# SAGE-5GC: Security-Aware Guidelines for Evaluating Anomaly Detection in the 5G Core Network

Cristian Manca[1,*,†], Christian Scano[1,2,*,†], Giorgio Piras[1], Fabio Brau[1], Maura Pintor[1] and Battista Biggio[1]

[1]University of Cagliari, Cagliari, Italy
[2]Sapienza University, Rome, Italy

**Abstract**

Machine learning-based anomaly detection systems are increasingly being adopted in 5G Core networks to monitor complex, high-volume traffic. However, most existing approaches are evaluated under strong assumptions that rarely hold in operational environments, notably the availability of independent and identically distributed (IID) data and the absence of adaptive attackers. In this work, we study the problem of detecting 5G attacks *in the wild*, focusing on realistic deployment settings. We propose a set of Security-Aware Guidelines for Evaluating anomaly detectors in 5G Core Network (SAGE-5GC), driven by domain knowledge and consideration of potential adversarial threats. Using a realistic 5G Core dataset, we first train several anomaly detectors and assess their baseline performance against standard 5GC control-plane cyberattacks targeting PFCP-based network services. We then extend the evaluation to adversarial settings, where an attacker tries to manipulate the observable features of the network traffic to evade detection, under the constraint that the intended functionality of the malicious traffic is preserved. Starting from a selected set of controllable features, we analyze model sensitivity and adversarial robustness through randomized perturbations. Finally, we introduce a practical optimization strategy based on genetic algorithms that operates exclusively on attacker-controllable features and does not require prior knowledge of the underlying detection model. Our experimental results show that adversarially crafted attacks can substantially degrade detection performance, underscoring the need for robust, security-aware evaluation methodologies for anomaly detection in 5G networks deployed in the wild.

**Keywords**

Cybersecurity, Detection, 5G, 6G, Machine Learning, Monitoring, Network Security



## 1. Introduction

The 5G Core network is the central component of the 5G architecture, responsible for control and user plane functions such as authentication, mobility management, and session control. Its cloud-native, service-based design generates large volumes of dynamic and heterogeneous traffic that is difficult to monitor using traditional rule-based security mechanisms. These characteristics enable flexibility and scalability, but also introduce new attack surfaces and operational challenges for network monitoring and protection. To protect this central component, machine learning (ML) and, in particular, anomaly detection systems are increasingly employed to analyze large volumes of heterogeneous traffic and to identify malicious activities that may not be captured by traditional rule-based or signature-based mechanisms. Ensuring the robustness of these learning-based detectors is essential for secure network operation. Despite their promise, most existing machine-learning approaches for anomaly detection in 5G networks are evaluated under assumptions that are difficult to satisfy in real deployments. In particular, they often rely on the availability of independent and identically distributed (IID) data and consider static threat models in which attackers do not adapt to the presence of learning-based defenses. In operational environments, however, 5G traffic exhibits strong temporal correlations, evolving service

*Joint National Conference on Cybersecurity (ITASEC & SERICS 2026), February 09-13, 2026, Cagliari, IT*
*Corresponding author.
†These authors contributed equally.
cristian.manca@unica.it (C. Manca); christian.scano@unica.it (C. Scano); giorgio.piras@unica.it (G. Piras); fabio.brau@unica.it (F. Brau); maura.pintor@unica.it (M. Pintor); battista.biggio@unica.it (B. Biggio)
0000-0002-0877-7063 (C. Manca); 0000-0001-7116-9338 (C. Scano); 0000-0001-7116-9338 (G. Piras); 0000-0001-7116-9338 (F. Brau); 0000-0001-7116-9338 (M. Pintor); 0000-0001-7116-9338 (B. Biggio)

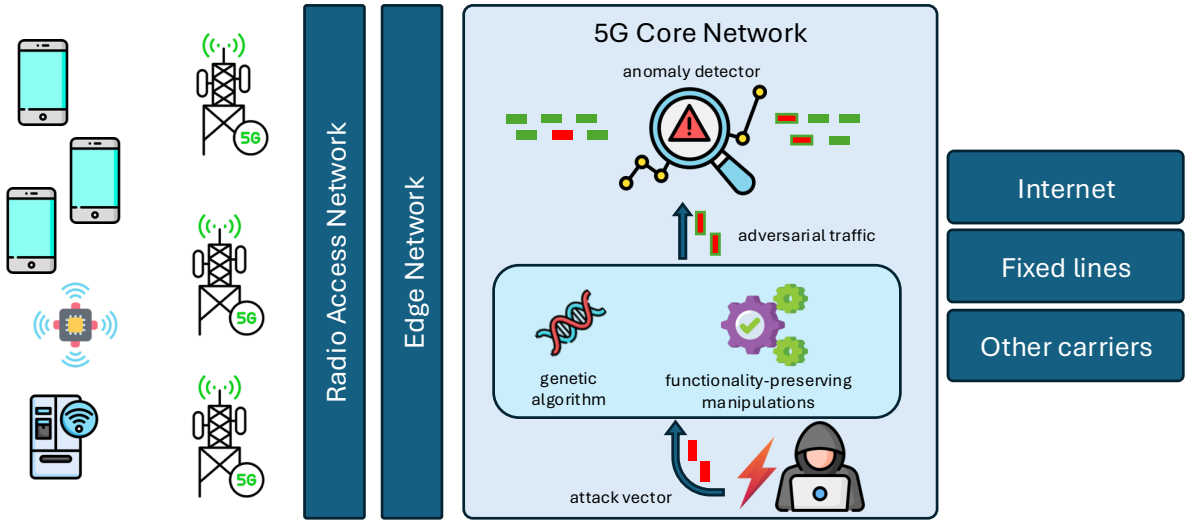


**Figure 1:** An overview of our proposed realistic SAGE-5GC evaluation, which confronts anomaly detection models with attackers that manipulate traffic to evade the system.

patterns, and configuration-driven changes that violate the IID assumption. At the same time, the integration of ML into critical network components introduces a new class of vulnerabilities. ML models are known to be susceptible to adversarial manipulation [1], where an attacker subtly alters inputs to mislead the model with the intent of evading detection.

This gap between evaluation practices and real-world conditions raises concerns about the reliability of current anomaly detection systems when deployed *in the wild*. High detection accuracy reported under controlled experimental settings does not necessarily translate into robust performance in the presence of traffic drift, non-stationarity, and adaptive attackers. In particular, adversarial attacks against machine-learning–based detectors pose a significant threat to system integrity, as they aim to induce misclassification through subtle, protocol-compliant feature manipulations.

In this work, we address these challenges by studying anomaly detection in realistic 5G Core network scenarios. We propose **SAGE-5GC**, a set of *Security-Aware Guidelines for Evaluating anomaly detection in the 5G Core network*, driven by domain knowledge of 5G protocols and operational constraints, and explicitly accounting for adversarial threats. Using a realistic 5G Core dataset, we first evaluate the baseline performance of several anomaly detection models against *traditional* cyberattacks targeting control-plane network services (PFCP). We then extend the analysis to adversarial settings, in which attackers preserve the semantics and effectiveness of malicious traffic while manipulating observable features to evade detection. We study the sensitivity of the models to feature perturbations and assess their robustness under both random and optimized adversarial strategies. To achieve the latter, we introduce a practical, model-agnostic optimization approach based on genetic algorithms that operates solely on attacker-controllable features and does not require prior knowledge of the underlying detection model. An overview of our method is presented in Fig. 1, which depicts the attacker-in-the-loop setting considered in a realistic 5G Core environment. Starting from a malicious attack vector targeting 5G Core control-plane traffic, the adversary applies functionality-preserving and protocol-compliant manipulations to a restricted set of attacker-controllable features through a genetic–algorithm–based strategy. The resulting adversarial samples are then fed to the anomaly detector to assess its robustness. The results of our experimental evaluation demonstrate that adversarially crafted attacks can significantly degrade detection performance, even for models that perform well under conventional evaluation settings. These findings highlight the need for adopting security-aware and adversarially informed evaluation methodologies for anomaly detection in 5G Core networks, particularly when systems are deployed in real-world operational environments. Our contributions are the following:

1. We provide a set of security-aware guidelines for training and testing anomaly detection systems

for the 5G Core network, explicitly considering adversarial threats;

2. We evaluate a diverse set of anomaly detection algorithms on a realistic 5G Core dataset representative of real deployment traffic; and

3. We extend the evaluation to adversarial scenarios, in which the attacker actively manipulates features to evade detection while keeping the malicious activity functional and effective.

