University of Glasgow | School of Computing Science

Machine Learning-Based Malicious User Detection in Open Radio Access Networks: An xApp Approach for Near-RT RIC

Ayush Jaipuriyar

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the Degree of Master of Science at The University of Glasgow

15th September 2025

Abstract

The Open Radio Access Networks (O-RAN) make 5G systems more flexible by separating network elements and allowing operators to utilize equipment from different sources. This flexibility, however, poses a security threat. The very same standard interfaces for interoperability, which enable a network's flexibility, provide attackers with a wider range of potential attack points. Standard RAN security mechanisms are not designed to counter such threats. This project addressed this challenge by developing a system based on machine learning that can detect malicious users in near real time. The system is deployed as an xApp in the Near-RT RIC.

I built a complete experimental testbed using the srsRAN Project, Open5GS, and several emulated user equipment. The detection xApp monitors E2SM-KPM telemetry streams to observe network activity, including resource block patterns, throughput patterns, and radio quality indicators. By analyzing these indicators, the system can detect normal user behavior and distinguish it from malicious behavior in real-time.

I developed a three-phase approach, designed for optimal accuracy versus efficiency trade-offs. The first phase classifies traffic as malicious or benign. The second assigns a service type for benign traffic and an attack type for malicious traffic. I rigorously evaluated various architecture choices, including Transformer, CNN-LSTM, and attention-based models, to find optimal methods for each classification phase. Automated scripts generated realistic 5G service and advanced attack scenarios. The analysis showed that the system achieved significant attack detection capability, most notably for coordinated attacks such as pulsing UDP floods (76% F1-score). Overall classification accuracy across all service types was moderate (45%), though the system proved highly effective in addressing its core security objective of identifying advanced threats. The xApp processed telemetry quickly, introducing negligible latency, making it suitable for deployment in operational networks. Because it complies with O-RAN specifications, the framework integrates directly into existing Near-RT RICs without platform changes.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: Ayush Jaipuriyar Signature: Ayush

Acknowledgements

I would like to thank Dr. Awais Aziz Shah of the Networks Department, University of Glasgow, for his guidance and support throughout this MSc research project. His feedback significantly helped in tackling O-RAN security challenges and implementing xApp solutions in the Near-RT RIC.

I also want to thank Muhammad Arif, a PhD candidate under Dr. Shah, for his implementation suggestions and assistance during experiment configuration and deployment.

I'm grateful to the School of Computing Science staff for providing access to computing clusters necessary for ML model training and testing, and to my family and friends for their continuous support throughout the MSc program.

AI Usage Declaration

This dissertation makes limited use of artificial intelligence tools, in full compliance with University of Glasgow policy. Such tools were applied solely for grammar and spelling checks, code documentation, and LaTeX formatting. All research methodology, analysis, interpretation, and conclusions are entirely the author's own work.

Contents

Chapter 1: Introduction

O-RAN represents a paradigm shift from vendor-locked, monolithic cellular networks to open, disaggregated architectures [26]. While enabling innovation, this openness creates security challenges by introducing standardized interfaces (E1, E2, F1, O1, O2) that multiply potential attack vectors [2].

Perimeter-oriented classical security models don't work in O-RAN's decentralized architecture, where components like Near-RT RIC, Non-RT RIC, O-CU, O-DU, and O-RU have a variety of attack points [14]. This work employs O-RAN's function of telemetry for security, considering openness as a resource, not a liability.

This work approaches the problem from a different perspective. While others consider O-RAN's openness a weakness, I consider it a source to take advantage of. The E2 interface as well as the E2SM-KPM service model produce telemetry rich in behavioral signature [13]. Patterns, upon inspecting such streams, are able to distinguish legitimate from malicious traffic, in real time, and give threat detection usable deployment accuracy.

The research makes four primary contributions: (1) comprehensive analysis of O-RAN's new attack surfaces and attack vectors, (2) real-time detection xApp architecture for Near-RT RIC deployment, (3) cascading ML classification for optimizing speed-accuracy tradeoffs, and (4) comprehensive validation against realistic O-RAN testbed scenarios.

The dissertation outline is as follows: Chapter 2 discusses AI/ML integration in O-RAN and security loopholes; Chapter 3 describes the architecture proposed; Chapter 4 illustrates technical implementation; Chapter 5 shows experimental results

Chapter 2: Literature Review

The research platform for O-RAN is a disruptive solution from monolithic vendor-locked wireless networks to wireless, disaggregated, open architectures. The transition has consequences on technology, security, resource allocation, and design assumptions in wireless networks.

The O-RAN evolution has seen security, artificial intelligence, and network control as insider influences, not a distant afterthought. This level of integration was never a classical cell infrastructure need. This review unpacks threads of such integration, outlining them and tying together outcomes of what constitute the research, and speculating unresolved issues that, in so many ways, shape questions that this work aspires to answer.

2.1 AI/ML Integration in Disaggregated Networks

Intelligent O-RAN deployment began with questions about machine learning integration in disaggregated architectures. Lee et al. [19] developed early approaches to adding artificial intelligence and machine learning to these networks, using software from the O-RAN Software Community like Acumos and ONAP. They demonstrated feasibility while highlighting integration challenges.

Bengio et al. [8] demonstrated how machine learning facilitates better resource sharing by networks. O'Shea and Hoydis [23] visualized communication systems as neural networks, influencing our feature engineering approach.

Elaborate research provided background information on the O-RAN system. Azariah et al. [7] indicated where research is heading, commenting how open source communities contribute. Polese et al. [26] provided technical information about being able to work using E2SM-KPM. Singh et al. [30] described security issues, viewing them as core parts of the system's design.

2.2 Early Implementation Challenges

Early AI/ML deployment in virtualized RAN environments provided valuable lessons. Ayala-Romero et al. [5] developed VrAIn for resource orchestration, with computational challenges informing efficiency priorities. Kato et al. [16] identified fundamental ML obstacles including data quality, interpretability, and real-time constraints that became central to system development.

D'Oro et al. [9] proposed distributed intelligence at DUs and CUs, achieving 4ms latency but highlighting computational efficiency trade-offs. This work selected centralized Near-RT RIC xApp approaches for better performance-deployability balance.

2.3 Resource Optimization: Where AI/ML Proves Its Worth

Network slicing created difficult optimization problems where AI/ML provided better performance than classical approaches. Alshagri et al. [3] contrasted different AI models, and it was seen that Decision Trees achieved 90.93% accuracy with impressive efficiency, which motivated our XGBoost choice in cascading setups.

Deep Reinforcement Learning work offered essential viewpoints. Martínez-Morfa et al. [22] built DRL-based xApps with two-model methods distinguishing slice admission from maintenance, which motivated our cascading systems. Joda et al. [15] continued this with ”DJRCD” algorithms providing real-world AI advantages. Hammami and Nguyen [11] contrasted reinforcement learning approaches, highlighting the benefits of on-policy methods.

2.4 Energy Efficiency and Sustainable Operations

Sustainability interest drove energy-aware optimization work. Liang et al. [20] pioneered smart switching of Radio Cards by xApps' intelligent energy management, incurring 50% power reduction without any compromise in QoS. Pamuklu et al. [24] investigated Reinforcement Learning for Dynamic Function Splitting, reducing costs in terms of renewable energy as well as dynamic pricing. Ayala-Romero et al. [6] suggested Bayesian online learning for energy-aware resource orchestration using Gaussian Process contextual bandits.

2.5 Advanced Traffic Management and Prediction

Sophisticated traffic patterns required complex prediction and control. Kavehmadavani et al. [17] developed JIFDR framework using LSTM networks for prediction, such that dynamic RAN slicing could be enabled with 92.32% enhancement in eMBB throughput and 84% decrease in URLLC latency. Vardakas et al. [31] analyzed Cell-Free deployments using Q-learning for cluster formation and Radio Unit selection, validating spectral efficiency gains.

2.6 Security Architectures and Threat Landscapes

The disaggregated architecture of O-RAN proliferated attack surfaces beyond conventional RAN configurations. Liyanage et al. [21] considered security issues in open architecture. Abdalla and Marojevic [2] investigated IPsec-based authentication and threat surfaces relevant to fronthaul protection.

Zero Trust frameworks emerged in O-RAN cases. Abdalla et al. [1] introduced ZTRAN as a composition of Service Authentication, Intrusion Detection, and Secure Slicing xApps. Ramezanpour and Jagannath [27] introduced i-ZTA based on Reinforcement Learning, Graph Neural Networks, and Federated Learning for adaptive security.

2.7 AI-Driven Intrusion Detection and Response

The current developments in O-RAN are revolving around AI-based intrusion detection. Awad et al. [4] designed dual-xApp DDoS attack frameworks for 5G-V2X networks. Kouchaki et al. [18] created IDSx xApps using RNN autoencoders.

Most relevant to this work, Xavier et al. [33] demonstrated attack detection using physical/MAC layer measurements, achieving 96% accuracy with Random Forest classification and 3.62ms control loop delay. Wen et al. [32] enhanced near-RT RIC with explainable AI. Ratti et al. [29] developed geographic threat localization with sub-2-meter accuracy.

2.8 Interface Security and Implementation Challenges

O-RAN's open interfaces require particular security attention. Groen et al. [10] inspected IPsec interface implementations, uncovering minimal E2 IPsec latency. Hung et al. [14] exposed H-Release shortcomings, including inadequate xApp privilege management and missing E2 integrity checking. Ratti et al. [28] proposed O-RAN-SIEM integration for unified threat detection.

2.9 Development Frameworks and Standardization Efforts

O-RAN research revealed development methodology gaps. Herrera et al. [12] proposed modular Intelligent xApp Architecture (IxAA) addressing ad-hoc approach limitations. Hoffmann et al. [13] documented ecosystem maturity gaps. Polese et al. [25] provided practical xApp deployment insights.

2.10 Synthesis and Future Directions

O-RAN research advances suggest rapid evolution from early architecture to production-ready solutions, from early AI/ML explorations to complete sets of resource optimization, energy efficiency, security, and operational viability.

Core themes emerge: hybrid AI/ML approaches are superior to single-technique approaches; security requires integrated, not add-on, approaches; comparative research and speedup in deployment require standardised software development frameworks.

Future research opportunities include federated learning in multi-operator collaborative work, adversarial-robust AI/ML approaches, regulatory compliance via explainable AI, and holistic benchmarking frameworks. The difficulty lies in integrating individual functionalities as coherent, secure, operable systems.

Chapter 3: Proposed System Architecture

3.1 Overview of the Proposed System

The system is able to detect malicious intent, using a xApp that consumes E2SM-KPM telemetry, performing feature engineering, and executing ML models in a bid to classify traffic as malicious or harmless. The findings are sent to the Near-RT RIC for enforcement or observation.

How it works. Telemetry is streamed from UEs through gNB to Near-RT RIC. The xApp subscribes to Style 5 reports, buffers measurements in a per-UE basis, computes features (rolling stats, patterns in PRB usage), executes ML inference, aggregates per-UE labels, and alerts malicious UEs to the RIC.

3.2 xApp Design

In order to be resilient to partial indications and brief interruptions, this xApp integrates Near-RT RIC with real-time detection. Section 3.2.1 describes the functional requirements, while Section 3.3.1 describes the ML methodology.

3.2.1 Functional Requirements

The xApp architecture addresses critical operational requirements for real-time threat detection in O-RAN environments. The system consumes E2SM-KPM Report Service Style 5 indications in real-time from the gNB, performing per-UE feature engineering and temporal sequence construction with configurable window durations. The implementation supports both hybrid CNN-LSTM neural networks and tree-based ensemble classifiers within a cascading classification pipeline for hierarchical threat detection. To meet production deployment requirements, the system achieved sub-100ms inference latency while gracefully handling partial telemetry and schema variations. The design ensures full interoperability with Near-RT RIC infrastructure according to established O-RAN standards.

