

Article

AI-Driven Forensic Analysis and Threat Detection for Open RAN and 5G Core Vulnerabilities: An Experimental Study with srsRAN and Open5GS

Akhmet Tussupov ^{1,*}, Yedil Nurakhov ², Danil Lebedev ², Madi Shayakhmetov ^{3,4,*}, Leila Rzayeva ^{3,4}, Ulykbek Shambulov ⁵ and Ibraheem Shayea ¹

- ¹ School of Cybersecurity, Astana IT University, Astana 010000, Kazakhstan; i.shayea@astanait.edu.kz
² Faculty of Information Technologies and Artificial Intelligence, al-Farabi Kazakh National University, Almaty 050040, Kazakhstan; y.nurakhov@gmail.com (Y.N.); dan-lebedev@mail.ru (D.L.)
³ Research Innovation Center “CyberTech”, Astana IT University, Astana 010000, Kazakhstan; l.rzayeva@astanait.edu.kz
⁴ National Institute for Cyber Policy and Research, RSE on REM “Digital Government Support Center”, Astana 010000, Kazakhstan
⁵ Head NISCC (The National Information Security Coordination Center), Astana 010000, Kazakhstan; u.shambulov@sts.kz
* Correspondence: akhmet.tussupov@astanait.edu.kz (A.T.); 242722@astanait.edu.kz (M.S.)

Abstract

(1) Background: The disaggregated and software-defined nature of fifth-generation (5G) core networks and the Open Radio Access Network (O-RAN) architecture increase the attack surface and produce large volumes of heterogeneous evidence that must be analyzed in real time to support incident reconstruction. Open-source 5G stacks (including Open5GS and srsRAN) have become reference platforms in the literature, yet recent research, such as the RANsacked study that reported 119 vulnerabilities and 97 unique CVEs across multiple LTE/5G implementations, have highlighted the pressing need for AI-based detection and forensic capabilities specific to these stacks. (2) Methods: We introduce an experimental framework consisting of a reproducible srsRAN+Open5GS testbed and an AI-driven forensic and detection pipeline. The pipeline receives control-plane (NAS, NGAP, F1AP) and Service-Based Interface (SBI) traffic, extracts protocol- and statistically grounded features and classifies traffic into seven attack types using a hybrid CNN–LSTM model. Integrity-protected and timeline-correlated forensic artifacts (PCAP, logs, memory dumps) assist in reconstructing an incident. (3) Results: The proposed hybrid model achieves a macro F1-score of 0.972 and an AUC-ROC of 0.995 (5-fold CV) and degrades gracefully under load. (4) Conclusions: We show that AI-based detection can be coupled with a scientifically sound evidence chain in open-source 5G stacks deployed as disaggregated mobile networks.



Academic Editors: Yichuang Sun,
Haeyoung Lee and Oluyomi Simpson

Received: 7 June 2026

Revised: 9 July 2026

Accepted: 14 July 2026

Published: 1 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: 5G security; Open RAN; srsRAN; Open5GS; intrusion detection; deep learning; digital forensics; vulnerability assessment; CNN-LSTM; Service-Based Architecture

1. Introduction

Fifth-generation (5G) mobile networks have restructured telecommunications infrastructure around cloud-native, service-based and disaggregated design principles. Service-Based Architecture (SBA) breaks down the core into modular Network Functions (NFs) including the Access and Mobility Management Function (AMF), Session Management

Function (SMF), Network Repository Function (NRF) and User Plane Function (UPF), all through standardized HTTP/2 interfaces over the Service-Based Interface (SBI) [1,2]. Simultaneously, the Open Radio Access Network (O-RAN) Alliance has redefined the radio-access tier into a disaggregated stack of Centralized Units (CUs), Distributed Units (DUs) and Radio Units (RUs) connected via open fronthaul interfaces and orchestrated by Near- and Non-Real-Time RAN Intelligent Controllers (Near-RT RIC, Non-RT RIC) that host third-party xApps and rApps [3,4].

These design decisions open the door to vendor independence, accelerated innovation and operational agility, but they also bring an order-of-magnitude increase in the attack surface. The O-RAN WG11 threat model has seven categories that comprise more than eighty concrete attack types, such as malicious xApp/rApp injection, fronthaul tampering, and container escape and AI/ML poisoning [4]. This is compounded by the cloud-native nature of the 5G core: vulnerabilities in NF implementations, weak authentication of SBI calls, and risks in the supply-chain across containerized deployments all contribute to making the SBA a prime target for adversaries [5,6]. Their complexity, coupled with limited operator visibility into virtualized components, has led to regulators in many jurisdictions classifying mobile networks as critical information and communications infrastructure [7].

Open-source 5G stacks have become the de facto platform for in-laboratory security testing: they provide a Release-17-compliant 5G core and a CU/DU implementation aligned with O-RAN principles, and together they form a widely used reference testbed [8–10]. Their vulnerabilities, however, are correspondingly consequential, since a single proof-of-concept against a public NF implementation can have wide-reaching effects. A recent study dubbed RANsacked found 119 distinct vulnerabilities and 97 CVE identifiers across seven LTE and three 5G open-source implementations, including Open5GS and srsRAN, with many of them being capable of disrupting cellular service at the city scale [11]. Open5GS and OpenAirInterface (OAI) deployments have also been shown to be practically vulnerable to denial-of-service (DoS) attacks, replay attacks, and SBI injection attacks by independent assessments [5,12].

Most existing security research on 5G is either focused on vulnerability discovery in protocol stacks or on training ML classifiers for network-level intrusion detection [13–15]. Each of these lines of work addresses only a partial aspect of the problem. Detection-only systems are typically tested on synthetic features and do little or nothing to deliver an actionable forensic pipeline. Recent solutions that are O-RAN-resident, including 5G-SPECTOR [16] and AI-driven xApps to detect anomaly traffic [17,18] bridge the two views to a degree, but their forensic affordances, such as chain of custody, integrity proofs, timeline reconstruction, etc., remain underdeveloped relative to the standards expected in the practice of digital forensics [19,20].

This work addresses that gap with an integrated, AI-driven forensic and detection framework for Open RAN and the 5G core, instantiated on a reproducible srsRAN+Open5GS testbed. We make the following contributions:

- We describe the threat surface of the srsRAN+Open5GS open-source stack in the context of the SBA and O-RAN architectures, mapping representative CVEs (CVE-2025-1893, CVE-2024-24432, CVE-2025-5935) and RANsacked-class vulnerabilities to attack categories that can be reproduced safely in a closed laboratory [11,21–23].
- We design and implement a protocol- and flow-level AI-driven detection pipeline based on a hybrid Convolutional–Long-Short-Term-Memory (CNN–LSTM) classifier that uses protocol- and flow-level features derived on N1/N2 NAS, NGAP, F1AP and HTTP/2 SBI traffic and compares it to eight baselines. The hybrid is chosen because 5G/O-RAN control-plane attacks manifest jointly as short local protocol motifs (e.g., a malformed Authentication Response immediately following a Regis-

tration Request) and as longer-range temporal regimes (bursty fuzzing, sustained flooding): the convolutional front-end captures the former and the recurrent layer the latter, a complementarity that single-paradigm baselines (pure CNN, pure LSTM, Transformer) do not exploit jointly (Section 2).

- We combine detection with a forensically sound evidence chain, that is, with artifacts that can be investigated—PCAP capture with cryptographic hashing, log preservation, memory snapshots and timeline reconstruction—so that every alert is accompanied by investigation-ready artifacts.
- We test the framework on ten scenarios (nine attack and one benign baseline) on a real srsRAN+Open5GS testbed and report the precision of detection, end-to-end latency, scalability of throughput and forensic completeness and compare against five state-of-the-art publications.

Beyond integrating existing building blocks, the central contribution of this work is the coupling of real-time multi-class detection with a forensically sound, chain-of-custody-preserving evidence pipeline on a disaggregated open-source 5G/O-RAN stack—a combination that, to our knowledge, no prior system provides (Table 1). Detection-only systems produce alerts but no investigation-ready, integrity-protected artifacts; forensic and lawful-interception work predates the SBA/O-RAN attack surface; and the trust-boundary placement of the detector as a side-car *outside* the RIC (Section 3) is itself a design contribution that structurally rules out training-time poisoning and post hoc evidence tampering. The feasibility of porting this design to commercial deployments, together with its concrete limitations, is examined explicitly in Section 7.

Table 1. Comparative positioning of this work against representative state of the art. F1-scores reported by the original authors are listed in parentheses where directly available; “–” indicates the work does not report a comparable multi-class F1; a checkmark (✓) indicates that the capability is provided by the work, “partial” indicates that it is only partially provided, and bold denotes the framework proposed in this study.

Work	Approach	Real-Time Detection	Forensic Evidence	Open-Stack Testbed
Samarakoon et al. [13]	5G-NIDD dataset + classical/DL ML (F1 = 0.97)	✓	–	partial
Bitton et al. [14]	Deep transfer learning, BiLSTM (F1 = 0.987, binary DDoS only)	✓	–	partial
5G-SPECTOR [16]	O-RAN xApp + MobiFlow (F1 $\approx$ 0.95)	✓	partial	✓
Tsourdinis et al. [17]	AI-driven ATD xApp (F1 $\approx$ 0.94, DoS only)	✓	–	✓
6G-XSec [18]	Unsupervised + explainable (F1 $\approx$ 0.94)	✓	partial	✓
Dolente et al. [5]	Manual vulnerability assessment (no ML)	–	✓	✓
This work	Hybrid CNN-LSTM + chain-of-custody (F1 = 0.972, 7-class)	✓	✓	✓

The rest of the paper will be structured in the following way. Section 2 reviews the related literature in 5G/O-RAN security and AI-driven detection. The threat model is developed in Section 3. The suggested framework is introduced in Section 4 and the experiment setup and evaluation protocol are presented in Section 5. In Section 6 the results and ablation studies are reported. Implications, limitations and future directions are discussed in Section 7 and a conclusion is provided in Section 8.

2. Background and Related Work

This section situates the proposed work at the intersection of three research threads: the architectural and security foundations of 5G core and Open RAN deployments, empirical vulnerability assessments of open-source 5G stacks such as Open5GS and srsRAN, and AI-driven intrusion detection systems tailored to cellular environments. We additionally review digital-forensics practice as it applies to mobile networks, since our framework couples real-time detection with a forensically sound evidence chain. Each subsection summarizes the state of the art, identifies the gaps relevant to our contribution, and motivates the design decisions detailed in Sections 3–5. A concluding subsection positions this work explicitly against representative state-of-the-art systems.

2.1. 5G Core and O-RAN Architectures

The security architecture of the 5G System (5GS) [1,2,24] specifies end-to-end protection between the User Equipment (UE) and the data network, mutual authentication between the UE and the network using EAP-AKA', SUPI/SUCI subscriber-privacy mechanisms, and Service-Based Architecture (SBA) protections including TLS-protected SBI calls and OAuth 2.0 authorization between NFs. The standard also requires control and user plane integrity protection and detection of man-in-the-middle and replay attacks. Although this comprehensive baseline exists on paper, in practice 5G core deployments often disable TLS for performance and operational reasons, exposing the SBI to API injection, replay and DoS attacks [5,12].

The O-RAN Alliance defines an open, disaggregated radio-access architecture that complements the 3GPP standards [3]. By dividing the gNB into CUs, DUs and RUs, defining open fronthaul (7.2x) and management (E2, O1, O2, A1) interfaces, and exposing programmable RICs, O-RAN enables vendor neutrality but introduces new threat vectors: malicious xApps that misuse the E2 interface, compromise of orchestrators (SMO), and AI/ML model poisoning of RIC-resident inference services [4,25,26]. The O-RAN WG11 working group has published a detailed set of security requirements that mandate SSH/TLS/IPSec/DTLS to protect interfaces, the Certificate Management Protocol (CMP) to enroll certificates and OAuth 2.0 to authorize services [4].

