# PUL-Inter-slice Defender: An Anomaly Detection Solution for Distributed Slice Mobility Attacks

Ricardo Misael Ayala Molina<sup>1</sup>, Hyame Assem Alameddine<sup>2</sup>, Makan Pourzandi<sup>2</sup>, Chadi Assi<sup>1</sup> 

<sup>1</sup> Concordia University, Montreal, Canada <sup>2</sup> Ericsson, Montreal, Canada

Abstract—Network Slices (NSs) are virtual networks operating over a shared physical infrastructure, each designed to meet specific application requirements while maintaining consistent Quality of Service (QoS). In Fifth Generation (5G) networks, User Equipment (UE) can connect to and seamlessly switch between multiple NSs to access diverse services. However, this flexibility, known as Inter-Slice Switching (ISS), introduces a potential vulnerability that can be exploited to launch Distributed Slice Mobility (DSM) attacks, a form of Distributed Denial of Service (DDoS) attack. To secure 5G networks and their NSs against DSM attacks, we present in this work, PUL-Inter-Slice Defender; an anomaly detection solution that leverages Positive Unlabeled Learning (PUL) and incorporates a combination of Long Short-Term Memory Autoencoders and K-Means clustering. PUL-Inter-Slice Defender leverages the Third Generation Partnership Project (3GPP) key performance indicators and performance measurement counters as features for its machine learning models to detect DSM attack variants while maintaining robustness in the presence of contaminated training data. When evaluated on data collected from our 5G testbed based on the open-source free5GC and UERANSIM, a UE/ Radio Access Network (RAN) simulator; PUL-Inter-Slice Defender achieved F1-scores exceeding 98.50% on training datasets with 10% to 40% attack contamination, consistently outperforming its counterpart Inter-Slice Defender and other PUL based solutions combining One-Class Support Vector Machine (OCSVM) with Random Forest and XGBoost.

*Index Terms*—Positive-unlabeled learning, contaminated datasets, network slicing, 5G, security, inter-slice switching, anomaly detection, machine learning.

#### I. INTRODUCTION

Network slicing is a cornerstone technology and a vital enabler for Fifth Generation (5G) networks, allowing them to deliver services that accommodate a wide range of specifications including high data rates, ultra low latency, diverse security requirements, dense connectivity, among others [1]. The critical importance of network slicing is demonstrated by its central role in the development of virtual end-to-end networks, designated as Network Slices (NSs), which are individually configured to serve specific vertical industry applications [1]–[3].

Network slicing enables User Equipment (UE) to seamlessly transition between various NSs [4] to address specific operational requirements such as Quality of Service (QoS), security levels, and service cost, among other factors [5]. UEs can be smoothly reallocated from one NS to another either based on a choice they opted for (e.g., enhance their QoS, access another service, etc.) or based on the network state and configuration by the Mobile Network Operator (MNO) (e.g., based on NS load and management configuration, etc.) [5].

The intrinsic versatility of NSs opens up new security risks, making them vulnerable to attacks [6], [7]. Notably, each time

a UE switches between NSs, a process known as Inter-Slice Switching (ISS) is launched [5]. ISS requires the UE to reinitiate the authentication and registration procedures among others [8]. These procedures generate a substantial increase in signaling traffic within the 5G Control Plane (CP) and between the CP and the UE. Malicious actors can induce numerous ISS events to cause a flood of signaling messages that can overload the 5G Network Functions (NFs) such as the Access Mobility Management Function (AMF) and the Session Management Function (SMF). This scenario, known as Distributed Slice Mobility (DSM) attack, can culminate in a Distributed Denial of Service (DDoS) attack and disrupts the connection for legitimate UEs [9].

DSM attack along with its economical and performance damage was first discussed in [9]. The same authors then proposed a protocol to secure the network against DSM attacks [10]. The proposed protocol automatically selects an NS for a UE based on its subscription and services offered by external networks. Subsequently, [11] proposed a method for detecting DSM attacks by evaluating the average waiting time and switching rate. Lately, we devised two variations of the DSM attack and created Inter-Slice Defender [12], a binary classifier based on a Long Short-Term Memory (LSTM)-Autoencoder model to detect them, leveraging 3GPP Key Performance Indicators (KPI) and Performance Measurement (PM) counters. To the best of our knowledge, [11] and [12] are the only works which addressed DSM attack detection while other works [13]–[15] focused on DDoS attack detection in general using Machine Learning (ML) techniques without evaluating the effectiveness of their solutions in detecting the DSM attack.

Furthermore, none of the aforementioned works examined the robustness of their ML models under contaminated training conditions, raising concerns about their effectiveness in real-world deployments. In practice, training data often contains noise, mislabeled instances, or samples from overlapping distributions, all of which can undermine model accuracy and reliability [16]. Addressing this challenge is essential for developing DSM attack detection solutions that remain resilient and effective in operational environments, underscoring the need for further research into methods capable of handling contaminated datasets.

To address this gap, we build upon our previous work on Inter-Slice Defender [12], which lacked robustness under such conditions, and propose a novel network-slicing anomaly detection solution that integrates Positive-Unlabeled Learning (PUL) into the training process [17]. Our model is designed to

detect two DSM attack variants: Random Slice Attack (RSA), where UEs are switched to random NSs, and Target Slice Attack (TSA), where UEs are redirected to pre-selected NSs.

PUL assumes a set of labeled positive data and a set of unlabeled data that includes both positive (i.e., benign data in our work) and negative (i.e., DSM attack) samples, in order to build a classifier capable of distinguishing between them [18]. This solution is well-suited for real-world anomaly detection, where benign data can be reliably identified by experts during normal network operations. The assumption of unlabeled data aligns with typical unsupervised learning scenarios involving contaminated datasets, where training data predominantly consists of benign samples but may include undetected attacks. Therefore, PUL provides a solid methodological foundation for improving the robustness of Inter-Slice Defender in detecting DSM attacks, leading to the development of our proposed PUL-Inter-Slice Defender.

Our contributions are summarized as follows:

- We develop PUL-Inter-Slice Defender, a novel network slicing anomaly detection solution that incorporates PUL to detect DSM attack variations in contaminated training datasets comprising a blend of normal behavior samples and attack instances. Our approach employs an LSTM-Autoencoder to extract latent features profiling ISS events across key 5G procedures, including UE authentication, registration, PDU session establishment, and deregistration. These features are then clustered using K-Means to differentiate between benign and attack patterns.
- We train and test the PUL-Inter-Slice Defender using a dataset containing both benign and attack samples within a 5G network slicing environment. To the best of our knowledge, this is the first study to assess the effectiveness of PUL for detecting slicing-related attacks under such conditions.
- Using the open source free5GC testbed [19] and the UERANSIM [20] simulator, we build a 5G network with four different NSs and we adapt it to emulate different variations of the DSM attack.
- We emulate both normal and DSM attack traffic on our 5G testbed and extract the corresponding data to compute 3GPP KPIs and PM counters, which serve as input to the PUL-Inter-Slice Defender. To the best of our knowledge, the resulting dataset is the first of its kind for network slicing anomaly detection.
- We developed two PUL-based solutions: one combining One-Class Support Vector Machine (OCSVM) with Random Forest (RF), hereafter called PUL-OCSVM-RF, and another combining OCSVM with XGBoost, subsequently termed PUL-OCSVM-XGBoost. We use these as benchmarks, along with the Inter-Slice Defender, to compare against PUL-Inter-Slice Defender.
- Our experimental results demonstrate PUL-Inter-Slice Defender robustness in detecting RSA and TSA, with an average F1-score exceeding 98% across training datasets containing 10% to 40% attack samples. This performance substantially surpasses that of the benchmark models, including Inter-Slice Defender, PUL-OCSVM-RF, and PUL-OCSVM-XGBoost, none of which exceed an F1-

score of 89.87% under the same conditions.

The remainder of the paper is structured as follows: Section III reviews related work. Section III describes the DSM attack variations and threat model, while Section IV summarizes the original Inter-Slice Defender. Section V presents the PUL-Inter-Slice Defender architecture and assumptions. Section VI outlines the emulation setup and datasets. Section VII examines the impact of RSA and TSA. Section VIII reports experimental results; Section IX discusses implementation considerations in real-world 5G environments, and Section X concludes the paper.