In addition, the framework code is available at https://github.com/pralab/sage-5gc.

## 2. Anomaly Detection in 5G Core Networks

In this section, we first provide an overview of the 5GC network and its vulnerabilities and formalize the task of anomaly detection (Sect. 2.1). We then describe the anomaly detection algorithms relevant to our work (Sect. 2.2), and conclude with a set of guidelines to guide the training of these systems for deployment in a realistic and applicable scenario (Sect. 2.3).

### 2.1. Overview

**Overview of the 5G Core network.** The 5G Core (5GC) network is the central component of the 5G architecture, responsible for key control-plane and user-plane functions, including authentication, mobility management, session establishment, and service orchestration. The 5GC adopts a service-based, cloud-native architecture built on Network Function Virtualization (NFV) and edge-cloud components. While this design enables flexibility and scalability, it also generates high-volume, heterogeneous, and highly dynamic traffic, making continuous and automated monitoring essential for maintaining network security.

**Attacks against the 5GC network.** The 5GC is exposed to a wide range of traditional cybersecurity attacks targeting network services and protocols, including control-plane signaling protocols such as PFCP. These include denial-of-service (DoS) attacks, network scanning, and reverse shells (for lateral movement), as well as attacks that exploit vulnerabilities specific to the 5GC. Such attacks aim to disrupt service availability, degrade network performance, or compromise user data. We defer the description of these PFCP-based attacks to Sect. 4, where we provide details of the dataset used.

**Anomaly detection.** Anomaly detection provides a natural approach for monitoring the 5GC in scenarios where comprehensive labeling of malicious traffic is not available. The general pipeline for anomaly detection in the 5GC consists of: (i) raw traffic collection from core network functions (i.e., collecting packets at the transport level), (ii) feature extraction and preprocessing, (iii) training an anomaly detection model on benign traffic, and (iv) evaluating new incoming traffic to identify deviations from normal behavior.

In this work, we consider anomaly detection in the one-class learning setting, where the training data consists primarily of benign 5GC traffic, i.e., the detector learns a representation of normal behavior and identifies deviations as potential attacks. The anomaly detection is performed on features extracted from packets obtained by raw network traffic captures.

Packets are represented as byte strings of variable length over an alphabet $[256]$, where $[n] := 1, \dots, n$ for $n \in \mathbb{N}$. A feature extraction function $\mathcal{F} : [256]^* \to \mathbb{R}^d$ maps each packet $\mathbf{p}$ into a finite-dimensional feature space suitable for learning-based analysis. The resulting feature vector $\mathbf{x} = \mathcal{F}(\mathbf{p})$ is non-homogeneous and can be decomposed as $\mathbf{x}^\top = [\mathbf{x}_1^\top, \mathbf{x}_2^\top] \in \mathbb{N}^{d_1} \times \mathbb{R}^{d_2} \subseteq \mathbb{R}^d$, where $\mathbf{x}_1$ and $\mathbf{x}_2$ represent categorical and numerical features, respectively. An anomaly detector $\mathcal{D} : \mathbb{R}^d \to \mathbb{R}$ assigns an anomaly score to each feature vector, with higher values indicating greater deviation from learned normal behavior. An observation is classified as anomalous if its score exceeds a threshold $\tau$, i.e., if $\mathcal{D}(\mathcal{F}(\mathbf{p})) > 0$.

## 2.2. Anomaly Detection Algorithms

Algorithms for anomaly detection can be categorized into different families based on their underlying principles and detection mechanisms. We evaluate representative methods from each category to assess their effectiveness in the selected scenario.

**Statistical methods (HBOS, COPOD, ECOD, `Feature Bagging`).** These approaches assume that normal data can be effectively characterized by probabilistic or statistical models, and anomalies are identified as rare or unlikely observations under such models. Common strategies include constructing feature-wise histograms or fitting empirical distributions. Examples within this family are the Histogram-Based Outlier Score (HBOS) [2], which scores anomalies based on individual feature histograms, Copula-Based Outlier Detection (COPOD) [3], which captures feature dependencies through copula models, and Empirical-Cumulative-distribution-based Outlier Detection (ECOD) [4], which relies on empirical cumulative distribution functions to estimate tail probabilities without strict parametric assumptions. Finally, `Feature Bagging` [5], combines several base detectors trained on randomly selected feature subsets, capturing attack patterns that might be missed by any single view of the data.

**Density-based techniques (kNN, LOF, `IForest`, LODA, INNE).** In this category, the underlying intuition is that normal data points reside in dense regions of the feature space, while outliers or attacks are typically situated in sparser areas. Models adopt various ways to quantify local density or sample isolation. Typical approaches in this family are $k$-Nearest Neighbors (kNN) [6], which evaluates the distance to neighboring points, and Local Outlier Factor (LOF) [7], which compares the local density of a sample to those of its neighbors to identify context-dependent anomalies. More scalable or ensemble-based variants, such as Isolation Forest (`IForest`) [8], which isolates samples using random splits, Lightweight On-line Detector of Anomalies (LODA) [9], which projects data onto low-dimensional subspaces, and Isolation-based anomaly detection using Nearest-Neighbor Ensembles (INNE) [10], which combines isolation principles and nearest neighbor distances, further enhance robustness in high-dimensional scenarios.

**Geometric-based techniques (PCA, ABOD, GMM).** These methods focus on capturing the overall shape, spread, or geometric boundaries of the normal data distribution. They flag anomalies as points that fall outside of this learned structure. Among the commonly used strategies are Principal Component Analysis (PCA) [11], which identifies deviations from the principal axes of variation via reconstruction errors; Angle-Based Outlier Detection (ABOD) [12], which analyzes angular variance between points, particularly suited for high-dimensional data, and Gaussian Mixture Models (GMM) [13], which probabilistically cluster data and detect outliers as those with low likelihood under the learned mixture components.

**Ensemble techniques.** This family comprises methods that aggregate the outputs of multiple anomaly detectors, often trained on different feature subsets or with different initializations, to improve reliability and generalization.

## 2.3. Guidelines to Train Real-time Anomaly Detectors

To be effectively deployed, an anomaly detection system for the 5GC must satisfy the following guidelines for training (GT), which we summarize here and detail below:

- GT1 *Environmental independence*, the system should rely exclusively on protocol and behavioral features, explicitly removing environment-dependent information such as IP addresses, port numbers, and deployment-specific identifiers.
- GT2 *Protocol-specific design*, feature selection should be guided by domain knowledge of the targeted 5GC protocols and attack surfaces, eliminating irrelevant traffic and redundant fields.
- GT3 *Robust feature representation*, all retained features must be systematically transformed into stable, numeric representations suitable for machine learning, with careful handling of missing values and heterogeneous scales.

GT4 *Anomaly-agnostic training*, malicious packets should therefore be excluded from the training phase and used only for validation and testing.

**GT1. Environmental independence.** Data representation in the network traffic domain plays a critical role in the performance of anomaly detection methods in machine learning [5]. Therefore, it is essential to avoid any representation that relies on specific packet header values, such as the value of a source or destination IP address or port number, the value of a protocol (e.g., TCP, UDP), and so on, to avoid developing models that would define normal behavior as specific to a particular network connection point (e.g., IP address or port). This is important because training models based on specific values, such as IP addresses or port numbers, would simply define anomalies as any deviation in network destinations from those detected in the training data. Consequently, a model trained on traffic traces collected in one part of the network would be unable to perform general novelty detection when applied or deployed to other parts of the network. Furthermore, the model would fail to detect new behaviors originating from the same IP addresses over time, as would happen if a device were compromised or changed its behavior as a result of changes in the environment or in the way users interact with the device. For similar reasons, it is important to avoid relying on port numbers. While a significant amount of traffic, both normal and abnormal, traverses common ports (for example, port 443 or HTTPS), client traffic, in contrast, originates from ports that are constantly changing (for example, increasing). Defining normal traffic based on client ports would result in a significant number of false positives, simply due to the normal behavior of clients originating connections from different ports over time.