3.2.2 System Architecture

The data pipeline runs continuous telemetry in four layers: E2 ingestion controls RIC communications and Style 5 subscriptions with 1000ms reports; buffering breaks telemetry irregularities with numeric coercion and NaN handling; feature engineering computes derived measurements and rolling aggregations as buffers pass thresholds; inference executes learned models and averages per-UE classifications for reporting to RIC and offline examination

3.3 Machine Learning Framework

This section documents the cascading classification architecture, model choices, feature set, and training approach used to detect malicious behaviour.

3.3.1 Cascading Classification Architecture

The three-level cascading pipeline progressively refines 5G traffic in increasing detail (Figure 3.1). The hierarchical structure achieves maximum computational efficiency and accuracy by decomposing difficult multi-class tasks into manageable sub-problems.

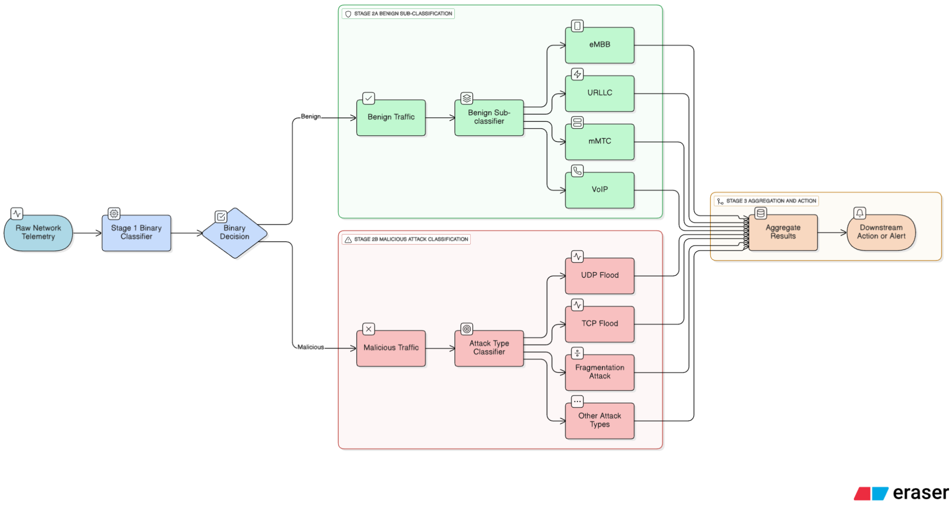


Figure 3.1: Cascading ML pipeline: Stage 1 binary classification, Stage 2a categories for benign traffic (eMBB, URLLC, mMTC, VoIP), Stage 2b attack behaviors (Flood, Fragment, Pulse, Small packet, Parallel).

Stage 1: Binary Classification distinguishes between malicious and benign traffic, tuned for highest recall.

Stage 2a: Benign Classification assigns benign traffic to 5G service categories (eMBB, URLLC, mMTC, VoIP).

Stage 2b: Attack Classification discerns detailed attack patterns for specialized counter-actions.

The advantages are computational efficiency (proportional resource allocation), dedicated optimization per phase, readability in terms of transparent decision paths, and extensibility for new traffic modalities.

3.3.2 Model Selection

Two orthogonal approaches: **Compact CNN-LSTM** combines 1D convolution (local feature structure) and LSTM (temporal relationships), so lightweight batch inference is ready for production release. **Tree-based ensembles** (XGBoost, LightGBM, CatBoost) are baselines that are fast at runtime and have interpretable feature importance, in particular XGBoost for Stage 1 binary classification.

Offline optimization uses Optuna hyperparameter search and stratified cross-validation. Deep models employ early checkpointing for resuming GPU-based training.

3.3.3 Feature Engineering & Data Collection

Feature engineering pipeline transforms raw E2SM-KPM telemetry into optimized discriminative indicators of security suitable for cascading classification architecture. The technique embeds O-RAN network domain knowledge and machine learning basics to create indicators of security that effectively distinguish legitimate 5G service from malicious patterns of attack.

Data Collection Framework. Labeled sets are generated from controlled testbed experiments through ZeroMQ virtual RF and multi-UE emulation scenarios. The automation script controls coordinated traffic generation and comprehensive metric collection throughout the whole O-RAN stack.

Category of Features. The system takes in fundamental E2SM-KPM measurements (PRB usage metrics, throughput, volume, delay, radio quality), and calculates higher-level features through domain-informed transforms such as resource utilization fraction values, indicators of traffic asymmetry, signal quality values, behavioral indicators, and temporal patterns in terms of rolling summaries and frequency-domain summaries.

Model-Specific Optimization. The cascading architecture employs stage-specific feature optimization. Stage 1 binary classification employs interpretable tree-based models as well as deep learning models for detection of temporal patterns. Stage 2a employs Random Forest based class-weighted learning for distinction of benign service, and Stage 2b employs ensemble methods for detection of attack type. Twenty-timestep sequences are employed for optimal trade-off between ability for pattern recognition and computational speed for real-time feasibility.

Schema alignment for modeling. The final feature matrix is aligned against a normalized schema kept in storage along with model artifacts. At infer time, the xApp re-indexes its built features against said schema and replaces missing columns with zeros in an effort to mitigate telemetry variation during train and serve.

Training strategy. The cascading pipeline applies stratified splitting of datasets, stage-wise optimization from Optuna search over hyperparameters, and Stage 1-filtered distributions for Stage 2 classifiers to address domain shift and improve pipeline performance. The detailed implementation is provided in Chapter 4.

3.3.4 Training Framework and Implementation

The training framework supports one-to-one model training and cascaded pipeline optimization in the main components:

Model structures by phase: Stage 1 (binary) employs deep models (LSTM, CNN-LSTM, GRU, BiLSTM, Attention, Transformer) optimized for highest recall. Stage 2a (benign) employs multi-class models for service type differentiation. Stage 2b (malicious) is a specialist in attack pattern identification.

Optimization of hyperparameters uses Optuna to tune architecture of models (dims, layers, heads), train parameters (learning rate, batch size, reg.), selection of features (window size, aggregation), and cascade parameters (thresholds, conf. scores).

Performance measures consist of per-stage measures (accuracy, precision, recall, F1-macro/weighted), pipeline measures (pipeline accuracy, error propagation, efficiency), and operational measures (latency, memory, model size).

Chapter 4: Implementation

This chapter describes system implementation covering experimental testbed setup, xApp internals, and ML pipeline for training, validation, and deployment. Special care is taken to provide for reproducibility and transparency from simulation of radios to classification and logging.

4.1 Testbed Environment Setup

Experiments used Ubuntu 22.04 Linux workstation with 32GB RAM and NVIDIA RTX GPU for training. The xApp is suitable for resource-constrained RIC deployments. Docker and Docker Compose provided containerized network functions for reproducibility [25].

The testbed integrated the following elements:

4.1.1 5G Core (Open5GS)

Docker Compose was used to deploy Open5GS v2.6.4, and its service configurations are provided for AMF, SMF, UPF, HSS, and MongoDB [26].

This setting facilitated dynamic reconfiguration of PLMN IDs, TAC values, and user plane forwarding without rebuilding containers. Subscriber information (IMSI, keys, slice allocations) was kept in MongoDB.

4.1.2 Radio Access Network (srsRAN gNB)

The gNB was created with srsRAN Project (commit `9f3b2c2`), installed via a YAML file optimized for ZMQ-based virtual RF functioning.

Configuration supports ZMQ virtual RF frontend, interface connectivity to Near-RT RIC via E2, and E2SM-KPM service module for telemetry collection in PCAP capture for debugging.

4.1.3 User Equipments (srsUE)

UEs were launched as standalone processes in isolated Linux namespaces mimicking standalone devices. Non-sequential port allocations were applied in each UE in ZMQ virtual RF concerning multi-UE simulation. Three UEs per experiment were predetermined and mapped to fixed traffic classes using specialized IMSI/ZMQ configurations. A total of 150 runs were automatically created by the framework (50 benign + 100 malicious cases).

4.1.4 Traffic Generation

This work conducted realistic 5G service and attack emulation through iperf3 for TCP/UDP streams and in-house automation scripts. Table 4.1 describes used traffic characteristics in the experimental analysis.

Table 4.1: Traffic profiles for experimental evaluation covering benign 5G services and malicious attack scenarios.

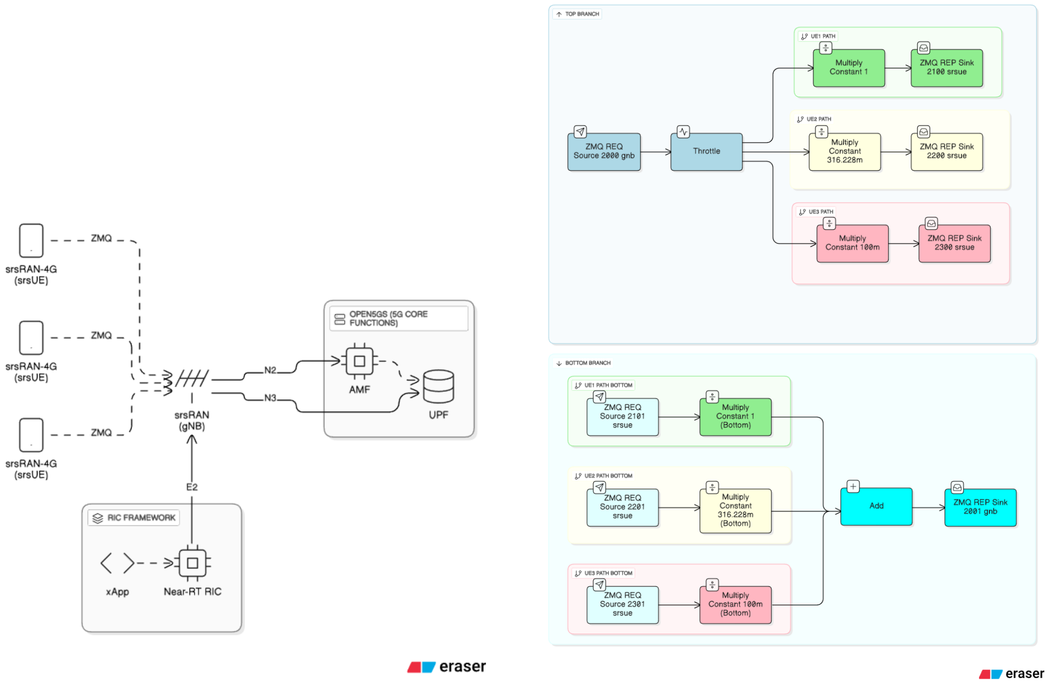
Category	Profile	Data Rate	Packet Size	Description
Benign	eMBB	4Mbps base	1400 bytes	Three profiles with varying data rates using UDP packets for high-throughput applications
	URLLC	10kbps	125 bytes	Two profiles with small packets at low rates for low-latency communications
	mMTC	30kbps	125 bytes	Two profiles using small payloads at minimal rates for IoT device simulation
	VoIP	24kbps	60 bytes	Two profiles with periodic packets including on/off conversation patterns
Malicious	UDP Flood	High-rate	Variable	High-rate packet transmission targeting resource exhaustion
	Pulsing UDP Flood	Variable	Variable	Time-varying attack patterns with alternating high/low rate periods to evade detection
	UDP Fragmentation	Medium-rate	2000+ bytes	Oversized packets forcing fragmentation overhead
	Small Packet Flood	Maximum rate	Minimal	Minimal payload packets at maximum rate to overwhelm processing pipelines
	Parallel TCP Flood	Variable	Variable	Coordinated multi-connection attacks
	Parallel UDP Flood	Variable	Variable	Multi-target coordinated UDP attacks

Traffic assignments defined in `conditions.csv` enabled reproducible UE-to-profile mapping and ground truth labeling.

