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

Nathalie Wehbe[1], Hyame Assem Alameddine[2], Makan Pourzandi[2], Chadi Assi[1]
[1] *Concordia University, Montreal, Canada* [2] *Ericsson, Montreal, Canada*
{nathalie.wehbe, chadi.assi}@concordia.ca, {hyame.a.alameddine, makan.pourzandi}@ericsson.com

***Abstract*—Fifth Generation (5G) core network leverages the application-layer Hypertext Transfer Protocol version 2 (HTTP/2) to enable the communication between the Network Functions (NFs) of its Service-Based Architecture (SBA). 5G SBA adopts the security-by-design principle, yet, the usage of HTTP/2 introduces some vulnerabilities related to its features exploitation. For instance, the HTTP/2 stream multiplexing attack exploits the stream multiplexing feature, which allows carrying multiple requests over a single TCP connection, and causes a Denial of Service (DoS) on 5G SBA. HTTP/2 attacks can be detected using traditional flow-based anomaly detection solutions in a web environment. Nonetheless, these solutions fall short in detecting these attacks in a 5G network, as we show in this work. To reinforce 5G core network security against HTTP/2 attacks, we propose 5GShield, a novel application-layer anomaly detection framework that uses neural networks, namely, Autoencoder, for anomaly detection. To evaluate our approach, we deploy a 5G testbed, simulate the HTTP/2 stream multiplexing attack and collect HTTP/2 data. Our experimental results show that 5GShield can detect HTTP/2 stream multiplexing attack 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.**

***Index Terms*—5G SBA, security, HTTP/2, stream multiplexing attack, anomaly detection, application-layer features**

I. Introduction

The 5G network revolution is driven by the increasing number of IoT devices and evolving services in the telecommunications sector. To cater to these services, the 5G Core (5GC) network adopts a Service-Based Architecture (SBA) in its design, leveraging cloud-native applications. The implementation of virtualization technologies like Network Functions Virtualization (NFV) and Software-Defined Networking (SDN) [1], [2] enables flexible 5G SBA design consisting of interconnected Network Functions (NFs) that provide access to network resources and capabilities via Service Based Interface (SBI) [3]. The communication between NFs is facilitated by RESTful Application Programming Interfaces (APIs) over HTTP/2 [3]. The HTTP/2 protocol ensures secure, reliable, efficient, and bidirectional communication [4]. The third Generation Partnership Project (3GPP) leverages the use of this harmonized protocol in 5GC Control Plane (CP) signaling [3], as well as other web-based technologies such as Transport Control Protocol (TCP), Transport Layer Security (TLS), and JavaScript Object Notation (JSON).

ISBN 978-3-903176-57-7 © 2023 IFIP

HTTP/2 enables the communication between an NF Service Consumer (NFc) (i.e., HTTP/2 client) and an NF Service Producer (NFp) (i.e., HTTP/2 server) [3] in the form of a *request/response* or *subscribe/notify*. It introduces new features such as stream multiplexing, flow control, header compression, and server push which allow it to meet the low-latency requirement of 5G services. However, recent studies [5]–[8] have shown that HTTP/2 is vulnerable to Distributed Denial of Service (DDoS) attacks such as slow-read, stream multiplexing, and HPACK bomb attacks, exploiting its features. Following these studies, the works in [7], [9], [10] evaluated the security risks posed by HTTP/2 in 5G SBA. This highlights the need for a robust security solution to enforce 5G network security against HTTP/2 attacks. While many works [6], [11] developed anomaly detection solutions to secure the web against HTTP/2 attacks using Machine Learning (ML) techniques, HTTP/2 attacks on 5G SBA were only assessed theoretically in [9], [10]. 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.

In this work, we argue that 5G networks are vulnerable to HTTP/2 attacks and demonstrate that HTTP/2 stream multiplexing attacks can occur between two 5G NFs (e.g., a compromised NFc can launch an HTTP/2 attack on an NFp). 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 [12]. As a result, existing anomaly detection methods that rely on flow-based features fail to discover such application-layer attacks [13].