#### II. LITERATURE REVIEW

#### A. DSM Attack

A limited number of research studies have been dedicated to analyzing and detecting DSM attacks. For instance, Sathi et al. [9] were the first to propose the DSM attack and theoretically study the economic and performance damage it can cause on the network. They note that such damage exceeds those resulting from other DDoS attacks. Sajjad et al. [5] detailed 3GPP-based procedures enabling UE mobility and its transition between NSs. They highlighted critical UE mobility challenges and suggested potential avenues for future research. Bisht et al. [11] illustrated the consequences of ISS events that result in a DDoS attack. They employed an algorithm based on two metrics, average waiting time and switching rate, to identify and block compromised UEs. From a different perspective, Sathi et al. [10] introduced a preventive approach through a novel NS selection protocol. The protocol suggests that the network chooses the best NS for the UE based on its subscription and data network services which contradicts 3GPP 5G standards that allow the UE to request the services of a specific NS [25].

The work tackling DSM attack is limited to theoretical studies and does not consider the practical implementation of this attack in a real 5G network. Further, DSM attack detection solutions are limited to [11], which followed a simplistic algorithmic approach based on two metrics that were not tested on a real 5G network slicing-based dataset, thus overlooking the complexity of this attack.

#### B. Traditional DDoS Attacks Targeting Network Slices

DDoS attacks are known to be the most disruptive attacks in the realm of cybersecurity [26], [27]. While DSM has been identified as a DDoS attack on NSs, DDoS flooding attacks, such as User Data Protocol (UDP) lag and Transmission Control Protocol (TCP) SYN, among others, have also gained attention. DDoS flooding attacks can be caused by exploiting vulnerabilities in the high number of UEs in 5G networks and targeting its NSs. To detect these attacks, Deep Learning (DL) and mathematical models have been proposed.

For example, Khan et al. [13] used a bidirectional-LSTM model to detect DDoS attacks (i.e., UDP flooding and TCP SYN attacks) in two NSs. They evaluated the impact of these attacks by measuring the incurred bandwidth and latency. Similarly, the authors in [14] devised a solution leveraging LSTM to identify DDoS attacks and detect if UEs' NS requests are either legitimate or malicious. Further, Thantharate et al.

TABLE I: State-of-the-art on DSM attack detection and PUL (X = Not Supported,  $\sqrt{\ }$  = Supported, N/A=Not Applicable).

Reference	Approach	ML Model(s)	Type of Feature	Robustness in the Presence of Contamination	5G	F1-score
Approaches to detect D	SM attacks ( [11], [12]) a	and DDoS attacks based on	flow-based features ( [13]–[	[15]) in 5G network witho	ut incorp	porating PUL
Bisht et al. [11]	Two metrics	N/A	N/A	X	✓	91%-94%
R.M.A. Molina et al. [12]	DL Model	LSTM-Autoencoder	PM Counters, KPIs	X	$\checkmark$	98.75%
Khan et al. [13]	DL model	BiLSTM	Flow-based features	X	$\checkmark$	99.99%
Kuadey et al. [14]	DL model	LSTM	Flow-based features	X	$\checkmark$	99.965%
Thantharate et al. [15]	DL model	Deep Neural Network	Flow-based features	X	$\checkmark$	N/A
	Approaches	to detect DDoS attacks usi	ing PUL in technologies othe	r than 5G		
G. Long et al. [21]	PUL, DL Model	VRNN	Latent feature represen- tation from raw traffic	$\checkmark$	X	N/A
R. Dilworth et al. [22]	PUL, DL Model	XGBoost, RF SVM, Naïve Bayes	Flow-based features	✓	X	40.30%-99.05%
Z. Fan et al. [23]	PUL, DL Model	Binary classifier	Flow-based features	✓	X	94.51%
S. Lv. et al. [24]	PUL, DL Model	Custom neural network	Flow-based features	$\checkmark$	X	88.71%-99.65%
Our ap	proach uses PUL in 5G no	etworks, leveraging NS-spec	cific features and 5G-specific	datasets to detect DSM a	ttacks	
This work	PUL, DL/ML models	LSTM-Autoencoder and K-means	PM counters and KPIs	$\checkmark$	✓	*98.50%-99.33

\*F1-score values across variable contamination/negative sample proportions ranging from 10% to 40% in the training dataset.

[15] developed Secure5G that employs a Convolutional Neural Network (CNN) for early DDoS attack detection by analyzing network traffic patterns collected from a 5G testbed. Secure5G detects anomalous UEs' requests targeting multiple NSs and redirects suspicious UEs to a quarantine NS.

Most of the aforementioned work on NS attack detection focuses on general DDoS attacks that are not NS specific. Despite the efforts presented in some of these works to test these attacks on a 5G testbed and use the generated dataset for DDoS attack detection, their anomaly detection models remain limited to flow-based features that do not capture any NS-specific characteristics. Moreover, these studies do not examine nor evaluate the performance of their detection solutions in detecting the DSM attack and its variations.

It is important to emphasize that there are other studies [28]–[30] that employ autoencoders, LSTM, clustering algorithms, or combinations of these methods for attack detection in domains such as traditional Internet-based networks. While these studies may give the impression of similarity to our proposed approach, they neither incorporate 5G-specific features nor adopt PUL in their methodologies. These two aspects are central to our approach, and their absence makes those studies unsuitable for direct comparison.

# C. Positive-Unlabeled Learning in DDoS Attack Detection

Although PUL has been widely studied in various domains, its potential in 5G technology remains underexplored, particularly for anomaly detection. Conversely, the concept of learning from positive and unlabeled data has gained significant attention in the machine learning community for anomaly detection applications in other technologies, due to its practical relevance in deployment scenarios where obtaining fully clean or comprehensively labeled datasets is infeasible.

For example, [21] proposed PUNet, a PUL-based semisupervised anomaly detection model that addresses distribution shift and data imbalance using Variational Recurrent Neural Network (VRNN)-based features and reconstruction-lossguided pseudo-labeling. It uses CatBoost for classification and effectively detects DDoS and encrypted traffic anomalies. [22] applies PUL to cloud-based DDoS detection using the BCCC-cPacket-Cloud-DDoS-2024 dataset. It frames DDoS attacks as positive samples and evaluates multiple classifiers. The study highlights PUL's effectiveness in handling limited labeled data for detecting DDoS attacks in cloud environments. [23] This work proposes SRPU, a PUL-based solution for malicious traffic detection under class imbalance. It combines self-paced learning with a reweighting strategy to gradually select confident samples and manage noisy data. SRPU effectively detects various attack types, including DDoS, even with limited labeled samples. [24] introduced a method that uses non-negative risk estimator learning for intrusion detection in network systems. This approach treats cyber-attacks as positive samples in a PUL scenario and utilizes a risk estimator to calculate binary classification loss, specifically enhanced with focal loss to address data imbalance issues.

Given the demonstrated success of PUL in various anomaly detection applications, we adopt this approach in our methodology for detecting DSM attacks in 5G networks, as obtaining clean, fully labeled datasets is often impractical.

# D. Overall Comparison to Our Approach

In this paper, we highlight the lack of previous works that detect DSM attacks while considering deploying their solutions in real-world scenarios where obtaining a clean dataset is not always feasible. We discuss several key limitations of the related works in literature and provide a comprehensive comparison in Table I with our work against them. For instance, the works presented in this table [11]-[15] operate under the assumption that the training dataset is clean, meaning it exclusively contains samples from the class of interest without any noise or irrelevant data. This assumption simplifies the learning process, as the model can focus on distinguishing between well-defined classes without the interference of mislabeled or out-of-distribution samples. Nevertheless, in many real-world scenarios, this assumption might not always be valid. In practice, training datasets are often contaminated with a certain percentage of unwanted instances [31], such as mislabeled data, samples from other classes, or outliers. These contaminants can distort the learned model, leading to reduced performance, biased predictions,

or even incorrect conclusions. Furthermore, some studies [12]–[14] achieved high F1-scores over 98% by proposing approaches that leverage DL models. However, these results are based on ideal, uncontaminated datasets. Consequently, the robustness and generalization of these works against data contamination remain significant gaps in existing research, underscoring the importance of developing models that can maintain high performance in real-world conditions.

