, reflecting a degradation of the network availability and QoS without observing a total DoS. Although the PCF CPU consumption is comparable in both, *A.3.1.SR-Setting* and *A.3.3.SR-Win-Size*, we notice from Figure 6.7d and Figure 6.7f that the number of messages exchanged between UDR and PCF is higher in *A.3.3.SR-Win-Size* than in *A.3.1.SR-Setting*. This explains that the high CPU consumption is not a result of the number of exchanged messages but rather of resources allocated during the waiting times at the PCF for a *SETTINGS* acknowledgment in case of *A.3.1.SR-Setting* and for a *WINDOW_UPDATE* in case of *A.3.3.SR-Win-Size*.

A.3.2.SR-Con-Pref does not exhibit a significant impact on the CPU of the different NFs, as illustrated in Figure 6.6e. In contrast, although the trend is different, the CPU consumption is comparable to the benign network behavior shown in Figure 6.3. This indicates that detecting this attack may be more challenging. By observing Figure 6.7d, Figure 6.7e, and Figure 6.7f, we note that the total number of messages exchanged during the different HTTP/2 slow rate attacks is lower than that exchanged during normal network operations (Figure 6.3). This demonstrates that simply counting messages between NFs is insufficient to detect manipulation in the HTTP/2 flow, particularly in HTTP/2 frames.

6.3.2 Discussion & Protection Measures

The analysis of HTTP/2 attacks on our 5G testbed reveal that HTTP/2 SMA attacks are the most damaging due to the DoS impact they cause on the targeted NF and on the 5G network as a whole. More specifically, degradation of the performance of the 5G NFs not directly involved in the attack is observed and is highly related to the type of the targeted NFs. For instance, the emulated 5G procedures (Table 6.1) involve many interactions between the AMF and SMF that cause a bottleneck for the completion of these procedures which was reflected by a degradation of the CPU consumption of other NFs which were not receiving as many messages as during the normal network traffic emulations. Similar

performance degradation was also observed on the NFs not involved in the attack during the rapid reset and slow rate attacks. However, these attacks exploit the inherent limitations in timeout and rate-handling mechanisms within the network, pushing the 5G SBA towards a slow degradation rather than a sudden failure. This subtlety can lead to longer periods of unnoticed impact, allowing significant damage over time.

It is worth noting that the value assigned to the SETTINGS_MAX_CONCURRENT_STREAMS plays an important role in protecting the network against SMA and rapid reset attacks. Here, an intelligent solution for setting the value of this SETTINGS_MAX_CONCURRENT_STREAMS is highly important to achieve the trade-off between network security and network performance. A high value of this setting can maximize the benefits of the stream multiplexing feature in terms of latency, however, it can increase the SMA impact on the network. Thus, an intelligent and adaptive SETTINGS_MAX_CONCURRENT_STREAMS value adjustment solution based on network state can be efficient in protecting the network against SMA attacks. The stealthy nature of slow-rate attacks can make their detection challenging, requiring intelligent anomaly detection solutions. In contrast, they can be prevented by intelligent monitoring solutions and timer values to drop or close malicious connections with long inactivity time at the HTTP/2 client. Although these attacks are exploited in the web, exploiting these attacks in a 5G environment requires attacking 5G specific API, making these attacks more challenging to perform than in the web where APIs are not necessarily used. Furthermore, their impact on 5G network can be more disruptive than in a web environment given the dependencies and interactions existing between 5G NFs.

6.4 Datasets

To generate a 5G dataset that mirrors real 5G network traffic, we emulate normal and attack traffic as noted in Section 6.2 and collect the generated data. Using Wireshark The

Wireshark Team. (2021), a network monitoring tool, we capture network traffic data within our 5G testbed. These captures are raw packets stored as PCAP files The Wireshark Team. (2021), and document raw network interactions between various entities like UE, RAN, and 5G NFs. We collect benign data and execute HTTP/2 attacks at different times. We refine our dataset by processing raw network-layer data with CICFlowMeter Cybersecurity (2020), which generates 84 flow-based features (Appendix A) capable of distinguishing normal and malicious behaviors Hussain et al. (2020); Pourahmadi et al. (2022); Salahuddin et al. (2021). Each row in the resulting CSV file represents a single flow, defined as packets with the same source IP, destination IP, source port, and destination port within a specified time interval. Separate CSV files were created for benign traffic and each emulated HTTP/2 attack.

Following the extracted features, we perform feature normalization and select the most relevant ones. At the feature selection stage, we use the variance threshold scikit learn (2021) function to determine the most relevant variance value of the features. We choose this selection function, as it is well known for its usage in unsupervised models scikit learn (2021). The purpose of its usage is to help in removing features with minimal variations or those deemed as noise. As the model is highly dependent on 5G SBA behavior patterns, the features selected to train the model must be accurately represented (i.e., have high variance) and provided to the anomaly detection module, as a result, we consider 54 features that have high variance, as shown in Table 6.3.

For each emulated scenario, we divide the flow-based dataset (Table 6.4) into two categories: benign and attack, with the total duration of each emulation. Although emulations were planned for two hours, *A.1.1.SMA-Req/Resp* and *A.1.2.SMA-Sub/Not* lasted for 1 hour 40 minutes and 1 hour 50 minutes, respectively, as the network went down due to the attack. It is worth noting that the reported datasets in Table 6.4 are mutually exclusive and do not include any redundant records. For the data to be usable for anomaly detection, we

Selected Flow-based Features
Src Port, Dst Port, Protocol, TotLen Fwd Pkts, TotLen Bwd Pkts, Fwd Pkt Len Max, Fwd Pkt Len Min, Fwd Pkt Len Mean, Fwd Pkt Len Std, Bwd Pkt Len Max, Bwd Pkt Len Min, Bwd Pkt Len Mean, Bwd Pkt Len Std, Flow Byts/s, Bwd IAT Std, Fwd PSH Flags, Bwd PSH Flags, Fwd URG Flags, Bwd URG Flags, Fwd Header Len, Bwd Header Len, Pkt Len Min, Pkt Len Max, Pkt Len Mean, Pkt Len Std, Pkt Len Var, FIN Flag Cnt, SYN Flag Cnt, RST Flag Cnt, PSH Flag Cnt, ACK Flag Cnt, URG Flag Cnt, CWE Flag Count, ECE Flag Cnt, Pkt Size Avg, Fwd Seg Size Avg, Bwd Seg Size Avg, Fwd Byts/b Avg, Fwd Pkts/b Avg, Fwd Blk Rate Avg, Bwd Byts/b Avg, Bwd Pkts/b Avg, Bwd Blk Rate Avg, Subflow Fwd Byts, Subflow Bwd Pkts, Subflow Bwd Byts, Init Fwd Win Byts, Init Bwd Win Byts, Fwd Act Data Pkts, Fwd Seg Size Min, Active Mean, Active Std, Active Max, Active Min

Table 6.3: Selected flow-based features

label our flows as benign (0) and attack (1) based on our knowledge of the compromised UEs used and the time of the attack emulations were launched.

Table 6.4: Flow-based dataset in 5G networks

Emulation Type	Benign Rows	Attack Rows	Duration
Benign	129 367	-	2 hours
A.1.1.SMA-Req/Resp	90 163	55 954	1 hour 40 minutes
A.1.2.SMA-Sub/Not	91 010	54 188	1 hour 50 minutes
A.2.Rapid-Reset	135 120	314 866	3 hours
A.3.1.SR-Setting	74 540	11 696	2 hours
A.3.2.SR-Con-Pref	53 722	10 278	2 hours
A.3.3.SR-Win-Size	68 134	16 738	2 hours

6.5 HTTP/2 Anomaly Detection

In this section, we evaluate the performance of three unsupervised models using flow-based features as an anomaly detection solution in 5G SBA, focusing on their ability to identify HTTP/2-5G-specific attacks.

6.5.1 Anomaly Detection Benchmark Models

To detect HTTP/2 attacks and anomalies in 5G networks, we use unsupervised machine learning models given the difficulties in obtaining labeled data from real network traffic along with the advantage of unsupervised models in detecting anomalies and zero-day attacks Li et al. (2023). Deep Neural Networks (DNNs) are known for their capability to learn the best features that represent the data Hussain et al. (2020). Therefore, we employ DNNs to classify the flow-based feature records as either legitimate or malicious. AE are engineered to compress input data into a lower-dimensional representation and reconstruct it, thereby learning to encapsulate normal data behavior Mirsky et al. (2018). This ability allows AEs to reconstruct normal network traffic and identify deviations as anomalies. Furthermore, AE can be adapted to capture temporal dependencies by integrating recurrent layers, such as LSTM Said Elsayed, Le-Khac, Dev, and Jurcut (2020). As 5G network data will have inherent time-based patterns, choosing a model that can capture these temporal dependencies is crucial. This is where LSTM-AE can be particularly useful, as it also requires relatively lower memory and processing resources compared to more complex architectures like transformers Said Elsayed et al. (2020). Furthermore, Isolation Forest (IF) excels in isolating outliers indicative of malicious behavior, by its ability to partition data points in feature space Laskar et al. (2021). Models must detect subtle anomalies that do not significantly deviate from normal behavior, requiring detailed data representation. Thus, we train and compare the performance of LSTM-AE, AE, and IF in detecting HTTP/2 attacks.

6.5.2 Experimental Results

We evaluate the performance of the three aforementioned models in detecting the six HTTP/2 attacks that we emulated. We use the F1-score, an effective evaluation metric to assess the models' precision and recall capabilities and hence, their detection performance.

We first train and validate multiple architectures for each of the three models, selecting the one with the best detection performance. The final chosen architecture and hyperparameters for each model are reported in Table 6.5. To accomplish this, we allocate 20% of the training dataset of size $100,000$ rows as a validation dataset, and we train the models using the remaining training dataset. To test the three models and to check their performance over different HTTP/2 attacks, we select from each attack file (Table 6.4) $30,000$ benign rows and $10,000$ attack rows.

Table 6.5: Hyperparameters

Hyperparameter	Architecture	Dropout	Batch size	Optimizer	Hidden activation	Estimator
LSTM-Autoencoder	[32,16,16,32]	0.1	16	Adam	ReLU	-
Autoencoder	[42,2,42]	-	32	Adam	ReLU	-
Isolation Forest	-	-	-	-	-	50

Table 6.6: F1-score of LSTM-AE, AE, IF across HTTP/2 attacks

Attack Type	**LSTM-Autoencoder**	**Autoencoder**	**Isolation Forest**
A.1.1.SMA-Req/Resp	93.78%	80.16%	82.63%
A.1.2.SMA-Sub/Not	97.58%	86.08%	88.42%
A.2.Rapid-Reset	96.08%	81.73%	82.32%
A.3.1.SR-Setting	88.21%	84.50%	83.22%
A.3.2.SR-Con-Pref	89.45%	85.09%	85.23%
A.3.3.SR-Win-Size	88.32%	82.11%	84.22%
Average	**92.24%**	**83.28%**	**84.34%**

After training and validating the models, we test each model using the test dataset. As shown in Table 6.6, LSTM-AE outperforms AE and IF across all HTTP/2 attacks. LSTM-AE achieves an average F1-score of 92.24% across HTTP/2 attacks, reflecting its robustness in anomaly detection. IF follows with a lower F1-score. AE shows a comparable average F1-score to IF, but a lower F1-score particularly in more complex attack scenarios (i.e., A.1.1.SMA-Req/Resp, A.3.3.SR-Win-Size), indicating its relative difficulty in capturing all anomalies compared to other models. Figure 6.8 shows a detailed performance of LSTM-AE which consistently achieved the highest F1-score across all HTTP/2 attacks,

highlighting its superior ability to capture temporal dependencies in 5G network data. Notably, it has higher precision for *A.1.1.SMA-Req/Resp* or *A.3.1.SR-Setting* depicting the ability of LSTM-AE to detect them. However, a higher recall is obtained for the remaining attacks, showing the model struggles to correctly detect them. The results suggest that although LSTM-AE is the most robust, further fine-tuning and optimization of all models are necessary to enhance their performance, especially in complex scenarios.