**GT2. Protocol-specific design.** Since the focus is on control-plane security in the 5GC, attention should be restricted to the Packet Forwarding Control Protocol (PFCP), which governs session and forwarding state between core network functions and is exclusively transported over UDP. Traffic unrelated to this control-plane interaction, such as TCP- and ICMP-based packets, should therefore be excluded at the packet level, and all features associated with these protocols should be removed. This restriction ensures that the analysis focuses on the protocol layers that are directly relevant to control-plane attack scenarios, namely IP, UDP, and PFCP, while avoiding the introduction of noise from unrelated transport or application-layer behaviors.

**GT3. Robust feature representation.** To ensure that the feature space remains both minimal and informative, constant, empty, and redundant protocol fields should be removed. This includes features that take a single unique value across all observations or encode no meaningful variability. Such fields do not contribute to anomaly detection and may introduce unnecessary noise or bias. Features that present missing values, instead, should be explicitly handled and imputed when necessary, since many anomaly detection models cannot operate on feature vectors containing null values, while non-numerical features should be transformed into a suitable numerical representation.

Finally, feature scaling should be considered as a recommended preprocessing step, particularly in the presence of heterogeneous feature ranges and extreme values, which are common in network traffic data. Robust scaling methods that rely on distribution quantiles can mitigate the influence of outliers without assuming normality. Although scaling is not strictly required for all anomaly detection models, its impact should be explicitly evaluated to ensure a transparent and reproducible assessment.

**GT4. Anomaly-agnostic training.** Malicious packets should be excluded from the training phase of anomaly detection models and used exclusively for validation and testing. Including malicious traffic during training undermines the novelty-detection objective and leads to overly optimistic performance estimates that do not reflect real deployment conditions, where attacks are unknown at training time.

## 3. Attacking Anomaly Detectors for the 5GC Network

Traditional cyberattacks on 5GC networks aim to compromise regular system functionality, regardless of the presence of ML-based defenses. In contrast, adversarial attacks explicitly target the integrity of ML models. However, directly applying gradient-based adversarial methods such as the Projected

Gradient Descent (PGD) [14] to packet-level feature representations is generally infeasible [15]. PGD perturbs input features in the direction of the loss gradient and enforces an $\ell_p$ constraint to maintain the perturbation small. While this approach works well with images, applying PGD to the packet features without accounting for protocol semantics is not suitable for our case. In fact, when applied to features derived from PFCP control-plane traffic, such perturbations may result in invalid protocol fields, inconsistent flag combinations, or malformed message structures that violate 5G specifications. As a consequence, the modified packets would no longer be accepted by network functions, thereby breaking the attack's functionality and thus invalidating the adversarial objective. Therefore, our goal is to test the detectors when they face malicious traffic that is subtly modified to appear benign to the detector while still performing the intended harmful activity. In this section, we first outline the formalization (Sect. 3.1) and detail the solving algorithms (Sect. 3.2) of our attack. Then, we draw the guidelines for evaluating anomaly detection for the 5GC network (Sect. 3.3).

### 3.1. Adversarial Attacks against Anomaly Detectors for 5GC Network

To formalize the attack, we first identify our assumptions and then provide definitions that enable the attack to occur at the packet level. We build on these observations to design our strategies for finding adversarial perturbations that are both feasible and compliant with the protocol, first relying on a random search, and then on a more efficient genetic algorithm.

**Threat Model.** We assume a model-agnostic attacker, capable of injecting or spoofing PFCP traffic in the 5GC network. From an evaluation perspective, assuming a model-agnostic attacker provides a conservative and robust assessment of anomaly detection systems. Model-specific adversaries may exploit vulnerabilities tied to a particular model, leading to conclusions that do not generalize across different detection models and complicate comparisons. In contrast, a model-agnostic attacker relies solely on observable behavior and feature-level knowledge, reflecting realistic deployment scenarios in which the internal details of the detector are not exposed.

We make the following assumptions. The attacker: (i) has no access to the internal parameters or architecture of the anomaly detector $\mathcal{D}$; (ii) can observe the detector's outcome, such as identifying which traffic is flagged as anomalous, and the detection score (therefore, has also access to the threshold $\tau$ that can be, in practice, inferred from a set of triggered detections and the observed scores); and (iii) has access to the feature extractor $\mathcal{F}$.

The third assumption enables the adversary to operate in the feature space while deriving the constraints required to map admissible perturbations back to the input (packet) space, ensuring that the resulting attacks are feasible in practice. Feasible modifications are restricted to traffic fields whose alteration does not violate protocol compliance or interfere with the core functionality of the malicious activity. This reflects realistic constraints in attacks on the 5GC network, where attack effectiveness must be preserved while remaining compliant with protocol specifications. A formalization of these constraints is provided in the remainder of this section.

**Definition 1 (Feasibility).** *Let $J \subseteq [d]$ be a set of feature indices. We say that a modification constrained to the features in $J$ is* feasible *if it produces a realizable feature vector. Formally, for each packet $\mathbf{p}$ with feature representation $\mathbf{x} = \mathcal{F}(\mathbf{p}) \in \mathbb{R}^d$, any modified feature vector $\mathbf{x}' \in \mathbb{R}^d$ such that $\mathbf{x}'_i = \mathbf{x}_i \quad$ for all $i \notin J$ is said to be feasible if the modification is* invertible*, i.e., if there exists a packet $\mathbf{p}'$ such that $\mathcal{F}(\mathbf{p}') = \mathbf{x}'$. With a slight abuse of notation, we say that $J$ is feasible if it allows for feasible modifications.*

**Definition 2 (Compliance).** *We say that two packets $\mathbf{p}, \mathbf{q}$ are compliant (namely $\mathbf{p} \equiv \mathbf{q}$) if $\mathbf{p}$ preserves protocol compliance and attack integrity (or functionality) of $\mathbf{q}$.*

Let $J \subseteq [d]$ be a set of feasible features, given a malicious packet $\mathbf{p}$, which we assume is successfully detected, i.e., $\mathcal{D}(\mathbf{p}) \geq \tau$, the aim of the attacker is to deduce a new packet $\mathbf{p}' \equiv \mathbf{p}$ which preserve the functionality of $\mathbf{p}$ but such that $\mathcal{D}(\mathbf{p}') < \tau$. Formally, this involves solving the following minimization problem (MP):

$$\min_{\mathbf{p}' \equiv \mathbf{p}} \left[\mathcal{D}(\mathcal{F}(\mathbf{p}')) - \tau\right]_+, \quad \text{s.t.} \quad \mathcal{F}(\mathbf{p})_i = \mathcal{F}(\mathbf{p}')_i, \, \forall i \notin J, \tag{MP}$$

where $[\cdot]_+$ indicates the positive part of a function, $\tau$ is the detection threshold.

### 3.2. Attack Optimization Algorithms

We emphasize that we specifically avoid minimization methods that leverage the gradient of $\mathcal{D}$ to maintain the model-agnostic nature of the attack. Hence, we tackle Problem MP through two algorithms, namely *random search* and *genetic algorithm*. Finally, we note that throughout the paper, we assume the set of feasible indices $J$ is manually selected by the attacker, based on their previous knowledge of the network system, and is therefore not optimized.

**Random search (RS).** In the random attack strategy, each malicious sample is perturbed by randomly modifying only the set of feasible features that produce a compliant packet, thus, those realizable modifications permitted by protocol constraints and that preserve the integrity of the attack. During a preparatory stage, a feasible set $J$ is selected, and, for each feature $j \in J$, the distributions $\mu_j$ are deduced on previously spoofed data. The minimum problem is then estimated by considering $\mathbf{p}' = \mathcal{F}^{-1}(\mathbf{x}')$ where $\mathbf{x}'_j \sim \mu_j$ for each $j \in J$ and $\mathbf{x}'_j = \mathbf{x}_j = \mathcal{F}(\mathbf{p})_j$ otherwise, and sampling $\mathbf{p}'$ until the objective of MP reaches 0.