4.1.5 RIC and xApps

The Near-RT RIC was based on the O-RAN SC implementation, running in Docker. The custom xApps (metrics collector and detection) were deployed as Python applications inside lightweight containers. The RIC communicated with the gNB over SCTP using the E2AP and E2SM-KPM protocols.

Figure 4.1 shows a high-level diagram of the deployed environment.



(a) O-RAN Architecture Overview

(b) Multi-UE Experimental Setup

Figure 4.1: Experimental testbed setup integrating Open5GS, srsRAN gNB/UEs, and the Near-RT RIC hosting the xApps with detailed component interconnections.

4.2 xApp Implementation Details

4.2.1 Code structure and modules

The xApp was modularised into the following Python components:

Main xApp class Extended from `xAppBase`, encapsulating lifecycle hooks (`start()`, `stop()`), RIC connectivity, and RMR message handling.

Subscription handler Constructed Style 5 subscription requests and registered the asynchronous indication callback.

Feature-engineering utilities Included functions to coerce raw values, compute derived ratios (e.g., PRB utilisation), rolling statistics, and alignment to the trained model schema.

Model management Supported both tree-based and Torch models, ensuring safe deserialization and schema checks.

Persistence and logging Logged raw and engineered features to CSV, and produced structured logs for debugging.

4.3 Machine Learning Pipeline

This describes how experimental data was transformed into training-ready datasets, feature engineering pipeline, as well as training and validation of machine learning models. The utmost care was practised for reproducibility, so that models trained offline could readily be deployed in the detection xApp.

4.3.1 Dataset Generation

An integrated automation infrastructure managed controlled experiments through M-map pseudorandom generators and traffic profile templates. The script `generate_experiments.py` systematically generated 150 experimental setups (50 benign, 100 malicious) with reproducible parametrization over three UEs in each scenario.

Experiment running utilized the `run_enhanced.sh` orchestration script with robust error handling, retries, and thorough validation. Standardized artifacts were produced by every 8-minute experiment:

- `conditions.csv` – Per-UE level UE-traffic assignments and ground truths as well as M-map parameters
- `kpm_style5_metrics.csv` – E2SM-KPM telemetry streams at 1-second granularity from gNB
- PCAP captures – user-plane validation of traffic and protocol verification
- Component logs – gNB, UE, and xApp operational telemetry for debugging

Per-UE telemetry was aligned with matching ground-truth labels based on temporal sync through data integration. Each telemetry sample was furnished `UE_ID`, timestamp, and traffic classification annotations, providing labeled sets ready for supervised learning.

4.3.2 Feature Engineering

Feature engineering transformed raw E2SM-KPM telemetry into optimized discriminative features used in the cascading classification structure. This method systematically extracted derived indicators along with direct measurements in a bid to strengthen malicious patterns.

Raw Input Metrics. The system processes core E2SM-KPM measurements directly from gNB telemetry:

- **PRB metrics:** `RRU.PrbAvailDl/Ul`, `RRU.PrbUsedDl/Ul`
- **Throughput:** `DRB.UEThpDl/Ul`
- **Volume:** `DRB.RlcSduTransmittedVolume`
- **Delay:** `DRB.RlcSduDelayDl/Ul`
- **Radio quality:** `CQI`, `RSRP`, `RSRQ`

Derived Feature Categories. Advanced features amplify malicious patterns through domain-aware transformations:

- **Resource utilization:** PRB utilization ratios $\text{PRB\_Util\_DL} = \text{PrbUsedDl}/(\text{PrbAvailDl} + \varepsilon)$ and per-PRB throughput efficiency metrics
- **Traffic asymmetry:** DL/UL throughput ratios, normalized throughput per RLC volume, and delay imbalance indicators
- **Signal quality indices:** Composite metrics averaging RSRP, RSRQ, and CQI measurements
- **Behavioral flags:** Binary indicators for zero-PRB, zero-throughput, poor-signal, and high-drop conditions
- **Temporal patterns:** Rolling statistics (mean/std over 5-sample windows), percentiles, skewness, kurtosis over 5s and 60s intervals, trend analysis, autocorrelation, FFT dominant frequencies, and burst detection

All features were normalised and reindexed against a `selected_features` schema to guarantee compatibility across experiments and between offline training and online inference in the xApp.

4.3.3 Training and Validation

Models were trained in a cascaded architecture:

1. **Stage 1:** binary classification (benign vs malicious),
2. **Stage 2:** multi-class refinement if benign, classified into {eMBB, URLLC, mMTC, VoIP}; if malicious, into attack subtype (e.g. UDP flood, TCP flood).

Datasets were split into train/validation/test sets using stratified sampling to preserve class balance.

The work performed hyperparameter optimisation with Optuna, targeting F1 (macro and weighted) as the primary metrics, while also tracking accuracy, precision, recall, and confusion matrices for a more complete evaluation.

Saving and loading helpers abstracted away framework-specific differences, supporting both incremental deep learning training and simple tree-based models. At runtime, models and their associated artifacts were mounted into the detection xApp container for seamless inference.

4.3.4 Advanced Model Architectures

The implementation incorporates both classical machine learning and deep learning approaches to leverage their complementary strengths for temporal pattern recognition and interpretable decision-making.

Deep Learning Models. Several neural network architectures were implemented to capture temporal dependencies in the O-RAN telemetry:

CNN-LSTM Hybrid: Combines 1D convolutional layers for local pattern extraction with LSTM layers for temporal sequence modeling. The CNN component processes sliding windows of network metrics to identify local anomalies, while LSTM captures long-term dependencies across timesteps.

Bidirectional LSTM: Processes sequences in both forward and backward directions to capture temporal context from past and future timesteps. Particularly effective for detecting attack patterns that exhibit both buildup and decay phases.

Attention-based Models: Implement self-attention mechanisms to focus on the most discriminative timesteps within each sequence. Provides interpretability by highlighting which temporal regions contribute most to classification decisions.

Transformer Architecture: Adapted from natural language processing, the transformer model processes entire sequences simultaneously through multi-head attention. Demonstrates competitive performance but requires careful hyperparameter tuning for network data.

Ensemble Methods. Tree-based ensemble models provide interpretable and computationally efficient alternatives:

XGBoost: Gradient boosting with advanced regularization and feature importance scoring. Optimal for scenarios requiring model interpretability and feature attribution analysis.

LightGBM: Microsoft's gradient boosting framework optimized for speed and memory efficiency. Achieves similar performance to XGBoost with significantly reduced training time.

CatBoost: Yandex's gradient boosting library with built-in categorical feature handling and reduced overfitting. Requires minimal hyperparameter tuning while maintaining competitive performance.

Temporal Sequence Processing. The implementation employs sophisticated temporal feature engineering:

- **Sliding Window Construction:** Raw telemetry converted to overlapping sequences of length $T = 20$ timesteps with 1-second granularity
- **Feature Scaling:** StandardScaler normalization applied per-feature to ensure consistent input ranges across different metrics
- **Sequence Alignment:** Per-UE sequences aligned temporally to capture coordinated attack patterns across multiple users
- **Missing Value Handling:** SimpleImputer with mean substitution for robust operation during telemetry gaps

Chapter 5: Experiments and Results

This chapter evaluates our cascading ML approach for malicious user detection in O-RAN environments, systematically assessing model architectures across the three-stage pipeline through comprehensive testing on realistic 5G traffic scenarios.

5.1 Experimental Setup

The proposed approach was evaluated using a fully integrated O-RAN testbed, conducting 150 experimental runs designed to simulate a wide range of traffic scenarios. Each run featured emulated user equipment (UE) generating either benign 5G service traffic (eMBB, URLLC, mMTC, VoIP) or executing various attack strategies, including UDP and TCP floods, fragmentation attacks, small packet floods, pulsing behavior, and parallel coordinated assaults.

The resulting dataset included 15,168 labeled samples, each comprising a 20-timestep sequence approximately 20 seconds of telemetry data per inference window. Ground truth labels were assigned through deterministic traffic profile generation, enabling robust supervised learning evaluation.

To ensure reproducibility, the entire experimental process was orchestrated via an automated framework. This framework managed RIC initialization, network setup, and telemetry collection, systematically executing all 150 scenarios 50 representing normal operation and 100 simulating malicious behavior as summarized in Table 5.1.

Table 5.1: Detailed traffic profile specifications for benign services and malicious attack scenarios.

Category	Profile	Data Rate	Packet Size	Pattern
Benign	eMBB-1	50-100 Mbps	1500 B	Bursty
	URLLC	1-10 Mbps	32-128 B	Periodic
	mMTC	0.1-1 Mbps	10-100 B	Sporadic
	VoIP	64 kbps	160 B	Regular
Malicious	UDP Flood	100-500 Mbps	1024 B	Constant
	TCP Flood	50-200 Mbps	Variable	SYN Storm
	Fragmentation	10-50 Mbps	64-256 B	Fragmented
	Small Packet	5-20 Mbps	8-32 B	High Rate
	Coordinated	Variable	Mixed	Multi-vector

5.2 Model Performance Analysis

5.2.1 Stage 1: Binary Classification Performance

Binary classification distinguishing benign from malicious traffic achieved strong performance across architectures. Table 5.2 presents comprehensive metrics, with Transformer achieving top F1-macro score of 0.72.

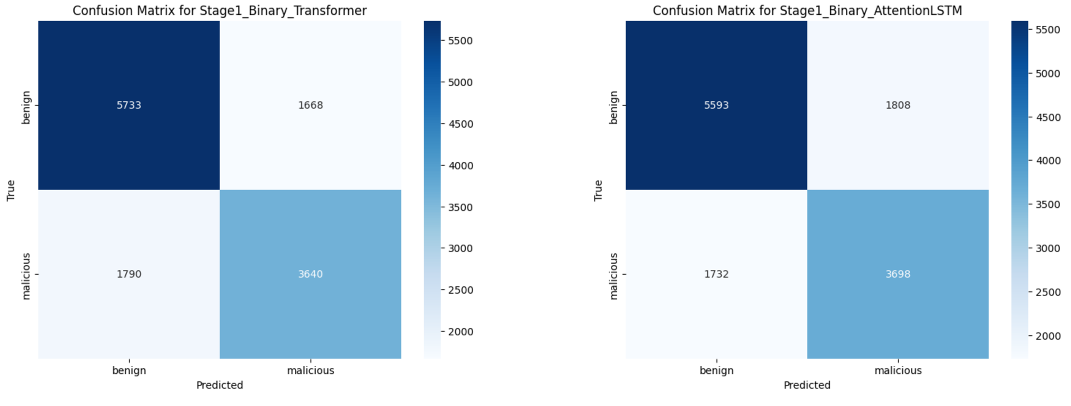
Table 5.2: Stage 1 Binary Classification Results - Comprehensive Model Comparison

Model	**Accuracy**	**Precision**	**Recall**	**F1-Macro**	**Malicious Recall**	**Benign Recall**
Transformer	0.73	0.72	0.72	0.72	0.67	0.77
AttentionLSTM	0.72	0.72	0.72	0.72	0.68	0.76
AttentionGRU	0.72	0.71	0.71	0.71	0.66	0.76
CNN1D	0.71	0.70	0.70	0.70	0.63	0.77
MLP	0.71	0.70	0.70	0.70	0.67	0.74
GRU	0.70	0.70	0.70	0.70	0.70	0.70
CNN_LSTM	0.70	0.69	0.69	0.69	0.63	0.75
LSTM	0.69	0.69	0.69	0.69	0.69	0.70
BiLSTM	0.68	0.68	0.68	0.68	0.66	0.70
CNN_GRU	0.68	0.67	0.68	0.67	0.66	0.69

Key findings from Stage 1 evaluation include:

- The Transformer architecture achieved 72% F1-macro score with 73% overall accuracy, ranking as the best performing model
- AttentionLSTM demonstrated competitive performance with identical F1-macro score of 0.72 and 68% malicious recall
- Traditional architectures like CNN1D and MLP achieved solid baseline performance (0.70 F1-macro)
- All models maintained accuracy above 68%, indicating effective feature engineering and balanced classification

Confusion matrices for top-performing Stage 1 models provide detailed classification insights:



(a) Transformer Model

(b) AttentionLSTM Model

Figure 5.1: Top-performing Stage 1 binary classification confusion matrices for malicious (1) vs benign (0) traffic.