In contrast to the assumptions made by previous studies, [21]–[24] acknowledge that obtaining clean, uncontaminated training datasets is rarely feasible in practical settings. These works incorporate the PUL paradigm into their anomaly detection approaches, enabling their models to learn from datasets composed of both labeled and unlabeled data. As shown in Table I, their methods achieve promising F1-scores exceeding 94%, detecting various types of attacks such as DDoS, thereby demonstrating the effectiveness of PUL in handling real-world data contamination. However, despite these strengths, their approaches remain limited in scope. First, their evaluations are not conducted on 5G-specific datasets nor within a 5G testbed environment, raising concerns about their applicability to current and future mobile network architectures. Second, similar to [11], [13]-[15], these studies do not consider NS-specific features or behaviors. As a result, they overlook critical NS-level dynamics that are essential for accurately detecting sophisticated slice-based attacks such as DSM. Furthermore, none of the studies [21]–[24] examine or evaluate the effectiveness of their detection models against DSM attack variations like RSA and TSA. Consequently, their robustness and generalization capabilities within the context of 5G and NS-oriented threat landscapes remain unaddressed. This highlights the need for specialized detection mechanisms, such as the one we propose, that are not only designed for contaminated data but are also tailored to detect DSM attacks within realistic 5G network environments using NS-aware features and real testbed implementations.

We provide a practical and comprehensive approach that combines the PUL paradigm with an LSTM-Autoencoder integrated with the K-means algorithm for feature extraction and clustering. This architecture is specifically designed to capture the complex and evolving patterns of DSM attacks in the presence of contaminated data. Unlike prior studies, our solution is implemented and evaluated in a realistic 5G testbed that supports network slicing (Section VI-A), from which we collect traffic to compute 3GPP-based features (e.g., KPIs and PM counters) for profiling the behavior of NSs and core NFs. We also examine the impact of DSM attacks on CPU usage in the CP NFs, demonstrating their disruptive effect on 5G operations. This end-to-end methodology underscores the robustness and practical value of our proposed solution for DSM attack detection in real-world 5G deployments.

#### III. DSM ATTACK - THREAT MODEL AND VARIATIONS

The DSM attack is a DDoS attack that exploits UEs ISS events to disrupt the performance of both the 5G network CP and its NSs. Such disruption is caused by the signaling associated with the high number of malicious ISS events. In fact, ISS triggers the PDU session release procedure to

release the UE PDU session in its current NS, registration, and PDU session establishment to a new desired NS, and other procedures, such as UE authentication to the network [5], [8]. These procedures involve a sheer volume of signaling messages within the 5G CP and the NSs. In the following, we detail the DSM attack assumptions, and threat model while highlighting two of its variations (i.e., RSA and TSA).

# A. Attack Assumptions

To perform RSA and TSA, these assumptions are followed:

- Access to a set of compromised UEs. UEs and Internet
  of Things (IoT) devices are known to be highly vulnerable to attacks such as those noted in [32]. An attacker,
  can thus compromise a set of UEs and use them as a
  botnet to perform a DSM attack.
- 2) UEs NS configurations and credential information. We assume that the attacker has access to compromised UEs credentials (i.e., cryptographic keys, NSs configuration, etc.) and can use them to successfully connect to the MNO network and its NSs.
- Remote activation of ISS. We assume that the attacker is able to remotely access and manipulate the compromised UEs in order to force them to trigger ISS to switch between their accessible NSs.

#### B. DSM Attack Variations - Threat Model

In this work, we exploit two variations of the DSM attack, illustrated in Fig. 1:

Random Slice Attack (RSA). To perform a RSA, the attacker selects different subsets of compromised UEs, and simultaneously switch them to randomly selected NSs. RSA creates varying loads on these random NSs as a result of the varying number of connected UEs.

**Target Slice Attack (TSA).** To perform a TSA, the attacker selects a target NS to attack and triggers ISS events to switch the compromised set of UEs to the selected NS.

According to Fig. 1, RSA and TSA can be performed as follows:

- 1) Select RSA/TSA start time. The attacker identifies the network's peak times, during which a substantial number of UEs are connected to the network in order to perform the RSA/TSA. This will impose a more pronounced impact on the CP's NFs by augmenting the peak network load with a high frequency of ISS events that will be initiated by the attack.
- 2) Compromise UEs. The attacker compromises a set of UEs and use them as botnet to launch the RSA/TSA. The strategic use of UEs as attack vectors exploits the inherent trust and legitimacy of these devices within the network, effectively masking the malicious intent and making the attack detection more difficult.
- 3) Launch RSA/TSA. To launch a RSA, the attacker identifies multiple random NSs (i.e., one per each compromised UE) and switches the compromised UEs to these NSs by triggering ISS for each of them. In contrast, to launch a TSA, the attacker switches the compromised UEs to a target, pre-selected NS.

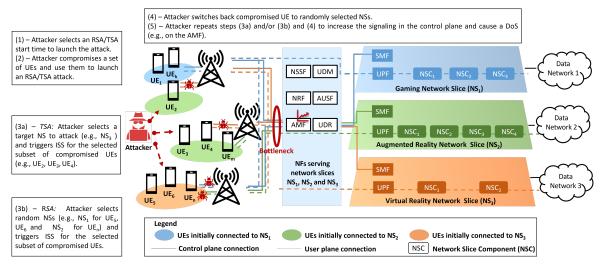


Fig. 1: Overview of DSM attack variations, illustrating how compromised UEs initiate ISS to perform RSA and TSA, resulting in excessive signaling load on 5G Core NFs. The colored line indicates the targeted NSs.

- 4) **Switch back compromised UEs to random NSs.** The attacker switches back some or all the compromised UEs to random NSs in order to introduce random ISS patterns for UEs and make the attack look stealthy.
- 5) Repeat steps (3) and (4). For increased and varying impact on the network and its NSs, the attacker can repeat steps (3) and (4) using the same or different subset of compromised UEs at different times, making the attack harder to detect while always causing disruptions to the network.

# IV. INTER-SLICE DEFENDER

As this study extends the research conducted in our previous work [12], this section presents its proposed solution, named Inter-Slice Defender (Fig. 2a), for detecting DSM attacks, with emphasis on its key findings and limitations. Fig. 2a illustrates this approach, which comprises three modules: data collection and pre-processing, feature extraction, and anomaly detection.

The data collection and pre-processing module gathers 5G network data and prepares it for the feature extraction module, which extracts, selects, and normalizes 3GPP KPIs [33] and PM counters [34] that are shown in Table II. The anomaly detection module is composed of two engines: the training and optimization engine, and the anomaly detection engine. The training and optimization engine is responsible for training the anomaly detection model based on an LSTM-Autoencoder [35], optimizing its architecture and hyperparameters, and selecting a threshold that ensures good detection performance. The choice of LSTM-Autoencoder is based on LSTM's suitability for processing and learning from sequential data that capture long-term dependencies, while the Autoencoder is chosen for its ability to compress input data into a lowerdimensional latent space and then reconstruct it. Inter-Slice Defender leverages the Autoencoder's ability as an unsupervised model to reconstruct input data [35]. Trained primarily on benign data, the model poorly reconstructs anomalous inputs, resulting in a high reconstruction error. This error, compared against a selected threshold  $\alpha$ , is used by the anomaly detection engine to perform real-time anomaly detection.

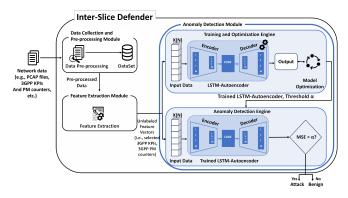
Inter-Slice Defender (Fig. 2a) achieved notable results, reaching an average F1-score of 98.75%, demonstrating its robustness in detecting DSM attacks and their variations [12]. However, this performance was based on training exclusively with benign data under the assumption of a clean training environment, which is unrealistic in real-world deployments where operational data may include noise or stealthy attacks. To better reflect practical scenarios, we retrained Inter-Slice Defender using contaminated datasets composed of typical operational data mixed with defined percentages of attack samples. This resulted in a significant performance drop, with the F1-score declining to 84.6% when only 5% of the training data was contaminated [12]. The presence of such anomalous data introduces conflicting patterns that compromise the model's ability to learn a reliable representation of normal behavior. This weakens the discriminative power of the reconstruction error, as the model begins to internalize features of both benign and attack inputs. Consequently, the error gap narrows, leading to false negatives and underscoring the need for further adjustments to ensure robust anomaly detection in real-world network environments.

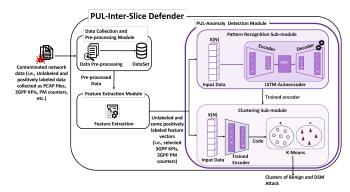
# V. PUL-INTER-SLICE DEFENDER

We present the PUL-Inter-Slice Defender (Fig. 2b), our novel NS anomaly detection solution that leverages LSTM-Autoencoder, K-Means, 3GPP KPIs [33], and PM counters [34] (Table II) to detect DSM attacks and their variations using a training dataset composed of positive and negative samples. Our approach relies on several key assumptions inherent to PUL, which are fundamental in handling this type of dataset. PUL-Inter-Slice Defender is composed of three modules: data collection and pre-processing module, feature extraction module, and PUL-anomaly detection module. We begin with a description of the PUL assumptions, followed by detailed explanations of each module.