**Genetic algorithms (GA).** We also address Problem MP by means of a genetic algorithm (GA), which is well suited for model-agnostic optimization under feasibility constraints, such as feasibility and compliance. The algorithm iterates by generating individuals in the population, which represent a candidate packet $\mathbf{p}'$ through its feature vector $\mathbf{x}' = \mathcal{F}(\mathbf{p}') \in \mathbb{R}^d$, where genes corresponding to indices in $J$ are free to vary, while all remaining genes are fixed, i.e., $x'_i = x_i$ for all $i \notin J$. By construction, individuals are restricted to feasible modifications, and the decoding step $\mathbf{p}' = \mathcal{F}^{-1}(\mathbf{x}')$ ensures that, if inverted, these manipulations can correspond to real packets. The algorithm iterates the following steps:

(i) The initial population is generated by sampling features in $J$ according to the empirical distributions $\mu_j$ introduced above, while keeping all other features unchanged.

(ii) Given a population $\{\mathbf{p}'_k\}_{k=1}^N$, the fitness of each individual is defined as

$$f(\mathbf{p}'_k) = \left[\mathcal{D}(\mathcal{F}(\mathbf{p}'_k)) - \tau\right]_+,$$

which directly mirrors the objective in (MP). Lower fitness values correspond to more successful evasion attempts.

(iii) At each generation, individuals are selected according to their fitness and combined through crossover operators acting only on the indices in $J$, so as to preserve feasibility and compliance. Mutation is then applied by randomly resampling a subset of features in $J$ from the corresponding distributions $\mu_j$, while leaving all indices $i \notin J$ untouched.

(iv) The algorithm iterates until either an individual with zero fitness is found, corresponding to a successful evasion, or a maximum number of generations is reached. In this way, the GA explores the space of feasible and compliant modifications more efficiently than pure random sampling, while remaining fully compatible with the model-agnostic threat model assumed throughout this work.

### 3.3. Guidelines for Evaluating Anomaly Detectors

Evaluating anomaly detection systems in the 5G Core network requires consideration of several implementation constraints, including novelty-based operation, class imbalance, and adaptive attacks.

To this end, we define a set of guidelines for evaluating anomaly detectors (GEs), which complement the training guidelines provided in Sect. 2.3 and outline a principled framework for evaluating both the effectiveness and robustness of detection in realistic 5G scenarios. Again, we summarize these guidelines here and discuss them in detail below:

GE1 *Realistic data and novelty-driven setup,* evaluation must reflect realistic 5GC network conditions, enforcing novelty-detection assumptions and preserving the natural imbalance between benign and malicious traffic.

GE2 *Alignment with imbalanced and security-critical settings,* evaluation should rely on performance indicators that capture detector behavior under class imbalance and heterogeneous attack patterns, combining global measures of detection capability with class-level analyses to reveal security-relevant weaknesses.

GE3 *Adversarial robustness,* since anomaly detectors in the 5GC operate in adversarial environments, evaluation must explicitly assess robustness against adaptive attackers. This includes testing the detectors against adversarial manipulations to ensure their robustness.

**GE1. Realistic data and novelty-driven setup.** Evaluation protocols should reflect the conditions under which anomaly detectors are expected to operate once deployed in a real 5GC network. In practice, this implies adopting a strict novelty-detection setting, where models are trained primarily or exclusively on benign traffic and are required to identify deviations corresponding to previously unseen attacks. Benign traffic used for training and evaluation should originate from distinct flows, clients, or operational instances, rather than from overlapping or trivially related samples. For instance, testing on packets from the same PFCP sessions observed during training leads to near-perfect performance by construction, rather than genuine anomaly detection. Moreover, evaluation datasets should preserve the strong class imbalance that characterizes real 5GC environments, where malicious events are rare compared to legitimate traffic. By enforcing these characteristics, evaluation results more accurately reflect the detector's ability to generalize and operate effectively in real deployment scenarios.

**GE2. Alignment with imbalanced and security-critical settings.** In security-critical environments such as the 5GC, evaluation must capture detector behavior beyond aggregate performance summaries. Due to the heterogeneity of attack types and their uneven prevalence, a detector may appear effective overall while systematically failing to detect specific attacks or generating excessive false alarms on benign traffic. A good practice in this scenario is to achieve the best trade-off between detection and false alarms. For this reason, evaluation should adopt performance indicators that jointly characterize global detection capability and class-level behavior. This allows practitioners to identify security-relevant weaknesses, such as the poor detection of specific attack categories or unstable behavior across traffic conditions, which would otherwise remain hidden when relying solely on aggregate measures.

**GE3. Adversarial robustness.** Anomaly detectors deployed in the 5GC must be evaluated not only against traditional cybersecurity attacks, but also under adversarial conditions in which attackers actively manipulate traffic to evade the learning-based detectors. Since these systems operate in a security-sensitive context, robustness to adversarial behavior is a fundamental evaluation criterion. Evaluation should therefore include the assessment of detector robustness against adversarial manipulations that respect protocol compliance and preserve the functionality of the attack. Perturbations must be limited to attacker-controllable features and reflect realistic constraints imposed by 5G Core protocols and attack semantics. This form of adversarial evaluation helps distinguish detectors that are robust by design from those whose performance relies on fragile correlations in the feature space.

## 4. Experiments

Following the structure of the previous section, we report in Sect. 4.1 the experimental setup, we discuss in Sect. 4.2 the evaluation of the detector's performance on clean data, while in Sect. 4.3 we discuss the evaluation of the detectors under the adversarial attacks formulated in Sect. 3.1.

| Characteristics | Train Dataset | Validation Dataset | Test Dataset |
|---|---|---|---|
| **Packets** | | | |
| Normal | 21341 | 4731 | 4732 |
| PFCP Restoration-TEID | 0 | 13 | 22 |
| PFCP Flood | 0 | 1039 | 1026 |
| PFCP Deletion | 0 | 7 | 13 |
| PFCP Modification | 0 | 16 | 12 |
| UPF PDN-0 Fault | 0 | 10 | 12 |

**Table 1**
Sample Distribution of Train, Validation, and Test datasets.

## 4.1. Experimental Settings

In this section, we present the experimental settings adopted during the experiment phase and discuss the results obtained.

**Dataset.** In our experiments, we use 5G-Attacks [16] a recently published dataset containing several attack categories targeting the 5G Core network.[1] This dataset provides labeled network traffic, including realistic attack scenarios, making it suitable for machine learning-based intrusion detection solutions. The resulting datasets include 5G-oriented attack scenarios that exploit specific vulnerabilities in next-generation mobile networks. An overview of the dataset is reported below (see Sect. A.1 for further details). The dataset includes five representative PFCP-based attacks targeting the 5GC. All attacks exploit the absence of authentication, integrity protection, and strict input validation in PFCP signaling, and are generated by manipulating protocol fields while preserving syntactic validity.

*PFCP Restoration-TEID.* This attack disrupts PFCP session recovery by injecting forged restoration messages with out-of-range TEID values and corrupted F-TEID parameters. The resulting desynchronization between control-plane and user-plane tunnel state leads to traffic misrouting, session drops, or UPF crashes.

*PFCP Flood.* A control-plane denial-of-service attack that overwhelms PFCP endpoints with unsolicited Session Establishment and Heartbeat messages. Randomized identifiers and sequence numbers lead to excessive state processing, exhausting control-plane resources and degrading session management performance.

*PFCP Deletion.* This attack forges Session Deletion Requests targeting active SEIDs, resulting in the premature removal of PDRs, FARs, and tunnel state. The immediate teardown of sessions results in the abrupt interruption of user-plane traffic.

*PFCP Modification.* Counterfeit Session Modification messages are injected to corrupt forwarding behavior at the UPF. By disabling forwarding actions and altering encapsulation and interface parameters, the attack induces packet drops, misrouting, or invalid tunneling.