2.2. Vulnerability Assessment of Open-Source 5G Stacks

Empirical evaluations of open-source 5G core implementations continually reveal security vulnerabilities. A systematic vulnerability analysis of Open5GS and OAI focusing on the AMF and NRF/NEF SBI endpoints was conducted by Dolente et al. [5] and it demonstrated that both stacks do not have default TLS and are susceptible to SQL injection and inconsistent input validation, and that OAI in particular crashes when subjected to HTTP/2 flooding. Park et al. [6] studied the resilience of real 5G test networks and reported a diverse range of authentication, encryption and signaling-plane vulnerabilities. In the RAN-core Vulnerabilities study [11], a fuzzer that is aware of domain information was used to test LTE/5G RAN-core interfaces and disclosed 119 vulnerabilities (97 unique CVEs) across Open5GS, Magma, OpenAirInterface, Athonet, SD-Core, NextEPC and srsRAN.

The practical risk is exemplified by specific CVEs in the Open5GS stack. Open5GS is affected by CVE-2025-1893, which targets Open5GS up to and including 2.7.2, where a single UE can invoke the `gmm_state_authentication` function in the AMF and cause a process crash, stopping mobility management and session activation [21]. CVE-2024-24432, which affects the Open5GS 2.6.4 and older, allows DoS via a crafted NAS packet that is processed by the `ogs_kdf_hash_mme` function [22]. CVE-2025-5935 targets the `common_register_state` function in the AMF, where a crafted value of the `ran_ue_id`

parameter triggers a crash; a public exploit has been published [23]. The three CVEs are the minimum reproduced in our testbed.

2.3. AI-Driven Intrusion Detection for 5G and O-RAN

The 5G-NIDD dataset was created on the 5G test network at the University of Oulu, and it is now used as a reference benchmark when investigating intrusion detection in a 5G environment (with labeled flows) [13]. A deep transfer-learning approach, called DTL-5G, and implemented using BiLSTM, was introduced by Bitton et al. [14] and achieved 98.74% accuracy on 5G-NIDD; later work has explored Transformer-based designs (Deep-TransIDS [15]) and bidirectional temporal models (BiTAD [27]). Various O-RAN-native detection systems have been proposed, such as 5G-SPECTOR [16], which is an O-RAN-compliant L3 attack-detection xApp that introduces the MobiFlow telemetry stream and a stateful detection engine; an AI-driven Anomaly Traffic Detector (ATD) xApp embedded in O-RAN that mitigates DoS in real time, described by Tsourdinis et al. [17]; and 6G-XSec [18], which investigated explainable, unsupervised security telemetry analysis at the edge. In their study, Yungaicela-Naula et al. [25] evaluated the effects of AI/ML misconfiguration on O-RAN security policies. None of these works, however, considers the artifacts that are generated by detection as forensic evidence that has guarantees of chain-of-custody.

The architectural choices in this body of work also motivate our model selection. Purely convolutional detectors [13] capture local n -gram-like protocol patterns but are weak at modeling the long-range temporal dependencies of sustained or regime-shifting attacks; purely recurrent detectors such as the BiLSTM of DTL-5G [14] and BiTAD [27] model temporal structure but treat each per-window feature vector as an opaque token, discarding the short local protocol motifs that distinguish, for example, a malformed authentication exchange from a benign one; and Transformer encoders [15] attend globally but are data-hungry and, as we show in Section 6, scale poorly under the per-window latency budget of a Near-RT RIC control loop. A hybrid CNN-LSTM directly addresses this gap: a 1D convolutional front-end first encodes local protocol motifs within each window, and a recurrent back-end then models their evolution across windows. This division of labor is the explicit reason we adopt the hybrid rather than any single-paradigm model, and the detection results and feature-group ablation in Section 6 quantify the resulting gain over each baseline.

Federated and decentralized learning for cross-operator detection.

A complementary line of work studies how detection models can be trained *across* operators or O-RAN nodes without centralizing raw traffic, which is attractive both for privacy and for resilience. Classical federated averaging is vulnerable to single points of failure and to model poisoning when a node is compromised. To mitigate this, hierarchical federated learning (FL) organizes clients into edge-aggregator tiers and shapes the data distribution at the edge to improve communication efficiency [28], while heterogeneous-client variants transfer knowledge across tiers with differing model capacities [29]. Fully decentralized, peer-to-peer designs remove the central aggregator entirely: BlockDFL uses a blockchain to coordinate trustless peer-to-peer FL [30], and FedRA applies a fully decentralized FL scheme to robust intelligence sharing across O-RAN dApps [31]. These architectures are directly relevant to our framework: because our detection side-car is deployed per-site, hierarchical or fully decentralized FL would let multiple operators jointly improve detection while the system keeps operating even if some nodes become unavailable or are attacked, avoiding the single point of failure of centralized aggregation. We return to this as a concrete extension in Section 7.8.

2.4. Digital Forensics in Mobile Networks

Cellular-network digital forensics has traditionally been centered around Lawful Interception, acquisition of mobile-devices, and analysis of call detail records [32,33]. The 5G transition has led to new evidence types and challenges: encrypted SBI traffic, ephemeral virtualized network functions, network slicing and lawful interception across federations of operators [32,34]. Packet-level forensics surveys [35] highlight the role of deep packet inspection and AI-augmented traffic classification, and emerging frameworks instrument 5G core NFs to capture SBI messages falling outside the boundaries of transport-layer encryption to support timeline reconstruction [34]. Other domain-specific contributions in Kazakhstan have related to forensic recovery of heterogeneous storage and devices: Rzayeva et al. developed a methodology of automated forensic recovery of Hikvision and Dahua DVR/NVR systems [36]; Imanberdi et al. designed a data decryption method for raw mobile-phone disk image files [37]; and complementary work has explored intelligent browser-history forensics for analyzing user activity logs and behavioral profiles of user activity logs [20]. Such contributions guide our forensic-ready design but are not specific to the radio-access or SBA layers of 5G.

2.5. Positioning of This Work

This work sits at the intersection of these threads. In contrast to vulnerability-focused studies [5,11], we offer a deployable detection pipeline. In comparison to dataset-centric ML detectors [13–15], we run on traffic obtained in real-time on an actual srsRAN+Open5GS testbed and incorporate forensically validated evidence collection. Compared to O-RAN xApp detection systems [16–18], we couple detection with chain-of-custody guarantees on PCAPs, logs and memory dumps that can be used to investigate an incident that has occurred. This positioning can be summarized as in Table 1.

3. Threat Model

We are using an open-source 5G testbed which combines *srsRAN_Project* as the gNB CU/DU, the *Open5GS* as the 5G core, and srsUE or UERANSIM as the User Equipment. Figure 1 summarizes the relevant 5G core Service-Based Architecture (SBA) and O-RAN reference architecture, highlighting the three acquisition vantage points used by our framework: the N1/N2 interface between the UE/gNB and the AMF, the F1 interface between the CU and the DU, and the HTTP/2 SBI message bus interconnecting AMF, NRF, SMF and UPF. The defender is in charge of the testbed and is allowed to instrument any component but does not alter upstream software.

The adversary $\mathcal{A}$ either resides within the administrative domain (insider or supply-chain) of the operator or has obtained access to network-internal interfaces (compromised xApp, malicious VNF or rogue gNB on the fronthaul), according to the O-RAN WG11 threat model and earlier insider-threat analyses [4,12].

Adversary capabilities:

$\mathcal{A}$ can: (a) send malformed NAS packets over a controlled UE to the AMF, (b) replay captured NAS Security Mode Command (SMC) sequences on the N1 interface using tools such as 5Greplay [5,38], (c) flood the AMF or NRF SBI endpoints with HTTP/2 requests using MHDDoS-class tools [39], (d) probe NRF endpoints with API injection payloads from the OWASP Top 10 [40], (e) fuzz the RAN–core interface following the RANsacked methodology [11], and (f) mount privacy-violating IMSI-catcher style probes against unencrypted radio segments.

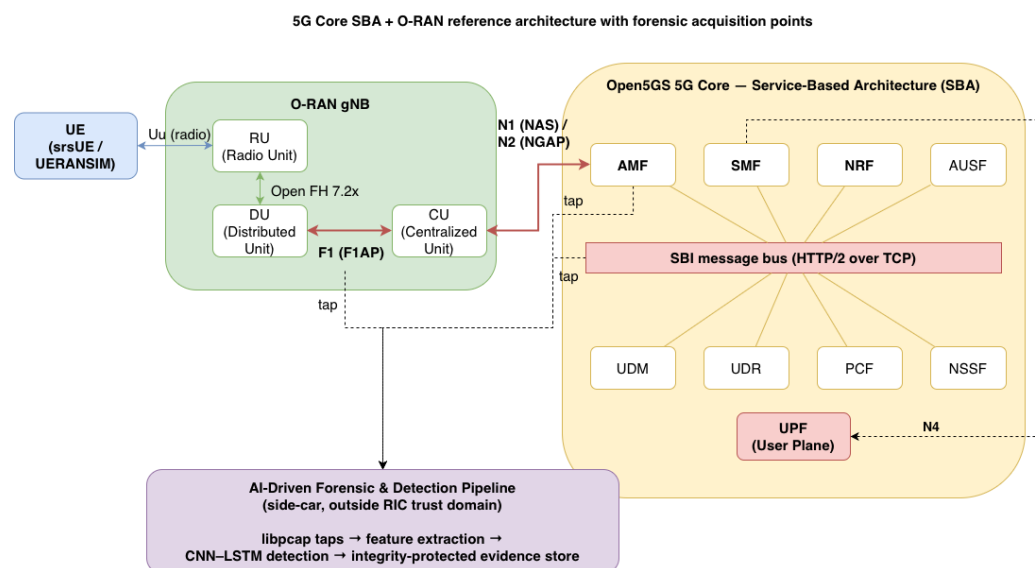


Figure 1. Reference 5G core SBA and O-RAN architecture annotated with the interfaces relevant to this work. The UE attaches to the gNB (split into O-RAN CU, DU and RU over the open fronthaul); the gNB connects to the 5G core over N1 (NAS) and N2 (NGAP), while the CU/DU split exposes the F1 (F1AP) interface. Within the core, Network Functions (AMF, SMF, NRF, AUSF, UDM, UDR, PCF, UPF) communicate over the HTTP/2 Service-Based Interface (SBI). Dashed taps mark the three forensic acquisition points (N1/N2, F1, SBI) instrumented by the proposed pipeline.

Defender capabilities:

The 5G stack side-car, the proposed AI-based forensic and detection pipeline, deploys on the defender. The pipeline (i) captures packets on N1/N2, F1 and SBI interfaces, (ii) logs component logs and triggers memory snapshots on anomaly, (iii) runs a CNN-LSTM classifier in near-real time, and (iv) keeps evidence in an integrity-guaranteed store using cryptographic hashing.

Trust boundary of the detection pipeline:

An imperceptible yet significant aspect is where the detection pipeline itself executes. We explicitly place it *outside* the trust domain of the RAN Intelligent Controller (RIC), as a side-car to the data-plane components instead of an xApp on the Near-RT RIC trust domain. An adversary that breaches an xApp, i.e., one of the threat vectors listed above, does not, therefore, obtain direct read or write access to the detection model, its training data, or the integrity-protected evidence store. This separation does not preclude the adversarial-ML risks (a sophisticated attacker still able to design evasive traffic patterns against the deployed model), but it excludes the more dangerous capability of *model poisoning during training* or of *tampering with stored evidence post hoc*. We do not consider attacks on the evidence store between hashing and signing, attacks on the host kernel, nor the side-car process boundary, which outlines the scope of rootkit countermeasures.

Out of scope:

Attacks at the air interface (jamming, GPS-spoofing) and radio-frequency forensics are not within the scope of this study, although our framework's evidence-chain can be used with downstream PHY-layer analysis [34,41].

4. Proposed Framework

The proposed framework integrates real-time AI-driven attack detection with a forensically sound evidence chain for open-source 5G stacks, addressing the gap between vulnerability-focused studies and dataset-centric ML detectors identified in Section 2.

The overall architecture is shown in Figure 2 and is organized into four cooperating layers: (i) the open-source 5G data plane comprising the srsRAN gNB and the Open5GS core, (ii) a forensic acquisition layer that captures control-plane and SBI traffic together with logs and memory snapshots, (iii) an AI-driven detection layer implementing a hybrid CNN–LSTM classifier, and (iv) an integrity-protected evidence store and reporting layer that turns detections into investigation-ready artifacts with chain-of-custody guarantees.