# A. PUL Assumptions

The PUL is a binary classification problem where only a subset of labeled positive and unlabeled samples are available. This approach addresses the practical limitations of acquiring





(a) Inter-Slice Defender

(b) PUL-Inter-slice Defender

Fig. 2: Architecture of Inter-Slice Defender and PUL-Inter-Slice Defender. Inter-Slice Defender uses an LSTM-Autoencoder trained on benign data to detect attacks via reconstruction error and a threshold, while PUL-Inter-Slice Defender adopts PUL by leveraging the latent space for feature extraction to classify benign and DSM attack samples in the presence of contaminated training data through K-means.

fully labeled datasets due to factors such as time, expertise, and financial investment [16], [18]. The aim of PUL is to construct a classifier that can effectively distinguish between positive and negative samples, even when the training data consists of only a subset of labeled positive examples and a large set of unlabeled examples, thereby alleviating the cost of fully labeling data. It is important to note that the labeling process in PUL requires satisfying specific assumptions, as outlined in [18]. In light of this, our approach incorporates several of these assumptions, which are enumerated subsequently:

- Separability: The data used for PUL is assumed to be separable, that is positive and negative classes are naturally separable, indicating the existence of a classifier that can perfectly distinguish between the two. This assumption facilitates the development of approaches that concentrate on delineating a clear boundary between these two classes.
- 2) Smoothness: This assumption states that samples close to each other in the feature space are likely to share the same label, thereby enabling the application of techniques such as clustering. This enhances learning by assuming that proximity in spatial or feature dimensions is indicative of label similarity.
- 3) Unlabeled data contains positive and negative samples: In PUL, only a subset of positive samples is labeled, while negative samples remain without explicit labels. Consequently, our unlabeled dataset comprises both positive samples that were not chosen for labeling and negative samples. This constitutes a core principle of PUL, differentiating it from other learning methods such as supervised learning, where all instances are explicitly labeled.
- 4) **Selected Completely At Random (SCAR):** This assumption implies that the labeled samples are a uniform subset of the positive samples, selected without bias towards any specific attributes, which simplifies the application of PUL by enabling it to be treated like a binary classification problem.

## B. Data Collection and Pre-processing Module

The data collection and pre-processing module (Fig. 2b) collects unlabeled data alongside a subset of positive labeled data (e.g., PCAP files, 3GPP KPI, PM counters, etc.) from the 5G network and pre-processes it to be ready for use by the feature extraction module. The collected data can include PCAP files depicting signaling messages between the 5G network components (i.e., Radio Access Network (RAN) and core NFs) and which pertain to the different 5G procedures such as ISS, registration, etc. It can also account for 3GPP KPIs and PM counters (Table II) that are usually calculated at the different 5G NFs, and can be available at the Network Data and Analytics function (NWDAF) [36]. NWDAF is a 5G NF that facilitates data collection and analysis. It can collect KPIs from the 5G NFs and calculate others such as those KPIs related to NSs [36].

The data collected by the data collection and pre-processing module then undergoes a transformation that simplifies the extraction of features. For instance, in this work, we use TShark [37] to collect PCAP files from the network. We merge these files into a single one while maintaining their chronological order. This will serve the accurate calculation of time series based features by the feature extraction module. Then, we pre-process the unified PCAP file by extracting the most relevant information (e.g., IP source, IP destination, port destination, HTTP/2 header, etc.) for feature extraction.

#### C. Feature Extraction Module

The feature engineering module (Fig. 2b) acquires the preprocessed data from the data collection and pre-processing module and uses it for feature extraction. This data is used for feature extraction, selection, and normalization to extract 3GPP KPI [33] and PM counters [34] such as those listed in Table II. These features capture the 5G network and its NSs normal behavior, along with the abnormalities that can be caused by a DSM attack. The calculated features are time-series based, suitable for LSTM-Autoencoder required input format. To select the most pertinent features, a variance threshold process is employed which eliminates features characterized by minimal variations or perceived as noise.

TABLE II: Features of PUL-Inter-Slice Defender model.

Туре	3GPP KPI features	Definition		
3GPP-NS	*Registration success rate of one single NS	Success ratio of registration procedures (i.e., ratio of number of successful registrations over total number of attempted registrations) within a single NS for a specific AMF set.		
	*PDU session establishment success rate of one NS	Rate of successful PDU session establishment request over total number of attempted requests across all SMFs associated with a specific NS.		
	*Mean number of PDU sessions of network and NS	Average number of successful PDU session within a specific NS.		
	*Maximum number of PDU sessions of NS	Maximum number of successfully established PDU sessions within a single NS.		
	3GPP PM Counter features	Definition		
	Number of initial registration requests	Total number of initial registration requests that AMF receives.		
3GPP-AMF	Number of successful initial registrations	Count of successful initial registrations processed by the AMF.		
3GPP-AMF	Total number of attempted service requests	Number of attempted service requests including those initiated by the network		
		and those initiated by UEs.		
	Total number of successful service requests	Cumulative count of successful service requests accounting for those initiated by both the network and by UEs.		
	+Number of PDU session creation requests	Number of PDU session creation requests received by the SMF.		
	+Number of successful PDU session creations	Number of PDU sessions successfully established by the SMF.		
3GPP-SMF	+Number of failed PDU session creations	Count of PDU sessions successfully created by the SMF.		
	*Max time of PDU session establishment	Maximum time for PDU session establishment in each granularity period divided into sub-counters for each NS.		
	Number of released PDU sessions (AMF initiated)	Number of PDU sessions released at SMF with initiation originating from the AMF.		
3GPP-NSSF	Number of NS selection requests	Total number of NS selection requests that the NSSF receives.		
	Number of successful NS selections	Total successful NS selections executed by the NSSF.		
	Number of failed NS selections	Number NS selection attempts that failed at the NSSF.		
	Non-3GPP PM Counter feature	Definition		
AMF	Number of failed initial registrations	Number of unsuccessful registrations.		

\* Feature computed per each NS; + Feature computed per each SMF

Consequently, a total of 17 distinctive features (Table II) are selected to comprehensively scrutinize, analyze, and establish the normal and attack behavioral patterns exhibited by 5G networks. They can be classified per type that reflects their calculation. For instance, some features are calculated per NF (i.e., AMF, SMF, NSSF), mainly those NF involved in ISS related procedures (e.g., "Number of PDU session creation requests" feature is calculated at the SMF as it is involved in PDU session establishment procedure) while others are calculated per NS. With the exception of "Number of failed initial registrations" feature that is not standardized by 3GPP, the computation of the remaining features follows 3GPP specifications. These features can be collected from the NWDAF or the NFs if available or can be calculated from the PCAPs as in this work. Finally, it is worth noting that in the case where any of the aforementioned NFs is dedicated to an NS, its related features will then reflect the NS it serves. The total number of features processed by this module contains information that represents the 5G network in normal conditions and under DSM attacks.

#### D. PUL-Anomaly Detection Module

The PUL-anomaly detection module (Fig. 2b) leverages our approach's training process, incorporating PUL. This module utilizes an LSTM-autoencoder architecture combined with K-means clustering, guided by a PUL strategy that can be considered a variation of the two-step PUL technique. Standard two-step PUL techniques comprise two steps [18]. The first step involves the extraction of negative, and optionally positive samples from the unlabeled dataset. The second step involves training a model using the identified negatives alongside labeled positive samples [18]. However, in our approach, we first use the encoder of the LSTM-autoencoder to extract features (step 1) that characterize positive and negative samples, and then apply K-means clustering to separate these features into positive and negative clusters (step 2). These two steps

are realized in conjunction with the pattern recognition submodule and the clustering sub-module of the PUL-anomaly detection module, detailed hereafter.