*UPF PDN-0 Fault.* This attack exploits the insufficient validation of PDN Type 0 session contexts by inducing inconsistent PFCP state through malformed identifiers and invalid F-TEID flag combinations. The resulting inconsistencies cause session establishment failures, unstable state transitions, or traffic misrouting.

The raw datasets extracted via `tshark` contain multi-protocol packet-level information spanning IP, UDP, TCP, and the 5GC-specific PFCP protocol. Features include packet-level attributes (e.g., header lengths, flags, timestamps), protocol-specific fields (e.g., PFCP TEID values), and source/destination data such as IP addresses and transport-layer port numbers. The original dataset consists of three specific splits: one for unsupervised training, one for supervised training, and the last for testing. The detailed sample distribution for each attack class and dataset partition is reported in Table 1.

[1] https://github.com/clem272001/5G-Attacks

| Protocol | Before Preprocessing | After Preprocessing |
|---|---|---|
| | **Number of Features** | |
| IP | 16 | 4 |
| UDP | 9 | 1 |
| PFCP | 45 | 28 |

**Table 2**
Number of Features before and after applying Preprocessing

*Following GT4, we have to modify the setup to perform anomaly detection training in an unsupervised manner, providing only the benign traffic.* Starting from the cleaned dataset version provided with the original 5G-Attacks repository, we construct our own train and test splits by merging the entirety of dataset 1 with all legitimate samples (i.e., those without an attack label) from dataset 2, while our test set comprises all labeled attack samples from dataset 2 together with all samples from dataset 3.

*Following GT2, we need to remove irrelevant traffic from the dataset.* To focus our analysis on relevant control-plane traffic, and in line with the focus on PFCP-based attacks, we further filter out all TCP and ICMP packets, retaining only UDP flows. Table 2 summarizes the feature distribution across protocols (excluding TCP, since PFCP communication in the 5GC relies on UDP) before and after applying our preprocessing pipeline, reducing the feature space and preserving only the relevant and discriminative information for anomaly detection.

*In line with GT3, missing values were imputed using robust and well-established methods from the `scikit-learn` library [17]*: For categorical features, we apply the `SimpleImputer` with the most frequent category strategy, ensuring consistent treatment of common protocol or flag fields; for numerical features, we use the `IterativeImputer` with a `Random Forest Regressor` as estimator to exploit correlations among features and generate plausible imputations.

These choices help maintain the statistical properties of the data and reduce the risk of introducing bias or spurious patterns in subsequent modeling steps. Finally, we apply the `RobustScaler` from `scikit-learn` independently to each numerical feature. The resulting dataset is a compact, protocol-based, and environment-independent representation of 5G traffic.

**Models.** We consider a diverse set of state-of-the-art anomaly detectors, imported from the PyOD library [18], as well as custom pipelines tailored for the 5G network data. Specifically, we evaluate: ABOD [12], COPOD [3], ECOD [4], `FeatBagg` [5], GMM [13], HBOS [2], `Isolation Forest` [8], INNE [10], kNN [6], LODA [9], LOF [7], and PCA [11]. For each detector, we tune the hyperparameters via grid search on the validation set. For each model, the grid search spans detector-specific parameters such as neighborhood size for density-based methods (e.g., k in kNN/LOF), number of bins or splits for histogram/statistical models (e.g., HBOS), and contamination (expected outlier proportion).

We also use ensemble models, which exploit the complementarity of heterogeneous detection strategies. In each ensemble, the scores deduced from the outcome of a subset of base detectors are fed into a binary classifier (specifically, a Support Vector Machine) to deduce an aggregated output score. The ensemble-model parameters are trained on the validation set, while the parameters of the base detectors are kept fixed. *Following the GT4 guideline, the validation set is free of malicious packets.* All ensembles adopt a Support Vector Classifier (SVC) with an RBF kernel, implemented using the `scikit-learn` python library. The hyperparameters of the SVC (i.e., the regularization parameter $C$ and the scaling factor $\gamma$) are selected on different runs to maximize the F1-score on the validation set. We consider the following ensemble models:

- `Ens-HKAIP`: SVC with $C = 10$ and $\gamma = 10$, combining HBOS, kNN, ABOD, INNE, and PCA;
- `Ens-HKGIP`: SVC with $C = 10$ and $\gamma = 10$, combining HBOS, kNN, GMM, INNE, and PCA;
- `Ens-HKLIP`: SVC with $C = 10$ and $\gamma = 10$, combining HBOS, kNN, LOF, INNE, and PCA;
- `Ens-HKLIF`: SVC with $C = 100$ and $\gamma = 100$, combining HBOS, kNN, LOF, INNE, and `FeatBagg`.

We expect this design to enable the system to compensate for the weaknesses of individual detectors in specific attack categories, leading to more stable and accurate anomaly detection performance compared to standalone models.

*Following GT3, we evaluate also the effect of scaling the features.* Therefore, we measure results in two cases, i.e., with and without the application of feature scaling using `RobustScaler` applied to the features (fitted on the training set). Reporting both configurations clarifies the impact of the scaling on the detection performance and its influence on model stability and robustness.

**Evaluation metrics.** To assess the effectiveness of the detection, we employ a set of evaluation metrics commonly used in anomaly detection and classification tasks. The recall, which quantifies the proportion of attack samples correctly identified as anomalies. Precision measures the proportion of samples flagged as anomalous that correspond to actual attacks, reflecting the cost of false positives. The F1-score, defined as the harmonic mean of precision and recall, summarizes the trade-off between detection effectiveness and false alarm rate. Finally, the Area Under the Receiver Operating Characteristic curve (AUC) evaluates the model's ability to discriminate between benign and malicious traffic across all possible detection thresholds, independently of a specific operating point.

**Attack Algorithms.** For reproducibility, we report the hyperparameters used in the implementation of the attacks. The random search RS attack is performed by sampling $\mathbf{p}'$ a single time, without attempting multiple attacks on the same packet. For the Genetic Algorithm (GA), we leverage population-based evolutionary optimization using the `Nevergrad` library.[2] We use two algorithms:

*Differential Evolution ($GA_{\mathrm{DE}}$).* The optimizer is instantiated with a population size of 20 (`popsize=20`), a two-point crossover scheme (`crossover="twopoints"`), and ensures heritage propagation (`propagate_heritage=True`). For every attack sample, the search proceeds for a fixed query budget of 100, thus allowing up to 100 candidate adversarial samples can be generated and tested against the detector.

*Evolution Strategy ($GA_{\mathrm{ES}}$).* The alternative optimizer, available by switching the relevant code block, also uses a population size of 20 (`popsize=20`) and a recombination ratio of 0.9, with a query budget of 100. All other parameters are set as defaults.

Both algorithms evolve a population of adversarial samples, but their internal logic for generating new candidates differs. $\mathrm{GA_{DE}}$ relies on combining individuals via differential perturbations and explicit crossover; thus, new solutions are created by adding scaled differences between members of the current population, with two-point crossover further mixing feature values. $\mathrm{GA_{ES}}$, in contrast, emphasizes random mutation and optional recombination of "parent" solutions, focusing on stochastic local sampling of the search space. For all runs, we set the random seed to 42 to ensure experiment reproducibility. Each optimization process is further bounded by a strict query budget per attack sample, guaranteeing that all experiments are both controlled and repeatable across all attack types.

## 4.2. Evaluation of Detectors

Evaluations of the detection are summarized in Table 3, where, *following guideline GE1*, all evaluation metrics are computed consistently across the two settings on the same test set for a fair comparison.