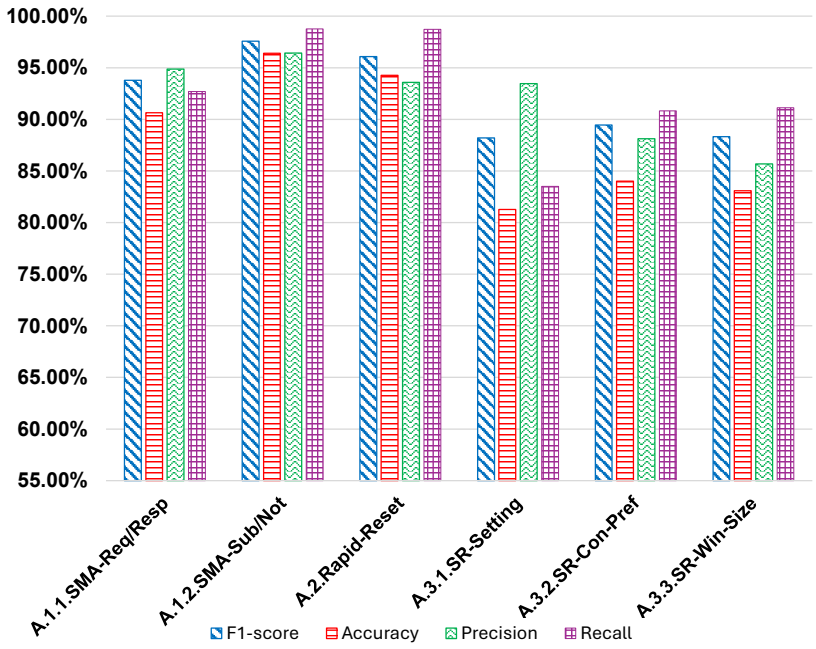


Figure 6.8: LSTM-AE performance across HTTP/2 attacks

To better evaluate the LSTM-AE model across six distinct attacks, we rely on the ROC curves. An ROC curve illustrates the trade-off between the FPR and the TPR across all thresholds Dalianis (2018). The AUC is a commonly used metric in conjunction with the ROC curve, providing an aggregated measure of the model's performance over all thresholds. An $AUC = 1$ indicates a perfect model capable of achieving $TPR = 1$ and $FPR = 0$ with an ideal threshold. Figure 6.9 showcases the performance of the LSTM-AE model tested over six attacks. With AUC values ranging from 0.87 to 0.97 across HTTP/2 attacks, the results highlight the model's ability to detect anomalies effectively. However,

the variation in AUC across attacks emphasizes the impact of attack complexity and feature relevance on detection performance.

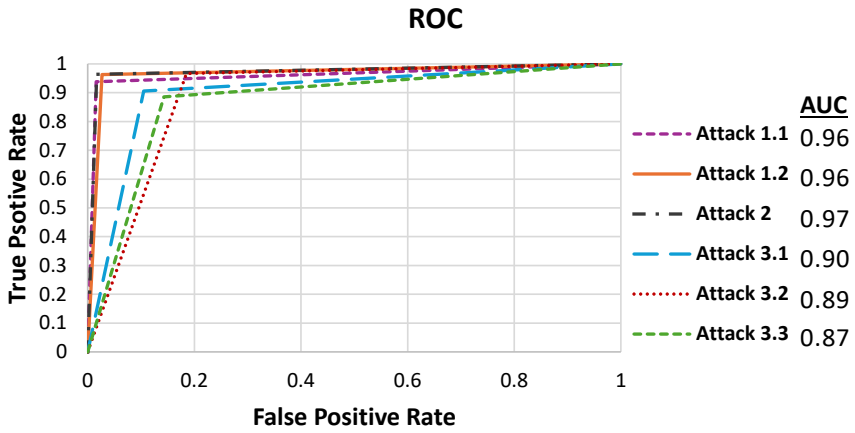


Figure 6.9: AUC-ROC of LSTM-AE across HTTP/2 attacks

6.6 Discussion

While HTTP/2 enhances efficiency in 5G SBA, it introduces security, performance, and operational limitations. The protocol's stream multiplexing feature, while reducing latency, makes 5G networks vulnerable to DoS attacks such as stream multiplexing, rapid reset, and slow-rate attacks, which exploit HTTP/2's request handling mechanisms to overload NFs. These attacks can cause cascading failures, impacting multiple services and leading to network-wide outages. Performance-wise, HTTP/2 increases CPU and memory usage due to its multiplexing, header compression, and flow control mechanisms, which can strain resource-constrained environments. Adaptive rate-limiting and intelligent monitoring are necessary to balance security with network efficiency, but these solutions add computational overhead. Operationally, the lack of standardized HTTP/2-specific security measures for 5G SBA makes detection and mitigation challenging. Traditional security tools are insufficient for 5G-specific signaling interactions, requiring custom anomaly

detection models, which are costly and require frequent updates. Deploying real-time AI-driven anomaly detection such as LSTM-AE models improves attack detection. Additionally, compliance with 5G security regulations further complicates implementation. Effective defense requires a multi-layered security strategy, integrating machine learning-based anomaly detection, dynamic rate control frameworks to mitigate HTTP/2 threats while maintaining network performance. Future research should explore lightweight, adaptive detection mechanisms to enhance security without significantly impacting operational costs.

Chapter 7

Kraken: Multi-Layer Ensemble Learning Detection of HTTP/2 Attacks in 5G and Beyond

In this chapter, we address the limitations of single-feature detection approaches, such as 5GShield and 5GGuardian, which are tailored to detect only HTTP/2 SMA at individual 5G SBA NFs. While effective for SMAs, these solutions do not account for other critical HTTP/2 attacks, such as slow-rate and rapid-reset attacks, leaving significant gaps in 5G security. To overcome these limitations, we propose Kraken, a multi-layer ensemble learning solution designed to enhance anomaly detection in 5G SBA. Kraken leverages three feature sets, namely, flow-based, 5G-stream, and HTTP/2 event-frame to train base models such as an LSTM-Autoencoder, a time-series transformer, and an Autoencoder, respectively, at each 5G SBA NF. The outputs of the base models are aggregated within a meta-model at each NF. Subsequently, feature vectors from the meta-models across all eight NFs are integrated through another meta-model across 5G SBA, to detect sophisticated multi-stage attacks. Kraken achieves an average F1-score of 0.98 across six variations of HTTP/2

attacks, outperforming existing solutions that rely solely on flow-based, stream-based, or HTTP/2 event-frame features.

7.1 Motivation

Based on previous studies in the literature Tripathi (2022); Tripathi and Shaji (2022); Wehbe et al. (2023, 2025), we tested the aforementioned HTTP/2 attacks in a 5G testbed and evaluated the performance of various unsupervised ML models, each trained using 5GC dataset on one of the following feature sets, namely flow-based, 5G-stream, and HTTP/2 event-frame features, in detecting them (i.e., details on the testbed and feature sets are provided in Section 7.3). Figure 7.1 shows that flow-based features used to train an LSTM-AE, are better suited to detect *A.1.2.SMA-Sub/Not* and *A.2.Rapid-Reset*, achieving F1-scores of 0.97 and 0.96 respectively. Flow-based features are widely used for anomaly detection Imperva (2016); Praseed and Thilagam (2020); Wehbe et al. (2023) especially for detecting flooding attacks at the network layer (i.e., TCP syn, UDP flood, etc.) that exhibit anomalies in the flows and packets statistics. In contrast, 5G-stream features capture application layer information that are specialized for HTTP/2 streams. Thus, when used to train a 5GGuardian Wehbe et al. (2025) (5G-stream time-series transformer in Figure 7.1), they exhibit a good detection performance for *A.1.1.SMA-Req/Resp* and *A.1.2.SMA-Sub/Not* with an F1-score of 0.96 for each of them. This is because SMA exploits a high number of streams and related APIs along with the long open HTTP/2 connections that are very well captured through these HTTP/2 and 5G-specific features. Finally, Figure 7.1 shows that HTTP/2 event-frame features used to train an AE, are the best in detecting slow-rate attacks, respectively achieving F1-scores of 0.96, 0.93, and 0.98 for *A.3.1.SR-Setting*, *A.3.2.SR-Con-Pref* and *A.3.3.SR-Win-Size*. These results confirm existing works Tripathi (2022); Tripathi and Shaji (2022) conclusion detailing that HTTP/2 event-frame features are effective in detecting HTTP/2 slow-rate attacks as they capture the intricate

behavior of the HTTP/2 protocol by analyzing its operation at the frame level Tripathi (2022); Tripathi and Shaji (2022). This is valid as event-frame features include frame types such as HEADERS, SETTINGS, and WINDOW_UPDATE that are exploited to perform these slow-rate attacks. Nonetheless, in terms of average performance across the different attacks, we notice that flow-based LSTM-AE outperforms 5G-stream time-series transformer which in turn outperforms the HTTP/2 event-frame AE as they respectively exhibit average F1-scores of 0.92, 0.86 and 0.88.

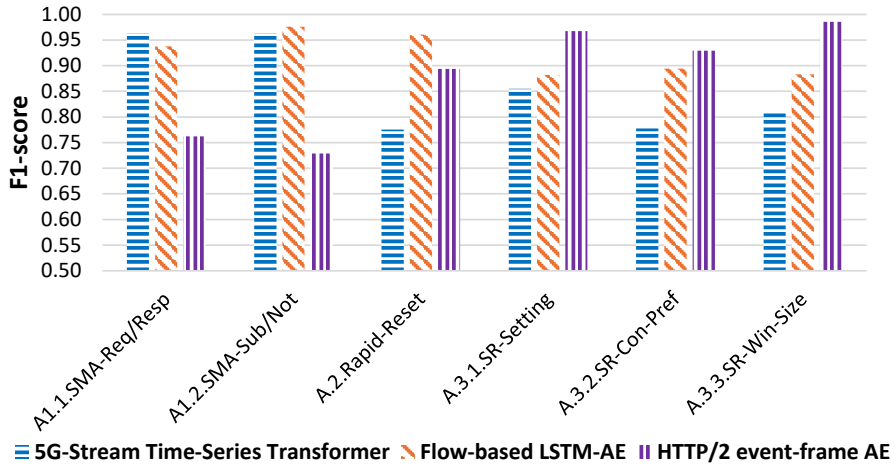


Figure 7.1: F1-scores for HTTP/2 attacks across three feature sets using 5GC dataset.

These results indicate that the accuracy of HTTP/2 attack detection depends significantly on the attack type, the selected feature set, and the choice of ML model. Moreover, analyzing the entire 5G NFs data as one dataset is insufficient to detect HTTP/2 attacks effectively. This is because each NF exhibits distinct behavioral patterns, meaning that treating all NFs together dilutes important attack indicators and increases misclassification rates. Building on these results, we address in this work, the shortcomings of the work in the literature and present Kraken, an ensemble learning based solution, that leverages and combines the strengths of multiple ML models trained on different feature sets in order to provide superior detection performance of HTTP/2 attacks in comparison to single-model anomaly detection solutions leveraging a single feature set.

7.2 Methodology - Kraken: Multi-Layer Ensemble Learning

In this section, we introduce Kraken, our novel and intelligent multi-layer ensemble learning solution designed to detect HTTP/2 attacks via signaling discrepancies between 5G SBA NFs. Kraken maximizes the benefits of ensemble learning within 5G SBA to create a resilient and adaptable detection system designed for multi-dimensional network traffic. Kraken integrates specialized ML models for each feature set, enabling it to precisely, and efficiently detect attacks.