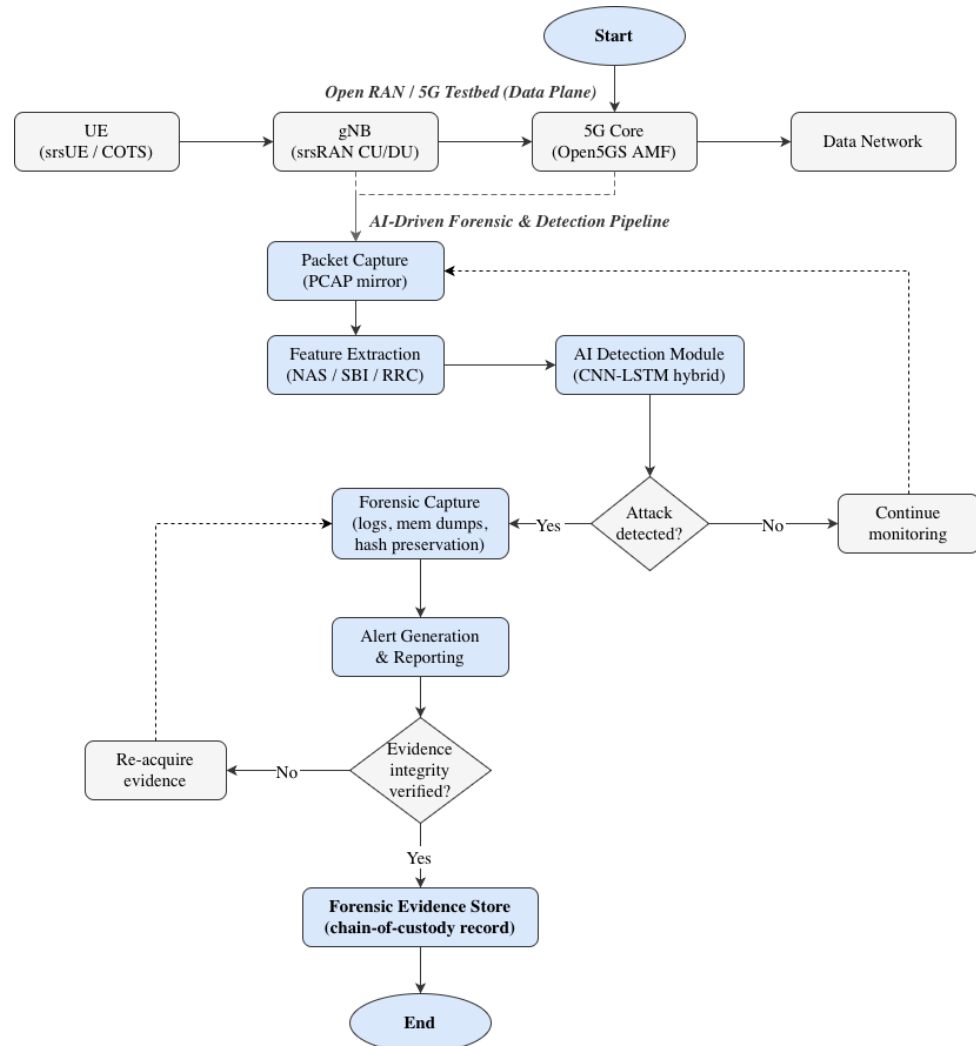


Figure 2. A general structure of the proposed AI-based forensic and detection system. Dashed lines are used to represent acquisition taps, and solid arrows represent the detection and forensic pipeline.

4.1. Forensic Acquisition Layer

The implementation of acquisition is carried out through libpcap-based taps at three vantage points: the N1/N2 interface between the UE/gNB and AMF, the F1 interface between xCU and DU, and the SBI message bus that interconnects AMF, NRF, SMF and UPF. We run containerized tshark instances and processes that run rotating PCAP files using tcpdump. The PCAP files P_i are each further augmented with an integrity envelope:

$$H_i = \text{SHA-256}(P_i \parallel t_i \parallel n_i), \quad (1)$$

The variables here are t_i , the start time as obtained by a monotonic clock, and n_i , a per-file nonce. Hashes are added to a manifest that is signed with an Ed25519 key belonging to the forensic subsystem, according to the principles of chain-of-custody as expressed by [33,36].

At the same time, the component logs (Open5GS NF logs, srsRAN CU/DU logs, container runtime logs) are streamed to the evidence store. Resource measurements on a per-process basis (resource sampling at 1 Hz: read `/proc/[pid]/stat` and read `/proc/[pid]/status`) are used to support analysis of post hoc behavior. Upon being alerted by the detection layer, a memory snapshot of the impacted NF process is taken using `gcore` and stored alongside the associated PCAP slice, subject to the per-NF deduplication policy described in Section 4.4.

4.2. Feature Extraction

We extract a feature vector of the aggregated flow $\mathbf{x} \in \mathbb{R}^d$ of a sliding flow window of length $\tau = 1$ s. The features can be grouped into three:

Statistical features:

The statistical features are the number of packets, the number of bytes, the mean and standard deviation of the packet length, the mean and standard deviation of the interval between packets, the Shannon entropy of the payload bytes, and the ratio of TCP packets to UDP packets and SCTP packets. The Shannon entropy of a payload b with length L is

$$H(b) = - \sum_{x=0}^{255} p(x) \log_2 p(x), \quad (2)$$

In which $p(x)$ is the empirical frequency of the value of a byte, x .

Protocol features:

In the case of NAS we log the message-type histogram (Registration Request, Authentication Response, Security Mode Complete, etc.) and the integrity and ciphering algorithm indicators. In the case of NGAP/F1AP, we store the procedure code and the number of UE-Context-Modification messages. In the case of SBI (HTTP/2 over TCP) we capture distributions of path and method decoded by HPACK and we detect malformed frames using `nghttp2`.

Temporal features:

These features include the temporal properties of bursty and chaotic behavior of many fuzz- and DoS-class workloads, such as a lag- k autocorrelation of message arrival rate, $r_k = \text{Corr}(\lambda_t, \lambda_{t-k})$, with $k \in \{1, 2, 5, 10\}$. Figure 3 shows this difference between a typical set of legitimate and adversarial workloads.

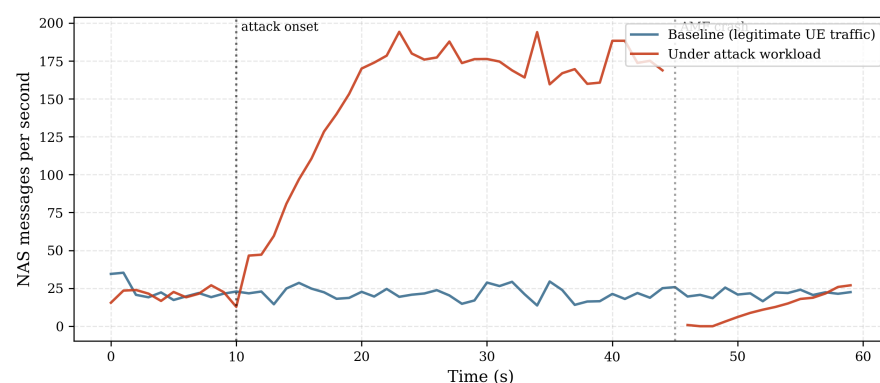


Figure 3. NAS-layer message rate at AMF: legitimate vs adversarial workload. The rate at which NAS messages are received at the AMF within a 30 s time frame. The workload, under domain-informed fuzzing (red), has chaotic, regime-shifting bursts, unlike the legitimate workload (blue), which follows a low-variance, smoothed Poisson-like envelope.

The resulting features' discriminative value is shown in Figure 4, which demonstrates the kernel-density estimates (computed with `scipy.stats.gaussian_kde`) of two informative features, NAS inter-arrival time and payload Shannon entropy, under benign and attack workloads. The two distributions are quite distinct on the two features, thus explaining the need to have them in the feature vector.

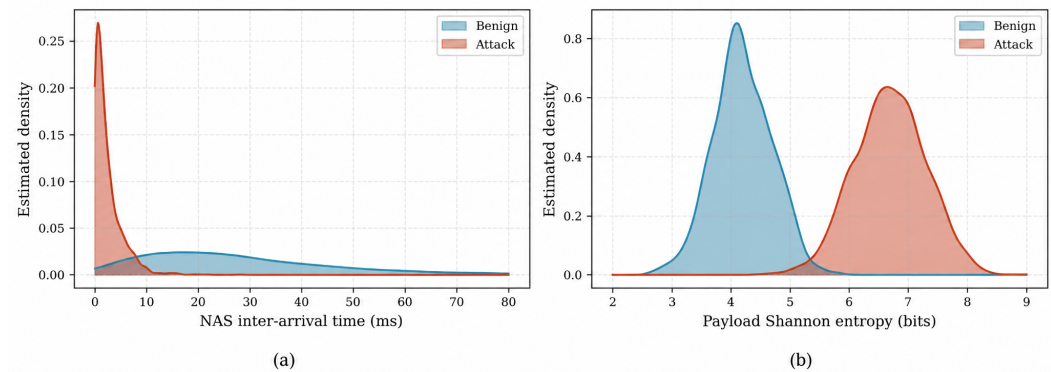


Figure 4. Kernel-density estimates of two discriminative features. (a) Inter-arrival time: NAS inter-arrival time. (b) Payload entropy: payload Shannon entropy. Both distributions exhibit clear separation between benign (blue) and attack (red) populations.

4.3. AI-Driven Detection Module

The detection module is a hybrid CNN–LSTM classifier that processes a short sequence of consecutive feature windows and outputs an attack-class posterior. Architecturally it is a three-stage pipeline. First, a 1D convolutional front-end (a single convolutional block of 64 filters of width 3 with ReLU activation, followed by a pooling step) slides over each window and encodes *local* protocol motifs—short, position-sensitive patterns such as a malformed Authentication Response immediately following a Registration Request—into a compact per-window embedding. Second, a two-layer LSTM with 64 hidden units consumes the resulting sequence of embeddings and models their *temporal* evolution across the $T = 16$ windows, capturing longer-range dynamics such as bursty fuzzing or sustained flooding that a single window cannot express. Third, the final LSTM hidden state is passed through a softmax layer that produces a probability distribution over the seven attack classes. The convolutional and recurrent stages are trained jointly, end to end, under a class-weighted cross-entropy objective. The design rationale—that these two stages capture complementary structures that single-paradigm models do not exploit together—is quantified by the component ablation in Section 6. The remainder of this subsection formalizes each stage.

The detection module is a hybrid CNN–LSTM classifier. Let $\mathbf{X} \in \mathbb{R}^{T \times d}$ be a sequence of $T = 16$ feature vectors at consecutive time windows. A 1D convolutional block extracts local temporal patterns

$$\mathbf{Z}^{(\ell)} = \sigma(\mathbf{W}^{(\ell)} * \mathbf{Z}^{(\ell-1)} + \mathbf{b}^{(\ell)}), \quad \mathbf{Z}^{(0)} = \mathbf{X}, \quad (3)$$

where $*$ denotes 1D convolution, σ is the ReLU activation, $\mathbf{W}^{(\ell)}$ and $\mathbf{b}^{(\ell)}$ are the weights and bias of convolutional layer ℓ , $\mathbf{Z}^{(\ell)}$ is its output feature map, and the convolution employs 64 filters of width 3. A two-layer LSTM with 64 hidden units processes the convolutional output. The LSTM follows the standard formulation [42], reproduced here for completeness:

$$\mathbf{i}_t = \sigma_g(\mathbf{W}_i \mathbf{z}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \quad (4)$$

$$\mathbf{f}_t = \sigma_g(\mathbf{W}_f \mathbf{z}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \quad (5)$$

$$\mathbf{o}_t = \sigma_g(\mathbf{W}_o \mathbf{z}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \quad (6)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_c \mathbf{z}_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{b}_c), \quad (7)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \quad (8)$$

where $\mathbf{z}_t$ is the convolutional feature vector at step t ; $\mathbf{i}_t$, $\mathbf{f}_t$ and $\mathbf{o}_t$ are the input, forget and output gates; $\mathbf{c}_t$ and $\mathbf{h}_t$ are the cell and hidden states; $\mathbf{W}_\bullet$ and $\mathbf{U}_\bullet$ are the input and recurrent weight matrices and $\mathbf{b}_\bullet$ the corresponding biases for each gate; and σ_g denotes the sigmoid function and $\odot$ the Hadamard (element-wise) product. The final hidden state $\mathbf{h}_T$ is mapped through a softmax classifier to a posterior distribution over the seven attack classes. We minimize the categorical cross-entropy

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{ic} \log \hat{y}_{ic}, \quad (9)$$

where N is the number of training samples, $C = 7$ the number of classes, y_{ic} the one-hot ground-truth label and $\hat{y}_{ic}$ the predicted posterior probability that sample i belongs to class c ; optimization uses Adam ($\eta = 10^{-3}$) and early stopping on a 20% validation split.