We show that, among the base detectors, `HBOS`, `IForest`, and `ABOD` achieve the highest F1-scores in the Statistical, Density, and Geometrical categories, respectively. Among the ensemble methods, `Ens-HKLIP` achieves the best tradeoff between accuracy and F1-score in both its versions, making it the most powerful detector overall in all categories. Overall, all ensemble combinations show better overall performance than the individual detectors. *To provide deeper insight into per-class performance, in accordance with the guideline GE2, we evaluate the detection rates for individual attack categories and normal traffic* using a heatmap, as shown in Fig. 2. This representation highlights the strengths and weaknesses of each approach in recognizing specific types of attacks, as well as their ability to minimize false alarms on benign traffic and identify specific categories of hard-to-detect attacks. Taken together,

[2] https://facebookresearch.github.io/nevergrad/

| | | Without RobustScaler | | | | With RobustScaler | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **Model** | **AUC** | **Prec** | **Recall** | **F1** | **AUC** | **Prec** | **Recall** | **F1** |
| **Stat.** | HBOS | **0.996** | 0.979 | 0.977 | 0.978 | 0.989 | 0.864 | 0.946 | 0.903 |
| | COPOD | 0.860 | 0.542 | 0.527 | 0.534 | 0.871 | 0.583 | 0.520 | 0.549 |
| | ECOD | 0.870 | 0.808 | 0.514 | 0.628 | 0.884 | 0.833 | 0.514 | 0.636 |
| **Density** | kNN | 0.631 | 0.998 | 0.533 | 0.695 | 0.717 | 0.998 | 0.533 | 0.695 |
| | LOF | 0.878 | 0.451 | 0.738 | 0.560 | 0.835 | 0.354 | **1.000** | 0.523 |
| | IForest | 0.966 | 0.646 | 0.958 | 0.771 | 0.952 | 0.657 | 0.969 | 0.783 |
| | FeatBagg | 0.772 | 0.953 | 0.489 | 0.647 | 0.844 | 0.256 | **1.000** | 0.408 |
| | LODA | 0.370 | **1.000** | 0.054 | 0.103 | 0.511 | 0.746 | 0.236 | 0.359 |
| | INNE | 0.640 | 0.998 | 0.532 | 0.694 | 0.850 | **1.000** | 0.533 | 0.695 |
| **Geom.** | PCA | 0.860 | 0.363 | 0.782 | 0.495 | 0.858 | 0.363 | 0.782 | 0.496 |
| | ABOD | 0.619 | **1.000** | 0.533 | 0.695 | 0.749 | 0.997 | 0.533 | 0.694 |
| | GMM | 0.731 | 0.345 | 0.533 | 0.419 | 0.860 | 0.418 | **1.000** | 0.589 |
| **Ensemble** | Ens-HKAIP | 0.999 | 0.995 | **1.000** | 0.998 | 0.999 | 0.988 | 0.988 | 0.988 |
| | Ens-HKGIP | 0.999 | 0.995 | **1.000** | 0.997 | **1.000** | 0.995 | 0.988 | **0.992** |
| | Ens-HKLIP | 0.999 | 0.995 | **1.000** | 0.998 | **1.000** | 0.995 | 0.988 | **0.992** |
| | Ens-HKLIF | **1.000** | 0.997 | **1.000** | **0.999** | 0.996 | 0.949 | 0.997 | 0.973 |

**Table 3**
Overall performance comparison of anomaly detection models.

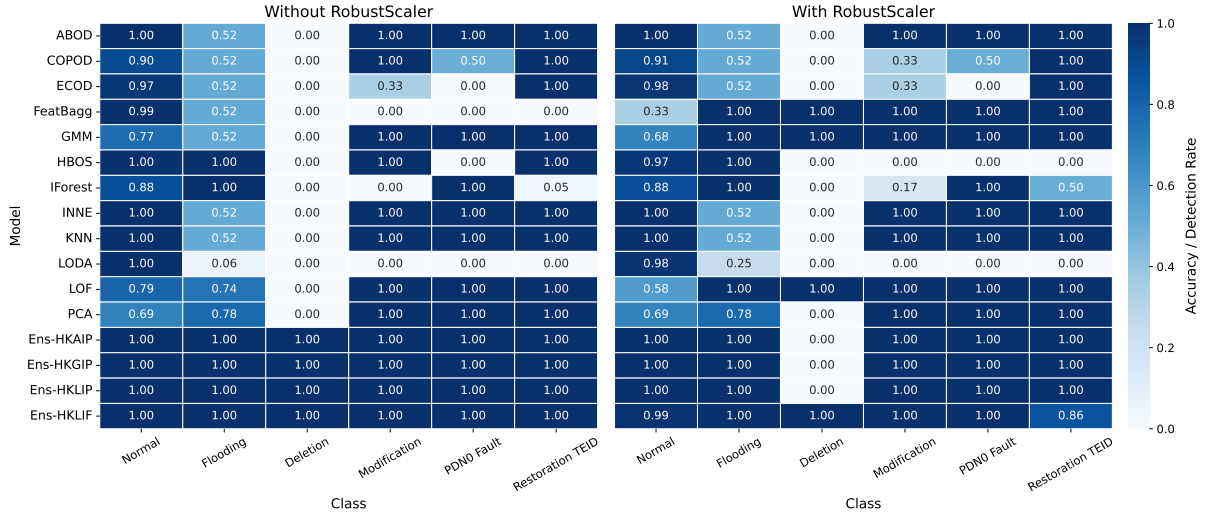


**Figure 2:** Heat Map of the Detection Rate of the Obtained Models

these evaluation strategies ensure a comprehensive and interpretable assessment of IDS capabilities in realistic, imbalanced 5G attack scenarios.

## 4.3. Evaluation of Detectors Under Attack

*After evaluating the performance of the obtained detectors under normal conditions, we assess the robustness under adversarial (feasible and compliant) manipulations, as stated in the guideline GE3.* Specifically, for each model, we generate adversarial samples using both random and optimization-based model-agnostic attack algorithms, as described in Sect. 3.1. In this context, one of the primary evaluation metrics in the adversarial context is the evasion rate, defined as the proportion of adversarially modified attack samples that successfully bypass the detector (i.e., are misclassified as benign). This metric directly quantifies the vulnerability of each IDS to evasion under different attack strategies. To facilitate comparison, the evasion rates are summarized in Table 4. This provides an immediate overview of model robustness

| | | Without Scaler | | | With Scaler | | |
|---|---|---|---|---|---|---|---|
| | **Model** | RS | $GA_{DE}$ | $GA_{ES}$ | RS | $GA_{DE}$ | $GA_{ES}$ |
| **Stat.** | HBOS | 0.0% | 98.4% | 98.4% | 0.0% | **0.0**% | **0.0**% |
| | COPOD | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| | ECOD | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| **Density** | kNN | 0.0% | 100.0% | 100.0% | 0.0% | **0.0**% | **0.0**% |
| | LOF | 0.0% | 100.0% | 100.0% | 0.0% | 100.0% | 100.0% |
| | IForest | 0.0% | 49.0% | 0.0% | 0.0% | **0.2**% | 0.0% |
| | FeatBagg | 47.1% | 100.0% | 100.0% | **0.0**% | **98.4**% | **1.3**% |
| | LODA | **5.3**% | 100.0% | 100.0% | 22.6% | 100.0% | 100.0% |
| | INNE | 0.0% | 100.0% | 100.0% | 0.0% | 100.0% | 100.0% |
| **Geom.** | PCA | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| | ABOD | 0.0% | 100.0% | 100.0% | 0.0% | **0.0**% | **0.0**% |
| | GMM | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| **Ensemble** | Ens-HKAIP | 0.0% | 98.4% | 98.4% | 0.0% | **50.0**% | 98.4% |
| | Ens-HKGIP | 0.0% | 98.4% | 98.4% | 0.0% | **50.0**% | **48.7**% |
| | Ens-HKLIP | 0.0% | 98.4% | 98.4% | 0.0% | **50.0**% | **48.7**% |
| | Ens-HKLF | 0.0% | 100.0% | 100.0% | 0.0% | 100.0% | 100.0% |

**Table 4**
Evasion rates under random and adaptive model-agnostic attack strategies.

and the relative effectiveness of different attack techniques. Results highlight not only the overall susceptibility of the models but also the comparative resilience to simple versus sophisticated adversarial manipulations, supporting a nuanced evaluation of IDS security in a 5G environment. Notably, the results indicate that applying the `RobustScaler` generally improves robustness against adversarial evasion, particularly under optimization-based model-agnostic attacks, by stabilizing the feature space and reducing the impact of extreme values.