Pattern Recognition Sub-module. Pattern recognition submodule is developed to extract the most critical patterns, correlations, temporal dependencies, and informative features from those provided by the feature extraction module. It employs an LSTM-Autoencoder [35] to this end, training and preparing its encoder for effective feature extraction in the subsequent submodule (i.e., the clustering sub-module). The selection of an LSTM-Autoencoder is based on its capability to effectively manage sequences and time-based patterns through LSTM. This capability is coherently integrated with an Autoencoder, which learns an abstract representation of its inputs (i.e., the normal behavior of the network and attack scenarios), accomplishing this through the compression and reconstruction of the input data [35]. The encoder effectively reduces the input dimensionality, thereby generating a latent space representation that captures the most significant characteristics and relationships in the training dataset, while the decoder is tasked with reconstructing the original input data from the compressed representation generated by the encoder. Thus, LSTM-Autoencoder yields a good depiction of the 5G network under both normal conditions and DSM attack scenarios, where the dependencies and precedence constraints of the various 5G procedures that represent ISS events are thoroughly detailed. For instance, an ISS event to a new NS cannot occur before deregistering from the current NS and registering with the new one (Section VI-B). Furthermore, the frequency of normal network behavior and attack patterns over time can be captured by an LSTM through its cell capabilities, which can remember values over various time intervals [35].

Clustering sub-module. The clustering sub-module is designed to identify and classify unlabeled data as positive or negative samples, alleviating the problem of learning the decision boundary without a fully labeled dataset and achieving

the goal of the PUL approach. Additionally, this module is employed to detect DSM attacks.

Upon completing the training phase of the LSTMautoencoder model in the pattern recognition sub-module, its encoder component is employed by the clustering submodule to extract features from the data processed by the feature extraction module, which characterized the unlabeled data and the available subset of positively labeled data. Inside the encoder, these features are compressed and encoded into a compact and meaningful representation that preserves the essential characteristics of the sequential data that represent normal and attack behavior. This refined representation is leveraged by the K-means clustering algorithm. K-means clustering is an unsupervised machine learning algorithm, simple and easy to implement, and widely used in anomaly detection applications [38]. It is designed to partition a dataset into K distinct, non-overlapping clusters, assigning each data point to one of the K clusters based on predefined similarity measures. By iteratively updating cluster centroids and reallocating data points, K-means effectively identifies the underlying structure within the data.

In our approach, K-means clustering algorithm leverages the extracted features from the encoder to enhance its performance by improving clustering accuracy, thereby potentially augmenting the separability of the samples, which aligns with the smoothness and separability assumptions described in Section V-A. This process facilitates the grouping of the data into two distinct clusters: one comprising positive samples (i.e., benign instances), and the other encompassing negative samples (i.e., attack instances). Furthermore, given the inclusion of an initial subset of positively labeled data within the training dataset (Section VI-C), it becomes feasible to assign labels to each cluster. By mapping these labeled samples to the resulting clusters, this subset facilitates the determination of which cluster has to be designated as positive and which as negative.

Once the K-means training is completed, new data entries are processed through the predefined sequence involving LSTM-Autoencoder's encoder for feature extraction followed by clustering. The classification of a new data point, whether it is anomalous or not, is then assessed based on its association with a specific cluster. This assessment relies on the premise that the clusters distinctly represent either normal or anomalous instances.

# VI. 5G Environmental Setup And Data Generation

Given the lack of a 5G dataset suitable for training and testing our PUL-Inter-Slice Defender solution, we present, in this section, our environmental setup and data generation strategy. We highlight our 5G testbed that we deploy to emulate normal network traffic in addition to RSA and TSA.

#### A. 5G Testbed

We employ the open source free5GC-compose and free5GC all-in-one implementation [19], adhering to the 3GPP standard [39] to build our testbed, depicted in Fig. 3. The testbed runs free5GC-compose version 3.3.0 as the CP on a Virtual Machine (VM) operating Ubuntu 20.04 – Focal. The VM features 8 virtual CPUs, 8 GB of RAM, and a 60 GB of hard drive, with each NF encapsulated within this virtualized environment. To

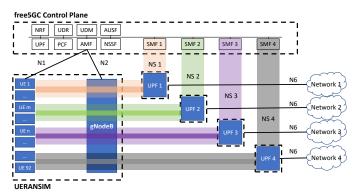


Fig. 3: 5G testbed comprising four NSs, designed in accordance with the 3GPP standard.

enhance the realism of our emulation environment, we opt to segregate the RAN from the CP. This segregation involves installing UERAMSIN 3.2.6 [20], a UE and RAN simulator, on a distinct VM. Our designed testbed comprises four NSs, each featuring a dedicated UPF installed on a separate VM, and a dedicated SMF. For the installation and configuration of all UPFs, we use the free5GC all-in-one version 3.3.0 [19] while the SMFs are deployed on containers in the VM hosting the CP. Furthermore, our testbed hosts 92 UEs configured to be able to connect to the existing four different NSs. The creation and management of the VMs within our solution are orchestrated through OpenStack [40].

TABLE III: Logical dependency between 5G procedures.

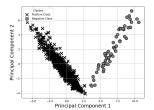
Triggered procedure	Possible subsequent procedures		
ISS	ISS, Uplink, Downlink, UE release PDU session, gNodeB release PDU session		
Registration	ISS, Uplink, Downlink, UE release PDU session, gNodeB release PDU session		
Uplink	ISS, Downlink, UE release PDU session, gNodeB release PDU session		
Downlink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

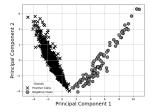
# B. Emulation of Normal and Attack Network Traffic

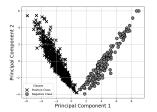
To collect benign and attack data, we perform three different emulations. The initial emulation replicates a 5G network operating under normal conditions without any external attacks. In contrast, the subsequent two emulations were conducted to assess the impact of RSA and TSA.

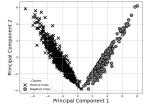
Normal network traffic emulation. Throughout the 120minute duration dedicated to this emulation, 92 UEs are used to emulate varying loads of normal network traffic. Each UE randomly triggers many 5G procedures from those detailed in Table III. Notably, the 5G procedures listed under the "Possible subsequent procedures" column are executable solely and exclusively if the 5G procedure indicated under the "Triggered procedure" column has been successfully completed. This logical interdependence underscores the sequential nature of these procedures. For instance, as outlined in Table III, following the successful completion of a registration procedure, several events can be triggered: ISS, Uplink, Downlink, UE release PDU session, and gNodeB release PDU session. Finally, the benign activity of the UEs is recorded in PCAP files. The latter constitutes the benign dataset that will be used for our LSTM-Autoencoder model training and testing.

**RSA** and **TSA** emulations. To conduct the RSA and TSA emulations, we adhere to the same approach as in the benign









- (a) 10% of negative samples
- (b) 20% of negative samples
- (c) 30% of negative samples
- (d) 40% of negative samples

Fig. 4: 2D PCA plot showing positive and negative classes in the training dataset with different levels of negative samples

emulation. We perform the emulation of each attack using a total of 92 connected UEs out of which 28 are compromised and used for the attacks. Following our threat model (Section III), the attacker strategically decides on the time to launch the RSA or TSA such that it coincides with the network's peak activity. As a result, during the 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 at its peak. When the attacks start, the 28 compromised UEs are connected to any of the four NSs and are used to perform ISS events in the quest of overloading the CP.

TABLE IV: Datasets statistics.

Dataset type	Contamination	No. of records	Benign records	Attack records	
				RSA	TSA
Training Dataset	10%	40000	36000	2000	2000
Training Dataset	20%	40000	32000	4000	4000
Training Dataset	30%	40000	28000	6000	6000
Training Dataset	40%	40000	24000	8000	8000
RSA Test dataset	_	20000	10000	10000	0
TSA Test dataset	_	20000	10000	0	10000

# C. Datasets for Anomaly Detection Model

The data generated from the aforementioned emulations is used to create different datasets to facilitate the training and evaluation of PUL-Inter-slice Defender model.

1) Training Dataset: To develop an effective anomaly detection solution for contaminated data, we employ four training datasets (Table IV) encompassing different proportions of both benign samples, representing the positive class, and DSM attack samples (i.e., RSA and TSA samples), representing the negative class. Specifically, the proportions of attack samples are set at 10%, 20%, 30%, and 40% of the total training dataset, with the remaining samples being benign. Each dataset includes 10% labeled benign data, identified by a security expert who randomly selects a diverse and representative subset of positive samples, ensuring unbiased and high-quality data. This enhances the effectiveness of our PUL-anomaly detection module's binary classifier, as established by the SCAR assumption (Section V-A). It also aligns with the PUL assumption, which asserts that a dataset contains a subset of labeled positives and unlabeled samples, where the latter comprises both positive and negative samples (Section V-A).