This work answers the following questions: *How can we exploit the HTTP/2 stream multiplexing feature in a 5G network? How will 5G NFs behave under the HTTP/2 stream multiplexing attack? How can we detect this attack in 5G SBA?* Our contributions can be summarized as follows:

- We propose **5GShield**, an application-layer anomaly detection framework based on Autoencoder (AE) [14]. 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 stream multiplexing attack 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. Note that 5G NFs behavior remains hidden to flow-based anomaly detection solutions.

- We simulate and study the impact of HTTP/2 stream multiplexing attack in a 5G network using the open-source Free5GC [15] testbed and UERANSIM [16], a User Equipment (UE)/ Radio Access Network (RAN) simulator.
- We generate a 5G SBA HTTP/2 dataset that captures both normal and abnormal 5G SBA network behavior under the HTTP/2 stream multiplexing attack in both stealthy and non-stealthy modes. We simulate the attack from a compromised Session Management Function (SMF) towards an Access and Mobility Management Function (AMF) given that the AMF is a valuable target for attackers, as we explain in the sequel. Nonetheless, our work applies to HTTP/2 attacks between any other NFs of the 5G SBA.
- Our experimental results show that 5GShield can detect HTTP/2 stream multiplexing attack with an F1-score of 0.992, outperforming a flow-based anomaly detection solution that exhibits an F1-score of 0.78.

The rest of the paper is organized as follows. We present an overview of related works in Section II. We provide background information on 5G signaling and discuss the HTTP/2 stream multiplexing threat model in Section III and Section IV respectively. Details of the proposed 5GShield framework are presented in Section V. We explain our environment setup and data collection in Section VI. We present our main findings and experimental results in Section VII. We discuss in Section VIII 5GShield deployment options in a 5G network. We conclude and highlight our future work in Section IX.

II. Related Works

A. HTTP/2 and 5G SBA Security

The security of HTTP/2 was discussed in [8] 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 [5] presented five versions of slow rate DoS attack 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 [7] 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 [9], [10], [17] assessed HTTP/2 security in 5G SBA. Authors of [9] 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. In [10], the authors investigated the applicability of HTTP/2 attacks in 5G SBA. They noted that some attacks, e.g., HTTP/2 stream multiplexing attack, can only be addressed by sophisticated application-layer security and anomaly detection systems, which are currently unavailable. The work in [17] 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.

B. HTTP Anomaly Detection

Few works in the literature addressed HTTP/2 anomaly detection problem. Authors of [6] developed a real-time detection strategy based on event sequence analysis to detect HTTP/2 slow-read attacks in real-time using an HTTP/2 web server dataset. [11] detected HTTP/2 multiplexed AL-DDoS attacks by relying on an HTTP/1.1 dataset. Other works presenting anomaly detection solutions [12], [18]–[21] focused on DDoS attacks including HTTP/1.1 flooding attack rather than HTTP/2 attacks. Authors of [21] employed statistical methods to detect HTTP AL-DDoS attacks. [12] 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 [6] 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 [11] is the only work that addressed HTTP/2 stream multiplexing attack detection. However, the authors used an HTTP/1.1 dataset just like the remaining works which focused on HTTP/1.1 attacks.

Flow-based anomaly detection techniques are widely used in the literature. AE with flow-based features [18] and Convolutional Neural Network (CNN) [19] 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 [12], we argue that flow-based features are inefficient in detecting application-layer attacks (e.g., HTTP/2 attacks) as HTTP/2 may not exhibit malicious activities when their network traffic is observed from the network or transport layers.

C. HTTP/2 Dataset

The works in the literature [11], [18], [19], [21], [22] utilized various datasets such as CICIDS2018, CICDDoS2019, 4G-LTE, modified HTTP/1.1 to HTTP/2 dataset, and HTTP/2 web server dataset. HTTP/2 datasets are private and not collected in a 5G network while the others are public and not suitable for further research on anomaly detection in 5G. Unlike these works, we simulate and collect 5G HTTP/2 data from a 5G testbed.