7.2.1 Kraken Architectural Overview

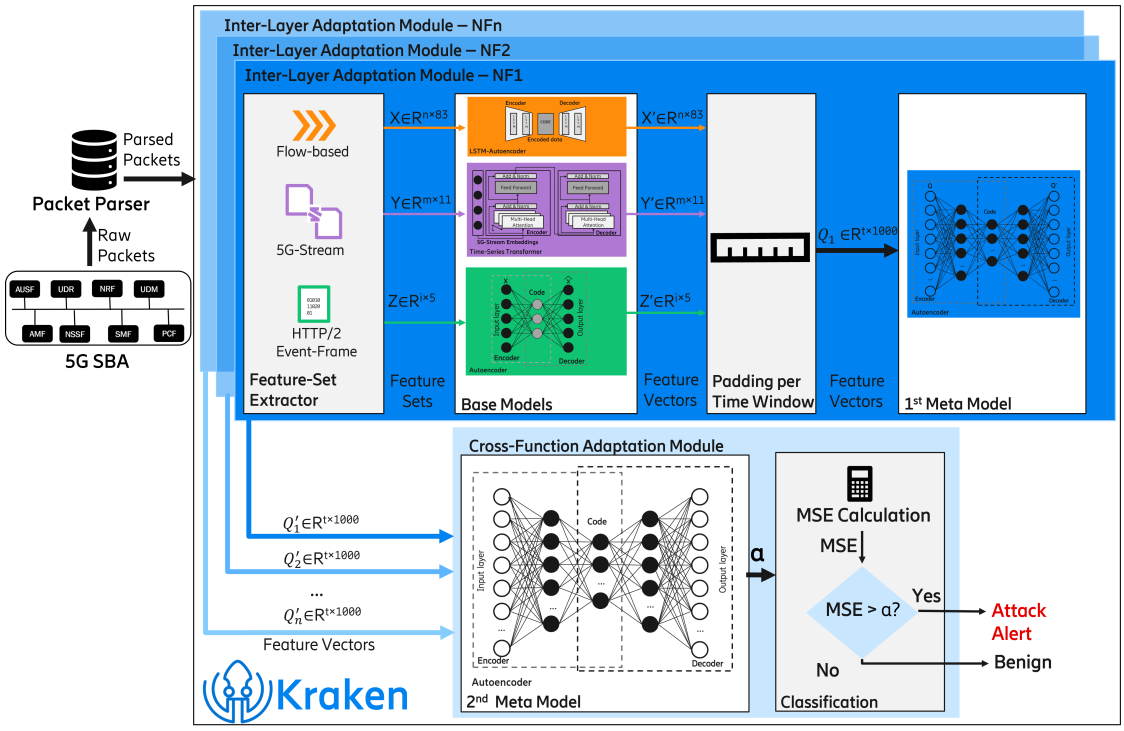


Figure 7.2: Kraken architecture.

Figure 7.2 illustrates the architecture of Kraken that encompasses two modules: 1) The *Inter-Layer Adaptation Module* acts at the NF level. It collects raw packet data from the

network and pre-processes it to extract multiple feature sets that are then used to train three base ML models. The outputs of these base models are then aggregated as input to the 1^{st} meta-model which enables a high level of accuracy in detecting attacks at each NF; 2) The *Cross-Function Adaptation Module*, aggregates feature vectors from the 1^{st} meta-models of the inter-layer adaptation module of different NFs and input them to a 2^{nd} meta-model, allowing Kraken to detect sophisticated attacks exploiting the SBA interconnected nature.

7.2.2 Inter-Layer Adaptation per NF Module

This module operates at the NF level, and accounts for anomaly detection by extracting three distinct feature sets: flow-based, 5G-stream, and HTTP/2 event-frame features. These features are independently processed and used to train specialized unsupervised models, generating feature vectors flattened to one dimension and padded for consistency across time windows. The standardized feature vectors are then used to train a meta-model for each NF, enabling accurate detection by capturing temporal dynamics and NF-specific behavior. The components composing the inter-layer adaptation module are detailed in the following.

A. Feature-Sets Extractor

ML models used in Kraken integrate diverse features derived from network traffic, such as flow-based, 5G-stream, and HTTP/2 event-frame features. We leverage the following feature sets (Table 7.1) strengths to identify specific attack patterns:

Flow-Based Feature Set: Using CICFlowMeter for Cybersecurity (2020), we extract a comprehensive set of 83 flow-based features Cybersecurity (2020) including packet size, flow duration, inter-packet arrival times, byte counts per flow and flow directionality, among others. These flow-based features represent the underlying behaviors of network traffic flows, making them invaluable for anomaly detection. They are particularly critical

for identifying network-layer anomalies, as they provide insights into traffic patterns and deviations from normal behavior Hussain et al. (2020); Pourahmadi et al. (2022); Salahuddin et al. (2021); Wehbe et al. (2023). For example, unusual flow duration spikes or irregular packet arrival times can indicate DoS attacks or congestion caused by malicious activity.

5G-Stream Feature Set: This feature set consists of 11 features derived from NF-to-NF interactions within the 5G SBA Wehbe et al. (2025). These features, such as HTTP/2 stream IDs, header paths, response code, latency, and response indicators, provide a fine-grained representation of NF-to-NF interactions, allowing precise modeling of NFs behaviors and improving anomaly detection performance. For example, abnormal response codes, such as excessive $4xx$ or $5xx$ errors, can signal service interruptions or malicious attempts to overload specific NFs. In addition, latency and response indicators are crucial for detecting anomalies, such as delayed signaling messages or failed communications between NFs. The fine-grained nature of these features makes them particularly effective in identifying deviations specific to 5G SBA operations.

Table 7.1: List of feature sets

Feature Type	**List of Features**
Flow-Based	83 features using CICFlowMeter Cybersecurity (2020) such as Flow Bytes/s, Flow duration, byte counts per flow, etc.
5G-Stream	Latency, http2_protocols, http2_headers_method, http2_headers_path, Header_request_size, Header_response_size, ResponseCode, IMSIfromAPI, Http2_max_concurrent_stream, Stream_time, HasResponse
HTTP/2 Event-Frame	Previous_Event_Type, Previous_Event_Time, Current_Event_Type, Current_Event_Time, Latency

HTTP/2 Event-Frame Feature Set: This feature set delves into the intricate behavior of the HTTP/2 protocol by analyzing its operation at the frame level, making it particularly effective for detecting protocol-specific attacks Tripathi (2022); Tripathi and Shaji (2022).

In HTTP/2, endpoints communicate using frames, which serve as the fundamental units of a stream IETF (2015). Common frame types include *HEADERS*, which initiate streams and convey metadata; *DATA*, which carries the actual content of the stream; and *SETTINGS*, which manage connection parameters. Thus, we extract five key event-frame features, including Previous_Event_Type and Current_Event_Type, which capture specific frame types within the HTTP/2 protocol. These features are extracted across multiple streams at each NF, meaning that Previous_Event_Type does not necessarily refer to the previous frame within the same stream but rather the most recent frame processed by the NF, regardless of its stream. Understanding the previous frame makes it possible to assess the sequence of operations leading up to the current event, which can be critical in identifying anomalies or unusual patterns. At the same time, the Current_Event_Type records the type of frame associated with the current event in the sequence. These features highlight irregularities that are indicative of protocol-specific attacks such as slow-rate attacks, where incomplete frame exchanges deliberately slow server responses to exhaust resources.

B. Base Models And Feature Vectors

For each NF in 5G SBA, we train a dedicated model per feature set to generate anomaly scores, referred to as feature vectors, tailored to the unique features of each set. The selection of models—LSTM Autoencoder, Transformer, and traditional Autoencoder—was driven by the distinct characteristics of each feature set and its role in detecting HTTP/2 attacks across different layers of 5G SBA. 1) LSTM-AE was chosen for flow-based features due to its ability to capture temporal dependencies within network flows. By reconstructing normal traffic patterns, the LSTM-AE model identifies anomalies through deviations in sequential data, making it particularly effective for detecting network-layer threats such as volumetric anomalies and traffic manipulation. 2) The time-series Transformer model was selected for 5G-stream features because it excels at handling long-range dependencies

and complex relationships in NF-to-NF interactions. Unlike traditional sequence models, transformers utilize self-attention mechanisms to weigh important interactions across different time steps, making them well-suited for capturing subtle variations in inter-NF communication patterns. This is crucial for identifying sophisticated anomalies that evolve over time, such as stealthy SMA variations. 3) A standard AE was applied to HTTP/2 event-frame features, as its ability to learn the normal protocol behavior enables it to detect low-level deviations indicative of protocol-specific threats like slow-rate attacks and rapid-reset exploits. The reconstruction loss in the AE serves as an anomaly indicator, ensuring precise detection of application-layer inconsistencies. By leveraging these three distinct models, we ensure a multi-perspective anomaly detection framework, where each model specializes in detecting attacks within its corresponding feature space, thereby providing a comprehensive and layered security mechanism for 5G SBA. In the following, we refer to these models as base models.

The training of these three base models results in three distinct feature vectors: $X' \in \mathbb{R}^{n\times 83}$, $Y' \in \mathbb{R}^{m\times 11}$, and $Z' \in \mathbb{R}^{l\times 5}$, where n, m, and l represent the number of samples for flow-based features, 5G-stream features, and HTTP/2 event-frame features, respectively. Together, these feature vectors serve as input to a 1^{st} meta-model, forming the basis of a comprehensive anomaly detection system that effectively addresses diverse attack vectors across network and application layers.

C. Padding Feature Vectors

The feature vectors generated by the flow-based, 5G-stream, and HTTP/2 event-frame models need to be combined to input the fixed length to the 1^{st} meta-model per NF. However, combining them is not a straightforward mechanism as they have time and logical dependencies. For instance, a flow is composed of multiple streams and a stream encompasses multiple frames. The number of streams within a flow and frames within a stream

may vary across different HTTP/2 connections. Further, as HTTP/2 connections may remain open for a relatively long period of time, waiting for the connection to be closed in order to process the data, generate the feature sets, and input them to Kraken for anomaly detection may not be efficient as it will prevent early attack detection. Thus, to concatenate feature vectors from different models, each with a distinct shape, we resort to using a common time window which can be decided by the network operator. Using a time window seems plausible for 5G anomaly detection as network behavior in systems like the 5G often demonstrates temporal dependencies, where anomalies emerge as deviations over a period rather than isolated instances L.-P. Yuan, Choo, Yu, Khalil, and Zhu (2021). Hence, as an example, in the first time window, we might concatenate the flow-based feature set for flow1 with two feature sets for two of it 5Gstream exchanged within this time window, along with 18 event-frame feature sets corresponding to those streams. However, during the second time window, we might concatenate the flow-based feature set for the same flow1 with another 4 feature sets of 5G-stream that were exchanged within the second time window along with 45 event-frame feature sets corresponding to those 4 streams.

To effectively combine the feature vectors generated by the base models within a time window, we employ a padding technique. Although dimensionality reduction techniques such as Uniform Manifold Approximation and Projection (UMAP) Mittal et al. (2024) could be applied, we opted for padding to preserve the original feature vectors and prevent information loss associated with dimensionality reduction Azab, Khasawneh, Alrabaee, Choo, and Sarsour (2024); Mousa'B, Hasan, Sulaiman, Islam, and Khan (2023). Padding standardizes the feature vectors size across three feature sets, aligning them to a fixed length of $1,000$ dimensions (i.e., we assume this length based on feature vector analysis in our dataset) for each time window. The choice of 1000 dimensions for the feature vectors was determined through an in-depth analysis of the dataset, ensuring a standardized representation across all three feature sets: flow-based records, 5G-stream records, and HTTP/2

event-frame records. To establish a suitable fixed length, we assessed the maximum number of records that could be present per time window for each feature set. For instance, if a typical time window contained 3 flow-based records, 7 5G-stream records, and 38 HTTP/2 event-frame records, the total feature vector size would be calculated as follows: flow-based features: $3 \times 83 = 249$, 5G-stream features: $7 \times 11 = 77$, and HTTP/2 event-frame features: $38 \times 5 = 190$. Summing these values results in 597 feature dimensions for this specific case. Extending this analysis across different time windows, we found that the maximum feature vector size observed was approximately 900 dimensions. To ensure robustness and prevent potential data truncation in cases where the number of records per window slightly exceeds our observations, we incorporated a buffer margin of 100 dimensions, setting the final fixed dimension size to 1000. This padding strategy allows for flexibility in handling edge cases while maintaining computational efficiency in training the detection models. By standardizing the feature vectors size, the 1^{st} meta-model can process inputs efficiently and consistently across feature types, enhancing its ability to detect multi-dimensional threats.

D. 1^{st} Meta-Model

The 1^{st} meta-model, an AE, is trained on these concatenated and padded feature vectors. As an unsupervised learning model, the AE learns to reconstruct normal patterns by encoding the input feature vector into a compressed representation and decoding it back to its original form Mirsky et al. (2018); Wehbe et al. (2023). By leveraging the holistic, padded feature vectors, this 1^{st} meta-model provides a robust mechanism for anomaly detection, addressing diverse attack vectors in 5G SBA. Training the 1^{st} meta-model using these concatenated and padded feature vectors results in new feature vectors, one per NF, denoted as $Q' \in \mathbb{R}^{t \times 1000}$, where t represents the number of time windows. These newly generated feature vectors are the output of 1^{st} meta-model (i.e., reconstructed input) of the different NFs. They encapsulate refined anomaly insights captured through high deviation

of their values in comparison to the model's input feature vector. In fact, during training, the AE adjusts its weights to minimize such reconstruction deviation for typical network behaviors. When exposed to anomalous data, this deviation increases significantly, as the model struggles to replicate patterns it has not encountered before. Thus, high deviation between the model's output and its input allows to effectively identify complex, synchronized threats across all feature sets.