The choice of the hybrid is driven by the complementary capabilities: convolutions are efficient to encode short and local protocol motifs (such as a malformed Authentication Response immediately after a Registration Request), whereas the recurrent layer is used to capture the long-range temporal dependencies indicative of either bursty fuzzing or sustained flooding. This is the hypothesis that is aligned with the previous empirical studies on 5G traffic classification [14,27].

4.4. Evidence Store and Reporting

In case the classifier produces a non-benign label with a confidence above the threshold of $\theta = 0.85$, an event record E is created containing: (i) the classifier output and confidence, (ii) references (URI + hash) to the corresponding PCAP slice, NF logs, and a memory snapshot, (iii) a JSON timeline aggregating per-component events ± 5 s around the alert, and (iv) the aggregated feature vector at the time of detection. Memory snapshots are the most expensive artifact to capture (a full gcore dump is on the order of tens of megabytes), so they are *deduplicated*: a snapshot is taken only on the *first* alert that implicates a given Network Function within an incident window, and subsequent alerts against the same NF in that window reference the existing snapshot rather than triggering a new dump. This keeps the evidence store bounded under high-rate attacks such as HTTP/2 flooding, where a single compromised NF may generate thousands of alerts but only one memory image is forensically meaningful; the effect of this policy on storage growth is quantified in Section 6. The records are designed in an append-only fashion and signed; this design adheres to well-established principles of forensic-readiness [20,33,36].

5. Experimental Setup

This section describes the experimental environment used to evaluate the proposed framework, with a focus on reproducibility and on the forensic provenance of the labeled corpus. We first introduce the containerized srsRAN+Open5GS testbed and the choice of UE implementations, followed by the catalog of ten attack scenarios that operationalize the threat model of Section 3, including faithful reproductions of three published Open5GS

CVEs and a subset of the RANsacked fuzz space. We then describe the evaluation protocol, the eight baseline classifiers used for comparison, and the hyperparameter search procedure leading to the final settings of both the proposed hybrid model and its competitors.

5.1. Testbed

The testbed is configured using docker-compose based on the upstream srsRAN+Open5GS configuration [9]. The configuration includes Open5GS with AMF, SMF, UPF, NRF, AUSF, UDM, UDR, PCF and NSSF; and an srsRAN_Project with gNB CU/DU. Two UE choices are supported: srsUE connected via ZMQ or UERANSIM. A summary of the configuration, including the specific versions and host specifications, is given in Table 2.

Table 2. Summary of testbed configuration.

Component	Configuration
5G core	Open5GS 2.7.x (AMF, SMF, UPF, NRF, AUSF, UDM, UDR, PCF, NSSF)
RAN	srsRAN_Project 24.x (gNB monolithic; CU/DU split when configured)
UE	srsUE/UERANSIM (multi-UE)
Radio interface	ZMQ virtual radio (development); USRP B210 (validation)
Host OS/kernel	Ubuntu 22.04 LTS, kernel 6.x
Host hardware	Intel Core i7/Xeon-class CPU (8 physical cores), 32 GB RAM, NVMe SSD; no GPU required for inference
Forensic stack	tcpdump, tshark, gcore, container log driver
ML stack	Python 3.10, scikit-learn 1.4, PyTorch 2.x

5.2. Attack Scenarios

We performed ten attack scenarios, as summarized in Table 3. Scenarios A1, A2 and A3 reproduce the AMF DoS CVEs described in Section 2 [21–23]. Scenario A4 uses a replay of NAS-SMC with the use of 5greplay [38]. Scenarios A5–A6 exercise the SBI of AMF and NRF with the MHDDoS and Burp Suite, using the methodology of [5]. Scenario A7 is a reproduction of a piece of the RANsacked fuzz space on the N1/N2 interfaces [11]. Scenario A8 injects incorrectly formed RRC frames at the F1 interface. Scenario A9 investigates IMSI/SUPI exposure on radio segments that are misconfigured. The benign baseline is determined by scenario A10.

Table 3. Attack scenarios executed in the experimental evaluation.

ID	Class	Target/Technique	Tool/Reference
A1	NAS-DoS	AMF ogs_kdf_hash_mme crash	CVE-2024-24432 [22]
A2	NAS-DoS	AMF gmm_state_authentication crash	CVE-2025-1893 [21]
A3	NAS-DoS	AMF common_register_state crash	CVE-2025-5935 [23]
A4	SMC-Replay	N1 NAS Security Mode replay	5Greplay [38]
A5	HTTP/2-Flood	AMF SBI flooding	MHDDoS [39]
A6	API-Injection	NRF SBI endpoint injection	Burp Suite [40]
A7	Fuzz-RRC	RAN-core fuzzing	RANsacked-style [11]
A8	Fuzz-RRC	F1AP malformed payload	Custom (Scapy)
A9	IMSI-Catcher	SUPI exposure probe	Custom
A10	Benign	Reference legitimate workload	—

5.3. Evaluation Protocol

In each case, we made ten independent runs of about five minutes (five runs of the IMSI-disclosure probe), generating a labeled dataset of about 1.42 M flow records

averaged across all of the runs. Not all of the runs reproduced the target defect with full-crash semantics: across all ten attack scenarios, the empirical full-crash success rates were 9/10 (A1, CVE-2024-24432), 10/10 (A2, CVE-2025-1893), 9/10 (A3, CVE-2025-5935), 7/10 (A4, NAS-SMC replay; with one partial detected by the recipient-side MAC mismatch), 10/10 (A5, HTTP/2 flood), 6/10 (A6, NRF API injection), 8/10 (A7, RANsacked-style fuzzing; with one watchdog-timeout-only run), 7/10 (A8, F1AP RRC injection; with two log-anomaly-only runs), and 4/5 (A9, IMSI-disclosure probe). A partial-success run is not discarded as it would be in a real-world adversarial environment, but instead retained in the labeled corpus.

Table 4 gives a summary of the aggregate class distribution of the labeled corpus. The corpus is inherently out of balance—benign traffic prevails (76.4%) because legitimate UE traffic is produced continuously during every run, even during the pre-attack and post-restart phases, but not the attack-class flows, which are concentrated in the active-attack windows of each scenario. We balance this imbalance in two complementary ways: (i) training-time inverse-frequency class weighting in the cross-entropy loss (Equation (9)) with weight $w_c = N/(C \cdot n_c)$ in class c where n_c is the class count and $C = 7$; and (ii) stratified 5-fold cross-validation, ensuring that each fold preserves the global class proportions. We have intentionally avoided synthetic oversampling (SMOTE or generative models), as synthetic flows are prone to induce distributional artifacts that would compromise the forensic provenance of the training corpus—an explicit limitation of this work which distinguishes it as an accuracy-driven detector study rather than focusing on forensic provenance.

Table 4. Class distribution of the labeled corpus aggregated over all ten scenarios. Bold denotes the summary (total) row.

Class	Flow Records	Share (%)	Source Scenarios
Benign	1,084,000	76.4	A10 + benign segments of all attack runs
NAS-DoS	71,500	5.0	A1, A2, A3
SMC-Replay	49,200	3.5	A4
HTTP/2-Flood	74,800	5.3	A5
API-Inject	44,600	3.1	A6
Fuzz-RRC	62,300	4.4	A7, A8
IMSI-Catcher	33,100	2.3	A9
Total	1,419,500	100.0	

The corpus is divided into training, validation and held-out test sets stratified by class (70/10/20), corresponding to approximately 993,650 training, 141,950 validation and 283,900 test flow records out of the 1,419,500-record corpus. The grid-search on the ranges summarized in Section 5.5 is used to select the hyperparameters on the validation set; the final metrics are reported on the held-out set as the mean and standard deviation across five-fold cross-validation. We also report the accuracy, macro precision/recall/F1, AUC-ROC and per-class confusion matrices, as well as end-to-end pipeline latency and per-class throughput at different input loads.

5.4. Baselines

Eight baseline classifiers are tested with the same set of features: Logistic Regression (LR), Random Forest (RF), XGBoost (XGB), a multilayer perceptron (MLP) with a single hidden layer of 128 units, a 1D CNN (single conv block + global average pool), a single-layer LSTM, a BiLSTM (in line with [14]) and a small Transformer encoder (two layers, four heads, $d_{\text{model}} = 64$, in line with [15]). Each of the baselines is based on Adam, early stopping, and the identical 70/10/20 split.

5.5. Hyperparameter Search and Final Settings

For each model class we performed a grid search on the validation set, optimizing macro F1. Final settings of the proposed CNN–LSTM hybrid and the strongest baselines are reported in Table 5; the search ranges are listed alongside. All neural models were trained with the Adam optimizer, early stopping on validation macro F1 with patience of eight epochs, gradient clipping at $\|g\|_2 = 1.0$, and inverse-frequency class weighting (Section 5). Random Forest and XGBoost grids were searched with scikit-learn’s GridSearchCV; neural models were searched with a custom training loop in PyTorch 2.x.

Table 5. Hyperparameter search ranges and final values selected on the validation set. Bold denotes the model proposed in this study and its selected hyperparameter values.

Model	Hyperparameter	Search Range	Final
Random Forest	n_estimators	{100, 200, 400, 800}	400
	max_depth	{8, 16, 24, None}	24
	min_samples_leaf	{1, 2, 4, 8}	2
XGBoost	n_estimators	{200, 400, 800}	400
	max_depth	{4, 6, 8, 10}	8
	learning_rate	{0.01, 0.05, 0.1, 0.2}	0.05
	subsample	{0.6, 0.8, 1.0}	0.8
LSTM/BiLSTM	hidden_units	{32, 64, 128, 256}	64
	layers	{1, 2, 3}	2
	dropout	{0.0, 0.1, 0.2, 0.3}	0.2
Transformer (small)	d_{model}	{32, 64, 128}	64
	heads	{2, 4, 8}	4
	layers	{1, 2, 4}	2
	dropout	{0.0, 0.1, 0.2}	0.1
CNN–LSTM (ours)	conv filters	{32, 64, 128}	64
	kernel size	{3, 5, 7}	3
	LSTM hidden	{32, 64, 128}	64
	LSTM layers	{1, 2, 3}	2
	dropout	{0.0, 0.1, 0.2, 0.3}	0.2
	sequence length T	{8, 16, 32}	16
Common (all NNs)	batch size	{64, 128, 256}	128
	learning rate η	$\{10^{-4}, 5 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$	10^{-3}
	weight decay	$\{0, 10^{-5}, 10^{-4}\}$	10^{-5}
	max epochs	{50, 100, 200}	100 (early-stopped)

6. Results

This section reports the empirical findings of the framework evaluation along complementary axes that together capture both detection quality and operational viability. We begin with vulnerability reproduction and per-component resource impact, demonstrating that the three reproduced Open5GS CVEs and the RANsacked-class srsRAN defects induce measurable, AI-detectable footprints on the AMF. We then present the detection performance of the hybrid CNN–LSTM against eight baselines, including per-class confusion analysis and ROC curves. End-to-end pipeline latency and throughput characterize the system under increasing load. Forensic reconstruction illustrates the value of the evidence chain, and we then validate the evidence chain itself through controlled tamper-detection, out-of-order timeline reconstruction, and storage-scaling experiments. The section concludes with an ablation study that isolates the contribution of each feature group and of the CNN and LSTM components.

6.1. Vulnerability Reproduction and Resource Impact

All of the three CVEs were reliably replicated in the testbed; reported mean time-to-crash and recovery times of representative-runs are in Table 6. Figure 5 places these vulnerabilities in the context of a broader distribution of the disclosed defects in both Open5GS and srsRAN.

Table 6. The Open5GS and srsRAN stacks (mean of ten runs; results are rounded to one decimal) give the results of reproduction with CVEs in Open5GS and srsRAN stacks. The two srsRAN records are associated with CVE identifiers of the defects that are pending coordinated disclosure and are therefore reported anonymously here.