Furthermore, to analyze the distribution and assess the class separability and smoothness assumptions within our training dataset, we employed Principal Component Analysis (PCA) [41]. PCA reduces the dataset's dimensionality to two dimensions, enabling the visualization of data in a scatter plot to verify the separability of classes in a reduced-dimensional space. Fig. 4 illustrates a clear separation between positive and

negative classes across all four training datasets, indicating that they are distinguishable within the transformed feature space. This suggests that training a model with positive and unlabeled data could effectively differentiate between positive and negative samples. Fig. 4 also demonstrates that our training datasets meet the smoothness assumption across the different proportions of positive and negative samples. Both classes, positive and negative, exhibit compact clustering within each class, indicating similarity and supporting this principle. Furthermore, the boundary between classes is fairly distinct with minimal overlap, reinforcing that data points of the same class are closer to each other than to those of the other class.

2) Test datasets: To rigorously evaluate our anomaly detection model, we use two test datasets, each containing benign and malicious data from RSA and TSA attack emulations (Section VI-B). This approach thoroughly assesses the model's ability to distinguish between normal and anomalous patterns.

It is crucial to highlight that the training and test datasets are meticulously designed to be mutually exclusive, ensuring that there are no redundant records between them.

#### VII. DSM ATTACK IMPACT ON 5G CONTROL PLANE

To assess the impact of the RSA and TSA on the 5G network, we observe the CPU utilization of the different 5G CP NFs during RSA and TSA emulations and compare it with their CPU utilization during normal network traffic.

# A. Impact on AMF

We first focus on the impact of the attacks on the AMF given that it is the CP NF that is involved the most in UE to 5G network communication and is the first CP point of contact. In fact, despite the role AMF plays in UE registration, authentication, and NS selection and allocation, it is usually shared among different NSs. Thus, when a significant number of UEs simultaneously perform ISS, such as in the case of RSA and TSA, we observe a significant increase in the AMF CPU utilization which leads to a DDoS (Fig. 5).

As explained in Section VI-B, we emulate varying loads of normal network traffic and trigger the RSA and TSA at the peak load depicted at time=60~minutes. Fig. 5a shows the AMF CPU utilization during normal network traffic emulation and depicts the highest utilization of 49.95% at time=60~minutes. In contrast, Fig. 5b and 5c show that the AMF CPU consumption reaches 135.98% and 139.55% during the RSA and TSA respectively which are launched at the peak network load (i.e., time=60~minutes). The figures

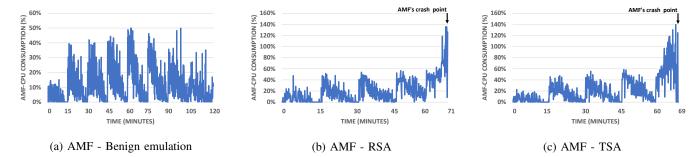


Fig. 5: AMF CPU consumption during benign and attack emulations.

show that RSA and TSA last for 11 *minutes* and 9 *minutes* respectively before the crash of the AMF, and hence the whole network. This shows the DDoS impact the DSM attack can have on the 5G network and further emphasizes the need to secure 5G networks against it. Finally, note that AMF CPU utilization exceeded 100% in our testbed due to containerized CP NFs sharing unused CPU resources. Without this sharing, the DDoS impact would have been observed earlier.

#### B. Overall Impact on 5G CP NFs

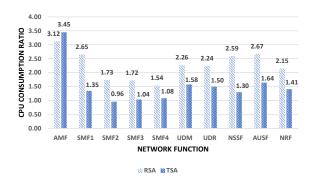


Fig. 6: CPU consumption ratio during RSA and TSA.

Despite the significant impact RSA and TSA have on the AMF, we observe a similar increase in the CPU consumption on other CP NFs as shown in Fig. 6. In fact, this figure shows the average CPU ratio for all the CP NFs during both RSA and TSA emulations. This ratio is calculated by dividing the average CPU consumption for each NF during the attack period (i.e., [60 minutes – 71 minutes] for RSA, [60 minutes – 69 minutes] for TSA) over that during the benign emulation for that same period.

Note that except for the AMF, the increase in the CPU utilization for all the NFs is greater during the RSA than during the TSA. Fig. 6 shows that RSA has a bigger impact on the 5G CP NFs. However, TSA has a higher impact on the AMF than the RSA, which results in degrading its performance in handling the requests, many of which ended up being dropped. Thus, fewer requests were forwarded to other CP NFs during TSA than during RSA which explains a smaller increase in their CPU utilization. Finally, unlike the AMF which CPU consumption exceeds 100% during the attacks, we observe that CPU consumption of other CP NFs remains under 50%.

#### VIII. EXPERIMENTAL RESULTS

In this section, we evaluate PUL-Inter-Slice Defender and assess its effectiveness in detecting RSA and TSA when its training dataset contains samples reflecting normal behavior in the 5G network and samples indicating DSM attacks.

#### A. LSTM-Autoencoder Architecture Selection

To determine the optimal architecture for the PUL-Inter-Slice Defender's LSTM-Autoencoder, which effectively learns patterns and dependencies in the sequential data representing normal and attack behaviors in our training dataset, we train and validate multiple architectures and evaluate their performance. To this end, we use a training dataset composed of 30% attack data and 70% benign samples (Table IV), as a dataset with a moderate proportion of negative samples allows our proposed approach to learn a more diverse representation of what constitutes a negative sample, which is crucial in PUL for avoiding false positives and preventing the model from overfitting to the positive class. Thus, we allocate 20% of the training dataset as a validation dataset and train the model using its remaining portion (Table IV). To select the model hyperparameters, we apply the K-fold cross-validation technique [42], which helps prevent overfitting and thoroughly evaluates model performance. Specifically, we train the model using a batch size of 32, a learning rate of 0.01, and the Adam optimizer, with a dropout rate of 0.2 to mitigate overfitting. We use ReLU as the hidden activation function and Mean Squared Error (MSE) as the loss function. The model is trained for up to 50 epochs. Additionally, we incorporate early stopping, which continuously monitors the cross-validation set's performance throughout the training process. If the validation error starts to increase, or if there is no improvement over a predetermined number of epochs, the training is stopped and the model with the best performance on the cross-validation set is selected as the final model.

We test and evaluate various LSTM-Autoencoder architectures and select {100, 50, 50, 50, 100} as it depicts the best performance in learning the characteristics of the training data and reconstructing them with low reconstruction errors. It is composed of two LSTM layers of 100 and 50 neurons, respectively, forming the encoder, and a decoder having the encoder's mirrored architecture. The code of the LSTM-Autoencoder is 50 neurons. We train our model with the selected architecture after fine-tuning its hyperparameters.

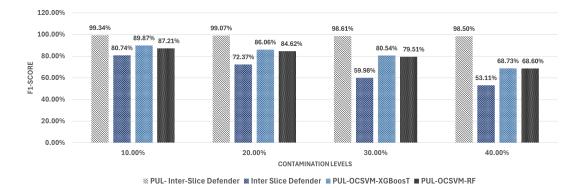


Fig. 7: F1-score comparison of PUL-Inter-Slice Defender, Inter-Slice Defender, PUL-OCSVM-RF, and PUL-OCSVM-XGBoost for detecting RSA and TSA under varying training data contamination levels (10%–40%).

#### B. K-means architecture

We apply the K-means clustering algorithm using the standard implementation provided by the scikit-learn library [43]. The number of clusters is set to two to align with the binary classification objective of our approach. To achieve consistent model behavior, the random seed is fixed to 42. The algorithm adopts the "k-means++" initialization method to enhance centroid selection and accelerate convergence. To improve clustering robustness, the model is configured to run ten times with different centroid seeds. The maximum number of iterations allowed for convergence is defined as 300, and the tolerance threshold for convergence is specified as 1e-4. These settings ensure a stable and effective clustering process, enabling the model to accurately separate benign and attack patterns within the latent feature space extracted by the LSTM-Autoencoder.

#### C. Benchmark Models

To conduct a thorough evaluation of our proposed solution, we compare the PUL-Inter-Slice Defender against three benchmark models. The first is the original Inter-Slice Defender [12], which serves as a natural baseline since our PUL-based model builds upon its architectural foundation.