7.2.3 Cross-Function Adaptation Module

The cross-function adaptation module aggregates the feature vectors $Q' \in \mathbb{R}^{t \times 1000}$ from the 1^{st} meta-model of the different NFs (i.e., 8 NFs in our case), thus, correlating anomalies over time windows within the 5G SBA. Through a 2^{nd} meta-model, this module detects system-wide anomalies by capturing cross-NFs interactions that may indicate coordinated attacks impacting the 5G system.

A. 2^{nd} Meta-Model

Feature vectors obtained from the 1^{st} meta-model of each NF are flattened and concatenated into a unified representation to form the input of the 2^{nd} meta-model that yields an AE. The 2^{nd} meta-model is trained to reconstruct normal behavior across the combined feature space from all NFs. During inference, the cross-function adaptation module evaluates the input reconstruction quality performed by the 2^{nd} meta-model in order to depict benign from attack data point. To this end, it computes the MSE and captures the squared differences between the input and its reconstruction (i.e., output of the AE) Mirsky et al. (2018). High MSE values, exceeding a defined threshold α (Section 7.2.2), signal deviations indicative of system-wide anomalies. Unlike other metrics, such as MAE, MSE exhibits high sensitivity to anomalies, thus better highlighting substantial deviations more prominently, hence, enabling better detection.

B. Classification

Given that the 2^{nd} meta-model is trained to learn benign behavior, it is expected to succeed in reconstructing benign data, which will lead to a low reconstruction error (i.e., MSE). In contrast, a high reconstruction error is expected in case of an anomaly. If the $MSE \leq \alpha$, the data is classified as benign; otherwise, it is labeled as malicious. By correlating anomalies across all NFs and leveraging MSE as the evaluation metric, this module enhances the robustness of the detection system, enabling the identification of sophisticated, cross-NF threats that involve attack vectors that exploit vulnerabilities across multiple NFs that cannot be detected at a single NF through the inter-layer adaptation module.

7.3 Environment Setup & Data Preparation

This section outlines the environmental setup used for the training environment for Kraken. Additionally, it discusses the data pre-processing and feature extraction applied to the data collected from the 5G testbed.

7.3.1 Emulation Setup

To train Kraken, we use data collected over 2 hours during normal network operations. We perform our experiments using Python (v3.8) ML libraries such as Tensorflow (v2.12.0) and Transformer (v4.27.4). Kraken training, testing, and experiments were performed on a separate VM equipped with an NVIDIA GPU and 28GB of RAM to ensure computational efficiency.

7.3.2 Network Surge Emulation

In a real 5G network deployment, a sudden surge in traffic can occur when a large number of UEs simultaneously connect and use the network, thus creating a lot of signaling

and consuming most of the network bandwidth Wei, Shi, and Dhelim (2022). This is known as Network Surge (NS) which is commonly seen during events such as sports games or concerts. NS is usually mistaken for a DoS attack. To this end, it is important to test the efficiency of Kraken in differentiating a normal increase of network signaling following an NS from a DoS attack.

Thus, we emulate an NS by accounting for the traffic of 50 legitimate UEs that join the network after an hour of normal network operations during which 100 UEs are served. This increase in UEs is emulated over an hour, resulting in a higher CPU load across the 5G NFs.

7.3.3 Dataset for Kraken

During our benign and attack emulations, we capture the exchanged signaling across 5G SBA using Wireshark The Wireshark Team. (2021) and store the captured traffic in PCAP files. The collected data is used to train and test Kraken after undergoing a pre-processing to extract and normalize the three feature sets, namely flow-based, 5G-stream, and HTTP/2 event-frame features that we discussed in Section 7.2.2 (Table 7.1). The data and hence, the extracted feature sets are classified per NF as shown in Table 7.2. The communication and behavior of each NF in our testbed during benign and attack emulation is represented by three different datasets, namely 5G-stream, flow-based, and HTTP/2 event-frame datasets. Each of these datasets is used to train and validate the base models (Section 7.2.2) of Kraken's inter-layer adaptation module.

Table 7.2 shows the variable number of records across the different datasets per NF. For instance, flow-based records are less than 5G stream dataset records, which in turn are less than those of HTTP/2 event-frame dataset records. This is expected because a flow encompasses multiple streams, and a stream includes multiple frames. In addition, the number of records of the different datasets between the NFs is variable as it is dependent

Table 7.2: Benign and attack datasets per NF across three feature sets

Features	**5G Stream**			**Flow-Based**			**HTTP/2 Event-Frame**		
NF	**Benign**	**Attack**	**NS**	**Benign**	**Attack**	**NS**	**Benign**	**Attack**	**NS**
AMF	93353	342010	64223	31456	168436	16879	472128	10326650	334554
SMF	64563	326926	45092	25938	163898	14208	458484	10513880	392442
NRF	40815	137642	29384	16172	33467	9572	285537	2323040	204770
AUSF	18420	51300	14102	3606	11180	3052	62724	761268	46354
NSSF	9417	25476	5303	1918	4437	1145	33254	336986	17236
PCF	52231	866574	33971	17876	341284	11219	251248	5917160	168686
UDM	64860	145588	45009	15669	29838	9328	258964	2063186	184222
UDR	43561	850730	31236	12672	39211	8174	270260	5790766	178400

on the communication it is involved in during normal network operations and also during attacks. Finally, tailoring different datasets to the unique nature of each feature set for each NF, accurately captures the specific operational patterns of the NF and enhances the accuracy and robustness of anomaly detection across all NFs. It is worth noting that the number of records under the benign columns in Table 7.2 represent the data captured during normal network operations. However, those shown under the attack columns depict the data collected during the different attacks and, hence, contain both benign and attack records. Finally, the NS columns depict the benign data collected during a NS.

7.3.4 Models Training, Validation and Testing

As the different ML models of Kraken are unsupervised models, they are usually trained on data collected during normal network operations. Thus, we allocate 1 hour and 20 minutes of data from the benign records (i.e., benign columns in Table 7.2) across the different datasets for training them and reserve the remaining 40 minutes as a validation dataset. The validation dataset serves two critical roles: validating the performance of the base models and providing the feature vectors resulting from the validation dataset to train both meta-models effectively (Section 7.2). We exclusively use the attack datasets (i.e., attack columns in Table 7.2), containing both benign and attack data, as test datasets to evaluate the performance of Kraken in detecting different attacks and accurately classifying

normal behavior. Finally, the NS datasets are used to evaluate the performance of Kraken in accurately classifying it as benign.

7.4 Experiments and Results

In this section, we evaluate the performance Kraken by leveraging the different training, validation and testing datasets described in Table 7.2.

7.4.1 Kraken Hyperparameter

To ensure the best detection capabilities, we fine-tune each ML model used in Kraken and its hyperparameters and retain those that maximize the F1-score. Thus, we train and validate multiple configurations, selecting the best performing architecture for each model. The final architectures and their corresponding hyperparameters are listed in Table 7.3.

Table 7.3: ML models and hyperparameters

ML Models	**Model Type**	**Architecture**	**Hidden Activation**
Base Models	LSTM-AE	[32,16,16,16,32]	ReLU
	LSTM-AE	[16,8,8,8,16]	ReLU
	AE	[5,5,5,5,5]	ReLU
	Time-Series Transformer	[11,1,1,1,11]	GeLU
1^{st} **Meta-Model**	AE	[1000,10,10,10,1000]	ReLU
2^{nd} **Meta-Model**	AE	[8000,8,8,8,8000]	ReLU

The base models are optimized for their respective feature sets to ensure effective anomaly detection. All models use a batch size of 16, a dropout rate of 0.1, and the Adam optimizer, with architecture differences tailored to each feature set. The LSTM-AE, using flow-based features, employs two distinct architectures with ReLU activation function: a [32, 16, 16, 16, 32] architecture tailored for AMF, SMF, PCF, UDR, and UDM, and a [16, 8, 8, 8, 6] architecture optimized for NSSF, AUSF, and NRF. The AE using HTTP/2 event-frame features leverages a compact [5, 5, 5, 5, 5] architecture with ReLU activation

function; and the time-series transformer for 5G-stream features employs a [11, 1, 1, 1, 11] architecture with 12 attention heads and GeLU activation function.

The 1^{st} meta-model uses aggregated feature vectors from its predecessors, base models, in Kraken. As base models may have feature vectors of different shapes with temporal and logical dependencies (Section 7.2.2), we select a two-seconds time window to determine the feature vectors that need to be aggregated as input to the 1^{st} meta-model as a single data point. The 1^{st} meta-model, an AE with a [1000, 10, 10, 10, 1000] architecture, learns NF-specific patterns, while the 2^{nd} meta-model, an AE with a [8000, 8, 8, 8, 8000] architecture, identifies cross-NF anomalies. Together, they provide a robust solution for detecting diverse threats in the 5G SBA.

7.4.2 Kraken Threshold

To assess the detection performance of Kraken and classify the data into benign and attack, there is a need to select a threshold to compare the MSE calculated based on the input/output of 2^{nd} meta-model against, as explained in Section 7.2.3. Thus, we select a threshold $\alpha = 3.0353$ that resulted in a good detection performance on the validation dataset. Based on the selected threshold $\alpha = 3.0353$, we evaluate in the following Kraken performance using test datasets for each attack scenario (Table 7.2).

7.4.3 Kraken Detection Performance

We assess Kraken's performance in detecting HTTP/2 attacks by: 1) Testing the base models (i.e., time-series transformer, LSTM-AE, and AE) on each NF, however, due to space limitation, we present the results only for the AMF; 2) Evaluating the inter-layer adaptation module per each NF; for this, we choose two attack scenarios, A.1.1.SMA-Req/Resp and A.3.1.SR-Setting, to showcase the efficiency of this module in capturing

temporal anomalies specific to each NF; 3) Evaluating the cross-function adaptation module that integrates insights from all NFs across attacks and NS.

A. Base Models Performance

Figure 7.3 compares the performance of three base models: 5G-stream time-series transformer, flow-based LSTM-AE, and HTTP/2 event-frame AE; trained, validated and tested on data collected at the AMF. Similar to the results in Figure 7.1 (i.e., that accounts for data collected across all the NFs to train, test and validate the base models), Figure 7.3 depicts that both variations of SMA attacks (A.1.1.SMA-Req/Resp and A.1.2.SMA-Sub/Not) are well-detected by 5G-stream time-series transformer and flow-based LSTM-AE with the 5G-stream time-series transformer slightly outperforming the flow-based LSTM-AE. However, by comparing Figure 7.1 and Figure 7.3, we can deduce that HTTP/2 event-frame AE can better detect SMA attacks when trained on data collected at the AMF, that is where the attack is occurring, rather than at data collected across all the NFs of 5G SBA. This clearly shows the value of fine grained and targeted behavioral training (i.e., NF profiling) in better detecting attacks.

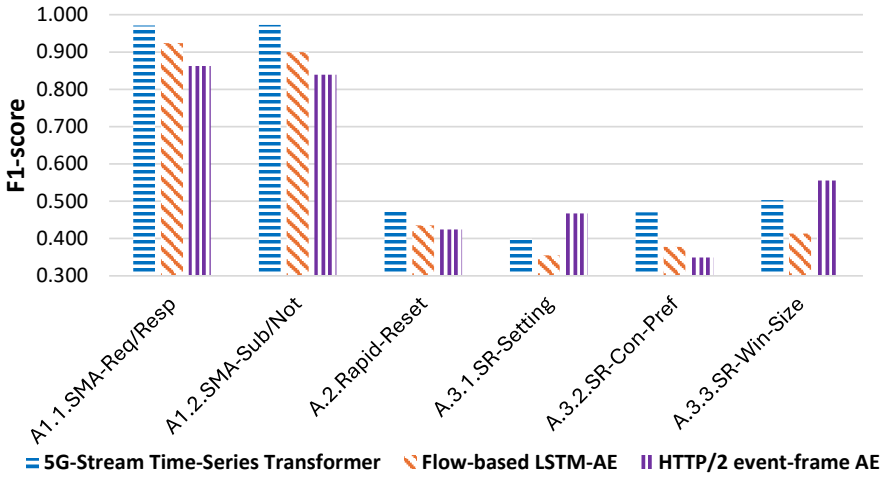


Figure 7.3: F1-score for HTTP/2 attacks detection across three feature sets using AMF dataset.