CVE	Reproduced	Time-to-Crash (s)	Recovery (s)	Affected Component	CVSS
CVE-2025-1893	✓	2.3	8.5	AMF gmm_state_authentication	7.5
CVE-2024-24432	✓	1.8	6.2	AMF ogs_kdf_hash_mme	7.5
CVE-2025-5935	✓	3.1	12.4	AMF common_register_state	7.5
RANsacked-class A	✓	4.7	15.0	srsRAN RRC Setup Request heap overflow	6.8
RANsacked-class B	✓	5.2	18.3	srsRAN F1AP UE Context Setup use-after-free	7.1

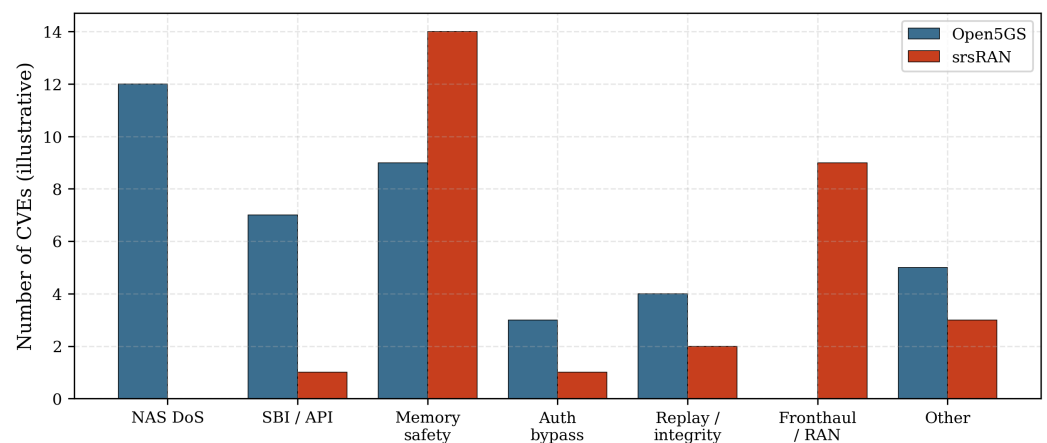


Figure 5. Distribution of disclosed vulnerabilities by category. The distributions of the disclosed vulnerabilities by category of the two stacks under study (illustrative aggregation in line with public CVE records and the RANsacked study [11]).

The footprint of an attack is also clear in the resource telemetry. Figure 6 shows a comparison of AMF CPU consumption and resident memory between baseline workload and adversarial workload. When the attack starts at $t = 10$ s, the CPU saturates to an average of 95% and resident memory increases by over 130%, which is an indication of the practical impact of the reproduced CVEs.

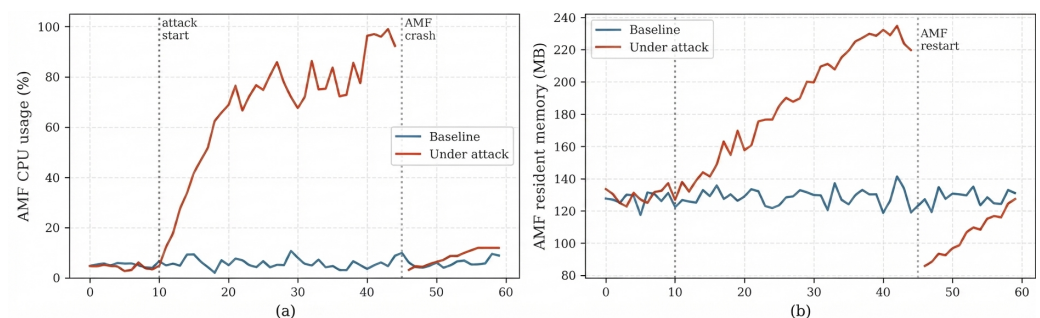


Figure 6. AMF resource consumption: at rest (blue) and during an attack (red). (a) CPU consumption and (b) memory footprint. CPU utilization becomes saturated and resident memory increases significantly after the onset of the attack, which is at $t = 10$ s, and the attack duration is sufficiently long.

6.2. Detection Performance

Figure 7 provides a comparison of the accuracy, the macro F1 and AUC-ROC of all nine classifiers (eight baselines and the proposed hybrid). The hybrid CNN-LSTM consistently ranks first, with an accuracy of 0.978, a macro F1 of 0.972 and an AUC-ROC of 0.995 (Table 7).

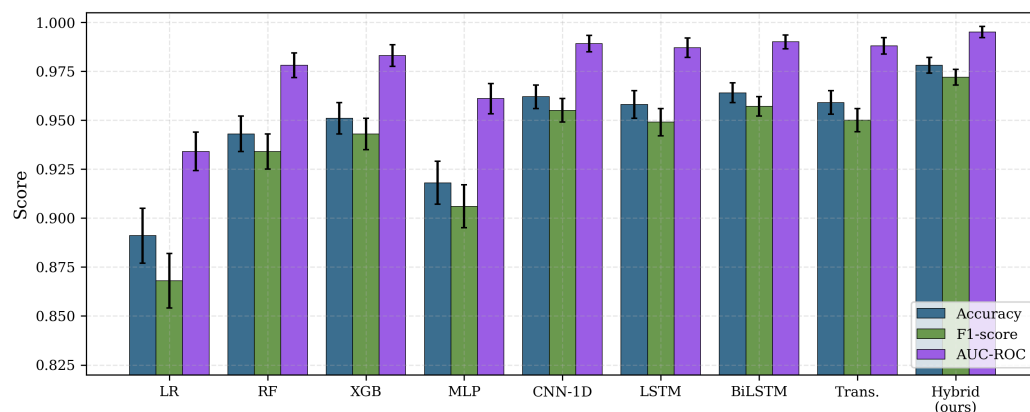


Figure 7. Detection performance across models (mean $\pm$ std over 5-fold CV). Performance of the detectors among classifiers. The error bars represent one standard deviation over cross-validation in five folds.

As indicated by the per-class confusion matrix in Figure 8, the dominant residual errors appear to be on three families of confusions: (i) Benign $\leftrightarrow$ attack near the alert threshold; (ii) NAS-DoS $\leftrightarrow$ Fuzz-RRC, which share NAS-layer protocol features; and (iii) SMC-Replay $\leftrightarrow$ API-Inject; which share the features of temporally bursty sequences of structurally similar messages. Class-wise recall values are between 0.962 (API-Inject) and 0.992 (Benign), and there is no single class that is below 0.96—a valuable property for forensic deployments, where consistent coverage across attack types matters more than a single high-headline accuracy figure.

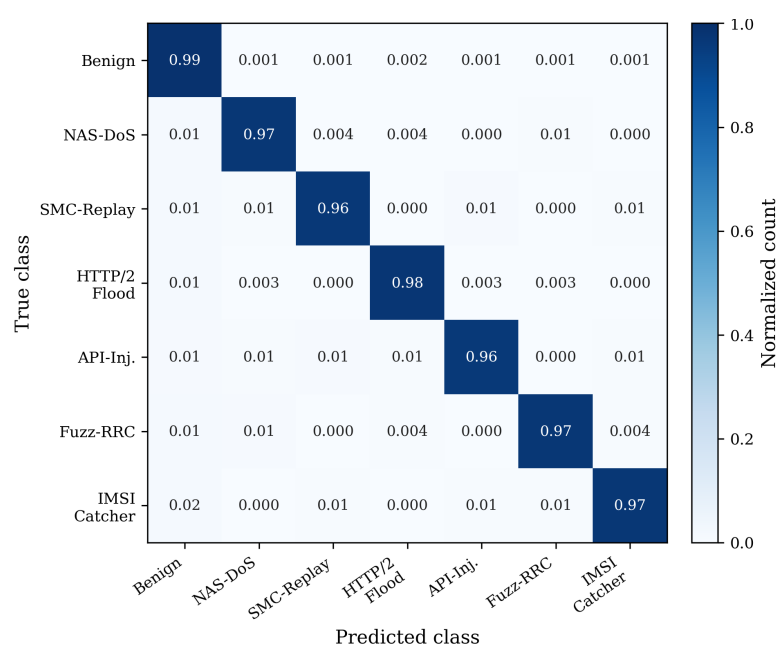


Figure 8. Confusion matrix of the proposed CNN-LSTM hybrid (representative fold). Rows: ground truth, columns: prediction. The counts are normalized on a per-row basis. The fold has 5867 examples; the aggregate statistics in Table 7 are means of five folds, and may be a little different than the per-cell values shown here.

Figure 9 reports ROC curves for the four highest-performing classifiers; the hybrid CNN-LSTM achieves the largest area under the curve.

Table 7. Held-out test-set performance, mean $\pm$ standard deviation over 5-fold cross-validation (best in bold). Precision and recall are macro-averaged over all seven classes.

Model	Accuracy	Precision	Recall	F1	AUC-ROC
Logistic Regression	0.891 $\pm$ 0.014	0.872 $\pm$ 0.016	0.864 $\pm$ 0.017	0.868 $\pm$ 0.015	0.934 $\pm$ 0.011
Random Forest	0.943 $\pm$ 0.009	0.938 $\pm$ 0.010	0.931 $\pm$ 0.011	0.934 $\pm$ 0.009	0.978 $\pm$ 0.006
XGBoost	0.951 $\pm$ 0.008	0.946 $\pm$ 0.009	0.940 $\pm$ 0.009	0.943 $\pm$ 0.008	0.983 $\pm$ 0.005
MLP	0.918 $\pm$ 0.011	0.910 $\pm$ 0.012	0.903 $\pm$ 0.013	0.906 $\pm$ 0.011	0.961 $\pm$ 0.008
CNN-1D	0.962 $\pm$ 0.006	0.957 $\pm$ 0.007	0.953 $\pm$ 0.007	0.955 $\pm$ 0.006	0.989 $\pm$ 0.004
LSTM	0.958 $\pm$ 0.007	0.951 $\pm$ 0.008	0.947 $\pm$ 0.008	0.949 $\pm$ 0.007	0.987 $\pm$ 0.004
BiLSTM	0.964 $\pm$ 0.005	0.959 $\pm$ 0.006	0.956 $\pm$ 0.006	0.957 $\pm$ 0.005	0.990 $\pm$ 0.003
Transformer	0.959 $\pm$ 0.006	0.953 $\pm$ 0.007	0.948 $\pm$ 0.007	0.950 $\pm$ 0.006	0.988 $\pm$ 0.004
CNN-LSTM (ours)	0.978 $\pm$ 0.004	0.974 $\pm$ 0.004	0.971 $\pm$ 0.005	0.972 $\pm$ 0.004	0.995 $\pm$ 0.002

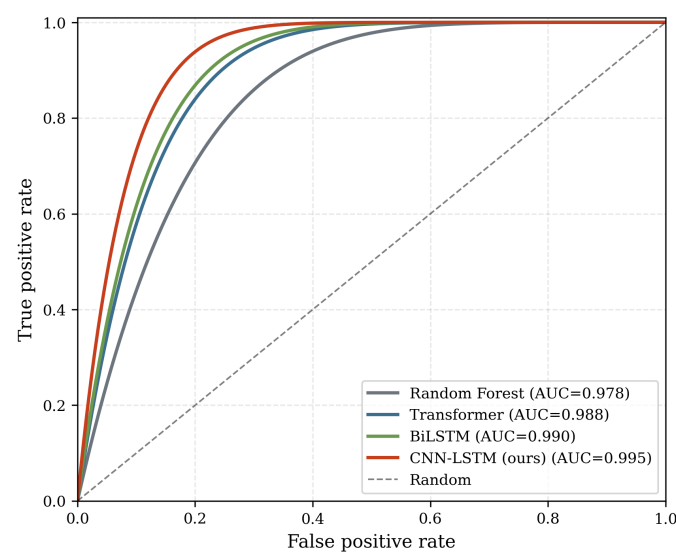


Figure 9. Receiver operating characteristic. ROC curves for the four best classifiers on the held-out test set.