III. Signaling in 5G SBA

A. Overview

5G network adopted an SBA for its CP, encompassing a set of interconnected NFs that communicate via an SBI (Figure 1). Such communication is enabled by HTTP/2 that was selected by 3GPP as the 5G signaling protocol, given its low latency, scalability, and extensibility [3]. In 5G SBA, the 5G CP NFs interact with each other using either *Request-Response*

or *Subscribe-Notify*. *Request-Response* is used when an NFc requests a service and an NFp responds, while *Subscribe-Notify* is employed when an NFc subscribes to an NFp event that causes the NFc to be called back when the event occurs (e.g., SMF subscribes to location report to get notified of the last known location of a UE or group of UEs by the AMF [23], etc.) [2], [3]. 5G NFs interactions are determined by the exchange of services via 3GPP standardized RESTful APIs [2], [3]. Thus, a 5G procedure (e.g., UE registration) is performed through HTTP/2 signaling and translated into a chain of API calls between 5G NFs. Hence, depending on the number of API calls and their processing, each 5G procedure can yield different communication and computation overheads in the network. Consequently, computationally expensive API calls can be used by an attacker to perform HTTP/2 attacks.

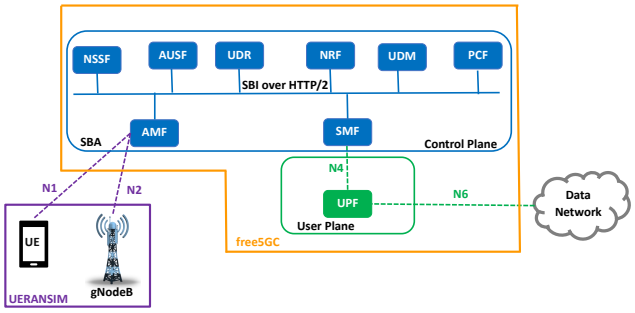

Fig. 1: 5G Service-Based Architecture [2], [15], [16]

B. 5G SBA Signaling Security

5G network was developed with security in mind, thus, exploiting it through HTTP/2 vulnerabilities while leveraging 5G procedures and API calls is not an easy task. To better assess the feasibility of HTTP/2 attacks in 5G networks, we highlight 5G SBA security as required by 3GPP. In fact, 5G signaling between NFs is secured through authentication and authorization mechanisms. Authentication for confidentiality and integrity protection of the messages exchanged between NFs is ensured through TLS [3], [24]. Service authorization is however, granted by the OAuth 2.0 protocol via the Network Repository Function (NRF) [4], [25]. NRF provides an OAuth 2.0 access token to 5G NFs to access each other services [4], [26]. Note that authorization and authentication exist in non-roaming and roaming scenarios [4]. In a roaming scenario, the protection of the 5G network from unauthorized access and attacks is performed by a Security Edge Protection Proxy (SEPP) that acts as a security gateway on the interconnections between roaming partners. SEPP provides end-to-end authentication, confidentiality, and integrity protection via signatures and encryption of HTTP/2 messages along with application-layer security between NFs and the roaming partners to enable their secure communication [4].

IV. HTTP/2 Stream Multiplexing Attack in 5G SBA

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

A. 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 [27]. The adoption of hyper-scale cloud by mobile operators extends the attack surface of their networks and makes their virtual NFs vulnerable [27]. 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 [28], and CVE-2019-5736 [29]) [1]. Attackers can take advantage of a breach of isolation between 5G network slices that share one or multiple NFs [30].
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 [4]. 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 [30] 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)) [26], we assume that the attacker can gain access to such information by monitoring NFc communications or even requesting such information from other NFs.

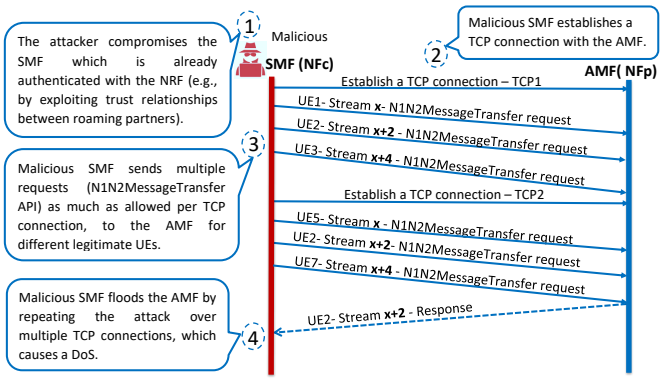

Fig. 2: HTTP/2 stream multiplexing attack on AMF

B. Threat Model - HTTP/2 Stream Multiplexing Attack

Given the prior assumptions, we simulate the HTTP/2 stream multiplexing attack 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 [23]. In addition, the AMF is exposed to external networks, which extend its attack surface and put it at risk [31]. A DDoS attack against the AMF can significantly reduce the availability of 5G services and even cause network outages [31]. 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 [23]. 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. [23]. We leverage this API to perform the HTTP/2 stream multiplexing attack in two forms:

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

In Figure 2, we illustrate the HTTP/2 stream multiplexing attack 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 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 [32]. Finally, as the attacker used a subset of legitimate UEs information and requests, the detection of this application-layer attack becomes challenging.

V. 5GShield Framework

In this section, we introduce the 5GShield framework (Figure 3), our novel and intelligent application-layer anomaly detection solution designed to detect HTTP/2 attacks including stream multiplexing attack.

A. Data Collection and Pre-processing Module

The data collection and pre-processing module aims at collecting application-layer information and pre-process it for

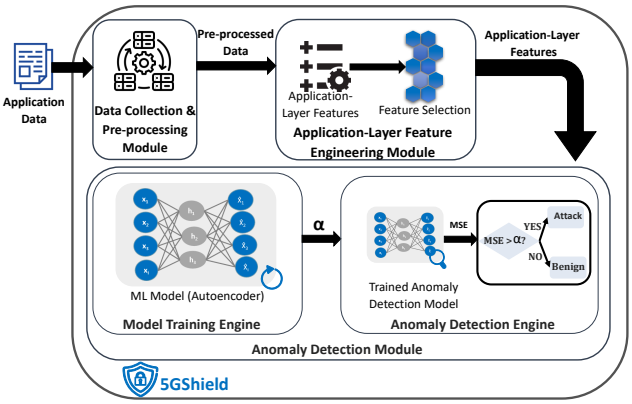

Fig. 3: An overview of 5GShield framework and its modules

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 [33] 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 [33]. 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.

B. 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 stream multiplexing attack, as 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 I the AMF features that we select. We distinguish the 3GPP-AMF features that are based on 3GPP guidelines [33], from which we choose the PM of AMF pertaining to the $Namf\_Communication\_N1N2MessageTransfer$

API (Section IV-B). 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 I), we perform feature normalization and then, we select the most relevant ones.

At the feature selection stage, we use the variance threshold [34] 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 [34]. 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.

C. Anomaly Detection Module

The anomaly detection module consists of a model training engine and an anomaly detection engine (Figure 3). 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 [14]. 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 [18]. 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. 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.

VI. Environment Setup

In this section, we present the environment that we use to simulate normal and malicious network traffic. It also includes details on the 5G testbed and discussions on the data pre-processing and the feature engineering that we perform on the data collected from our testbed.

A. Simulation Setup and 5G Testbed

We perform the evaluation using Free5GC [15], an open-source 5GC testbed, and UERANSIM [16] that provides a UE/RAN simulator. Our Free5GC and UERANSIM are installed on a Virtual Machine (VM) running on top of OpenStack [35]. 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 [36], in which the 5G NFs are deployed on different containers in the same VM. Using Python 3.8, we implement our 5GShield

TABLE I: 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, totalUnsuccessfulRequest

solution, while our anomaly detection AE model leverages PyOD library 1.0.6 [37].

B. 5G Network Simulation

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 stream multiplexing attack simulation. This requires simulating 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 II).

TABLE II: 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

Normal network behavior - Benign dataset generation — To simulate normal network traffic behavior, we consider 50 UEs and perform multiple 5G procedures selected randomly from those provided by the UERANSIM (Table II). 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.) [26]. 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) [26] 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 II 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.

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 IV-B). 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 Free5GC testbed. We initiate the malicious requests while other legitimate requests are being processed concurrently in the 5G network. We simulate both stealthy and non-stealthy versions of the HTTP/2 stream multiplexing attack. For stealthy attack simulation, 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 [26] while respecting their precedence constraints (Table II). In contrast, for the non-stealthy attack simulation, 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.