Further, the detection performance for rapid-reset and slow-rate attacks (A.2.Rapid-Reset, A.3.1.SR-Setting, A.3.2.SR-Con-Pref, and A.3.3.SR-Win-Size) are significantly lower across all feature sets in Figure 7.3 when compared to Figure 7.1. This is expected, as these attacks primarily exploit interactions between the PCF and UDR and do not directly impact the AMF. Consequently, the models trained on AMF-specific data struggle to identify patterns associated with these attacks, reflecting the attack's localized nature and the limited feature correlation at the AMF level.

These results underscore the necessity of leveraging inter-layer adaptation module for better fine grained detection at each NF, and also highlight the value of aggregate insights across the base models and 5G SBA to improve the detection of HTTP/2 attacks across the NFs. This entails that integrating the learning from multiple NFs within an ensemble approach through the second meta-model, could provide enhanced detection of attacks across the 5G SBA.

B. Inter-Layer Adaptation Module Performance

We are only reporting the results on these two attacks because they serve as representative cases for evaluating the detection capabilities of the inter-layer adaptation module across different NFs. *Attack 1 - SMA* variations are emulated between the SMF and AMF, while *Attack 2 - Rapid Reset Attack* and variations of *Attack 3 - HTTP/2 Slow-Rate Attack* were triggered between the PCF and UDR. By selecting A.1.1.SMA-Req/Resp and A.3.1.SR-Setting, we evaluate the detection capabilities of the inter-layer adaptation module across the NFs. We assess the detection performance of the 1^{st} meta-model of each NF in identifying these attacks (Table 7.4).

The evaluation of A.1.1.SMA-Req/Resp, initiated by the SMF against the AMF, highlights the ensemble model's effectiveness in detecting critical vulnerabilities within the 5G SBA, as presented in Table 7.4. As the primary NFs involved in the attack, AMF and SMF

Table 7.4: Performance of each NF in detecting A.1.1.SMA-Req/Resp and A.3.1.SR-Setting using the 1^{st} meta-model

Metrics	**TP**		**FN**		**FP**		**TN**		**Precision**		**Recall**		**F1-score**	
NFs	**A.1.1**	**A.3.1**	**A.1.1**	**A.3.1**	**A.1.1**	**A.3.1**	**A.1.1**	**A.3.1**	**A.1.1**	**A.3.1**	**A.1.1**	**A.3.1**	**A.1.1**	**A.3.1**
AMF	615	232	8	55	29	297	949	1216	0.95	0.43	0.98	0.80	0.97	0.56
SMF	578	297	0	45	14	423	1009	1035	0.97	0.41	1	0.86	0.98	0.55
NRF	7	6	5	3	262	312	1337	1459	0.02	0.01	0.58	0.66	0.049	0.03
NSSF	NA	NA	NA	NA	68	114	1532	1686	NA	NA	NA	NA	NA	NA
AUSF	4	NA	0	NA	158	196	1473	1551	0.02	NA	1	NA	0.04	NA
PCF	44	567	29	61	270	75	1257	1097	0.14	0.88	0.60	0.90	0.22	0.89
UDR	66	893	34	11	382	142	1118	754	0.14	0.86	0.66	0.98	0.24	0.92
UDM	8	49	4	89	162	60	1297	1402	0.04	0.44	0.66	0.35	0.08	0.39

exhibit excellent detection performance of A.1.1.SMA-Req/Resp with high F1-scores of 0.97 and 0.98, respectively. However, other NFs, such as NRF, UDR, PCF, and UDM, experience high False Positives (FP) and False Negatives (FN). The high FP is because these NFs process benign messages that are delayed due to the overwhelmed AMF to handle the attack load. Thus, the 1^{st} meta-model of each of these NFs has mistaken these benign records by being attacks. Meanwhile, FN increase as requests/responses initiated by the compromised SMF should be classified as attacks but are misclassified as benign due to the limitations of the per-NF meta-models. These findings underscore the cascading effects of such attacks and the challenges in ensuring accurate classification across 5G networks.

With respect to the detection of A.3.1.SR-Setting, we notice that the UDR, as the primary NF initiating the attack, achieves a high F1-score of 0.92 due to its direct role in generating malicious traffic (Table 7.4). The PCF, as the targeted NF being attacked, records a slightly lower F1-score of 0.89, mainly due to a low recall caused by the misclassification of attack messages as benign. Additionally, NFs such as NRF, UDM, AMF, and SMF detection are lower due to two reasons: 1) High FP as benign messages being delayed or disrupted by the attack, creating patterns that resemble a malicious behavior; 2) Increase in FN occurs because the attack records, though not part of the targeted attack on the PCF, represent messages forwarded from the compromised UDR and are considered malicious as they originate from a compromised NF.

Finally, NFs like NSSF and AUSF were not involved in A.1.1.SMA-Req/Resp and A.3.1.SR-Setting, and the data does not involve any communication between them, and the compromised (i.e., SMF, UDR) and attacked (i.e., AMF, PCF) NFs. Consequently, these NFs do not exhibit significant anomalies in their detection results.

The results of the inter-layer adaptation module demonstrate the effectiveness of the ensemble approach in detecting critical NF-specific attacks. In fact, for attacks directly involving specific NFs, such as A.1.1.SMA-Req/Resp (SMF and AMF) and A.3.1.SR-Setting (UDR and PCF), the 1^{st} meta-model achieves high F1-score, showcasing its ability to identify malicious behavior accurately in the most impacted NFs. However, the detection results highlight challenges, particularly for indirectly affected NFs, where increased FP arise from benign messages delayed by attack traffic, and high FN is due to the propagation of malicious traffic through compromised NFs, which results in low recall, precision, and F1-score. These findings emphasize the cascading effects of attacks and the necessity of accurate classification across both directly and indirectly impacted NFs.

C. Cross-Function Adaptation Module Performance

This module of Kraken aggregates and refines anomaly detection by combining feature vectors from the 1^{st} meta-model of the different NFs, thus addressing the shortcoming of the inter-layer adaptation module in detecting cross-NFs attacks. This module achieves high precision, recall, and F1-score across HTTP/2 attacks, demonstrating Kraken's robustness in identifying threats with minimal misclassification.

As shown in Table 7.5, A.1.1.SMA-Req/Resp achieves nearly perfect performance with a precision of 0.984, recall of 0.996, and an F1-score of 0.990, reflecting the 2^{nd} meta-model ability to precisely and comprehensively detect instances of this attack. Similarly, A.3.2.SR-Con-Pref shows excellent detection metrics, with an F1-score of 0.985, suggesting that the ensemble model captures intricate patterns from aggregated NF interactions.

Table 7.5: Kraken final detection performance

Attacks	TP	FP	FN	TN	Precision	Recall	F1-score
A.1.1.SMA-Req/Resp	579	9	2	1210	0.984	0.996	0.990
A.1.2.SMA-Sub/Not	783	6	25	1086	0.992	0.969	0.980
A.2.Rapid-Reset	568	1	2	1229	0.998	0.996	0.997
A.3.1.SR-Setting	1064	35	4	797	0.968	0.996	0.982
A.3.2.SR-Con-Pref	608	8	10	1174	0.987	0.983	0.985
A.3.3.SR-Win-Size	787	24	8	1081	0.970	0.989	0.980

Even in more complex attacks, such as A.3.3.SR-Win-Size, where distinguishing between normal and anomalous behavior is challenging due to similar traffic patterns, kraken maintains an average F1-score of 0.98.

D. Kraken Performance with Network Surge

To demonstrate the robustness of Kraken, we test our solution under NS conditions that are usually confused as DoS attacks. (Figure 7.4) shows that NS is accurately detected by the 1^{st} meta-model of the different NFs with F1-score above 0.97 for all them. Similarly, Kraken with its 2^{nd} meta-model shows superior performance in detecting NS with an F1-score of 0.993, indicating that the model can adapt to high-variance scenarios and accurately differentiate between a NS and a DoS attack. Kraken distinguishes NS from DoS attack by leveraging key flow-based such as Inter-packet arrival variance and packet size distribution, and 5G-stream features such as the response time. These features reveal the dynamic nature of NS and help detect its incurred temporary delays, compared to uniform traffic patterns and sustained service degradation during a DoS.

7.4.4 Comparison of Kraken against the State-of-the-Art

Kraken outperforms both base models and 1^{st} meta-model at the NF level by integrating flow-based, 5G-stream, and HTTP/2 event-frame features, addressing the shortcomings of 5GGuardian Wehbe et al. (2025) (5G-stream time-series transformer) and 5GShield

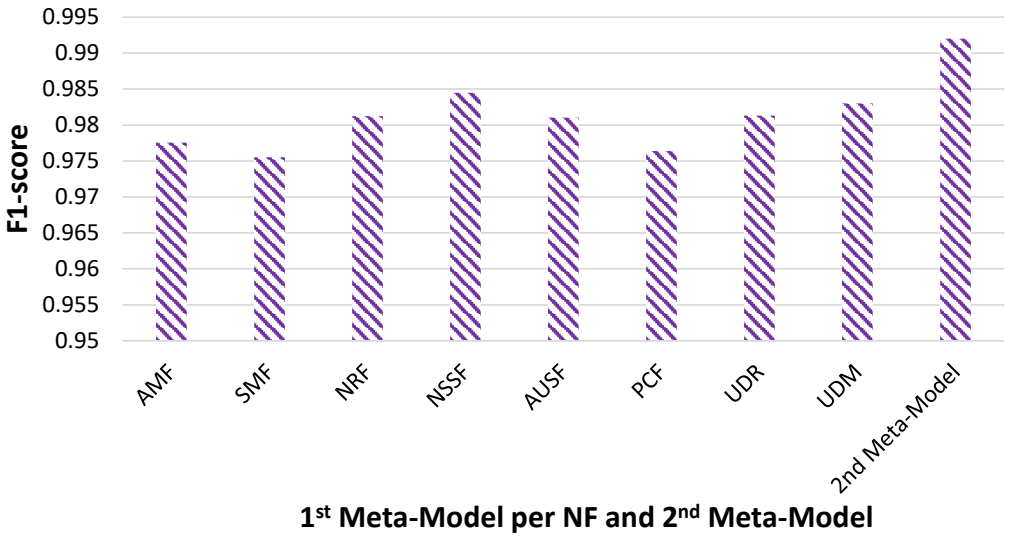


Figure 7.4: Kraken performance in the presence of network surge.

(flow-based LSTM-AE) Wehbe et al. (2023). While 5GGuardian excels in detecting SMA attacks, it lacks robustness against slow-rate and rapid-reset attacks, and 5GShield struggles with application-layer threats Figure 7.3. Kraken overcomes these limitations through multi-feature analysis through ensemble learning, achieving an average F1-score of 0.98, significantly surpassing 5GGuardian (0.86), 5GShield (0.92), and HTTP/2 event-frame AE (0.88) (Figure 7.1). By leveraging multi-layer ensemble learning, Kraken reduces FP, captures cross-NF correlations, and ensures comprehensive detection of localized and cascaded attacks.

7.4.5 Time Complexity

Kraken's design effectively balances time complexity and detection accuracy within its modules. The inter-layer adaptation module, which processes data independently for each of the eight NFs, achieves efficient analysis with an average training time of 4.5 seconds per NF for the 1^{st} meta-model. This module uses base models with practical training times. For instance, time-series transformer (5GGuardian) and AE complete training in 172.82 and

277.99 seconds, respectively, and the LSTM-AE requires 401.75 seconds to capture flow-based temporal dependencies. The cross-function adaptation module enhances anomaly detection by aggregating feature vectors across NFs, optimizing cross-NF correlations with minimal additional training overhead of just 2.71 seconds.