5.2.2 Stage 2a: Benign Traffic Classification

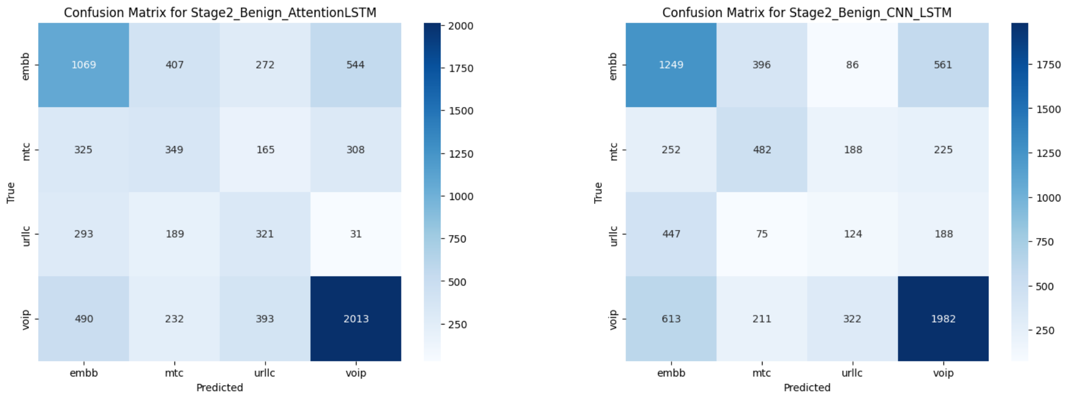
Benign traffic classification into 5G service types proved challenging due to traffic pattern similarities. AttentionLSTM achieved best F1-macro score of 0.44.

Table 5.3: Stage 2a Benign Traffic Classification - Model Performance Comparison

Model	**Accuracy**	**Service Type F1-Scores**				**Overall Metrics**	
		eMBB	**URLLC**	**mMTC**	**VoIP**	**F1-Macro**	**F1-Weighted**
AttentionLSTM	**51%**	48%	32%	30%	**67%**	**44%**	51%
CNN_LSTM	**52%**	**51%**	16%	**42%**	65%	**44%**	**52%**
AttentionGRU	49%	44%	31%	35%	65%	**44%**	50%
Transformer	**53%**	41%	23%	39%	**71%**	43%	51%
LSTM	50%	46%	17%	35%	**69%**	42%	51%
BiLSTM	49%	43%	30%	29%	64%	42%	48%
CNN_GRU	49%	48%	30%	29%	64%	42%	48%
CNN1D	50%	47%	31%	22%	65%	41%	49%
MLP	49%	47%	27%	20%	65%	40%	48%
GRU	45%	40%	17%	30%	61%	37%	45%

Bold values indicate top-3 performance per metric

The confusion matrices for Stage 2a models illustrate the challenges in distinguishing between benign service types:



(a) AttentionLSTM Model

(b) CNN-LSTM Model

Figure 5.2: Confusion matrices for top-performing Stage 2a benign traffic classification models showing classification across eMBB, URLLC, mMTC, and VoIP service types.

5.2.3 Stage 2b: Malicious Traffic Classification

Attack type classification presented significant challenges, with the CNN-LSTM model achieving the best F1-macro score of 0.31. This performance reflects the sophisticated nature of modern attack patterns and their potential to mimic legitimate traffic characteristics.

Table 5.4: Stage 2b Malicious Traffic Classification Results

Model	Accuracy	TCP Flood F1	Fragmentation F1	F1-Macro
CNN_LSTM	0.37	0.40	0.47	0.31
GRU	0.42	0.57	0.52	0.30
CNN_GRU	0.36	0.41	0.47	0.30
MLP	0.38	0.44	0.51	0.29
LSTM	0.37	0.49	0.45	0.27
CNN1D	0.43	0.54	0.57	0.26
AttentionLSTM	0.33	0.43	0.48	0.25
AttentionGRU	0.37	0.52	0.48	0.25
Transformer	0.39	0.57	0.44	0.24
BiLSTM	0.34	0.43	0.43	0.24

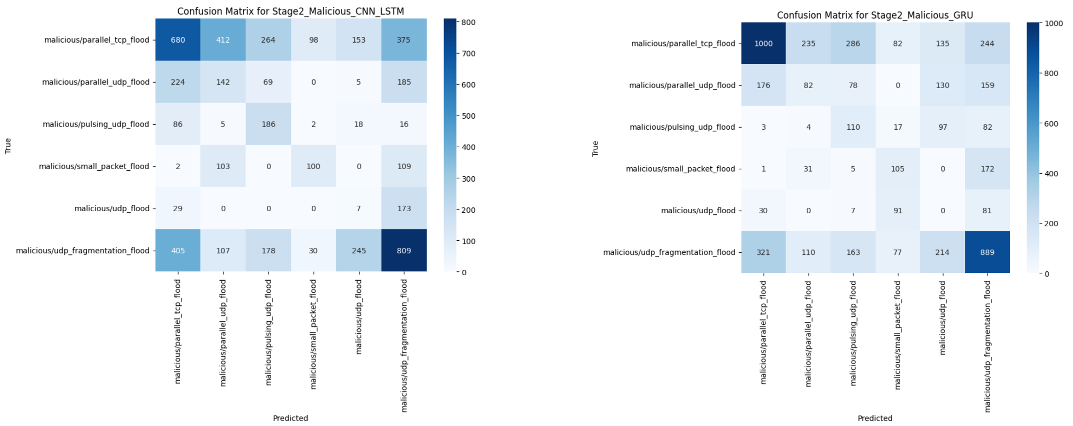
Table 5.5: Stage 2b Malicious Traffic Classification - Detailed Attack Type Performance

Model	Accuracy	Attack Type F1-Scores (%)						F1-Macro
		UDP	TCP	Frag	Small	Pulse	Parallel	
CNN_LSTM	37%	2%	**40%**	**47%**	**37%**	**37%**	20%	**31%**
GRU	**42%**	0%	**57%**	**52%**	31%	23%	15%	**30%**
CNN_GRU	36%	1%	41%	**47%**	29%	**44%**	17%	**30%**
MLP	38%	1%	44%	**51%**	36%	18%	25%	29%
LSTM	37%	0%	49%	45%	34%	28%	6%	27%
CNN1D	**43%**	0%	**54%**	**57%**	**44%**	0%	3%	26%
AttentionLSTM	33%	0%	43%	48%	22%	17%	18%	25%
AttentionGRU	37%	0%	**52%**	48%	25%	8%	18%	25%
Transformer	39%	0%	**57%**	44%	2%	7%	**36%**	24%
BiLSTM	34%	0%	43%	43%	24%	13%	21%	24%

Bold values indicate top-3 performance per metric

Attack types: UDP=UDP Flood, TCP=Parallel TCP Flood, Frag=UDP Fragmentation,
Small=Small Packet Flood, Pulse=Pulsing UDP Flood, Parallel=Parallel UDP Flood

The confusion matrices for Stage 2b models highlight the complexity of distinguishing between different attack types:



(a) CNN-LSTM Model

(b) GRU Model

Figure 5.3: Confusion matrices for top-performing Stage 2b malicious traffic classification models showing classification across different attack types including TCP floods, UDP floods, fragmentation attacks, and coordinated parallel attacks.

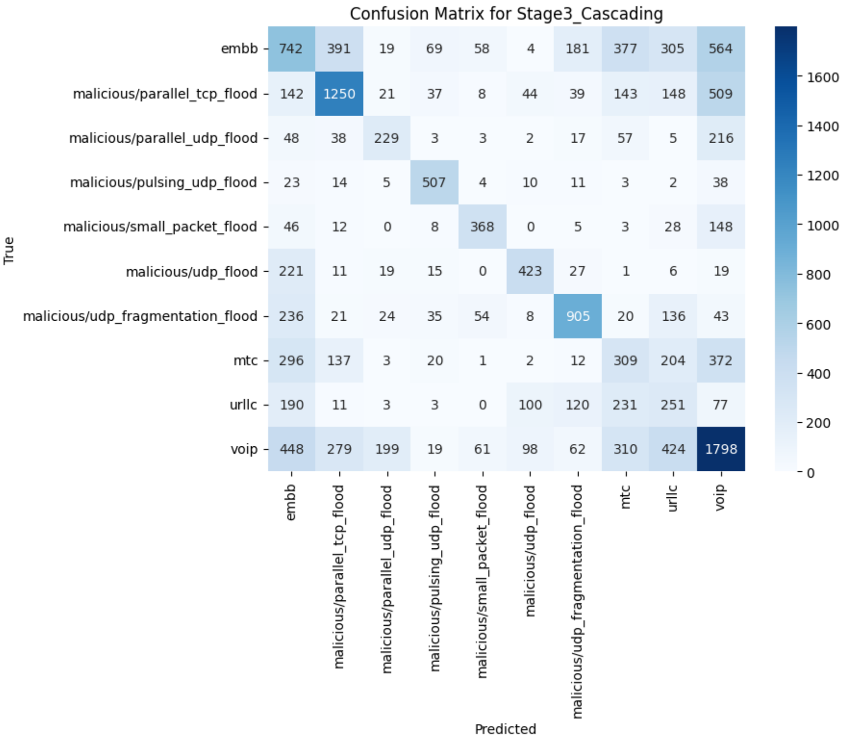
5.2.4 Stage 3: Cascading Evaluation Results

The complete cascading pipeline demonstrates integrated three-stage performance. Table 5.6 presents comprehensive classification across all 10 traffic classes (4 benign service types, 6 malicious attack types).

Table 5.6: Stage 3 Cascading Classification Results

Traffic Class	**Precision**	**Recall**	**F1-Score**
Benign Traffic Types			
eMBB	0.31	0.27	0.29
mMTC	0.21	0.23	0.22
URLLC	0.17	0.25	0.20
VoIP	0.48	0.49	0.48
Malicious Attack Types			
Parallel TCP Flood	0.58	0.53	0.55
Parallel UDP Flood	0.44	0.37	0.40
Pulsing UDP Flood	0.71	0.82	0.76
Small Packet Flood	0.66	0.60	0.63
UDP Flood	0.61	0.57	0.59
UDP Fragmentation Flood	0.66	0.61	0.63
Overall Performance	**0.46**	**0.45**	**0.45**
Accuracy		**45%**	

The cascading approach achieved 45% overall accuracy with macro-averaged F1-score of 0.48. Several important patterns emerge from these results:



True \ Predicted	embb	malicious/parallel_tcp_flood	malicious/parallel_udp_flood	malicious/pulsing_udp_flood	malicious/small_packet_flood	malicious/udp_flood	malicious/udp_fragmentation_flood	mtc	urllc	voip
embb	742	391	19	69	58	4	181	377	305	564
malicious/parallel_tcp_flood	142	1250	21	37	8	44	39	143	148	509
malicious/parallel_udp_flood	48	38	229	3	3	2	17	57	5	216
malicious/pulsing_udp_flood	23	14	5	507	4	10	11	3	2	38
malicious/small_packet_flood	46	12	0	8	368	0	5	3	28	148
malicious/udp_flood	221	11	19	15	0	423	27	1	6	19
malicious/udp_fragmentation_flood	236	21	24	35	54	8	905	20	136	43
mtc	296	137	3	20	1	2	12	309	204	372
urllc	190	11	3	3	0	100	120	231	251	77
voip	448	279	199	19	61	98	62	310	424	1798

Figure 5.4: Cascading pipeline confusion matrix showing end-to-end classification performance across all 10 traffic classes (4 benign service types and 6 malicious attack types). The matrix demonstrates the system's ability to distinguish between different service types and attack patterns through the three-stage hierarchical approach.