Data collection and attack impact — Using the benign and malicious network simulations described above, we collect from the Free5GC testbed the application layer information at the AMF (Section V-A). 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 VI-C. During the attack simulation, we observe an increase in the Central Processing Unit (CPU) consumption at the AMF once the attack starts at 576 seconds (Figure 4). 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).

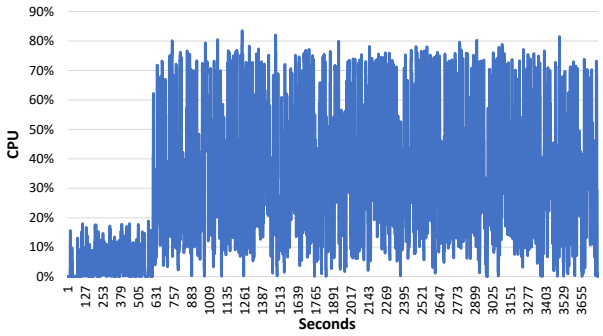

Fig. 4: AMF CPU consumption during the attack

C. Data Pre-processing and Feature Engineering

We pre-process the collected data to extract application-layer features to use in 5GShield, and flow-based features to

[1]UE PDU session establishment is automatically triggered in Free5GC [15] after a UE registration.

[2]UE PDU session establishment procedure is automatically triggered after the UE release PDU session procedure in Free5GC [15].

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 V-B), listed in Table I. 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 [34] (Section V-B). 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 [38]. This results in 84 features listed in [38]. We clean and normalize the collected features using oneHotEncoder. Then using the same variance threshold ML method [34] 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* [38], 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 simulated 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 simulations. 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.

D. 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.

VII. 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.

TABLE III: 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

A. 5GShield Application-layer Anomaly Detection Solution

AE architecture selection — To determine the architecture of the AE to use in our 5GShield anomaly detection module (Section V-C), and which better recognizes the HTTP/2 stream multiplexing attack, 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. Due to space limitation, we omit the performance of the different evaluated AE architectures. However, 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 III). We observe the average reconstruction loss across the different training epochs for the training model using benign unlabelled data. As shown in Figure 5, the training loss and the validation loss start to converge after 30 epochs and the AE depicts a reasonable convergence within 200 epochs.

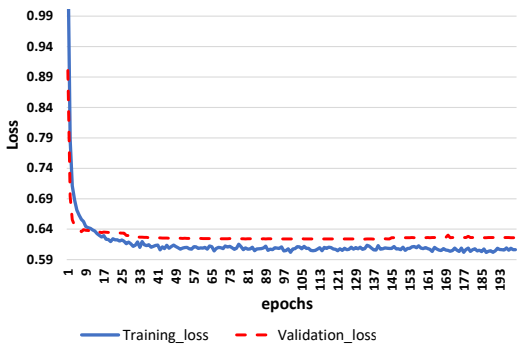

Fig. 5: Training and validation loss of 5GShield AE model

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 6, we evaluate the model performance using a test dataset of another (other than optimization dataset) 1400 benign and 4600 malicious records. Figure 6 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 6. As a result, 5GShield with AE using application-layer features achieves good detection performance with an F1-score of 0.992.

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

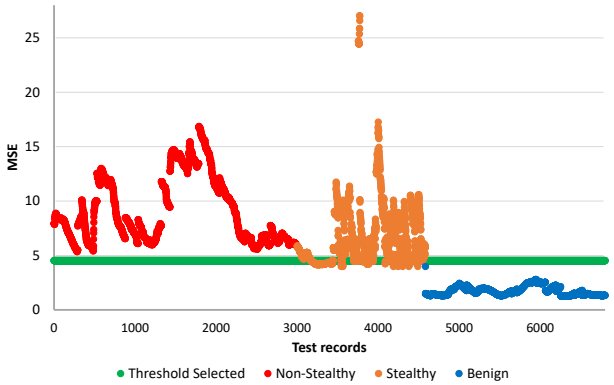

Fig. 6: Anomaly scores for test dataset records

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 III). We use the optimization and test datasets to select the threshold and test the model respectively. Figure 7 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.