Despite these training costs, Kraken maintains short detection time. The HTTP/2 event-frame model achieves a detection time of 0.08 seconds, flow-based model operates significantly faster at 0.01 seconds, and the 5G-Stream model is the fastest and detects anomalies in 0.003 seconds. The 1^{st} meta-model, which aggregates insights from base models per NF, achieves an average detection time of 0.09 seconds, ensuring real-time performance. Furthermore, the 2^{nd} meta-model, which consolidates outputs from the 1^{st} meta-models, demonstrate is faster and can detect attacks in 0.07 seconds.

Compared to single-feature models, Kraken continues to achieve competitive detection times for real-time deployments as it requires a total of 0.24 seconds for online detection. While single-feature models train faster, they lack the cross-layer adaptability and temporal awareness that Kraken achieves. By combining fast, per-NF and parallel processing of inter-layer adaptation module models with effective aggregation, Kraken ensures high detection performance with minimal computational overhead, making it an efficient and reliable 5G anomaly detection solution.

7.5 Deployemnt of Kraken

Kraken can be integrated as a built-in functionality within NWDAF TS.23.288 (2024), a 5G NF responsible for collecting and analyzing network data to detect anomalies and optimize performance. Kraken can be deployed centrally within a single NWDAF instance, where it processes data from multiple NFs to detect security threats. However, this centralized approach can introduce data collection and processing overhead, making real-time anomaly detection challenging.

To address this, we recommend a hierarchical NWDAF architecture Jeon and Pack (2024), where leaf NWDAFs are co-located with each NF and run Kraken's inter-layer adaptation module locally. This allows each NF to independently detect anomalies based on local traffic patterns before sharing its extracted feature vectors from the 1^{st} meta-model with a centralized NWDAF. This allows parallel anomaly detection at different NFs using an inter-layer adaptation module, thus reducing the Kraken detection time. The central NWDAF then executes Kraken's cross-function adaptation module, which aggregates and correlates data across multiple NFs, enabling more accurate and 5G SBA anomaly detection. This hierarchical and distributed deployment reduces data pre-processing and collection overhead. Moreover, it supports continuous protection against HTTP/2 attacks by facilitating periodic retraining and maintenance, ensuring that Kraken aligns with NWDAF's adaptive learning and update routines for enhanced 5G security.

Chapter 8

Conclusion and Future Directions

The security landscape of 5G networks, particularly within the SBA, demands a rigorous evaluation of vulnerabilities associated with HTTP/2 protocols. This thesis highlights the critical role of HTTP/2 as a potential attack vector, emphasizing the successful exploitation of its vulnerabilities through proof-of-concept demonstrations and the cascading effects of such exploits on interconnected NFs (Chapter 3). To address these challenges, we proposed advanced anomaly detection solutions tailored for HTTP/2 attacks in 5G SBA through this thesis. These include 5GShield (Chapter 4), an application-layer anomaly detection solution leveraging neural networks to identify HTTP/2 SMA using 5G-specific features, and 5GGuardian (Chapter 5), a time-series transformer-based solution designed for robust detection of SMA variations across NFs. Both solutions outperform traditional approaches, offering scalability and resilience even with contaminated training data. Further, recognizing the absence of practical studies and datasets, we contributed with the first 5G-compliant anomaly detection dataset, encompassing diverse HTTP/2 attack scenarios (Chapter 6). This dataset enabled the development of Kraken, a multi-layer ensemble learning solution integrating flow-based, 5G-stream, and HTTP/2 event-frame features across all NFs to detect sophisticated, multi-stage attacks (Chapter 7). By deploying these solutions within NWDAF, we envisioned a unified defense mechanism for securing 5G SBA.

Future research will focus on developing an advanced root cause analysis framework that leverages both network-wide data and historical attack patterns to trace back the origin of an HTTP/2-based attack. Unlike traditional web environments, where attack impact is typically localized, 5G networks present a unique challenge due to the interdependent nature of NFs. An attack on one NF can propagate across the network, disrupting multiple services and degrading overall QoS, making it difficult to pinpoint the origin of an attack. To address this, the framework will integrate causal inference models to analyze dependencies between NFs and understand how an attack propagates through the system. Additionally, we will explore graph-based anomaly detection, where each NF is represented as a node, and their interactions form edges, allowing the system to map and analyze attack propagation in real-time.

Beyond root cause analysis, intelligent mitigation techniques will be developed to dynamically respond to HTTP/2-based threats, such as SMAs, slow-rate attacks, and rapid-reset attacks. Instead of relying on static thresholds or rule-based filtering, we propose an adaptive mitigation system that continuously learns from attack attempts and network conditions. This system will adjust HTTP/2 parameters dynamically, such as HTTP/2 SETTINGS_MAX_CONCURRENT_STREAMS based on anomaly scores to limit attack impact while ensuring legitimate requests are not disrupted. It will also modify HTTP/2 flow control settings to prevent slow-rate DoS attacks without affecting normal traffic patterns. Another approach will be to assess the HTTP/2 custom headers to tag and verify the integrity of requests, allowing anomaly detection models to differentiate between benign and malicious traffic more effectively. By addressing these future research directions, we aim to transition from reactive detection to proactive and autonomous HTTP/2 attack mitigation, ensuring 5G networks remain resilient against ever-evolving security threats.

The bulk of the dissertation focuses on the work that has been performed by the student

as part of the Ph.D. program. The different contributions that have already been published/submitted for publication in top venues are summarized in Table 8.1.

Table 8.1: Contributions during the Ph.D. program

Title	Chapter	Citation
A Security Assessment of HTTP/2 Usage in 5G Service Based Architecture	Chapter 3	Wehbe, N., Alameddine, H. A., Pourzandi, M., Bou-Harb, E., & Assi, C. (2022). A security assessment of HTTP/2 usage in 5G service-based architecture. IEEE Communications Magazine, 61(1), 48-54.
5GShield: HTTP/2 Anomaly Detection in 5G Service-Based Architecture	Chapter 4	Wehbe, N., Alameddine, H. A., Pourzandi, M., & Assi, C. (2023, June). 5GShield: HTTP/2 Anomaly Detection in 5G Service-Based Architecture. In 2023 IFIP Networking Conference (IFIP Networking) (pp. 1-9). IEEE.
Empowering 5G SBA Security: Time Series Transformer for HTTP/2 Anomaly Detection	Chapter 5	Wehbe, N., Alameddine, H. A., Pourzandi, M.,& Assi, C. (2025). Empowering 5G SBA security: Time series transformer for HTTP/2 anomaly detection. Computers & Security, 148, 104114.
HTTP/2 DoS Attacks in 5G Networks: Impact Analysis and Anomaly Detection	Chapter 6	Wehbe, N., Alameddine, H. A.,& Assi, C. (2025). HTTP/2 DoS Attacks in 5G Networks: Impact Analysis and Anomaly Detection. Submitted to Transaction on Mobile Computing. Under Review.
Kraken: Multi-Layer Ensemble Learning Detection of HTTP/2 Attacks in 5G and Beyond	Chapter 7	Wehbe, N., Alameddine, H. A.,& Assi, C. (2025). Kraken: Multi-Layer Ensemble Learning Detection of HTTP/2 Attacks in 5G and Beyond. In 2025 IEEE/IFIP International Conference on Dependable Systems and Networks.

Other collaborations with different colleagues throughout my Ph.D. are summarized in Table 8.2.

Table 8.2: Other co-authorships during the Ph.D. program

Title	Citation
Inter-Slice Defender: An Anomaly Detection Solution for Distributed Slice Mobility Attacks	Molina, R. M. A., Wehbe, N., Alameddine, H. A., Pourzandi, M., & Assi, C. (2024, June). Inter-Slice Defender: An Anomaly Detection Solution for Distributed Slice Mobility Attacks. In 2024 IFIP Networking Conference (IFIP Networking) (pp. 432-440).
PUL-Inter-slice Defender: An Anomaly Detection Solution for Distributed Slice Mobility Attacks	Molina, R. M. A., Wehbe, N., Alameddine, H. A., Pourzandi, M., & Assi, C. (2025). PUL-Inter-slice Defender: An Anomaly Detection Solution for Distributed Slice Mobility Attacks. In 2025 Transactions on Information Forensics & Security.
A Reinforcement Learning-based Approach for Scaling the User Plane in 5G and Beyond Networks	Hurtado, J., Caicedo, O. M., Assi, C., Wehbe, N. & Suarez, L., (2025). A Reinforcement Learning-based Approach for Scaling the User Plane in 5G and Beyond Networks. In 2025 IEEE Open Journal of the Communications Society.
PEACE: Physics-Enabled Autoencoder Detection of Unknown Load-Altering Attacks in Smart Grids	M. A. Sayed, Nathalie Wehbe, K. Sarieddine, R. Atallah, C. Assi, & M. Debbabi. PEACE: Physics-Enabled Autoencoder Detection of Unknown Load-Altering Attacks in Smart Grids. IEEE Power & Energy Society General Meeting (PESGM).
GridWatch: Load-Altering Attack Detection and Localization Mechanism Powered by a Physics-Assisted Feature Fusion Hybrid Neural Network	M. A. Sayed, K. Sarieddine, Nathalie Wehbe, M. Arfaoui, R. Atallah, M. Debbabi, & C. Assi. GridWatch: Load-Altering Attack Detection and Localization Mechanism Powered by a Physics-Assisted Feature Fusion Hybrid Neural Network. Submitted IEEE transactions on Smart Grid.
EV-Shield: A Real-time Monitoring Tool to Detect and Mitigate Cyber-attacks in the EV Ecosystem	R. Reghunath, M. A. Sayed, N. Wehbe, R. Atallah, D. Jafarigiv, M. Kassouf, and C. Assi. EV-Shield: A Real-time Monitoring Tool to Detect and Mitigate Cyber-attacks in the EV Ecosystem. Submitted IEEE Transactions on Network and Service Management.

List of Abbreviations

3GPP	3^{rd} Generation Partnership Project
5G	Fifth Generation
5GC	5G Core
AE	Autoencoder
AL-DDoS	Application Layer DDoS
AF	Application Function
AI	Artificial Intelligence
AMF	Access and Mobility Management Function
API	Application Programming Interface
AUC	Area Under the Receiver Operating Characteristic Curve
AUSF	Authentication Server Function
CP	Control Plane
CPI	Central Processing Unit
DN	Data Network
DNN	Deep Neural Network
DoS	Denial of Service
DDoS	Distributed Denial of Service
eMBB	Enhanced Mobile Broadband
ETSI	European Telecommunications Standards Institute
FN	False Negative
FP	False Positive
FPR	False Positive Rate
FQDN	Fully Qualified Domain Name
GeLU	Gaussian error Linear Unit
HTTP/2	Hypertext Transfer Protocol version 2

IEs	Information Elements
IETF	Internet Engineering Task Force
IF	Isolation Forest
IMSI	International Mobile Subscriber Identity
IoT	Internet of Things
JSON	JavaScript Object Notation
KPIs	Key Performance Indicators
LCI	Load Control Information
LSTM-AE	Long Short Term Memory Autoencoder
MAE	Mean-Absolute Error
MitM	Man-in-the-Middle
ML	Machine Learning
MSE	Mean Squared Error
NAS	Non Access Stratum
NEF	Network Exposure Function
NF	Network Function
NFc	NF Service Consumer
NFp	NF Service Producer
NFV	Network Function Virtualization
NRF	Network Repository Function
NS	Network Surge
NSSF	Network Slice Selection Function
NWDAF	Network Data Analytics Function
OAM	Operations Administration and Maintenance
OCI	Overload Control Information
PCF	Policy Control Function

PCT	Procedure Completion Time
PDU	Packet Data Unit
PFCP	Packet Forwarding Control Protocol
PKI	Public-Key Infrastructure
PLMN	Public Land Mobile Network
PM	Performance Measurements
QoS	Quality of Service
RAN	Radio Access Network
RESTful	REpresentational State Transfer
ROC	Receiver Operating Characteristic
SBA	Service Based Architecture
SBIs	Service Based Interfaces
SCP	Service Communication Proxy
SDN	Software Defined Network
SEPP	Security Edge Protection Proxy
SMA	Stream Multiplexing Attack
SMF	Session Management Function
SMP	SBI Message Priority
SSL	Secure Sockets Layer
SUPI	Subscription Permanent Identifie
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TPR	True Positive Rate
UE	User Equipment
UMAP	Uniform Manifold Approximation and Projection
UP	Unified Data Management

UP	User Plane
UPF	User Plane Function
URI	Uniform Resource Identifier
VM	Virtual Machine

Appendix A

Flow-based Features

We utilized CICFlowMeter to extract flow-based features from our dataset. CICFlowMeter, an open-source tool, generates bidirectional flows (Biflows) from PCAP files and extracts relevant features from these flows. This network traffic flow generator creates bidirectional flows by determining the forward (source to destination) and backward (destination to source) directions based on the first packet observed. The extracted features are described in Table A.1.