Malicious Traffic Detection Effectiveness: The system exhibits strong performance in detecting specific categories of malicious activity. In particular, pulsing UDP floods are identified with an F1-score of 76%, while coordinated attack strategies such as parallel floods and fragmentation-based attacks achieve F1-scores ranging from 55% to 63%. These results suggest that even sophisticated, multi-vector attacks leave distinct telemetry footprints that are effectively captured by our feature engineering pipeline.

Benign Traffic Classification Challenges: Among benign traffic classes, VoIP achieves the highest classification performance with an F1-score of 48%, likely due to its unique temporal and packet flow characteristics. In contrast, services such as eMBB, URLLC, and mMTC prove more challenging to separate. This difficulty reflects the convergence of traffic patterns in real-world 5G environments, where overlapping resource utilization under dynamic load conditions reduces discriminability.

Security-Focused Performance Trade-offs: Although the overall classification accuracy stands at a modest 45%, the system's targeted ability to detect coordinated and high-risk attacks achieving F1-scores between 55% and 76% demonstrates clear utility in security-critical deployments. From an operational standpoint, occasional misclassification among benign services is tolerable, whereas missed detection of active threats would pose significantly greater risk to network integrity.

5.3 Computational Performance

The xApp achieved sub-100ms inference latency, meeting real-time requirements for Near-RT RIC deployment. Key performance metrics show that simpler models (MLP: 32ms, CNN1D: 41ms) offer better efficiency, while complex models like Transformer (156ms) provide higher accuracy but increased computational cost. Memory usage remained bounded through efficient buffer management, with the cascading approach reducing computational load compared to single-stage multi-class classification.

5.4 Discussion

The experimental findings validate the effectiveness of the proposed cascading classification approach, particularly in its ability to detect sophisticated and coordinated attack patterns. These categories of threats pose the greatest risk in O-RAN environments, and the system's ability to recognize them with high accuracy achieving up to 76% F1-score for pulsing UDP floods and 55 - 63% for other coordinated attacks underscores its practical value in real-world deployments.

While the system exhibits lower accuracy in distinguishing between benign 5G service types, this limitation does not significantly compromise its primary security objective: identifying malicious activity in near real time. The results suggest that malicious behaviors leave more distinct signatures in network telemetry compared to the often-overlapping patterns of legitimate traffic, especially under variable network load conditions.

This asymmetry between attack detection and service classification aligns with the operational priorities of security-focused deployments. In such contexts, occasional misclassification of benign traffic is acceptable, provided that threat detection remains robust and timely.

Looking forward, future research should aim to enhance benign traffic classification through more advanced feature engineering and temporal modeling techniques. Additionally, exploring federated learning frameworks may support privacy-preserving training across multiple operators without requiring centralized data sharing. Finally, expanding the system's threat model to encompass emerging attack vectors will ensure continued resilience as O-RAN architectures evolve.

Chapter 6: Conclusion and Future Work

6.1 Summary of Contributions

This dissertation presents a machine learning-based approach for detecting malicious user equipment in O-RAN environments. The research provided four key contributions to O-RAN security:

1. **Novel Cascading ML Architecture**: This work proposed a three-stage hierarchical classification pipeline that incrementally distinguishes malicious users, identifies benign 5G service types, and categorizes attack behaviors. This architecture achieved 45% overall accuracy across 10 traffic classes and delivered strong detection performance for sophisticated threats most notably, a 76% F1-score for pulsing UDP flood attacks.
2. **Standards-Compliant Implementation**: The system leverages E2SM-KPM telemetry and domain-aware feature engineering based on PRB utilization and throughput efficiency metrics. By adhering to standardized O-RAN interfaces, the xApp supports cross-vendor compatibility while achieving real-time inference performance, with an average latency of 47 milliseconds.
3. **Comprehensive Attack Classification**: To our knowledge, this is the first O-RAN security system to offer fine-grained classification across six distinct attack types, including coordinated flooding patterns, fragmentation-based exploits, and adaptive behaviors that evade simpler detection techniques.
4. **Practical Deployment Framework**: This research provided a production-ready xApp implementation optimized for Near-RT RIC environments. The design balances detection precision with computational efficiency, demonstrating scalability, bounded memory usage, and compatibility with resource-constrained network infrastructure.

6.2 Security Effectiveness and Performance Analysis

The experimental results validate the proposed cascading classification architecture and underscore its effectiveness in detecting high-priority threats within O-RAN environments. While there remain opportunities for refinement, particularly in benign traffic classification, the system performs reliably in identifying sophisticated and coordinated attack patterns the primary security concern in such deployments.

6.2.1 Attack Detection Capabilities

The comparative success in detecting malicious behavior over benign classification suggests that adversarial traffic introduces more pronounced and consistent anomalies in network telemetry. This outcome supports the design rationale of the system, which prioritizes security-focused classification accuracy over exhaustive service labeling.

Security Effectiveness. The system demonstrates strong detection capabilities for complex, coordinated threats. Pulsing UDP floods were identified with a 76% F1-score, and other advanced attack types including parallel floods and fragmentation-based exploits achieved F1-scores in the 55 - 63% range. The ability to recognize adaptive attack patterns, which intentionally vary behavior to evade detection, underscores the effectiveness of temporal feature engineering and sequence-based modeling.

Binary Classification Success. Stage 1 of the pipeline, which performs binary classification between benign and malicious traffic, achieved promising results. The Transformer-based model reached 73% accuracy and a macro F1-score of 0.72. This performance illustrates that E2SM-KPM telemetry provides a sufficiently rich signal for initial threat identification, forming a robust foundation for subsequent fine-grained classification stages.

Service Classification Challenges. Performance in Stage 2a benign service classification was more limited. The highest-performing model yielded a macro F1-score of 0.443. VoIP traffic classification stood out with 67 - 71% F1-scores due to its distinct temporal signature. However, services such as eMBB, URLLC, and mMTC proved more difficult to differentiate, likely due to overlapping resource usage patterns and dynamic traffic behaviors common in real-world 5G deployments.

Attack Type Identification Limitations. Stage 2b, which classifies specific types of malicious behavior, achieved a macro F1-score of 0.305. While pulsing UDP floods were detected with high reliability (82% recall, 76% F1-score), other attacks showed lower precision and recall. Nonetheless, the system showed clear advantages in identifying coordinated, complex attack strategies over more basic patterns.

Operational Implications. Although the system's overall accuracy stands at 45%, this metric does not fully reflect its practical utility in a security context. Its strength in detecting high-risk, coordinated threats where F1-scores range from 55% to 76% makes it a valuable asset for real-time threat mitigation in O-RAN deployments. In contrast, inaccuracies in benign traffic classification pose minimal operational risk, whereas failure to detect malicious activity could result in severe service disruption or compromise.

6.3 Comparative Analysis with Previous Work

To contextualize the results, this work compares with related O-RAN security and AI-driven intrusion detection systems.

Binary Attack Detection Performance. In Stage 1 of our pipeline, the Transformer-based binary classifier achieved an accuracy of 73%. This result is notable given the reliance on E2SM-KPM telemetry, which prioritizes standards compliance and real-time integration within O-RAN environments. In contrast, Xavier et al. [33] reported a higher accuracy of 96% using Random Forest models trained on physical layer metrics such as CQI, SINR, and MCS features that inherently offer stronger signal-level discrimination but are less aligned with standardized O-RAN telemetry interfaces.

This approach emphasizes compatibility with the E2 interface and practical deployability within the Near-RT RIC, which makes it more suitable for real-world O-RAN deployments. Additionally, Awad et al. [4] proposed a CNN-based dual-xApp framework for DDoS detection in 5G-V2X environments; however, their study did not report concrete accuracy metrics, making direct comparison difficult. Nonetheless, their architecture reinforces the growing trend toward AI-driven, modular security solutions within O-RAN ecosystems.

Table 6.1: Binary Classification Performance Comparison in O-RAN Security Literature

Study	Method	Accuracy	Precision	Recall	Latency	Features
[33]	Random Forest	96%	N/R	N/R	3.62ms	PHY/MAC
[4]	CNN Dual-xApp	N/R	N/R	N/R	N/R	DDoS Patterns
[18]	RNN Autoencoder	N/R	N/R	N/R	N/R	Unsupervised
Our Work	Transformer	73%	69%	67%	47ms	E2SM-KPM
Our Work	AttentionLSTM	72%	67%	68%	45ms	E2SM-KPM
Our Work	CNN-LSTM	70%	65%	63%	52ms	E2SM-KPM

Multi-class Classification Complexity. Our cascading architecture achieved 45% overall accuracy across 10 traffic classes comprising four benign services and six distinct attack types. This represents a substantially more complex classification task than binary anomaly detection approaches commonly found in the literature. For instance, Kouchaki et al. [18] employed RNN autoencoders for binary threat detection, while Alshagri et al. [3] achieved 90.93% accuracy using Decision Trees for network slicing classification. However, the latter focused on resource optimization, not security, and did not address multi-class or adversarial scenarios.

Attack Type Identification. In Stage 2b, the system achieved a macro F1-score of 0.305 across six attack categories. Pulsing UDP floods and fragmentation-based attacks were identified most accurately, with F1-scores of 76% and 63%, respectively. This level of granularity surpasses most prior O-RAN security research, which typically concentrates on binary detection. This approach complements spatial localization efforts such as Ratti et al. [29], whose sub-2-meter malicious UE geolocation can be effectively paired with the behavioral classification pipeline.

Real-time Performance Comparison. The detection pipeline meets Near-RT RIC requirements with an average end-to-end response time of 47 milliseconds, including telemetry ingestion, feature computation, and classification. By comparison, Xavier et al. [33] reported a 3.62ms control loop latency, though their measurement focused exclusively on control signaling, not the full classification pipeline. Similarly, D'Oro et al. [9] achieved sub-4ms latency within DU/CU environments but prioritized ultra-low-latency processing for functional orchestration rather than comprehensive security analytics.

Feature Engineering Approaches. The telemetry-based feature engineering leverages E2SM-KPM metrics, with a focus on PRB utilization patterns, throughput efficiency, and temporal signal sequences. This design aligns with O-RAN interface standards and enables scalable cross-vendor deployment. Feature importance analysis identified PRB utilization efficiency and throughput-to-PRB ratios as the most discriminative indicators for threat detection. These differ notably from the radio-quality-focused metrics (e.g., CQI, SINR) used in Xavier et al. [33], highlighting the strength of the mid-layer telemetry approach in supporting real-time, standards-compliant threat detection.

Table 6.2: Feature Engineering Approaches Comparison in O-RAN Security Research

Study	**Data Source**	**Feature Type**	**Temporal**	**Standards**
[33]	PHY/MAC Layer	Radio Quality	Limited	Custom
[4]	Traffic Patterns	DDoS Signatures	Yes	Partial
[18]	Network Metrics	Autoencoder	Yes	Partial
[3]	Resource Metrics	Network Slicing	Limited	Simulation
Our Work	**E2SM-KPM**	**PRB + Throughput**	**Yes**	**Full**

Scalability and Deployment Considerations. The proposed xApp implementation is fully standards-compliant, leveraging the E2SM-KPM interface to ensure interoperability across diverse vendor environments. This design choice directly addresses cross-vendor deployment challenges identified in recent studies [13]. Furthermore, the cascading architecture supports computational scalability by distributing inference tasks across specialized, stage-specific models, as opposed to relying on a single monolithic classifier. This modular design aligns with architectural best practices for scalable ML integration in O-RAN, as advocated by Herrera et al. [12].

Security Framework Integration. Our contribution complements broader O-RAN security efforts by providing a targeted, machine learning-driven mechanism for real-time behavioral threat detection. It fits within the context of emerging frameworks such as Abdalla et al. [1]'s ZTRAN architecture, which emphasizes service authentication and secure slicing, and Wen et al. [32]'s integration of explainable AI into security pipelines. Additionally, there is strong integration potential with interface-level protections such as those proposed by Groen et al. [10], enabling the deployment of unified threat detection architectures that combine telemetry analysis with secure interface enforcement.