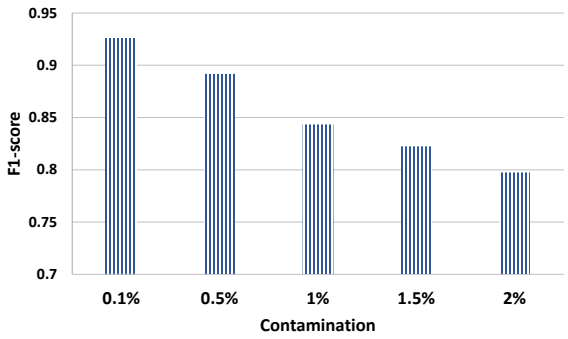

Fig. 7: F1-score of 5GShield model with contaminated data

B. 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 VI-B). 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 III. 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.

C. 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 [39]. 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 8 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.

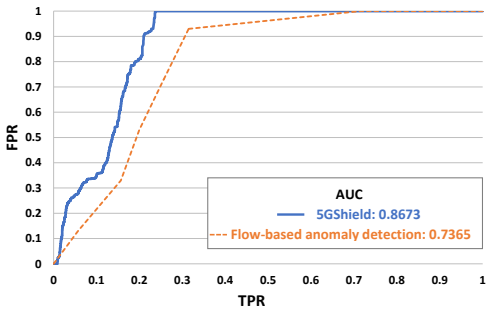

Fig. 8: AUC-ROC of 5GShield and flow-based anomaly detection solution

VIII. 5GShield Deployment Options

The 5GShield framework 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) [40]. 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 [40].

IX. Conclusion

In this work, we proposed 5GShield, an application-layer anomaly detection framework to protect 5G networks against HTTP/2 attacks. For that, we simulated the HTTP/2 stream multiplexing attack in a 5G testbed based on Free5GC and UERANSIM. Using an AE as our anomaly detection model, we showed that 5GShield achieves an F1-score of 0.992 and outperforms a flow-based anomaly detection solution that depicts an F1-score of 0.78. The superior detection performance of 5GShield shows the efficiency of using PM counters as application-layer features that capture 5G NFs profiles, services, behavior, and communications with their peers. As these counters can be collected by the NF itself or shared with a central NWDAF, 5GShield can thus, be deployed at any NF including the NWDAF.

As 5GShield can be deployed with any ML model, in future work, we aim at enhancing its detection performance by exploring additional ML techniques while also considering richer datasets depicting more 5G procedures (currently unavailable in UERANSIM) and various HTTP/2 attacks.

Acknowledgments

The authors would like to thank Dr. Amine Boukhtouta, Dr. Boubakr Nour, Dr. Luis Suárez, and Dr. Yosr Jarraya from Ericsson Research, for their invaluable feedback.

References

[1] T. Madi, H. A. Alameddine, M. Pourzandi, and A. Boukhtouta, "Nfv security survey in 5g networks: A three-dimensional threat taxonomy," *Computer Networks*, vol. 197, p. 108288, 2021.

[2] 3GPP, "5G; System architecture for the 5G System: TS 23.501 v.17.5.0," 2022.

[3] 3GPP, "5G; 5G System; Technical Realization of Service Based Architecture; Stage 3: TS 29.500 v.17.7.0," 2022.

[4] 3GPP, "5G; Security architecture and procedures for 5G System: TS 33.501 v.17.5.0."

[5] N. Tripathi and N. Hubballi, "Slow rate denial of service attacks against http/2 and detection," *Computers & security*, vol. 72, pp. 255–272, 2018.

[6] N. Tripathi and A. K. Shaji, "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)*. IEEE, 2022, pp. 194–198.

[7] A. Praseed and P. S. Thilagam, "Multiplexed asymmetric attacks: Next-generation ddos on http/2 servers," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1790–1800, 2019.

[8] Imperva, "HTTP/2: In-depth analysis of the top four flaws of the next generation web protocol," 2016.

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

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

[11] A. Praseed and P. S. Thilagam, "Fuzzy request set modelling for detecting multiplexed asymmetric ddos attacks on http/2 servers," *Expert Systems with Applications*, vol. 186, p. 115697, 2021.