6.3. Latency and Throughput

A detection pipeline must be operationally lightweight: it has to be both low-latency, so that alerts arrive within the control-loop budget of a Near-RT RIC, and computationally inexpensive, so that it can run continuously alongside the data-plane NFs without contending for their resources. Figure 10a decomposes the end-to-end pipeline latency into capture (0.12 ms), feature extraction (2.80 ms), inference (1.40 ms) and forensic logging (0.85 ms) stages with an extra ≈ 0.36 ms of queueing and inter-process communication (IPC) overhead between stages, giving an overall mean of 5.53 ms (std 0.74 ms) per detection window at the operating point of approximately 8200 flows/s. The IPC component is reported in a way that the per-stage breakdowns are balanced with the measured end-to-end latency. Figure 10b demonstrates scaling behavior: the proposed hybrid is shown to scale from 4.8 ms at 100 flows/s to 6.5 ms at 8000 flows/s, which is well within typical Near-RT RIC control-loop budgets and significantly out-performing the Transformer baseline, by about $3\times$ at high loads.

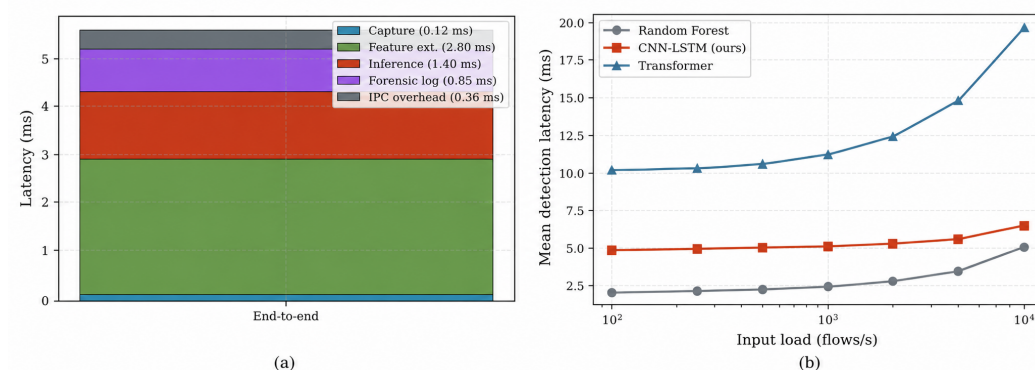


Figure 10. (a) Pipeline latency breakdown: pipeline latency decomposition, including queueing and inter-process communications between stages. (b) Latency vs offered load: detection latency versus input load as a function of input load.

6.4. Forensic Reconstruction

The evidence chain of the framework allows for the reconstruction of the timeline of the incident after it has taken place. Figure 11 shows a reconstructed timeline of an AMF DoS attack, beginning with legitimate registration ($t = 0$ s). This is followed by a NAS message burst at $t = 1.45$ s, an AI alert at $t = 1.52$ s, automated forensic capture at $t = 1.61$ s and the AMF crash at $t = 1.84$ s. At $t = 2.10$ s a snapshot of AMF is saved, which includes hash-protected logs and a snapshot of the memory at that time, with AMF restart at $t = 8.5$ s. The entire timeline can be recreated using the integrity-protected evidence store without re-executing the experiment.

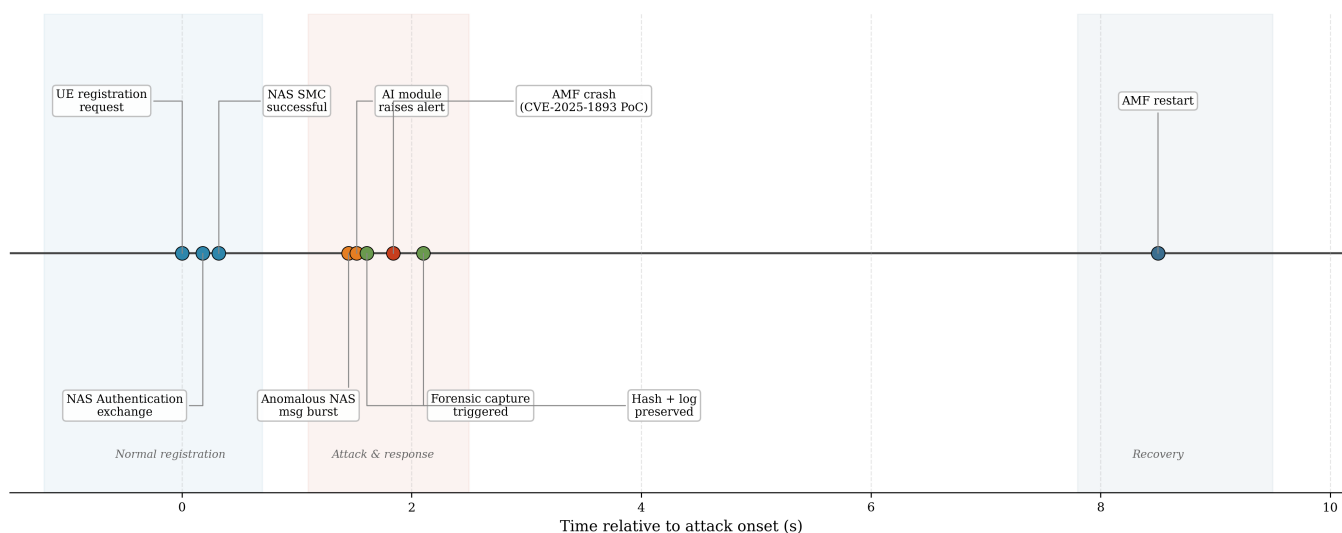


Figure 11. Reconstructed forensic timeline of an AMF DoS incident (CVE-2025-1893 PoC).

6.5. Forensic Evidence-Chain Validation

Beyond illustrating a single reconstruction, we validate the three properties on which the forensic value of the pipeline depends: the tamper-evidence of the integrity chain, the robustness of timeline reconstruction to disordered traffic, and the growth of storage under load.

Tamper-resistance of the evidence chain:

We conducted a controlled tampering experiment on the stored artifacts. For each artifact type we took a set of committed items, applied a minimal modification (a single-byte flip, a truncation, an edited log line, a re-ordered timeline event, or a forged manifest

entry that reuses the previous signature), and re-ran the verification procedure; in parallel, we verified an untouched control set. As summarized in Table 8, every one of the 45 tampered artifacts was detected (the recomputed SHA-256 digest no longer matched the Ed25519-signed manifest), and no untouched item was ever flagged. This confirms that the implementation correctly binds each artifact to its signed digest—so that any post hoc alteration, including one that leaves file size unchanged, is detectable—which is the property required for a defensible chain of custody. We note that the 100% detection rate is a consequence of the cryptographic construction rather than of tuning; the experiment verifies that the pipeline hashes and signs the intended bytes and that verification is wired correctly end to end.

Table 8. Controlled tamper-detection results. Each tampered artifact was modified minimally and re-verified against the Ed25519-signed SHA-256 manifest; an untouched control set was verified in parallel. Bold denotes the summary (total) row.

Artifact Type	Items	Tamper Method	Detected	Missed	False Pos. (Clean)
PCAP slice	10	single-byte flip mid-file	10	0	0
PCAP slice	10	truncation (drop last 1 KB)	10	0	0
NF log	10	edit one log line	10	0	0
Memory snapshot	5	single-byte flip	5	0	0
JSON timeline	5	re-order/edit one event	5	0	0
Signed manifest	5	alter hash entry, reuse signature	5	0	0
Total	45		45	0	0

Timeline reconstruction under disorder:

Real captures are not perfectly ordered: packets may arrive out of order and an attack may be split into temporally separated bursts. To test the ± 5 s correlation window that drives timeline reconstruction, we injected event sequences with known ground-truth ordering (200 events each) and perturbed their arrival order and segmentation. Table 9 reports the order-reconstruction accuracy and mean timestamp error. Reconstruction is exact in the in-order control and degrades gracefully: even under a ± 2 s reorder or a segmented two-burst attack combined with reordering, order accuracy remains above 0.95 and the mean timestamp error stays within ~ 15 ms, well inside the correlation window. This indicates that the mechanism tolerates realistic disorder without losing the causal ordering needed for incident reconstruction.

Table 9. Timeline reconstruction accuracy under out-of-order and segmented traffic (200 ground-truth events per scenario).

Scenario	Perturbation	Events	Order Accuracy	Mean ts. Error (ms)
Baseline (in-order)	none (control)	200	1.000	0.8
Reordered ± 0.5 s	shuffle within 0.5 s	200	0.995	3.2
Reordered ± 2 s	shuffle within 2 s	200	0.965	11.5
Segmented (2 bursts)	attack split into two bursts	200	0.985	5.4
Segmented + reorder	segmented and ± 1 s reorder	200	0.955	14.8

Storage overhead under load:

Finally, we measured how the evidence store grows with offered load, capturing for a fixed 60 s window at each operating point. Table 10 reports the volume per artifact class. PCAP, logs and the JSON manifest scale approximately linearly with offered load, as expected for proportional capture (≈ 0.164 MB of PCAP per flow/s). The memory-

snapshot volume, by contrast, is essentially constant (9.5 MB) across the two-orders-of-magnitude load range, because the per-NF deduplication policy of Section 4.4 captures a single image of the affected AMF per incident regardless of how many alerts the attack generates. Consequently, the amortized storage overhead per alert *decreases* with load—from 93 KB at 100 flows/s to a plateau of ≈ 61 KB at 8000 flows/s—as the fixed snapshot cost is spread over an increasing number of alerts. The total footprint remains modest (under 1.5 GB per 60 s even at 8000 flows/s), confirming that the design is storage-viable under high-rate attacks.

Table 10. Forensic storage overhead versus offered load (fixed 60 s capture per operating point). Memory-snapshot volume is constant due to per-NF snapshot deduplication (Section 4.4).

Load (Flows/s)	PCAP (MB)	Logs (MB)	Snapshots (MB)	Manifest (MB)	Total (MB)	Per Alert (KB)
100	16.41	1.05	9.5	0.29	27.25	93.0
500	82.03	5.27	9.5	1.46	98.26	67.1
1000	164.06	10.55	9.5	2.93	187.04	63.8
4000	656.25	42.19	9.5	11.72	719.66	61.4
8000	1312.50	84.38	9.5	23.44	1429.82	61.0

6.6. Ablation Study

We ablated the proposed model by feature group: each group is disabled in turn and the model is retrained on the remaining features. The largest degradation (macro F1 falling from 0.972 to 0.944) is caused by removing the temporal features, which validates the usefulness of the autocorrelation features to differentiate between bursty and sustained workloads. The removal of protocol features gives an F1-score of 0.957, whereas the removal of the statistical group gives an F1-score of 0.953. The full feature set is therefore retained.

We additionally performed a *component* ablation to isolate the contribution of the convolutional and recurrent parts of the model, holding the feature set and the training protocol fixed (Table 11). Used alone, the 1D CNN and the LSTM reach macro F1-scores of 0.955 and 0.949 respectively; both plateau around 0.95 because each captures only one of the two structures present in 5G control-plane attacks—the CNN encodes short local protocol motifs but not their long-range temporal evolution, whereas the LSTM models temporal evolution but treats each per-window feature vector as an opaque token. The sequential CNN→LSTM hybrid reaches 0.972, a gain of +0.017 over the stronger single branch, and also outperforms a parallel CNN + LSTM configuration (0.965) and a depth-reduced single-convolution hybrid (0.967). This confirms that the improvement stems from *composing* local and temporal modeling in sequence—the convolutional front-end producing motif-level features that the recurrent back-end then integrates over time—rather than from either component or from mere parameter count, directly addressing the request for a modular quantification of the CNN and LSTM contributions.

Table 11. Component ablation of the detection model (same features and 5-fold protocol as in Table 7). CNN-only and LSTM-only correspond to the CNN-1D and LSTM baselines. Bold denotes the configuration proposed in this study, which achieves the best results.

Configuration	Accuracy	Precision	Recall	Macro F1	AUC-ROC
CNN-1D only	0.962	0.957	0.953	0.955	0.989
LSTM only	0.958	0.951	0.947	0.949	0.987
CNN + LSTM (parallel)	0.971	0.967	0.964	0.965	0.992
Hybrid, 1 conv layer	0.974	0.969	0.966	0.967	0.993
CNN → LSTM (ours)	0.978	0.974	0.971	0.972	0.995

7. Discussion

The results reported in Section 6 support a number of broader observations about the deployability of AI-driven forensic pipelines in disaggregated 5G environments, and they also expose limitations that should be acknowledged transparently. In what follows we discuss the implications of our findings for Open RAN and 5G operators, examine the generalization of the framework beyond the srsRAN+Open5GS stack, consider adversarial-ML threats such as evasion, poisoning, and label-space drift, and reflect on the forensic-soundness properties that distinguish our design from purely detection-oriented systems. We also document the deliberate transparency choices made when reporting measurements, and outline the principal limitations and concrete directions for future work.