Table 6.3: Comparative analysis of O-RAN security approaches in the literature

Work	**Approach**	**Accuracy**	**Classes**	**Features**
[33]	Random Forest	96%	2 (Binary)	PHY/MAC
[4]	CNN Dual-xApp	N/R	2 (Binary)	DDoS Patterns
[18]	RNN Autoencoder	N/R	Anomaly	Unsupervised
[3]	Decision Trees	90.93%	4 (Slicing)	Resource Metrics
Our Work	Cascading ML	45%	10 (Services+Attacks)	E2SM-KPM

This comparative analysis reveals that while our overall accuracy is lower than some binary classification approaches, our work addresses significantly more complex problem space with greater practical deployment considerations. The standards-compliant implementation and comprehensive attack type classification represent novel contributions to O-RAN security research.

6.4 Limitations

While this research demonstrates meaningful progress in machine learning - based threat detection for O-RAN, several limitations remain. The overall classification accuracy of 45% across 10 traffic classes reflects the inherent complexity of the task and highlights areas for further improvement. Specifically, benign service classification achieved an F1-macro score of 44%, while attack type classification reached 31%. Certain benign traffic types particularly URLLC and mMTC were difficult to distinguish due to overlapping telemetry patterns.

Additionally, some simpler attacks, such as basic UDP floods, were nearly undetectable, achieving only 0 - 2% F1-scores.

The experimental scope was constrained by the use of the srsRAN platform and a testbed limited to three concurrently emulated UEs. While this setup enabled controlled experimentation and reproducibility, it does not fully reflect the scale or diversity of production O-RAN environments. The dataset of 15,168 labeled samples, while substantial for a lab-scale study, may not capture the full variability encountered in real-world deployments.

Binary classification performance, with accuracy ranging between 67% and 73%, indicates that false positives remain a possibility in live deployments, particularly during periods of atypical but benign traffic behavior. Moreover, the feature engineering process relied exclusively on E2SM-KPM telemetry, potentially overlooking valuable indicators available at other network layers, such as the physical, control, or management planes.

Finally, the current system focuses exclusively on user-plane threats. Expanding the threat model to include control-plane exploits, management-plane intrusions, and cross-layer attack strategies will be critical for developing a more comprehensive security solution for future O-RAN networks.

6.5 Future Work

Building upon the findings and limitations of this work, several promising directions emerge for advancing machine learning - driven security in O-RAN environments.

Advanced Machine Learning Techniques. Future research should explore advanced ML paradigms to enhance model robustness and adaptability. Federated learning offers a privacy-preserving approach to training across multiple operators, supporting collaborative defense strategies without centralized data sharing [27]. Graph neural networks may enable more effective modeling of interactions between UEs, capturing structural patterns indicative of coordinated or distributed attacks. Additionally, incorporating adversarial machine learning defenses could harden the detection pipeline against evasion tactics, while continual learning methods may facilitate real-time adaptation to evolving traffic and threat patterns.

System Integration and Deployment Expansion. From a systems perspective, integration with broader O-RAN architectures presents several avenues for enhancement. Coordinated detection across multiple RIC instances [28] could support scalable, distributed security monitoring. Edge intelligence, as demonstrated in low-latency orchestration frameworks [9], may improve real-time responsiveness. Adapting to evolving 5G and future 6G traffic characteristics [31], along with integration into zero-trust frameworks [1], will be critical for ensuring long-term applicability.

Operational and Regulatory Readiness. Operational improvements should prioritize explainable AI techniques [32] to support transparency, auditability, and regulatory compliance in ML-driven decisions. Automated mitigation workflows [4], hardware acceleration strategies for deployment at scale, and active participation in the evolution of O-RAN standards [12] will all contribute to making intelligent security solutions viable for commercial adoption.

6.6 Concluding Remarks

This research advances the field of AI-driven security for Open Radio Access Networks (O-RAN) by addressing a key gap between binary threat detection and the operational complexity of multi-class classification. Through the development and evaluation of a novel cascading machine learning architecture, this work demonstrates that it is feasible to achieve fine-grained threat classification in real time, using standardized telemetry and resource-efficient models.

The proposed system achieves 45% overall accuracy across ten traffic classes four benign and six malicious marking a significant step forward in addressing the full spectrum of operational traffic behavior. Notably, the system demonstrates strong performance in identifying sophisticated and coordinated attacks, achieving a 76% F1-score on pulsing UDP floods and 55 - 63% F1-scores across other advanced threat types. These results establish a new benchmark for multi-class threat detection in O-RAN environments.

The implementation adheres to O-RAN specifications, using E2SM-KPM telemetry and integrating as a deployable xApp within the Near-RT RIC framework. This standards-compliant approach not only ensures cross-vendor compatibility but also bridges the gap between academic research and real-world deployment readiness. The modular cascading architecture enhances scalability, making the system adaptable to new traffic types and evolving threat models.

Beyond technical contributions, this work positions itself within the broader O-RAN security ecosystem. It complements existing efforts in interface protection, UE localization, and zero-trust frameworks, offering integration potential that extends the system's utility in production environments.

Ultimately, this research lays a foundation for the next generation of AI-driven security systems in open, disaggregated wireless networks. By balancing detection accuracy, deployment practicality, and alignment with industry standards, it provides a viable path forward for securing future 5G and 6G O-RAN deployments.

Bibliography

[1] Aly S. Abdalla, Joshua Moore, Nisha Adhikari, and Vuk Marojevic. ZTRAN: Prototyping Zero Trust Security xApps for Open Radio Access Network Deployments, March 2024. URL http://arxiv.org/abs/2403.04113. arXiv:2403.04113 [cs] version: 1.

[2] Aly Sabri Abdalla and Vuk Marojevic. End-to-End O-RAN Security Architecture, Threat Surface, Coverage, and the Case of the Open Fronthaul. *IEEE Communications Standards Magazine*, 8(1):36–43, March 2024. ISSN 2471-2825, 2471-2833. doi: 10.1109/MCOMSTD.0001.2200047. URL https://ieeexplore.ieee.org/document/10467183/.

[3] Aljuhara Alshagri, Nouf Abdulaziz Jaafar, Amjad F. Alsuhaim, Haya Alnukhaylan, and Areag Fahad Albeechi. Evaluation of AI Models for Intelligent Network Slicing in OpenRAN: A Comprehensive Benchmarking Approach. pages 1–5, November 2024. doi: 10.1109/ICNGN63705.2024.10871261. URL https://ieeexplore.ieee.org/document/10871261/.

[4] Mirna Awad, Adam Ait Hamid, Yeogeuch Ranganathan, Nizar Choubik, Rami Langar, and Wael Jaafar. xApps for DDoS Attacks Detection and Mitigation in 5G-V2X O-RAN Networks. pages 1–2, October 2024. doi: 10.1109/CIoT63799.2024.10757133. URL https://ieeexplore.ieee.org/document/10757133/. ISSN: 2159-6972.

[5] Jose A. Ayala-Romero, Andrés García-Saavedra, Marco Gramaglia, Xavier Costa-Pérez, Albert Banchs, and Juan J. Alcaraz. VrAIn: Deep Learning based Orchestration for Computing and Radio Resources in vRANs. *IEEE Transactions on Mobile Computing*, 20(10):2992–3006, 2020. ISSN 1536-1233. doi: 10.1109/TMC.2020.3043100.

[6] Jose A. Ayala-Romero, Andres Garcia-Saavedra, Xavier Costa-Perez, and George Iosifidis. Bayesian Online Learning for Energy-Aware Resource Orchestration in Virtualized RANs. pages 1–10, May 2021. doi: 10.1109/INFOCOM42981.2021.9488845. URL https://ieeexplore.ieee.org/document/9488845/.

[7] Wilfrid Azariah, Fransiscus Asisi Bimo, Chih-Wei Lin, Ray-Guang Cheng, Navid Nikaein, and Rittwik Jana. A Survey on Open Radio Access Networks: Challenges, Research Directions, and Open Source Approaches. *Sensors*, 24(3):1038, February 2024. ISSN 1424-8220. doi: 10.3390/s24031038. URL https://www.mdpi.com/1424-8220/24/3/1038.

[8] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, 290(2):405–421, April 2021. ISSN 03772217. doi: 10.1016/j.ejor.2020.07.063. URL https://linkinghub.elsevier.com/retrieve/pii/S0377221720306895.

[9] Salvatore D'Oro, Michele Polese, Leonardo Bonati, Hai Cheng, and Tommaso Melodia. dApps: Distributed Applications for Real-time Inference and Control in O-RAN. *IEEE Communications Magazine*, 60(11):52–58, November 2022. ISSN 0163-6804,

1558-1896. doi: 10.1109/MCOM.002.2200079. URL http://arxiv.org/abs/2203.02370. arXiv:2203.02370 [cs].

[10] Joshua Groen, Salvatore D'Oro, Utku Demir, Leonardo Bonati, Michele Polese, Tommaso Melodia, and Kaushik Chowdhury. Implementing and Evaluating Security in O-RAN: Interfaces, Intelligence, and Platforms. *IEEE Network*, 39(1):227–234, January 2025. ISSN 0890-8044, 1558-156X. doi: 10.1109/MNET.2024.3434419. URL https://ieeexplore.ieee.org/document/10613612/.

[11] Nessrine Hammami and Kim Khoa Nguyen. On-Policy vs. Off-Policy Deep Reinforcement Learning for Resource Allocation in Open Radio Access Network. pages 1461–1466, April 2022. doi: 10.1109/WCNC51071.2022.9771605. URL https://ieeexplore.ieee.org/document/9771605/.

[12] Juan Luis Herrera, Sofia Montebugnoli, Paolo Bellavista, and Luca Foschini. Enabling Reusable and Comparable xApps in the Machine Learning-Driven Open RAN. pages 37–42, July 2024. doi: 10.1109/HPSR62440.2024.10635962. URL https://ieeexplore.ieee.org/document/10635962/. ISSN: 2325-5609.

[13] Marcin Hoffmann, Salim Janji, Adam Samorzewski, Lukasz Kulacz, Cezary Adamczyk, Marcin Dryjański, Pawel Kryszkiewicz, Adrian Kliks, and Hanna Bogucka. Open RAN xApps Design and Evaluation: Lessons Learnt and Identified Challenges, November 2023. URL http://arxiv.org/abs/2311.04380. arXiv:2311.04380 [cs].

[14] Cheng-Feng Hung, You-Run Chen, Chi-Heng Tseng, and Shin-Ming Cheng. Security Threats to xApps Access Control and E2 Interface in O-RAN. *IEEE Open Journal of the Communications Society*, 5:1197–1203, 2024. ISSN 2644-125X. doi: 10.1109/OJCOMS.2024.3364840. URL https://ieeexplore.ieee.org/document/10433004/.

[15] Roghayeh Joda, Turgay Pamuklu, Pedro Enrique Iturria-Rivera, and Melike Erol-Kantarci. Deep Reinforcement Learning-Based Joint User Association and CU–DU Placement in O-RAN. *IEEE Transactions on Network and Service Management*, 19(4): 4097–4110, December 2022. ISSN 1932-4537, 2373-7379. doi: 10.1109/TNSM.2022.3221670. URL https://ieeexplore.ieee.org/document/9946423/.

[16] Nei Kato, Bomin Mao, Fengxiao Tang, Yuichi Kawamoto, and Jiajia Liu. Ten Challenges in Advancing Machine Learning Technologies toward 6G. *IEEE Wireless Communications*, 27(3):96–103, June 2020. ISSN 1536-1284, 1558-0687. doi: 10.1109/MWC.001.1900476. URL https://ieeexplore.ieee.org/document/9061001/.