In addition, we implemented two custom PUL-based models: PUL-OCSVM-RF and PUL-OCSVM-XGBoost, to serve as comparative baselines within the PUL paradigm. Both models follow the same data pre-processing and feature extraction pipeline used in our main solution (Section V-B, Section V-C). In accordance with the standard two-steps PUL approach [18], both models first apply OCSVM to the positively labeled benign samples to identify reliable negatives from the unlabeled set. OCSVM has strong capability in detecting outliers and anomalies in high-dimensional data [44], making it particularly effective for identifying reliable negatives within the PUL context. It employs an Radial Basis Function (RBF) kernel with gamma='scale'. The nu parameter, which controls the trade-off between training errors and model complexity, was set to 0.025 for *PUL-OCSVM*-RF and 0.03 for PUL-OCSVM-XGBoost. Samples predicted as -1 (i.e., attacks) were selected as reliable negatives. These reliable negatives were then combined with the labeled benign data to train a supervised binary classifier. For the PUL-OCSVM-RF model, we utilize a RF classifier, whose ensemble nature helps improve classification accuracy and robustness when working with labeled data [45], making it particularly suitable for the second step of the two-step PUL technique [18]. It was configured with 100 estimators and a fixed random seed of 42 to ensure reproducibility. For the PUL-OCSVM-XGBoost model, we train an XGBoost classifier, selected for its demonstrated success in supervised anomaly detection and its ability to efficiently handle structured data, making it well suited for classification tasks within the PUL setting [22]. The model is configured with the label encoder disabled, a fixed random seed of 42, an evaluation metric based on log loss, 100 estimators, a maximum tree depth of 5, and a learning rate of 0.1.

These three benchmark models (Inter-Slice Defender, PUL-OCSVM-RF, and PUL-OCSVM-XGBoost) were trained and evaluated using the same datasets as the PUL-Inter-Slice Defender (Section VI-C), including training sets with varying levels of contamination and a common set of test datasets. The corresponding evaluation results are presented in the following subsection.

# D. PUL-Inter-Slice Defender Performance

We now present the performance of the proposed PUL-Inter-Slice Defender and compare it against the benchmark models (Section VIII-C) across varying contamination levels.

We retrain Inter-Slice Defender using each of our four training datasets (Section VI-C) to evaluate its performance under different levels of contamination (i.e., 10%, 20%, 30%, 40%). The model's lookback hyperparameter (i.e., timesteps) is fixed at 1, and the detection threshold is set to 0.1408, based on findings from [12], which identified these values as optimal for detection performance. Fig. 7 shows Inter-Slice Defender's F1-score values ranging from 80.74% to 53.11% as contamination levels increase from 10% to 40% for detecting RSA and TSA. This performance drop is attributed to the model's assumption of a clean training dataset (i.e., entirely benign), an assumption that does not hold in real-world deployments, where contamination is frequently present and compromises the model's practical effectiveness.

In contrast, the performance of PUL-OCSVM-XGBoost and PUL-OCSVM-RF reveals a clear advantage compared to the Inter-Slice Defender across all contamination levels (10%, 20%, 30%, and 40%), as shown in Fig. 7, highlighting the

benefits of PUL. Specifically, PUL-OCSVM-XGBoost maintains an F1-score of 89.87% at 10% contamination, which gradually decreases to 68.73% as the contamination level reaches 40%. Similarly, PUL-OCSVM-RF records an F1-score ranging from 87.21% to 68.60% over the same contamination range. Despite these improvements, both PUL-OCSVM-XGBoost and PUL-OCSVM-RF models exhibit a noticeable decline in performance as contamination increases, reflecting their limited capacity to capture temporal dependencies in the data. This poses a critical limitation for DSM attack detection, where RSA and TSA unfold through sequences of 5G control procedures such as registration, authentication, and PDU session establishment. These procedures generate time series patterns essential for distinguishing anomalous from benign behavior. While RF treats each input independently and is not tailored for capturing temporal dependencies; XGBoost does not account for sequential dependencies like LSTM. This limitation can lead to misclassifications when subtle temporal patterns become decisive under high contamination conditions.

On the contrary, PUL-Inter-Slice Defender demonstrates superior performance in detecting both RSA and TSA attacks, as illustrated in Fig. 7. It achieves F1-scores ranging from 99.34% to 98.50% across training datasets with contamination levels varying between 10% and 40%, significantly outperforming the benchmark models: Inter-Slice Defender, PUL-OCSVM-XGBoost, and PUL-OCSVM-RF. This performance improvement is attributed to three key architectural strengths:

- The integration of LSTM-autoencoder to effectively handle the sequential nature of the data. In the context of DSM attacks such as RSA and TSA, the behavior of the system unfolds over time through a series of control plane procedures. These procedures naturally generate time series data, where the order and timing of events are critical for identifying normal and abnormal patterns.
- The model leverages the LSTM-autoencoder's latent space to extract abstract features, facilitating effective feature clustering and pattern separation.
- 3) PUL enhances adaptability by enabling robust decision boundary learning from limited labeled positives and a large pool of unlabeled samples, making it well-suited for real-world scenarios with contaminated training data.

Overall, by combining PUL with the LSTM-autoencoder's latent space, the PUL-Inter-Slice Defender effectively captures the sequential structure and timing of 5G CP procedures. This enables robust detection under contaminated training conditions and allows the model to generalize well despite label uncertainty and the complexity of time-dependent behaviors during a DSM attack.

# IX. Implementation Considerations in Real-World 5G Environment

PUL-Inter-Slice Defender leverages 3GPP KPIs and PM counters as features for DSM attack detection. These metrics are usually calculated by different 5G NFs and shared with the Network Data Analytics Function (NWDAF), which is a core 5G NF designed to collect, process, and analyze data across various network components [36]. NWDAF provides a

granular view of network behavior and can be configured to monitor targeted NSs and NFs such as AMF, SMF, and NSSF, which are critical components for maintaining the operational integrity and security of the 5G network. By leveraging NWDAF's data collection and analytics capabilities, PUL-Inter-Slice Defender enhances its effectiveness in detecting DSM attacks, offering a data-driven approach that strengthens the overall resilience of the 5G network. Additionally, PUL-Inter-Slice Defender can complement existing intrusion detection systems to trigger automated responses such as rate limiting or slice isolation, mitigating threats before they escalate.

However, deploying PUL-Inter-Slice Defender in real-world 5G networks presents several challenges, particularly due to the fragmented nature of NS deployments across multiple administrative domains. These domains often enforce distinct security policies and data-sharing restrictions, limiting the holistic analysis of network behavior. Privacy concerns further complicate collaboration, as operators may be unwilling to share sensitive operational data. These limitations hinder centralized anomaly detection and necessitate approaches that balance accuracy, efficiency, and data privacy.

To address these constraints, PUL-Inter-Slice Defender's integration with NWDAF should support distributed monitoring, enabling each domain to maintain autonomy while contributing to a collective detection framework. A promising direction for future work is the use of federated learning [46], where local models are trained within each domain and only aggregated insights are shared. This allows for collaborative, privacy-preserving anomaly detection that is scalable, efficient, and aligned with the operational constraints of diverse 5G environments.

#### X. CONCLUSION

In this work, we introduced, tested, and evaluated two variants of the known DSM attack, namely RSA and TSA, that exploit ISS procedures. We analyzed the impact that these attacks have on the 5G network using our free5GC testbed and UERANSIM simulator. We showed that they cause a DDoS on the network due to the resulting overload on the AMF. Furthermore, we developed the PUL-Inter-Slice Defender, an innovative anomaly detection solution designed to identify inter-slice attacks such as RSA and TSA in the presence of contaminated data. PUL-Inter-Slice Defender employs the PUL approach, integrating LSTM-Autoencoder with K-Means. The proposed solution leverages 3GPP KPIs and PM counters for its training process. The integration of these 3GPP features enables easy deployment of our PUL-Inter-Slice Defender as part of the NWDAF, as these features are usually made/can be available in this 5G NF. PUL-Inter-Slice Defender was rigorously evaluated under various conditions, including different architectures, contamination levels, and test datasets. It was also compared against three benchmark models (Inter-Slice Defender, PUL-OCSVM-RF, and PUL-OCSVM-XGBoost), outperforming them and achieving an average F1score exceeding 98.50% across training datasets with varying proportions of negative samples, highlighting its robustness and generalization capability in detecting RSA and TSA.

Finally, as a future work, we aim at studying the performance and economic impact that the DSM attack and its variations can have on NSs when the AMF and CP NFs resources are large enough to contain the attack. Additionally, we plan to investigate the impact of concept drift, particularly under dynamic and evolving 5G network conditions where the statistical distribution of RSA, TSA, or benign traffic may change over time.