[12] B. Xie and Q. Zhang, "Application-layer anomaly detection based on application-layer protocols' keywords," in *Proceedings of 2012 2nd International Conference on Computer Science and Network Technology*. IEEE, 2012, pp. 2131–2135.

[13] T. Madi, H. A. Alameddine, M. Pourzandi, A. Boukhtouta, M. Shoukry, and C. Assi, "Autoguard: A dual intelligence proactive anomaly detection at application-layer in 5g networks," in *European Symposium on Research in Computer Security*. Springer, 2021, pp. 715–735.

[14] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, 2018.

[15] Free5GC, "Free5GC," 2021. [Online]. Available: https://www.free5gc.org/

[16] aligungr, "UERANSIM," 2021. [Online]. Available: https://github.com/aligungr/UERANSIM

[17] R. COMMUNICATIONS SECURITY and I. C. V. W. G. . G. S. P. SECURITY, "Report on security vulnerabilities in http/2," 2022.

[18] M. A. Salahuddin, V. Pourahmadi, H. A. Alameddine, M. F. Bari, and R. Boutaba, "Chronos: Ddos attack detection using time-based autoencoder," *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 627–641, 2021.

[19] B. Hussain, Q. Du, B. Sun, and Z. Han, "Deep learning-based ddos-attack detection for cyber–physical system over 5g network," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 860–870, 2020.

[20] J. Lam and R. Abbas, "Machine learning based anomaly detection for 5g networks," *arXiv preprint arXiv:2003.03474*, 2020.

[21] A. Praseed and P. S. Thilagam, "Modelling behavioural dynamics for asymmetric application layer ddos detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 617–626, 2020.

[22] V. Pourahmadi, H. A. Alameddine, M. A. Salahuddin, and R. Boutaba, "Spotting anomalies at the edge: Outlier exposure-based cross-silo federated learning for ddos detection," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–14, 2022.

[23] 3GPP, "5G; 5G System; Access and Mobility Management Services; TS 129.518 v.17.5.0," 2022.

[24] B. Christine Jost, "Security for 5G Service-Based Architecture: What you need to know," 2020, [Accessed 18-March-2022]. [Online]. Available: https://www.ericsson.com/en/blog/2020/8/security-for-5g-service-based-architecture

[25] IETF, "The OAuth 2.0 Authorization Framework RFC 6749," 2022.

[26] 3GPP, "5G; Procedures for the 5G System (5GS) TS 123.502 v.17.5.0," 2022.

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

[28] N. V. D. (NVD), "Cve-2016-5195," 2019. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2016-5195

[29] N. V. Database, "Cve-2016-5736," 2019. [Online]. Available: https://nvd.nist.gov/vuln/detail/CVE-2019-5736

[30] AdaptiveMobile, "A Slice in Time: Slicing Security in 5G Core Networks," 2021. [Online]. Available: https://info.adaptivemobile.com/network-slicing-security?hsLang=en#download

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

[32] IETF, "Hypertext Transfer Protocol Version 2 (HTTP/2) - RFC 7540," 2015.

[33] 3GPP, "5G; Management and orchestration; 5G performance measurements TS 28.552 V17.4.0," 2022.

[34] scikit learn, "scikit-learn," 2021. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html

[35] OpenStack, "Build the Future of Open Infrastructure." 2021. [Online]. Available: https://www.openstack.org/

[36] Free5GC, "Free5GC-compose," 2021. [Online]. Available: https://github.com/free5gc/free5gc-compose/tree/v3.0.5

[37] N. Z. L. Z. Zhao, Y., "PyOD: a python toolbox for scalable outlier detection," 2019.

[38] C. I. Cybersecurity, "Cicflowmeter," 2020. [Online]. Available: https://github.com/CanadianInstituteForCybersecurity/CICFlowMeter/blob/master/ReadMe.txt

[39] H. Dalianis, "Evaluation metrics and evaluation," in *Clinical text mining*. Springer, 2018, pp. 45–53.

[40] Y. Yuan, C. Gehrmann, J. Sternby, and L. Barriga, "Insight of anomaly detection with nwdaf in 5g," in *2022 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2022, pp. 1–6.