## 5. Related Work

Our work lies at the intersection between anomaly detection and adversarial machine learning in the 5GC network, with an explicit focus on realistic and practical approaches to the underlying attack problem. We thereby discuss the related work for each of these topics.

**ML-based intrusion and anomaly detection in the 5GC.** A growing body of work studies ML methods to monitor and secure the 5GC, motivated by the complexity and volume of control- and user plane-traffic. Early work has explored the application of ML for anomaly detection in emerging 5G networks, primarily from a flow-level perspective, demonstrating the feasibility of ML-based detection for 5G traffic and highlighting its potential to identify malicious traffic in yet another application scenario [19]. Recent approaches propose ML-powered Intrusion Detection System (IDS) pipelines tailored to 5GC interfaces and protocols, often emphasizing PFCP-related threats or focusing on explainability-based techniques to support operator decision making. For instance, 5GCIDS introduces an AI-based IDS specifically targeting PFCP and discusses the use of explainability-based techniques as an aid for security monitoring [20]. Related work has also investigated the design of learning-based intrusion detection pipelines for 5GC with a particular focus on efficiency and scalability [21]. In contrast to prior work, which evaluates ML-based 5GC intrusion detection under static and non-adversarial assumptions, we focus on realistic deployment scenarios and propose evaluation guidelines that explicitly consider adaptive attackers manipulating protocol-related traffic to evade detection.

**Adversarial machine learning in the 5GC.** Closer to our work, prior research has highlighted that 5G management and security tasks can be exposed to adversarial manipulation when ML components are deployed in operational pipelines. Specifically, in [22], the authors study this specific scenario,

emphasizing attacker constraints and threat models with adversaries and ML-based defenses. However, the focus of this line of work is on adversarial interference with ML systems that support radio access and network management tasks (e.g., slicing, CQI prediction, modulation recognition), rather than on evading ML-based anomaly detection systems operating on 5GC protocol traffic. In contrast, our work specifically addresses anomaly detection in the 5G Core control plane and complements existing research by providing security-aware evaluation guidelines that explicitly account for adaptive attackers capable of manipulating protocol-compliant traffic to evade detection in realistic deployment scenarios.

**Assumptions, guidelines, and realistic attacks in ML-based detection.** Recent systematization and survey works have highlighted that many ML-based network traffic analysis systems rely on fragile design choices and implicit assumptions—such as environment-dependent identifiers, legacy datasets, or shortcut features—that can lead to overfitting and limited generalization in real deployments [23]. Similar concerns regarding evaluation realism and deployment-oriented assumptions have been raised in recent methodological analyses of learning-based network security systems, further motivating the need for principled and security-aware evaluation practices [24]. Our work also builds on these observations, extending and instantiating their methodological principles in the context of 5GC. In particular, we translate high-level concerns about feature dependence and evaluation realism into concrete, protocol-aware guidelines for anomaly detection in 5GC environments, and further develop them by explicitly considering adversaries that manipulate protocol-compliant traffic to evade learning-based detectors. From a different standpoint, a line of work argues for realistic problem-space perturbations and demonstrates that such attacks can effectively degrade NIDS performance under practical constraints [15]. In contrast, our work does not involve traffic-space manipulation but adopts a complementary and deployment-oriented threat model for the 5GC. Hence, rather than arbitrary feature perturbations, we study adversarial evasion at the feature level under strict protocol and attack constraints, explicitly modeling the subset of related features that an attacker can realistically control without breaking protocol compliance or system functionality.

## 6. Conclusion

In this work, we addressed the gap between current evaluation methodologies for anomaly detection in the 5GC network and the operational reality of deployed networks. We introduced **SAGE-5GC**, a comprehensive set of guidelines for security-aware detection that prioritizes model independence and, crucially, robustness against adaptive attacks.

Through a comprehensive experimental campaign using realistic datasets for 5GC traffic, we demonstrated that high detection performance does not align with its security in the wild. While ensemble-based detectors achieved near-perfect accuracy against static attacks, our evaluation revealed their fragility under adversarial conditions. By employing constrained, model-agnostic optimization strategies (specifically random search and genetic algorithms) we showed that an attacker can generate protocol-compliant perturbations that successfully evade detection without compromising the efficacy of the attack. These findings underscore that standard metrics such as AUC and F1-score are insufficient for security-critical applications when used in isolation. We conclude that future evaluations of 5G anomaly detection systems must explicitly incorporate adversarial robustness assessments. Adopting the SAGE-5GC guidelines provides a necessary foundation for developing next-generation detectors that are not only accurate but also resilient to the evolving threat landscape of 5GC networks.

**Limitations and Future Work.** Despite the significance of these findings, our study has limitations. Our adversarial evaluation relies on a manually defined set of feasible features $J$ based on domain knowledge; however, in complex, multi-vendor environments, identifying these constraints may require automated inference. Furthermore, our analysis focuses exclusively on the PFCP protocol, leaving vulnerabilities related to other protocols unexplored. Future work will aim to bridge these gaps. Therefore, we plan to optimize the set of feasible features $J$ leveraging discrete optimization techniques. Finally, while our adversarial are feasible and compliant, they are not reconstructed fully to replayable

and fully-valid network traffic. Generating adversarial samples at the packet or flow level to produce executable traffic traces that can be validated in realistic testbeds.

## Acknowledgments

This work was partially supported by the EU-funded project Sec4AI4Sec (grant no. 101120393); and by the project SERICS (PE00000014) under the MUR NRRP funded by EU-NextGenEU. This work was carried out while C. Scano was enrolled in the Italian National Doctorate on AI run by the Sapienza University of Rome in collaboration with the University of Cagliari.

## Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

## References

[1] B. Biggio, F. Roli, Wild patterns: Ten years after the rise of adversarial machine learning, Pattern Recognition 84 (2018) 317–331. URL: https://www.sciencedirect.com/science/article/pii/S0031320318302565. doi:https://doi.org/10.1016/j.patcog.2018.07.023.

[2] M. Goldstein, A. Dengel, Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm, KI-2012: poster and demo track 1 (2012) 59–63.

[3] Z. Li, Y. Zhao, N. Botta, C. Ionescu, X. Hu, Copod: copula-based outlier detection, in: 2020 IEEE international conference on data mining (ICDM), IEEE, 2020, pp. 1118–1123. doi:https://doi.org/10.1109/ICDM50108.2020.00135.

[4] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, G. H. Chen, Ecod: Unsupervised outlier detection using empirical cumulative distribution functions, IEEE Transactions on Knowledge and Data Engineering 35 (2022) 12181–12193. doi:https://doi.org/10.1109/TKDE.2022.3159580.

[5] K. Yang, S. Kpotufe, N. Feamster, Feature extraction for novelty detection in network traffic, arXiv preprint arXiv:2006.16993 (2020). doi:https://doi.org/10.48550/arXiv.2006.16993.

[6] F. Angiulli, C. Pizzuti, Fast outlier detection in high dimensional spaces, in: European conference on principles of data mining and knowledge discovery, Springer, 2002, pp. 15–27. doi:https://doi.org/10.1007/3-540-45681-3_2.

[7] M. M. Breunig, H.-P. Kriegel, R. T. Ng, J. Sander, Lof: identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000, pp. 93–104. doi:https://doi.org/10.1145/342009.335388.

[8] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 eighth ieee international conference on data mining, IEEE, 2008, pp. 413–422. doi:https://doi.org/10.1109/ICDM.2008.17.

[9] T. Pevnỳ, Loda: Lightweight on-line detector of anomalies, Machine Learning 102 (2016) 275–304. doi:https://doi.org/10.1007/s10994-015-5521-0.

[10] T. R. Bandaragoda, K. M. Ting, D. Albrecht, F. T. Liu, Y. Zhu, J. R. Wells, Isolation-based anomaly detection using nearest-neighbor ensembles, Computational Intelligence 34 (2018) 968–998. doi:https://doi.org/10.1111/coin.12156.