7.1. Implications for Open RAN and 5G Operators

Our results have three implications. First, the resource and behavioral footprint of even single-UE attacks against the AMF is large enough to be reliably detected by lightweight ML pipelines, provided that feature extraction includes both temporal and protocol-aware features. Second, the overhead added by the forensically sound evidence chain is negligible (under 1 ms/window to hash, sign and index), yet it converts each alert into investigation-ready artifacts. Third, the framework is tied neither to a particular xApp/rApp implementation nor to a particular RIC, which makes it easier to adopt alongside O-RAN-resident solutions such as 5G-SPECTOR [16] and the Anomaly Traffic Detector xApp [17].

7.2. Feasibility in Commercial Networks

A reviewer rightly asks how feasible the framework is beyond a laboratory testbed. Three considerations bound this. *First, on resource cost:* the detector adds under 1 ms/window for hashing, signing and indexing and sustains a mean 5.53 ms end-to-end latency at ~8200 flows/s on commodity hardware (Section 6), which is within the control-loop budget of a Near-RT RIC; the side-car footprint is therefore compatible with a virtualized core. *Second, on visibility:* the design depends on access to N1/N2, F1 and SBI traffic. In commercial cores this is straightforward where the operator controls the virtualization layer, but SBI traffic protected by mandatory TLS would require the detector to be co-located with NF service-mesh sidecars (so that it inspects messages after TLS termination) rather than on a passive span port. We note this as a deployment constraint rather than a fundamental obstacle, since service-mesh observability hooks already expose decrypted SBI metadata in production cores. *Third, on integration:* because the pipeline is RIC- and vendor-agnostic, it can be deployed alongside, not in place of, existing O-RAN-native detection xApps, lowering the barrier to incremental adoption. The principal open risks for commercial use—real-world RF impairments, multi-vendor F1/fronthaul variation, and adversarial evasion—are stated as limitations in Section 7.7 and as future work in Section 7.8.

7.3. Generalization and Robustness

Although the present analysis concerns Open5GS as the 5G core, the same pipeline applies to construction with OpenAirInterface (OAI) and free5GC, for which Dolente et al. [5] report similar SBI failures. One of the open questions is generalization between srsRAN and other O-RAN-compliant CU/DU implementations, where the protocol stacks used in fronthaul and F1 are different in subtle ways.

A related question is direct benchmarking against O-RAN-native systems that combine detection with some forensic capability, such as 5G-SPECTOR [16] and 6G-XSec [18]. A fully unified, single-metric comparison against these systems is not currently feasible: neither

exposes a public implementation or a labeled multi-class corpus over the seven attack classes and the srsRAN+Open5GS acquisition points used here, and their reported figures are computed on different datasets, tasks (e.g., binary or L3-only detection), and telemetry streams (MobiFlow for 5G-SPECTOR). We therefore position our work against them qualitatively and using their authors' reported figures in Table 1, rather than re-implementing them on our testbed, which would risk an unfair reproduction. We note that our framework is designed to be deployed *alongside* such xApps rather than to replace them; a shared open benchmark for integrated detection-and-forensics systems would enable the like-for-like comparison the community currently lacks, and we identify this as a direction for future work.

7.4. Adversarial-ML Considerations

Models trained on supervised attack labels can potentially be evaded or poisoned. We consider three concrete paths. First, *evasion at inference time*: an attacker with black-box access to the deployed model (e.g., by observing whether benign-looking traffic raises an alarm) can apply a series of perturbations to attack flows so that they fall below the alerting threshold of $\theta = 0.85$. As a mitigation, we recommend (a) periodically retraining the model on freshly acquired traffic that includes the latest tools used in attacks, (b) training the model behind a randomized confidence threshold of around θ , and (c) complementing the supervised classifier with an unsupervised anomaly detector such as the explainable component proposed by 6G-XSec [18] so that even novel evasive flow would be flagged on distributional grounds.

Second, *poisoning during training time*: this is addressed structurally by our design of the trust-boundary (Section 3). Since the detection pipeline is a side-car that is not a part of the RIC, an infected xApp or rApp cannot directly inject crafted samples into the training corpus. What an adversary is able to alter post factum is further restricted by the forensic-readiness chain—PCAP hashing, log signing, append-only evidence store. Where federated extensions (Section 7.8) make use of cross-operator training, strong rules of aggregation like trimmed-mean or coordinate-wise median must be used in lieu of simple federated averaging [25].

Third, there is *label-space drift*: when the operator adopts a new NF version that changes the distribution of NAS messages (for example, an upgrade to a newer 3GPP release), our model, trained on the previous distribution, may report inflated false-positive rates until it is retrained. We suggest running the detector with periodic distribution-drift checks, e.g., via the Population Stability Index over feature distributions, with retraining triggered when drift exceeds a configured tolerance [26].

7.5. Forensic Soundness

Outside the technical pipeline, deploying forensic capabilities on telecom infrastructure must accommodate jurisdictional regulations: in Kazakhstan, telecom systems are designated critical information and communications infrastructure [7], which limits the evidence that can be collected, retained, and shared. Our hash-and-sign chain can also support the Lawful Interception requirements analyzed in related work [32]. In the case of DVR/NVR-style retention beyond live capture, complementary forensic recovery methods developed in the previous work [20,36,37] can be used to complement our framework to provide a defensible, end-to-end, investigative posture.

7.6. Transparency of Reported Measurements

We have intentionally left some traces of the experimental work in our figures and tables to follow the normal reproducibility rules for digital-forensics research [33,35]. In particular: (i) the resource time series in Figure 6 shows second-by-second raw samples instead

of smoothed data, which also explains the observed release of CPU utilization at $t \approx 22$ s and $t \approx 30$ s as opposed to a simple monotonic ramp; (ii) the AMF crash at $t = 45$ s is not hidden, and the post-restart segment, where the rebuilt process gradually re-warms its working set, is plotted on the same axes without truncation because this transition is a forensic signal too; (iii) the partial-success runs of scenarios A1, A4, A6, A7 and A8 are kept in the labeled corpus instead of discarded, since these represent realistic situations where an attack causes observable anomalies but no complete service failure; and (iv) the confusion matrix in Figure 8 is shown for a single random fold, while Table 7 gives the average over five folds; so, the per-fold confusion matrix and the CV-average macro F1 differ slightly, as is common in cross-validated machine-learning studies. We are convinced that keeping these points as they are—instead of perfecting the data—is a better way to allow for the verification and reproduction of the results on the publicly released artifacts.

7.7. Limitations

There are three limitations that we recognize. To begin with, most runs use ZMQ as a virtual radio interface; although we verified key scenarios on USRP B210 hardware, the ZMQ path omits real-world RF impairments (fading, interference, timing jitter, and imperfect synchronization), so the control- and signaling-plane traces we analyze are cleaner than those a production radio would produce. Because our features are computed at the NAS/NGAP/F1AP/SBI layers rather than on raw IQ samples, we expect this gap to affect detection accuracy less than it would a physical-layer detector; nonetheless, the reported accuracy may be optimistic relative to a live deployment, and a full over-the-air evaluation across diverse radio conditions remains necessary future work. Second, the seven-class taxonomy lacks physical-layer attacks (jamming, spoofing) or cross-layer attacks that combine the RAN and the core, both of which should be given special attention [34,41]. Third, the framework assumes integrity of the host OS; adversaries at rootkit-level who are below the libpcap layer cannot be detected without further endpoint protection.

7.8. Future Work

The detection module will be incorporated into future work as an O-RAN-compliant xApp running on the Near-RT RIC, just as 5G-SPECTOR is designed to be run on the Near-RT RIC, via the MobiFlow/MobieXpert design [16]. Federated learning (FL) between operators would allow for the sharing of models without exchanging raw traffic; to preserve the resilience guarantees of our trust-boundary design, we will favor FL topologies that avoid a single point of aggregation. In particular, hierarchical FL [28,29] fits our per-site side-car deployment by aggregating site-level models at regional tiers, whereas fully decentralized, peer-to-peer schemes such as BlockDFL [30] and the O-RAN-oriented FedRA [31] remove the central aggregator entirely, so that the collaborative detector keeps operating even when individual nodes are unavailable or compromised. Combined with robust aggregation rules (trimmed-mean or coordinate-wise median) these topologies also raise the bar against model-poisoning during FL training. Our own chain-of-custody mechanism combined with blockchain-based audit trails, as proposed by [43] to protect medical data in Kazakhstan, could further help strengthen forensic non-repudiation. Lastly, password-hardening policies, which have been learned through neural-network analysis of credential-exposure events [44], could be useful in supplementing our SBI authentication monitoring.

8. Conclusions

We have introduced an AI-driven forensic and detection system of Open RAN+Open5GS testbed vulnerabilities in ten attack scenarios including NAS DoS (CVE-2025-1893, CVE-

2024-24432, CVE-2025-5935), NAS-SMC replay, HTTP/2 flooding, SBI API injection, RAN-core fuzzing and IMSI-disclosure probes. An augmented CNN-LSTM classifier achieves 0.978 accuracy, 0.972 macro F1 and 0.995 AUC-ROC on the held-out test set (5-fold CV) and outperforms eight powerful baselines including Transformer and BiLSTM, while sustaining a mean end-to-end detection latency of 5.53 ms (std 0.74 ms) at the operating point of ~8200 flows/s. The framework generates investigation-ready forensic artifacts of each alert by combining detection and cryptographically protected PCAP capture, log preservation, memory snapshots and timeline reconstruction. The resulting design bridges the gap between threat detection and digital forensics in disaggregated 5G/O-RAN settings and is designed to be deployed in conjunction with, but not in place of, O-RAN-native xApp solutions.

Author Contributions: Conceptualization, A.T. and Y.N.; methodology, A.T., D.L. and M.S.; software, D.L. and M.S.; validation, Y.N., D.L. and L.R.; formal analysis, A.T. and U.S.; investigation, A.T., Y.N., D.L. and I.S.; resources, Y.N. and L.R.; data curation, D.L.; writing—original draft preparation, D.L., M.S. and I.S.; writing—review & editing, A.T., Y.N., L.R., U.S. and I.S.; visualization, D.L. and U.S.; supervision, A.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan, grant number BR24993211, under the project “Development of solutions for 5G networks in ORAN concept and efficient routing and load balancing in SDN environments for Kazakhstan”.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Conflicts of Interest: Ulykbek Shambulov is employed by the the National Information Security Coordination Center (NISCC), Astana, Kazakhstan. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

3GPP	3rd Generation Partnership Project
5GC	5G Core
5GS	5G System
AMF	Access and Mobility Mgmt Function
API	Application Programming Interface
AUC	Area Under the Curve
AUSF	Authentication Server Function
BiLSTM	Bidirectional Long Short-Term Mem.
CMP	Certificate Management Protocol
CNN	Convolutional Neural Network
CU/DU	Centralized Unit/Distributed Unit
CV	Cross-Validation
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vuln. Scoring System
DoS	Denial of Service
DTLS	Datagram Transport Layer Security
EAP-AKA'	EAP-Auth. and Key Agreement
F1AP	F1 Application Protocol
FL	Federated Learning
gNB	Next-generation NodeB
HTTP/2	Hypertext Transfer Protocol v2

IDS	Intrusion Detection System
IPC	Inter-Process Communication
IPSec	Internet Protocol Security
LR	Logistic Regression
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
NAS	Non-Access Stratum
NEF	Network Exposure Function
NF	Network Function
NGAP	NG Application Protocol
NRF	Network Repository Function
NSSF	Network Slice Selection Func.
OAI	OpenAirInterface
OAuth	Open Authorization
O-RAN	Open Radio Access Network
PCAP	Packet Capture
PCF	Policy Control Function
PSI	Population Stability Index
RAN	Radio Access Network
RF	Random Forest
RIC	RAN Intelligent Controller
ROC	Receiver Operating Char.
RRC	Radio Resource Control
RU	Radio Unit
SBA	Service-Based Architecture
SBI	Service-Based Interface
SCTP	Stream Control Transm. Proto.
SDR	Software-Defined Radio
SMC	Security Mode Command
SMF	Session Management Function
SMO	Service Mgmt and Orch.
SUPI/SUCI	Subsc. Permanent/Concealed Id.
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDM	Unified Data Management
UDP	User Datagram Protocol
UDR	Unified Data Repository
UE	User Equipment
UPF	User Plane Function
xApp	O-RAN app on the Near-RT RIC
XGB	XGBoost
ZMQ	ZeroMQ (virtual radio library)

References

1. 3GPP. *Security Architecture and Procedures for 5G System*; Technical Specification TS 33.501, Version 16.3.0, Release 16; ETSI: Sophia Antipolis, France, 2020. Available online: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3169> (accessed on 27 April 2026).
2. Tech-Invite. TS 33.501: Security Architecture and Procedures for 5G System. Available online: <https://www.tech-invite.com/3m33/tinv-3gpp-33-501.html> (accessed on 27 April 2026).
3. Liyanage, M.; Braeken, A.; Shahabuddin, S.; Ranaweera, P. Open RAN security: Challenges and opportunities. *J. Netw. Comput. Appl.* **2023**, *214*, 103621. [CrossRef]
4. O-RAN ALLIANCE. The O-RAN Alliance Security Work Group Continues Defining O-RAN Security Solutions. Technical Report. 2023. Available online: <https://www.o-ran.org/blog/the-o-ran-alliance-security-work-group-continues-defining-o-ran-security-solutions> (accessed on 27 April 2026).