#### REFERENCES

- [1] M. Chahbar et. al., "A comprehensive survey on the e2e 5g network slicing model," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 49–62, 2020.
- [2] K. Abbas, T. A. Khan, M. Afaq, and W.-C. Song, "Network slice life-cycle management for 5g mobile networks: An intent-based networking approach," *IEEE Access*, vol. 9, pp. 80128–80146, 2021.
- [3] S. Zhang, "An overview of network slicing for 5g," IEEE Wireless Communications, vol. 26, no. 3, pp. 111–117, 2019.
- [4] 3GPP, "System architecture for the 5G System(5GS): TS 23.501 V18.4.0," 2023.
- [5] M. M. Sajjad, C. J. Bernardos, D. Jayalath, and Y.-C. Tian, "Inter-slice mobility management in 5g: motivations, standard principles, challenges, and research directions," *IEEE Communications Standards Magazine*, vol. 6, no. 1, pp. 93–100, 2022.
- [6] R. F. Olimid and G. Nencioni, "5g network slicing: A security overview," IEEE Access, vol. 8, pp. 9999–100009, 2020.
- [7] C. De Alwis, P. Porambage, K. Dev, T. R. Gadekallu, and M. Liyanage, "A survey on network slicing security: Attacks, challenges, solutions and research directions," *IEEE Communications Surveys & Tutorials*, 2023.
- [8] 3GPP, "Procedures for the 5G System (5GS): TS 23.502 V18.4.0," 2023.
- [9] V. N. Sathi and C. S. R. Murthy, "Distributed slice mobility attack: A novel targeted attack against network slices of 5g networks," *IEEE Networking Letters*, vol. 3, no. 1, pp. 5–9, 2020.
- [10] V. N. Sathi and C. S. R. Murthy, "Dsm attack resistant slice selection in 5g," *IEEE Wireless Communications Letters*, vol. 10, no. 7, pp. 1469– 1473, 2021.
- [11] H. Bisht, M. Patra, and S. Kumar, "Detection and localization of ddos attack during inter-slice handover in 5g network slicing," in 2023 IEEE 20th Consumer Communications & Networking Conference (CCNC). IEEE, 2023, pp. 798–803.
- [12] R. M. A. Molina, N. Wehbe, H. A. Alameddine, M. Pourzandi, and C. Assi, "Inter-slice defender: An anomaly detection solution for distributed slice mobility attacks," in 2024 IFIP Networking Conference (IFIP Networking). IEEE, 2024, pp. 432–440.
- [13] M. S. Khan, B. Farzaneh, N. Shahriar, N. Saha, and R. Boutaba, "Slicesecure: Impact and detection of dos/ddos attacks on 5g network slices," in 2022 IEEE Future Networks World Forum (FNWF). IEEE, 2022, pp. 639–642.
- [14] N. A. E. Kuadey, G. T. Maale, T. Kwantwi, G. Sun, and G. Liu, "Deepsecure: Detection of distributed denial of service attacks on 5g network slicing—deep learning approach," *IEEE Wireless Communications Letters*, vol. 11, no. 3, pp. 488–492, 2021.
- [15] A. Thantharate, R. Paropkari, V. Walunj, C. Beard, and P. Kankariya, "Secure5g: A deep learning framework towards a secure network slicing in 5g and beyond," in 2020 10th annual computing and communication workshop and conference (CCWC). IEEE, 2020, pp. 0852–0857.
- [16] A. Papič, I. Kononenko, and Z. Bosnić, "Conditional generative positive and unlabeled learning," *Expert Systems with Applications*, vol. 224, p. 120046, 2023.
- [17] V. Sevetlidis, G. Pavlidis, S. G. Mouroutsos, and A. Gasteratos, "Dense-pu: Learning a density-based boundary for positive and unlabeled learning," *IEEE Access*, 2024.
- [18] J. Bekker and J. Davis, "Learning from positive and unlabeled data: A survey," *Machine Learning*, vol. 109, no. 4, pp. 719–760, 2020.
- [19] Free5GC, "Free5gc," https://free5gc.org/, 2023, [Online; accessed Jul-2023].
- [20] Aligungr, "Ueransim," https://github.com/aligungr/UERANSIM, 2023, [Online; accessed May-2023].
- [21] G. Long and Z. Zhang, "Punet: A semi-supervised anomaly detection model for network anomaly detection based on positive unlabeled data." *Computers, Materials & Continua*, vol. 81, no. 1, 2024.
- [22] R. Dilworth and C. Gudla, "Harnessing pu learning for enhanced cloud-based ddos detection: A comparative analysis," arXiv preprint arXiv:2410.18380, 2024.

- [23] Z. Fan, Y. Yao, Y. Du, and X. Du, "Self-paced and reweighting pu learning for imbalanced malicious traffic detection," in GLOBECOM 2023-2023 IEEE Global Communications Conference. IEEE, 2023, pp. 6018–6023.
- [24] S. Lv, Y. Liu, Z. Liu, W. Chao, C. Wu, and B. Wang, "Intrusion detection based on non-negative positive-unlabeled learning," in 2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS). IEEE, 2020, pp. 1015–1020.
- [25] 3GPP, "System Architecture for the 5G System (5GS): TS 23.501 V18.3.0," 2023.
- [26] M. A. Al-Shareeda, S. Manickam, and M. Ali, "Ddos attacks detection using machine learning and deep learning techniques: Analysis and comparison," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 2, pp. 930–939, 2023.
- [27] A. Bremler-Barr, M. Czeizler, H. Levy, and J. Tavori, "Exploiting miscoordination of microservices in tandem for effective ddos attacks," in *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE, 2024, pp. 231–240.
- [28] S. Narmadha and N. Balaji, "Improved network anomaly detection system using optimized autoencoder- lstm," Expert Systems with Applications, p. 126854, 2025.
- [29] Y. Wei, J. Jang-Jaccard, F. Sabrina, W. Xu, S. Camtepe, and A. Dunmore, "Reconstruction-based lstm-autoencoder for anomaly-based ddos attack detection over multivariate time-series data," arXiv preprint arXiv:2305.09475, 2023.
- [30] R. A. Shaikh and S. Shashikala, "An autoencoder and 1stm based intrusion detection approach against denial of service attacks," in 2019 1st international conference on advances in information technology (ICAIT). IEEE, 2019, pp. 406–410.
- [31] B. Tian, Q. Su, and J. Yu, "Leveraging contaminated datasets to learn clean-data distribution with purified generative adversarial networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 8, 2023, pp. 9989–9996.
- [32] N. Neshenko et. al., "Demystifying iot security: An exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.
- [33] 3GPP, "Management and orchestration; 5G end to end Key Performance Indicators (KPI): TS 28.554 V18.2.0," 2023.
- [34] 3GPP, "Management and orchestration; 5G performance measurements: TS 28.552 V18.3.0," 2023.
- [35] M. Said Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "Network anomaly detection using 1stm based autoencoder," in *Proceedings of the* 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, 2020, pp. 37–45.
- [36] G.-M. et. al, "Network automation and data analytics in 3gpp 5g systems," *IEEE Network*, vol. 38, no. 4, pp. 182–189, 2023.
- [37] TSHARK.DEV, "Tshark.dev," https://tshark.dev/, 2023, [Online; accessed Jul-2023].
- [38] M. Jain, G. Kaur, and V. Saxena, "A k-means clustering and svm based hybrid concept drift detection technique for network anomaly detection," *Expert Systems with Applications*, vol. 193, p. 116510, 2022.
- [39] 3GPP, "The 5g standard," https://www.3gpp.org/, 2023, [Online; accessed Aug-2023].
- [40] OpenStack, "An openinfra foundation project," https://www.openstack. org/, 2023, [Online; accessed Apr-2023].
- [41] M. Greenacre, P. J. Groenen, T. Hastie, A. I. d'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nature Reviews Methods Primers*, vol. 2, no. 1, p. 100, 2022.
- [42] T.-T. Wong and P.-Y. Yeh, "Reliable accuracy estimates from k-fold cross validation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 8, pp. 1586–1594, 2020.
- [43] scikit-learn developers, "Kmeans," https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html, 2024, [Online; accessed Aug-2024].
- [44] E. F. Agyemang, "Anomaly detection using unsupervised machine learning algorithms: A simulation study," *Scientific African*, vol. 26, p. e02386, 2024.
- [45] J. B. Awotunde, F. E. Ayo, R. Panigrahi, A. Garg, A. K. Bhoi, and P. Barsocchi, "A multi-level random forest model-based intrusion detection using fuzzy inference system for internet of things networks," *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, p. 31, 2023.
- [46] J. Wen, Z. Zhang, Y. Lan, Z. Cui, J. Cai, and W. Zhang, "A survey on federated learning: challenges and applications," *International Journal* of Machine Learning and Cybernetics, vol. 14, no. 2, pp. 513–535, 2023.