[11] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, L. Chang, A novel anomaly detection scheme based on principal component classifier, in: IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03), 2003.

[12] H.-P. Kriegel, M. Schubert, A. Zimek, Angle-based outlier detection in high-dimensional data, in: Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 444–452. doi:https://doi.org/10.1145/1401890.1401946.

[13] C. C. Aggarwal, Probabilistic and statistical models for outlier detection, in: Outlier analysis, Springer, 2016, pp. 35–64. doi:https://doi.org/10.1007/978-3-319-47578-3_2.

[14] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: International Conference on Learning Representations, 2018. doi:https://doi.org/10.48550/arXiv.1706.06083.

[15] M. Catillo, A. Pecchia, A. Repola, U. Villano, Towards realistic problem-space adversarial attacks against machine learning in network intrusion detection, in: Proceedings of the 19th International Conference on Availability, Reliability and Security, 2024, pp. 1–8. doi:https://doi.org/10.1145/3664476.3669974.

[16] C. D. A. B. P. B. V. V. Conan, 5g-attacks: A new dataset from realistic 5g-core attacks, in: 4th International Conference on 6G Networking, 2025.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[18] Y. Zhao, Z. Nasrullah, Z. Li, Pyod: A python toolbox for scalable outlier detection, Journal of Machine Learning Research 20 (2019) 1–7. URL: http://jmlr.org/papers/v20/19-011.html. doi:https://doi.org/10.48550/arXiv.1901.01588.

[19] J. Lam, R. Abbas, Machine learning based anomaly detection for 5g networks, arXiv preprint arXiv:2003.03474 (2020). doi:https://doi.org/10.48550/arXiv.2003.03474.

[20] P. Radoglou-Grammatikis, G. Nakas, G. Amponis, S. Giannakidou, T. Lagkas, V. Argyriou, S. Goudos, P. Sarigiannidis, 5gcids: An intrusion detection system for 5g core with ai and explainability mechanisms, in: 2023 IEEE Globecom Workshops (GC Wkshps), IEEE, 2023, pp. 353–358. doi:https://doi.org/10.1109/GCWkshps58843.2023.10464667.

[21] Y.-E. Kim, Y.-S. Kim, H. Kim, Effective feature selection methods to detect iot ddos attack in 5g core network, Sensors 22 (2022) 3819. doi:https://doi.org/10.3390/s22103819.

[22] G. Apruzzese, R. Vladimirov, A. Tastemirova, P. Laskov, Wild networks: Exposure of 5g network infrastructures to adversarial examples, IEEE Transactions on Network and Service Management 19 (2022) 5312–5332. doi:https://doi.org/10.1109/TNSM.2022.3188930.

[23] N. Wickramasinghe, A. Shaghaghi, G. Tsudik, S. Jha, Sok: Decoding the enigma of encrypted network traffic classifiers, in: 2025 IEEE Symposium on Security and Privacy (SP), IEEE, 2025, pp. 1825–1843. doi:https://doi.org/10.1109/SP61157.2025.00165.

[24] Y. Zhao, G. Dettori, M. Boffa, L. Vassio, M. Mellia, The sweet danger of sugar: Debunking representation learning for encrypted traffic classification, in: Proceedings of the ACM SIGCOMM 2025 Conference, 2025, pp. 296–310. doi:https://doi.org/10.1145/3718958.3750498.

# A. Further Experimental Details

This section provides additional details on the experimental settings.

## A.1. Datasets

The various cyberattacks implemented are:

**PFCP Restoration-TEID.** The PFCP Restoration-TEID attack (CVE-2025-29646) disrupts the PFCP session recovery process by injecting forged restoration messages containing manipulated TEID values. The attack specifically targets `pfcp.f_teid.teid`, setting it to values exceeding the TEID pool size (TEID $> 1024 \times 4 \times 16 = 65536$), alongside manipulated `pfcp.f_teid.ipv4_addr`, `pfcp.pdr_id`, and `pfcp.node_id_ipv4` fields. Additional fields modified include `pfcp.ie_type`, `pfcp.ie_len`, `pfcp.msg_type`, `pfcp.f_seid.ipv4`, `pfcp.flags`, the session flag `pfcp.s`, `pfcp.seid`, and the sequence number `pfcp.seqno`. Since TEIDs bind PFCP control-plane state to user-plane tunnel identifiers, corrupting them causes the UPF to reconstruct incorrect forwarding state, leading to traffic misrouting, session drops, or UPF crash. The attack exploits the lack of authentication and integrity checks in PFCP recovery procedures, allowing an attacker capable of spoofing PFCP packets to desynchronize the UPF's control-plane and user-plane state.

**PFCP Flood.** The PFCP Flood attack overwhelms PFCP control-plane entities with a high volume of unsolicited PFCP messages. The attack manipulates `pfcp.msg_type` (primarily Session Establishment Requests with value 50 and Heartbeat Requests), randomizes `pfcp.seqno` and `pfcp.ue_ip_addr_ipv4`, while also modifying `pfcp.seid` and `pfcp.s` flag unnecessarily. Because PFCP operates over UDP (port 8805) without built-in rate limiting, the receiving UPF or SMF is forced to process excessive traffic, exhausting control-plane resources and delaying or blocking legitimate signaling. This results in reduced session management performance and potential denial of service.

**PFCP Deletion.** In the PFCP Deletion attack, an attacker sends forged Session Deletion Requests by manipulating `pfcp.msg_type` (set to 54), `pfcp.s` flag, and targeting active `pfcp.seid` values to prematurely remove session state at the UPF. Additional fields modified include `pfcp.seqno`, `pfcp.length`, and `pfcp.flags`. This forces the immediate deletion of PDRs, FARs, and associated tunnel information, interrupting user-plane flows and causing abrupt service disruption. The attack is feasible because PFCP does not authenticate control messages, making it possible for a spoofing attacker to trigger unauthorized session teardown.

**PFCP Modification.** The PFCP Modification attack injects counterfeit Session Modification messages (`pfcp.msg_type = 52`) targeting active sessions identified by their `pfcp.seid` values. The attack operates by altering Forwarding Action Rules (FARs) within the UPF: specifically, it sets `pfcp.apply_action.forw` to 0 (disabling forwarding) while setting `pfcp.apply_action.buff` and `pfcp.apply_action.nocp` to manipulate buffering and notification behavior, effectively converting legitimate forwarding rules into DROP actions. Simultaneously, the attack corrupts forwarding parameters by modifying `pfcp.outer_hdr_creation.ipv4` (destination IP for encapsulated traffic), `pfcp.outer_hdr_creation.teid` (often set to invalid values to cause tunneling failures), and `pfcp.dst_interface`, redirecting packets to incorrect network interfaces. These changes are complemented by modifications to protocol-level fields such as `pfcp.s`, `pfcp.seqno`, `pfcp.length`, and `pfcp.flags`. By systematically corrupting these forwarding and encapsulation parameters, the attacker can cause user traffic to be dropped, misrouted to the wrong network segments, or encapsulated with invalid tunnel headers that result in packet discard. The attack exploits PFCP's lack of message authentication and integrity protection, allowing unauthorized modification of critical control-plane state that directly governs user-plane traffic processing at the UPF.

**UPF PDN-0 Fault.** The UPF PDN-0 Fault attack exploits weaknesses in the UPF's handling of PDN Type 0 session context (`pfcp.pdn_type=0`). The attack manipulates key session establishment fields including `pfcp.node_id_ipv4`, `pfcp.f_seid.ipv4`, `pfcp.ue_ip_addr_ipv4` and `pfcp.pdr_id`, cre-

ating an inconsistent state. It also sets F-TEID flags to invalid combinations: `pfcp.f_teid_flags.ch`, `pfcp.f_teid_flags.ch_id`, and `pfcp.f_teid_flags.v6`. Additional modified fields include `pfcp.msg_type`, `pfcp.flags`, `pfcp.s`, and `pfcp.seqno`. This attack induces state inconsistencies that can cause session establishment failures, unexpected teardown, or misrouting of user traffic, highlighting insufficient validation of PFCP parameters in PDN context management.