Table A.1: List of flow-based features and their descriptions.

Feature Name	**Description**
Flow ID	Identifier of the flow
SrcIP	Source ip of the flow
Src Port	Source port of the flow
DstIP	Destination ip of the flow
Dst Port	Destination port of the flow

Protocol	Protocol of the flow
Timestamp	Timestamp of the flow
Flow duration	Duration of the flow in Microsecond
total Fwd Packet	Total packets in the forward direction
total Bwd packets	Total packets in the backward direction
total Length of Fwd Packet	Total size of packet in forward direction
total Length of Bwd Packet	Total size of packet in backward direction
Fwd Packet Length Min	Minimum size of packet in forward direction
Fwd Packet Length Max	Maximum size of packet in forward direction
Fwd Packet Length Mean	Mean size of packet in forward direction
Fwd Packet Length Std	Standard deviation size of packet in forward direction
Bwd Packet Length Min	Minimum size of packet in backward direction
Bwd Packet Length Max	Maximum size of packet in backward direction
Bwd Packet Length Mean	Mean size of packet in backward direction
Bwd Packet Length Std	Standard deviation size of packet in backward direction
Flow Bytes/s	Number of flow bytes per second
Flow Packets/s	Number of flow packets per second
Flow IAT Mean	Mean time between two packets sent in the flow
Flow IAT Std	Standard deviation time between two packets sent in the flow
Flow IAT Max	Maximum time between two packets sent in the flow
Flow IAT Min	Minimum time between two packets sent in the flow

Fwd IAT Min	Minimum time between two packets sent in the forward direction
Fwd IAT Max	Maximum time between two packets sent in the forward direction
Fwd IAT Mean	Mean time between two packets sent in the forward direction
Fwd IAT Std	Standard deviation time between two packets sent in the forward direction
Fwd IAT Total	Total time between two packets sent in the forward direction
Bwd IAT Min	Minimum time between two packets sent in the backward direction
Bwd IAT Max	Maximum time between two packets sent in the backward direction
Bwd IAT Mean	Mean time between two packets sent in the backward direction
Bwd IAT Std	Standard deviation time between two packets sent in the backward direction
Bwd IAT Total	Total time between two packets sent in the backward direction
Fwd PSH flags	Number of times the PSH flag was set in packets travelling in the forward direction (0 for UDP)
Bwd PSH Flags	Number of times the PSH flag was set in packets travelling in the backward direction (0 for UDP)
Fwd URG Flags	Number of times the URG flag was set in packets travelling in the forward direction (0 for UDP)

Bwd URG Flags	Number of times the URG flag was set in packets travelling in the backward direction (0 for UDP)
Fwd Header Length	Total bytes used for headers in the forward direction
Bwd Header Length	Total bytes used for headers in the backward direction
FWD Packets/s	Number of forward packets per second
Bwd Packets/s	Number of backward packets per second
Packet Length Min	Minimum length of a packet
Packet Length Max	Maximum length of a packet
Packet Length Mean	Mean length of a packet
Packet Length Std	Standard deviation length of a packet
Packet Length Variance	Variance length of a packet
FIN Flag Count	Number of packets with FIN
SYN Flag Count	Number of packets with SYN
RST Flag Count	Number of packets with RST
PSH Flag Count	Number of packets with PUSH
ACK Flag Count	Number of packets with ACK
URG Flag Count	Number of packets with URG
CWR Flag Count	Number of packets with CWR
ECE Flag Count	Number of packets with ECE
down/Up Ratio	Download and upload ratio
Average Packet Size	Average size of packet
Fwd Segment Size Avg	Average size observed in the forward direction

Bwd Segment Size Avg	Average size observed in the backward direction
Fwd Bytes/Bulk Avg	Average number of bytes bulk rate in the forward direction
Fwd Packet/Bulk Avg	Average number of packets bulk rate in the forward direction
Fwd Bulk Rate Avg	Average number of bulk rate in the forward direction
Bwd Bytes/Bulk Avg	Average number of bytes bulk rate in the backward direction
Bwd Packet/Bulk Avg	Average number of packets bulk rate in the backward direction
Bwd Bulk Rate Avg	Average number of bulk rate in the backward direction
Subflow Fwd Packets	The average number of packets in a sub flow in the forward direction
Subflow Fwd Bytes	The average number of bytes in a sub flow in the forward direction
Subflow Bwd Packets	The average number of packets in a sub flow in the backward direction
Subflow Bwd Bytes	The average number of bytes in a sub flow in the backward direction
Fwd Init Win bytes	The total number of bytes sent in initial window in the forward direction
Bwd Init Win bytes	The total number of bytes sent in initial window in the backward direction
Fwd Act Data Pkts	Count of packets with at least 1 byte of TCP data payload in the forward direction
Fwd Seg Size Min	Minimum segment size observed in the forward direction
Active Min	Minimum time a flow was active before becoming idle

Active Mean	Mean time a flow was active before becoming idle
Active Max	Maximum time a flow was active before becoming idle
Active Std	Standard deviation time a flow was active before becoming idle
Idle Min	Minimum time a flow was idle before becoming active
Idle Mean	Mean time a flow was idle before becoming active
Idle Max	Maximum time a flow was idle before becoming active
Idle Std	Standard deviation time a flow was idle before becoming active

References

3GPP 5G Standard. (2025). *The 5g standard.* https://www.3gpp.org/. ([Online; accessed January-2025])

3GPP TS.29.500. (2024). 5G; 5G System; Technical Realization of Service Based Architecture; Stage 3: TS 29.500 v.18.7.0.

AdaptiveMobile. (2021). *A Slice in Time: Slicing Security in 5G Core Networks.* AdaptiveMobile Security. Retrieved from https://info.adaptivemobile.com/network-slicing-security?hsLang=en#download

Ahmad, I., Shahabuddin, S., Kumar, T., Okwuibe, J., Gurtov, A., & Ylianttila, M. (2019). Security for 5g and beyond. *IEEE Communications Surveys & Tutorials*, *21*(4), 3682–3722.

Alamr, A., & Artoli, A. (2023). Unsupervised transformer-based anomaly detection in ecg signals. *Algorithms*, *16*(3), 152.

aligungr. (2021). UERANSIM. Retrieved from https://github.com/aligungr/UERANSIM

Aljebreen, M., Mengash, H. A., Arasi, M. A., Aljameel, S. S., Salama, A. S., & Hamza, M. A. (2023). Enhancing ddos attack detection using snake optimizer with ensemble learning on internet of things environment. *IEEE Access*.

Alliance, N. (2018). Service-based architecture in 5G. *Final deliverable (approved-P Public)*.

Amponis, G., Radoglou-Grammatikis, P., Lagkas, T., Ouzounidis, S., Zevgara, M.,

Moscholios, I., ... Sarigiannidis, P. (2023). Generating full-stack 5g security datasets: Ip-layer and core network persistent pdu session attacks. *AEU-International Journal of Electronics and Communications*, *171*, 154913.

Anderson, M. W., Coldwell, C. W., Sgambati, M. R., Jacobson, B. G., Petersen, B. J., Spencer, D. R., ... Goodell, E. (2023). *Machine learning 5g attack detection in programmable logic* (Tech. Rep.). Idaho National Lab.(INL), Idaho Falls, ID (United States).

Azab, A., Khasawneh, M., Alrabaee, S., Choo, K.-K. R., & Sarsour, M. (2024). Network traffic classification: Techniques, datasets, and challenges. *Digital Communications and Networks*, *10*(3), 676–692.

Caccavale, F. G., Nguyen, H.-N., Cavalli, A., Montes De Oca, E., & Mallouli, W. (2023). Http/2 attacks generation using 5greplay. In *Proceedings of the 18th international conference on availability, reliability and security* (pp. 1–7).

Canadian Radio-television and Telecommunications Commission. (2024). *General information - service outages: 8000-c12-201909780.* Retrieved from https://crtc.gc.ca/otf/eng/2019/8000/c12-201909780.htm

Chatzoglou, E., Kouliaridis, V., Kambourakis, G., Karopoulos, G., & Gritzalis, S. (2023). A hands-on gaze on http/3 security through the lens of http/2 and a public dataset. In (Vol. 125, p. 103051). Elsevier.

Christine Jost, B. (2020). *Security for 5G Service-Based Architecture: What you need to know*. Ericsson. Retrieved from https://www.ericsson.com/en/blog/2020/8/security-for-5g-service-based-architecture ([Accessed 18-March-2022])

Cloudflare. (2023). *Http/2 rapid reset: deconstructing the record-breaking attack.* Retrieved from https://blog.cloudflare.com/technical-breakdown-http2-rapid-reset-ddos-attack/

CNN. (2024). *At&t says it has resolved software issue that caused an outage for some wireless customers.* Retrieved from https://www.cnn.com/2024/08/27/business/att-outage-software-issue-tuesday/index.html

Cybersecurity, C. I. (2020). Cicflowmeter. Retrieved from https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter/blob/master/ReadMe.txt

Dalianis, H. (2018). Evaluation metrics and evaluation. In *Clinical text mining* (pp. 45–53). Springer.

Dutta, A., & Hammad, E. (2020). 5g security challenges and opportunities: a system approach. In *2020 ieee 3rd 5g world forum (5gwf)* (pp. 109–114).

ENISA. (2021). Security In 5G Specifications Controls in 3GPP Security Specifications (5G SA). *ENISA*.

ETSI. (2020). Network Functions Virtualisation (NFV) Release 4; Security; Secure End-to-End VNF and NS management specification. Retrieved from https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=59208

for Cybersecurity, C. I. (2020). CICFlowMeter. *Canadian Institute for Cybersecurity*. Retrieved from https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter

Free5GC. (2021a). Free5GC. Retrieved from https://www.free5gc.org/

Free5GC. (2021b). Free5GC-compose. Retrieved from https://github.com/free5gc/free5gc-compose/tree/v3.0.5

Goshi, E., Jarschel, M., Pries, R., He, M., & Kellerer, W. (2021). Investigating inter-nf dependencies in cloud-native 5g core networks. In *2021 17th international conference on network and service management (cnsm)* (pp. 370–374).

GSMA. (2021). 5G Interconnect Security Version 2.0.

Haider, U., Waqas, M., Hanif, M., Alasmary, H., & Qaisar, S. M. (2023). Network load prediction and anomaly detection using ensemble learning in 5g cellular networks. *Computer Communications*, *197*, 141–150.

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hodson, T. O. (2022). Root-mean-square error (rmse) or mean absolute error (mae): When to use them or not. *Geoscientific Model Development*, *15*(14), 5481–5487.

Hu, X., Liu, C., Liu, S., You, W., & Zhao, Y. (2018). Signalling security analysis: Is http/2 secure in 5g core network? *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, 1–6.

Hussain, B., Du, Q., Sun, B., & Han, Z. (2020). Deep learning-based ddos-attack detection for cyber-physical system over 5g network. *IEEE Transactions on Industrial Informatics*, *17*(2), 860–870.

IETF. (2015). Hypertext Transfer Protocol Version 2 (HTTP/2) - RFC 7540.

IETF. (2022). The OAuth 2.0 Authorization Framework RFC 6749.

Imperva. (2016). HTTP/2: In-depth analysis of the top four flaws of the next generation web protocol. *Imperva*.

Jeon, Y., & Pack, S. (2024). Hierarchical network data analytics framework for 6g network automation: Design and implementation. *IEEE Internet Computing*.

Karim, I., Mubasshir, K. S., Rahman, M. M., & Bertino, E. (2023). Spec5g: A dataset for 5g cellular network protocol analysis. *arXiv preprint arXiv:2301.09201*.

Lam, J., & Abbas, R. (2020). Machine learning based anomaly detection for 5g networks. *arXiv preprint arXiv:2003.03474*.