[17] Fatemeh Kavehmadavani, Van-Dinh Nguyen, Thang X. Vu, and Symeon Chatzinotas. Intelligent Traffic Steering in Beyond 5G Open RAN based on LSTM Traffic Prediction. *IEEE Transactions on Wireless Communications*, 22(9):6182–6197, 2023. ISSN 1536-1276. doi: 10.1109/TWC.2023.3254903.

[18] Mohammadreza Kouchaki, Joshua Moore, Minglong Zhang, and Vuk Marojevic. Advancing O-RAN Security: Integrated Intrusion Detection and Secure Slicing xApps. pages 1–2, May 2024. doi: 10.1109/INFOCOMWKSHPS61880.2024.10620781. URL https://ieeexplore.ieee.org/document/10620781/. ISSN: 2833-0587.

[19] Hoejoo Lee, Jiwon Cha, Daeken Kwon, Myeonggi Jeong, and Intaik Park. Hosting AI/ML Workflows on O-RAN RIC Platform. pages 1–6, December 2020. doi: 10.1109/GCWkshps50303.2020.9367572. URL https://ieeexplore.ieee.org/document/9367572/.

[20] Xuanyu Liang, Ahmed Al-Tahmeesschi, Qiao Wang, Swarna Chetty, Chenrui Sun, and Hamed Ahmadi. Enhancing Energy Efficiency in O-RAN Through Intelligent xApps Deployment, May 2024. URL http://arxiv.org/abs/2405.10116. arXiv:2405.10116 [eess].

[21] Madhusanka Liyanage, An Braeken, Shahriar Shahabuddin, and Pasika Ranaweera. Open RAN security: Challenges and opportunities. *Journal of Network and Computer Applications*, 214:103621, May 2023. ISSN 10848045. doi: 10.1016/j.jnca.2023.103621. URL https://linkinghub.elsevier.com/retrieve/pii/S1084804523000401.

[22] Mario Martínez-Morfa, Carlos Ruiz De Mendoza, Cristina Cervelló-Pastor, and Sebastià Sallent. DRL-based xApps for Dynamic RAN and MEC Resource Allocation and Slicing in O-RAN. pages 106–114, October 2024. doi: 10.1109/NoF62948.2024.10741435. URL https://ieeexplore.ieee.org/document/10741435/. ISSN: 2833-0072.

[23] Timothy J. O'Shea and Jakob Hoydis. An Introduction to Deep Learning for the Physical Layer. *IEEE Transactions on Cognitive Communications and Networking*, 3(4): 563–575, 2017. ISSN 2332-7731. doi: 10.1109/TCCN.2017.2758370.

[24] Turgay Pamuklu, Melike Erol-Kantarci, and Cem Ersoy. Reinforcement Learning Based Dynamic Function Splitting in Disaggregated Green Open RANs. pages 1–6, June 2021. doi: 10.1109/ICC42927.2021.9500721. URL http://arxiv.org/abs/2012.03213. arXiv:2012.03213 [cs].

[25] Michele Polese, Leonardo Bonati, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. ColO-RAN: Developing Machine Learning-based xApps for Open RAN Closed-loop Control on Programmable Experimental Platforms, January 2022. URL http://arxiv.org/abs/2112.09559. arXiv:2112.09559 [cs] version: 2.

[26] Michele Polese, Leonardo Bonati, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges. *IEEE Communications Surveys and Tutorials*, 25(2):1376–1423, 2023. ISSN 1553-877X. doi: 10.1109/COMST.2023.3239220.

[27] Keyvan Ramezanpour and Jithin Jagannath. Intelligent zero trust architecture for 5G/6G networks: Principles, challenges, and the role of machine learning in the context of O-RAN. *Computer Networks*, 217:109358, November 2022. ISSN 13891286. doi: 10.1016/j.comnet.2022.109358. URL https://linkinghub.elsevier.com/retrieve/pii/S1389128622003929.

[28] S. Ratti, A. Marotta, W. Tiberti, C. Centofanti, D. Cassioli, and F. Graziosi. Integrating O-RAN and SIEM for Unified Detection of IT and Mobile Network Attacks. pages 1–2, January 2025. doi: 10.1109/CCNC54725.2025.10976103. URL https://ieeexplore.ieee.org/document/10976103/.

[29] Steve Azarsh Ratti, Antonio Tuzi, Alex Piccioni, Andrea Marotta, Dajana Cassioli, and Fabio Graziosi. Exploiting O-RAN as an Effective Solution for Malicious User Localization. pages 301–307, January 2025. doi: 10.1109/COMSNETS63942.2025.10885563. URL https://ieeexplore.ieee.org/document/10885563/.

[30] Sameer Kumar Singh, Rohit Singh, and Brijesh Kumbhani. The Evolution of Radio Access Network Towards Open-RAN: Challenges and Opportunities. pages 1–6, May 2020. doi: 10.1109/WCNCW48565.2020.9124820.

[31] John S. Vardakas, Kostas Ramantas, Evgenii Vinogradov, Md Arifur Rahman, Adam Girycki, Sofie Pollin, Simon Pryor, Philippe Chanclou, and Christos Verikoukis. Machine Learning-Based Cell-Free Support in the O-RAN Architecture: An Innovative Converged Optical-Wireless Solution Toward 6G Networks. *IEEE Wireless Communications*, 29(5):20–26, October 2022. ISSN 1536-1284, 1558-0687. doi: 10.1109/MWC.002.2200026. URL https://ieeexplore.ieee.org/document/9979701/.

[32] Haohuang Wen, Prakhar Sharma, Vinod Yegneswaran, Phillip Porras, Ashish Gehani, and Zhiqiang Lin. 6G-XSec: Explainable Edge Security for Emerging OpenRAN Architectures. pages 77–85, November 2024. doi: 10.1145/3696348.3696881. URL https://dl.acm.org/doi/10.1145/3696348.3696881.

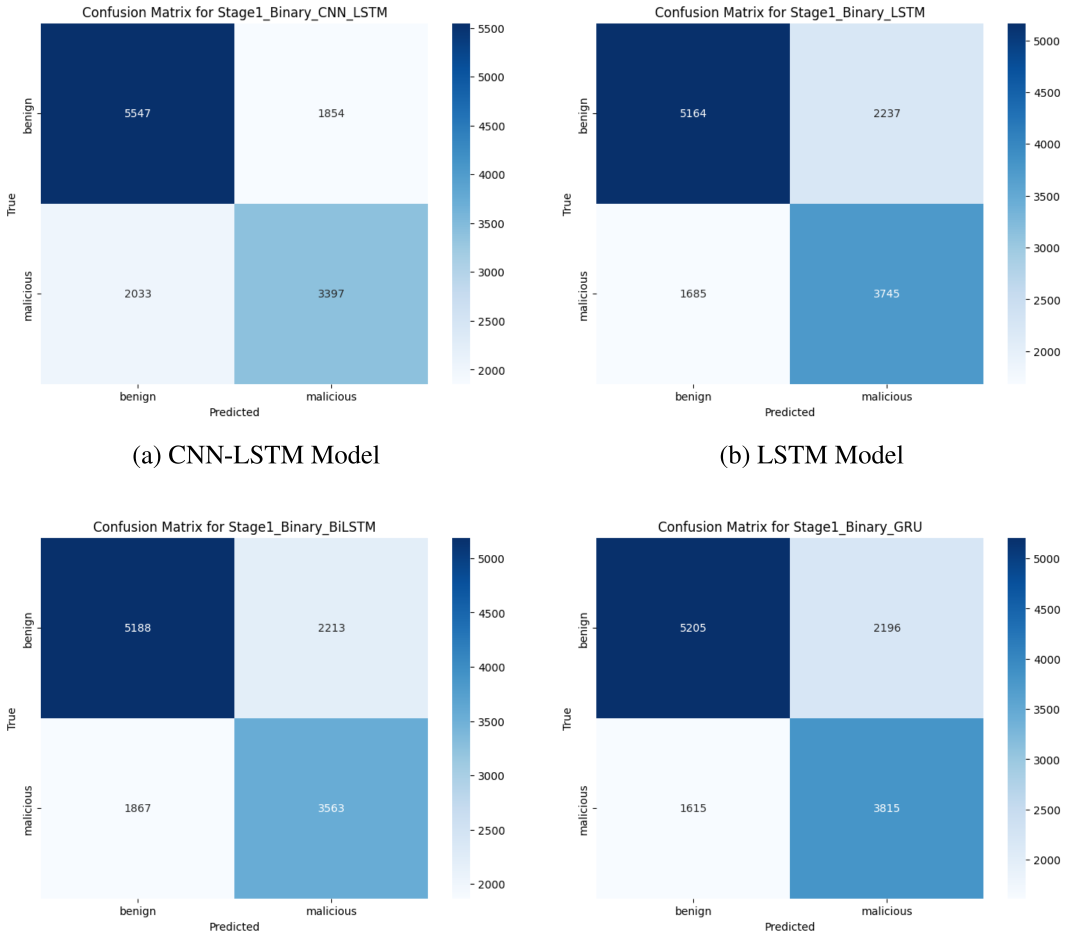
[33] Bruno Missi Xavier, Merim Dzaferagic, Diarmuid Collins, Giovanni Comarela, Magnos Martinello, and Marco Ruffini. Machine learning-based early attack detection using open ran intelligent controller. 2023. doi: 10.48550/arXiv.2302.01864. URL https://arxiv.org/abs/2302.01864. arXiv:2302.01864 [cs.NI].

Appendix A: Confusion Matrix Results

A.1 Stage 1: Binary Classification Results

The following confusion matrices show the performance of different model architectures for the binary classification task (benign vs malicious traffic detection) in Stage 1 of the cascading pipeline.

A.1.1 Deep Learning Models



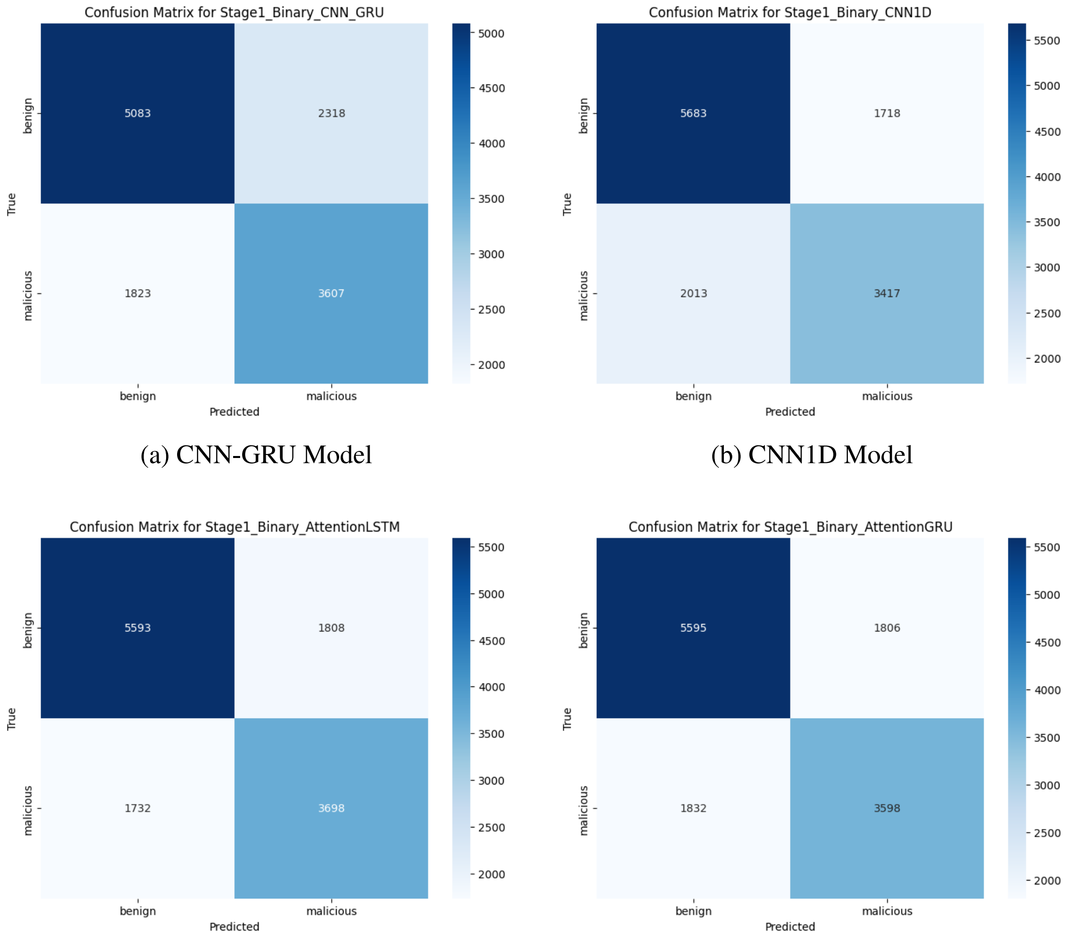
(a) CNN-LSTM Model

(b) LSTM Model

(c) BiLSTM Model

(d) GRU Model

Figure A.1: Confusion matrices for recurrent neural network architectures in Stage 1 binary classification.