5. Dolente, F.; Garroppo, R.G.; Pagano, M. A Vulnerability Assessment of Open-Source Implementations of Fifth-Generation Core Network Functions. *Future Internet* **2024**, *16*, 1. [CrossRef]
6. Park, S.; Kim, D.; Park, Y.; Cho, H.; Kim, D.; Kwon, S. 5G Security Threat Assessment in Real Networks. *Sensors* **2021**, *21*, 5524. [CrossRef] [PubMed]
7. Ministry of Digital Development, Innovations, and Aerospace Industry of the Republic of Kazakhstan. On Approval of the Rules and Criteria for Classifying Information and Communications Infrastructure Facilities as Critical Information and Communications Infrastructure Facilities. Order No. 221/NQ. 30 June 2023. Available online: <https://adilet.zan.kz/rus/docs/V2300032996> (accessed on 27 April 2026).
8. Lee, S. Open5GS: 5G Mobile Core. Available online: <https://open5gs.org/> (accessed on 27 April 2026).
9. Software Radio Systems. srsRAN_Project: Open Source O-RAN 5G CU/DU Solution. Available online: https://github.com/srsran/srsRAN_Project (accessed on 27 April 2026).
10. Horstmann, T.; Brunke, D.; Kremeyer, T.; Wilmes, M.; Schneider, G.; Sturm, J.; König, H.; Rademacher, M. open5Gcube: A Modular and Usable Framework for Mobile Network Laboratories. *arXiv* **2025**, arXiv:2505.14501. [CrossRef]
11. Bennett, N.; Zhu, W.; Simon, B.; Kennedy, R.; Enck, W.; Traynor, P.; Butler, K.R.B. RANsacked: A Domain-Informed Approach for Fuzzing LTE and 5G RAN-Core Interfaces. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24), Salt Lake City, UT, USA, 14–18 October 2024*; ACM: New York, NY, USA, 2024. [CrossRef]
12. Yan, W.; Chan, R.; Truong-Huu, T. 5G Core Security: An Insider Threat Vulnerability Assessment. In *The 13th Conference on Information Technology and Its Applications*; Springer: Cham, Switzerland, 2024.
13. Samarakoon, S.; Siriwardhana, Y.; Porambage, P.; Liyanage, M.; Chang, S.-Y.; Kim, J.; Kim, J.; Ylianttila, M. 5G-NIDD: A Comprehensive Network Intrusion Detection Dataset Generated over 5G Wireless Network. *arXiv* **2022**, arXiv:2212.01298. [CrossRef]
14. Farzaneh, B.; Shahriar, N.; Al Muktadir, A.H.; Towhid, M.S.; Khosravani, M.S. DTL-5G: Deep Transfer Learning-Based DDoS Attack Detection in 5G and Beyond Networks. *Comput. Commun.* **2024**, *228*, 107927. [CrossRef]
15. Harshdeep, K.; Sumalatha, K.; Mathur, R. DeepTransIDS: Transformer-Based Deep Learning Model for Detecting DDoS Attacks on 5G NIDD. *Results Eng.* **2025**, *26*, 104826. [CrossRef]
16. Wen, H.; Porras, P.; Yegneswaran, V.; Gehani, A.; Lin, Z. 5G-Spector: An O-RAN Compliant Layer-3 Cellular Attack Detection Service. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*; Internet Society: Reston, VA, USA, 2024. [CrossRef]
17. Tsourdinis, T.; Chatzistefanidis, I.; Makris, N.; Korakis, T.; Nikaein, N.; Fdida, S. AI-Driven Network Intrusion Detection and Resource Allocation in Real-World O-RAN 5G Networks. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking (MobiCom)*; ACM: New York, NY, USA, 2024. [CrossRef]
18. Wen, H.; Sharma, P.; Yegneswaran, V.; Porras, P.; Gehani, A.; Lin, Z. 6G-XSec: Explainable Edge Security for Emerging OpenRAN Architectures. In *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks (HotNets '24), Irvine, CA, USA, 2024*; ACM: New York, NY, USA, 2024; pp. 77–85. [CrossRef]
19. Khor, J.H.; Sidorov, M.; Woon, P.Y. Public Blockchains for Resource-Constrained IoT Devices—A State-of-the-Art Survey. *IEEE Internet Things J.* **2021**, *8*, 11960–11982. [CrossRef]
20. Rzaeva, L.; Zhetpisbayeva, A.; Batkuldin, A.; Nyssanov, N.; Ryzhova, A.; Saeed, F. An Intelligent Browser History Forensics Method for Automated Analysis of Web Activity Logs, Credentials, and User Behavioral Profiles. *Algorithms* **2026**, *19*, 75. [CrossRef]
21. National Vulnerability Database. CVE-2025-1893. Available online: <https://nvd.nist.gov/vuln/detail/CVE-2025-1893> (accessed on 27 April 2026).
22. National Vulnerability Database. CVE-2024-24432. Available online: <https://nvd.nist.gov/vuln/detail/CVE-2024-24432> (accessed on 27 April 2026).
23. National Vulnerability Database. CVE-2025-5935. Available online: <https://nvd.nist.gov/vuln/detail/CVE-2025-5935> (accessed on 27 April 2026).
24. ETSI. 5G; Security Architecture and Procedures for 5G System (3GPP TS 33.501 Version 16.3.0 Release 16). Available online: https://www.etsi.org/deliver/etsi_ts/133500_133599/133501/ (accessed on 27 April 2026).
25. Yungaicela-Naula, N.M.; Sharma, V.; Scott-Hayward, S. Misconfiguration in O-RAN: Analysis of the Impact of AI/ML. *Comput. Netw.* **2024**, *247*, 110455. [CrossRef]
26. Alnajim, A.; Habib, S.; Islam, M.; Thwin, S.; Alotaibi, F. A Comprehensive Survey of Cybersecurity Threats, Attacks, and Effective Countermeasures in Industrial Internet of Things. *Technologies* **2023**, *11*, 161. [CrossRef]
27. Lau, J.L.T.; Pang, Y.H.; Zarakovitis, C.; Lim, H.S.; Skordoulis, D.; Ooi, S.Y.; Chan, K.Y.; Pang, W.L. BiTAD: An Interpretable Temporal Anomaly Detector for 5G Networks with TwinLens Explainability. *Future Internet* **2025**, *17*, 482. [CrossRef]
28. Deng, Y.; Lyu, F.; Xia, T.; Zhou, Y.; Zhang, Y.; Ren, J.; Yang, Y. A Communication-Efficient Hierarchical Federated Learning Framework via Shaping Data Distribution at Edge. *IEEE/ACM Trans. Netw.* **2024**, *32*, 2600–2615. [CrossRef]

29. Deng, Y.; Ren, J.; Tang, C.; Lyu, F.; Liu, Y.; Zhang, Y. A Hierarchical Knowledge Transfer Framework for Heterogeneous Federated Learning. In *Proceedings of IEEE INFOCOM 2023—IEEE Conference on Computer Communications*; IEEE: New York, NY, USA, 2023.
30. Qin, Z.; Yan, X.; Zhou, M.; Deng, S. BlockDFL: A Blockchain-based Fully Decentralized Peer-to-Peer Federated Learning Framework. In *Proceedings of the ACM Web Conference 2024 (WWW '24)*; ACM: New York, NY, USA, 2024. [[CrossRef](#)]
31. Spantideas, S.T.; Giannopoulos, A.E.; Levis, G.A.; Trakadas, P. FedRA: A Fully Decentralized Federated Learning Framework for Robust Intelligence Sharing across O-RAN dApps. *IEEE Commun. Mag.* **2026**. [[CrossRef](#)]
32. Sharevski, F. Towards 5G Cellular Network Forensics. *EURASIP J. Inf. Secur.* **2018**, *2018*, 8. [[CrossRef](#)]
33. Casey, E. *Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet*, 4th ed.; Academic Press: Cambridge, MA, USA, 2024.
34. Wang, J.; Qin, W.; He, J.; Zhao, C.; Niyato, D.; Xiang, T. Intelligent Forensics in Next-Generation Mobile Networks: Evidence, Methods, and Applications. *arXiv* **2026**, arXiv:2603.29364. [[CrossRef](#)]
35. Sikos, L.F. Packet Analysis for Network Forensics: A Comprehensive Survey. *Forensic Sci. Int. Digit. Investig.* **2020**, *32*, 200892. [[CrossRef](#)]
36. Rzayeva, L.; Shayakhmetov, M.; Atanbayev, Y.; Budenov, R.; Mutaheer, H. Automated Forensic Recovery Methodology for Video Evidence from Hikvision and Dahua DVR/NVR Systems. *Information* **2025**, *16*, 983. [[CrossRef](#)]
37. Imanberdi, A.; Khassanova, A.; Yermekov, Y.; Shayakhmetov, M.; Alibek, A.; Konakbayev, O. Development of a Data Decryption Method for the Raw Disk Image of Mobile Phone. In *Proceedings of the 2025 IEEE/ACIS 29th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, Busan, Republic of Korea, 2025; IEEE: New York, NY, USA, 2025; pp. 106–110. [[CrossRef](#)]
38. Salazar, Z.; Zaidi, F.; Nguyen, H.; Mallouli, W.; Cavalli, A.R.; Montes de Oca, E. 5Greplay: A 5G Network Traffic Fuzzer—Application to Attack Injection. In *Proceedings of the 16th International Conference on Availability, Reliability and Security (ARES)*; ACM: New York, NY, USA, 2021. [[CrossRef](#)]
39. MatrixTM. MHDDoS: Best DDoS Attack Script with 56 Methods. Available online: <https://github.com/MatrixTM/MHDDoS> (accessed on 27 April 2026).
40. PortSwigger Ltd. Burp Suite: The Web Vulnerability Scanner. Available online: <https://portswigger.net/burp> (accessed on 27 April 2026).
41. Harvanek, M.; Bolcek, J.; Kufa, J.; Polak, L.; Simka, M.; Marsalek, R. Survey on 5G Physical Layer Security Threats and Countermeasures. *Sensors* **2024**, *24*, 5523. [[CrossRef](#)] [[PubMed](#)]
42. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
43. Batkuldin, A.; Rzayeva, L.; Toksanbay, B.; Kim, P.; Kozhakhmet, Z.; Imanberdi, A. Development of a Blockchain Platform for Protection and Security of Medical Data and Patient Identification in Kazakhstan. *Artif. Intell. Res. Appl.* **2025**, *1*, 189–203. [[CrossRef](#)]
44. Rzayeva, L.; Ryzhova, A.; Zhaparkhanova, M.; Myrzatay, A.; Konakbayev, O.; Imanberdi, A.; Ahmed, Y.; Kozhakhmet, Z. Development of a Method for Determining Password Formation Rules Using Neural Networks. *Information* **2025**, *16*, 655. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.