Laskar, M. T. R., Huang, J. X., Smetana, V., Stewart, C., Pouw, K., An, A., … Liu, L. (2021). Extending isolation forest for anomaly detection in big data via k-means. *ACM Transactions on Cyber-Physical Systems (TCPS)*, *5*(4), 1–26.

Li, Y., Peng, X., Zhang, J., Li, Z., & Wen, M. (2021). Dct-gan: Dilated convolutional transformer-based gan for time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*.

Li, Y., Sun, F., Hu, J., Liu, C., Wu, F., Li, K., ... others (2023). Self-supervised mafenn for classifying low-labeled distorted images over mobile fading channels. *IEEE Transactions on Mobile Computing*.

Liao, J., Teo, S. G., Kundu, P. P., & Truong-Huu, T. (2021). Enad: An ensemble framework for unsupervised network anomaly detection. , 81–88.

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*.

Madi, T., Alameddine, H. A., Pourzandi, M., & Boukhtouta, A. (2021). Nfv security survey in 5g networks: A three-dimensional threat taxonomy. *Computer Networks*, *197*, 108288.

Mathian, E., Liu, H., Fernandez-Cuesta, L., Samaras, D., Foll, M., & Chen, L. (2022). Haloae: An halonet based local transformer auto-encoder for anomaly detection and localization. *arXiv preprint arXiv:2208.03486*.

Mehmeti, F., & La Porta, T. F. (2022). Modeling and analysis of mmtc traffic in 5g base stations. In *2022 ieee 19th annual consumer communications & networking conference (ccnc)* (pp. 652–660).

Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*.

Mittal, M., Gujjar, P., Prasad, G., Devadas, R. M., Ambreen, L., & Kumar, V. (2024). Dimensionality reduction using umap and tsne technique. , *1*, 1–5.

Mousa'B, M. S., Hasan, M. K., Sulaiman, R., Islam, S., & Khan, A. U. R. (2023). An explainable ensemble deep learning approach for intrusion detection in industrial internet of things. *IEEE Access*, *11*, 115047–115061.

National Vulnerability Database. (2023). *Cve-2023-20864*. National Vulnerability

Database (NVD). Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2023-20864

National Vulnerability Database (NVD). (2023). *CVE-2023-44487*. National Vulnerability Database (NVD). Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2023-44487

Navarro-Ortiz, J., Romero-Diaz, P., Sendra, S., Ameigeiras, P., Ramos-Munoz, J. J., & Lopez-Soler, J. M. (2020). A survey on 5g usage scenarios and traffic models. *IEEE Communications Surveys & Tutorials*, *22*(2), 905–929.

(NVD), N. V. D. (2016). *Cve-2016-5736*. Author. Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2019-5736

(NVD), N. V. D. (2019). *Cve-2016-5195*. Author. Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2016-5195

(NVD), N. V. D. (2023). *Cve-2023-39325*. Author. Retrieved from https://nvd.nist.gov/vuln/detail/CVE-2023-39325

OpenStack. (2021). Build the Future of Open Infrastructure. *The Wireshark Team.*. Retrieved from https://www.openstack.org/

Pell, R., Moschoyiannis, S., Panaousis, E., & Heartfield, R. (2021). Towards dynamic threat modelling in 5g core networks based on mitre att&ck. *arXiv preprint arXiv:2108.11206*.

Pourahmadi, V., Alameddine, H. A., Salahuddin, M. A., & Boutaba, R. (2022). Spotting anomalies at the edge: Outlier exposure-based cross-silo federated learning for ddos detection. *IEEE Transactions on Dependable and Secure Computing*, 1-14. doi: 10.1109/TDSC.2022.3224896

Praseed, A., & Thilagam, P. S. (2018). Ddos attacks at the application layer: Challenges and research perspectives for safeguarding web applications. *IEEE Communications Surveys & Tutorials*, *21*(1), 661–685.

Praseed, A., & Thilagam, P. S. (2019). Multiplexed asymmetric attacks: Next-generation ddos on http/2 servers. *IEEE Transactions on Information Forensics and Security*, *15*, 1790–1800.

Praseed, A., & Thilagam, P. S. (2020). Modelling behavioural dynamics for asymmetric application layer ddos detection. *IEEE Transactions on Information Forensics and Security*, *16*, 617–626.

Praseed, A., & Thilagam, P. S. (2021). Fuzzy request set modelling for detecting multiplexed asymmetric ddos attacks on http/2 servers. *Expert Systems with Applications*, *186*, 115697.

Raaijmakers, Y., Mandelli, S., & Doll, M. (2021). Reinforcement learning for admission control in 5g wireless networks. In *2021 ieee global communications conference (globecom)* (pp. 1–6).

Report, E. M. (2022). *5G SA deployment: Moving beyond eMBB*. Ericsson. Retrieved from https://www.ericsson.com/en/reports-and-papers/mobility-report/dataforecasts/5g-standalone-deployment ([Accessed June-2022])

Rincy, T. N., & Gupta, R. (2020). Ensemble learning techniques and its efficiency in machine learning: A survey. *2nd international conference on data, engineering and applications (IDEA)*, 1–6.

Sagi, O., & Rokach, L. (2018). Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, *8*(4), e1249.

Saha, S., Priyoti, A. T., Sharma, A., & Haque, A. (2022). Towards an optimized ensemble feature selection for ddos detection using both supervised and unsupervised method. *Sensors*, *22*(23), 9144.

Said Elsayed, M., Le-Khac, N.-A., Dev, S., & Jurcut, A. D. (2020). Network anomaly detection using lstm based autoencoder. , 37–45.

Salahuddin, M. A., Pourahmadi, V., Alameddine, H. A., Bari, M. F., & Boutaba, R. (2021). Chronos: Ddos attack detection using time-based autoencoder. *IEEE Transactions on Network and Service Management*, *19*(1), 627–641.

Samarakoon, S., Siriwardhana, Y., Porambage, P., Liyanage, M., Chang, S.-Y., Kim, J., ... Ylianttila, M. (2022). 5g-nidd: A comprehensive network intrusion detection dataset generated over 5g wireless network. *arXiv preprint arXiv:2212.01298*.

Sattar, D., Vasoukolaei, A. H., Crysdale, P., & Matrawy, A. (2021). A stride threat model for 5g core slicing. In *2021 ieee 4th 5g world forum (5gwf)* (pp. 247–252).

scikit learn. (2021). scikit-learn. *scikit-learn Team.*. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html

Security considerations for the 5g era. (2020). 5G America. Retrieved from https://www.5gamericas.org/security-considerations-for-the-5g-era/ ([Accessed July-2020])

Shetty, R., Jangam, A., & Simlai, A. (2021). Intelligent strategies for overload detection & handling for 5g network. In *2021 ieee 4th 5g world forum (5gwf)* (pp. 135–140).

Sree Lekshmi. (2022). *5G Service Based Architecture (SBA).* Calsoft. Retrieved from https://calsoftinc.com/blogs/2022/09/5g-service-based-architecture-sba.html ([Accessed 21-September-2022])

Tang, Q., Ermis, O., Nguyen, C. D., De Oliveira, A., & Hirtzig, A. (2022). A systematic analysis of 5g networks with a focus on 5g core security. *IEEE Access*, *10*, 18298–18319.

telekom. (2021). 5g-trace-visualizer. Retrieved from https://github.com/telekom/5g-trace-visualizer

The Wireshark Team. (2021). Wireshark, Go Deep. *The Wireshark Team.*. Retrieved from https://www.wireshark.org/

The Wireshark Team. tshark. (2021). Wireshark, Go Deep. *The Wireshark Team.*. Retrieved from https://www.wireshark.org/docs/man-pages/tshark.html

Tian, Z., Patil, R., Gurusamy, M., & McCloud, J. (2023). Adseq-5gcn: Anomaly detection from network traffic sequences in 5g core network control plane. , 75–82.

Tripathi, N. (2022). Delays have dangerous ends: Slow http/2 dos attacks into the wild and their real-time detection using event sequence analysis. *arXiv preprint arXiv:2203.16796*.

Tripathi, N., & Hubballi, N. (2018). Slow rate denial of service attacks against http/2 and detection. *Computers & security*, *72*, 255–272.

Tripathi, N., & Shaji, A. K. (2022). Defer no time, delays have dangerous ends: Slow http/2 dos attacks into the wild. In *2022 14th international conference on communication systems & networks (comsnets)* (pp. 194–198).

TS.123.502, G. (2025). 5G; Procedures for the 5G System (5GS) TS 123.502 v.18.8.0.

TS.129.518, G. (2025). 5G; 5G System; Access and Mobility Management Services; TS 129.518 v.18.8.0.

TS.23.288, G. (2024). Technical Specification Group Services and System Aspects; Architecture enhancements for 5G System (5GS) to support network data analytics services (Release 18) TS 23.288 v18.6.0.

TS.23.501, G. (2024). 5G; System architecture for the 5G System: TS 23.501 v.18.8.0.

TS.28.552, G. (2024). 5G; Management and orchestration; 5G performance measurements TS 28.552 v18.8.0. *The 3rd Generation Partnership Project (3GPP)*.

TS.29.502, G. (2025). 5G; 5G System; Session Management Services; Stage 3 TS 29.502 v18.9.0.

TS.33.501, G. (2025a). 5G; Security architecture and procedures for 5G System: TS 33.501 v.18.8.0. *The 3rd Generation Partnership Project (3GPP)*.

TS.33.501, G. (2025b). 5G; Security architecture and procedures for 5G System: TS 33.501 v.18.8.0.

TS.38.413, G. (2024). 5G; NG-RAN; NG Application Protocol (NGAP TS 38.413 v.18.4. *The 3rd Generation Partnership Project (3GPP)*.

TSG-SA3, G. (2022). Key issue on misuse of OAuth 2.0 access token by anomalous Network functions, TSG-SA3 meeting #108e, S3-221787. Retrieved from https://www.3gpp.org/ftp/TSG_SA/WG3_Security/TSGS3_108e/Docs/S3-221787.zip

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

VIII, C. (2022). Report on security vulnerabilities in http/2.

Wehbe, N., Alameddine, H. A., Pourzandi, M., & Assi, C. (2023). 5gshield: Http/2 anomaly detection in 5g service-based architecture. In *2023 ifip networking conference (ifip networking)* (pp. 1–9).

Wehbe, N., Alameddine, H. A., Pourzandi, M., & Assi, C. (2025). Empowering 5g sba security: Time series transformer for http/2 anomaly detection. *Computers & Security*.

Wehbe, N., Alameddine, H. A., Pourzandi, M., Bou-Harb, E., & Assi, C. (2022). A security assessment of http/2 usage in 5g service based architecture. *IEEE Communications Magazine*.

Wei, D., Shi, F., & Dhelim, S. (2022). A self-supervised learning model for unknown internet traffic identification based on surge period. *Future Internet*, *14*(10), 289.

Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., & Sun, L. (2022). Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*.

Xie, B., & Zhang, Q. (2012). Application-layer anomaly detection based on application-layer protocols' keywords. In *Proceedings of 2012 2nd international conference on computer science and network technology* (pp. 2131–2135).

Xona Partners Inc. (2024). *Assessment of rogers networks for resiliency and reliability following the 8 july 2022 outage – executive summary.* Retrieved from https://crtc.gc.ca/eng/publications/reports/xona2024.htm

Xu, J., Wu, H., Wang, J., & Long, M. (2021). Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*.

Yuan, L.-P., Choo, E., Yu, T., Khalil, I., & Zhu, S. (2021). Time-window based group-behavior supported method for accurate detection of anomalous users. , 250–262.

Yuan, Y., Gehrmann, C., Sternby, J., & Barriga, L. (2022). Insight of anomaly detection with nwdaf in 5g. In *2022 international conference on computer, information and telecommunication systems (cits)* (p. 1-6). doi: 10.1109/CITS55221.2022.9832914

Zeng, F., Chen, M., Qian, C., Wang, Y., Zhou, Y., & Tang, W. (2023). Multivariate time series anomaly detection with adversarial transformer architecture in the internet of things. *Future Generation Computer Systems*, *144*, 244–255.

Zhang, H., Xia, Y., Yan, T., & Liu, G. (2021). Unsupervised anomaly detection in multivariate time series through transformer-based variational autoencoder. In *2021 33rd chinese control and decision conference (ccdc)* (pp. 281–286).

Zhao, N.-Z. L. Z., Y. (2019). PyOD: a python toolbox for scalable outlier detection. *The Wireshark Team.*.