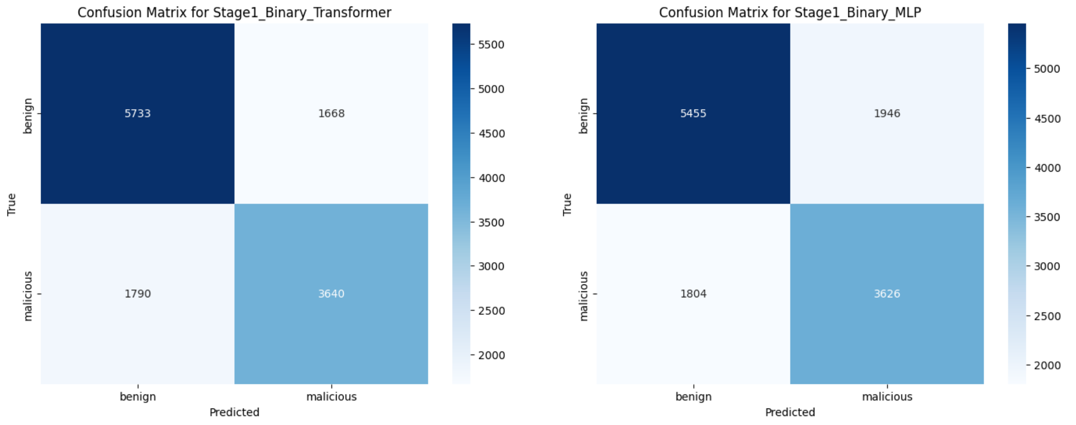
(a) CNN-GRU Model

(b) CNN1D Model

(c) Attention-LSTM Model

(d) Attention-GRU Model

Figure A.2: Confusion matrices for hybrid and attention-based architectures in Stage 1 binary classification.

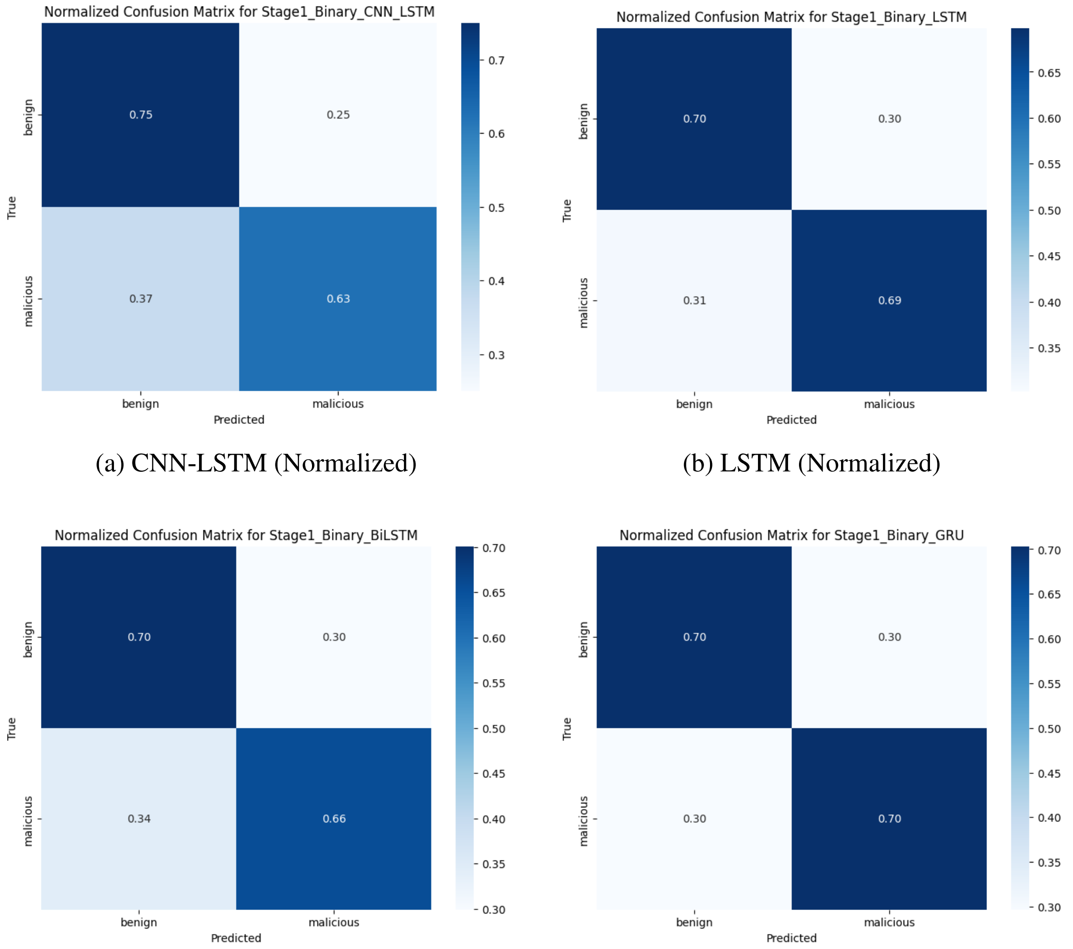


(a) Transformer Model

(b) MLP Model

Figure A.3: Confusion matrices for transformer and feedforward architectures in Stage 1 binary classification.

A.1.2 Normalized Confusion Matrices



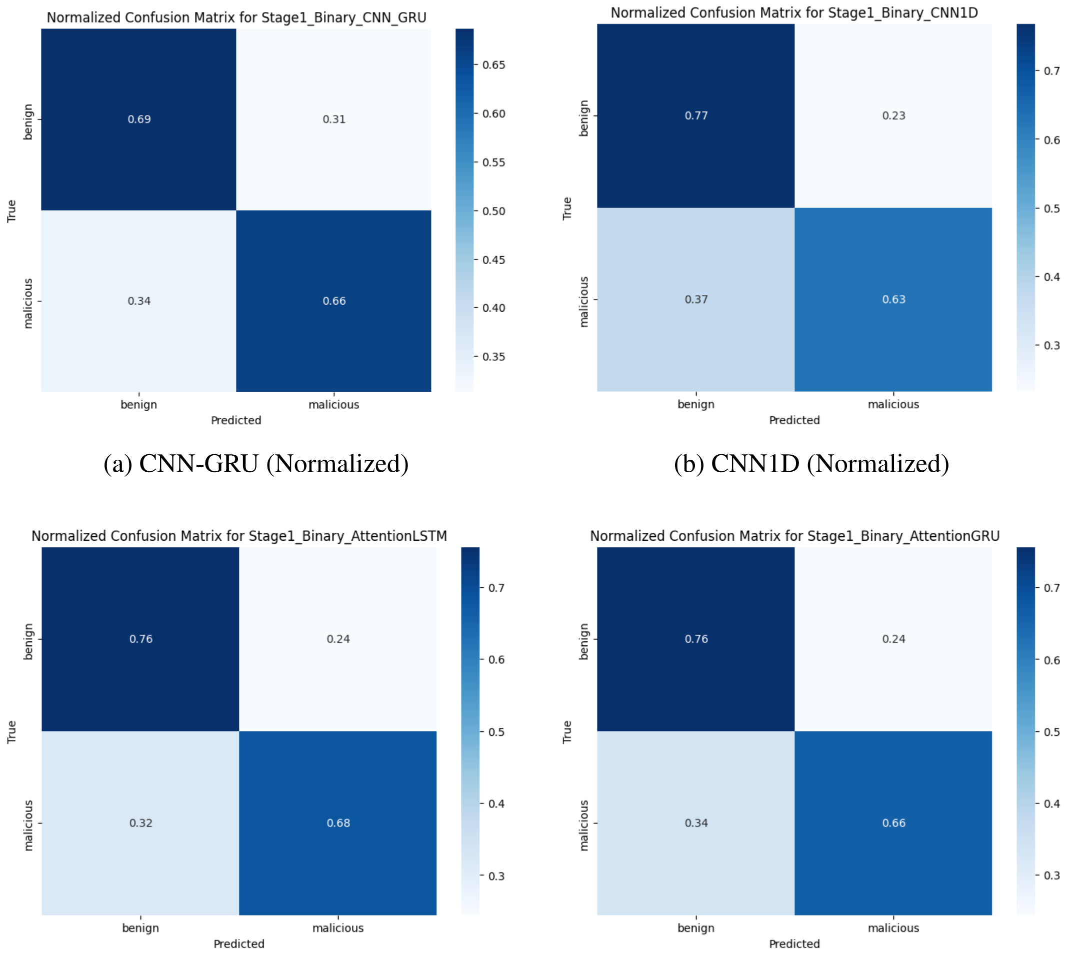
(a) CNN-LSTM (Normalized)

(b) LSTM (Normalized)

(c) BiLSTM (Normalized)

(d) GRU (Normalized)

Figure A.4: Normalized confusion matrices for recurrent neural network architectures in Stage 1.



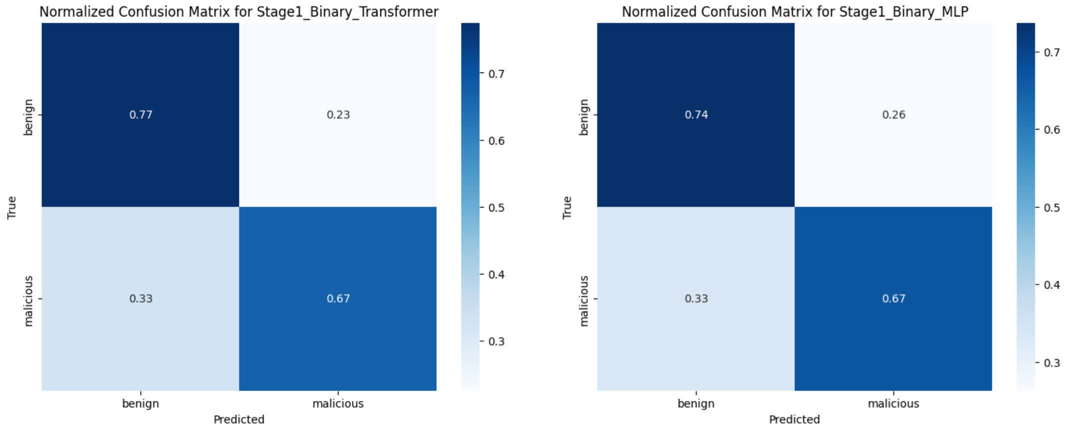
(a) CNN-GRU (Normalized)

(b) CNN1D (Normalized)

(c) Attention-LSTM (Normalized)

(d) Attention-GRU (Normalized)

Figure A.5: Normalized confusion matrices for hybrid and attention-based architectures in Stage 1.



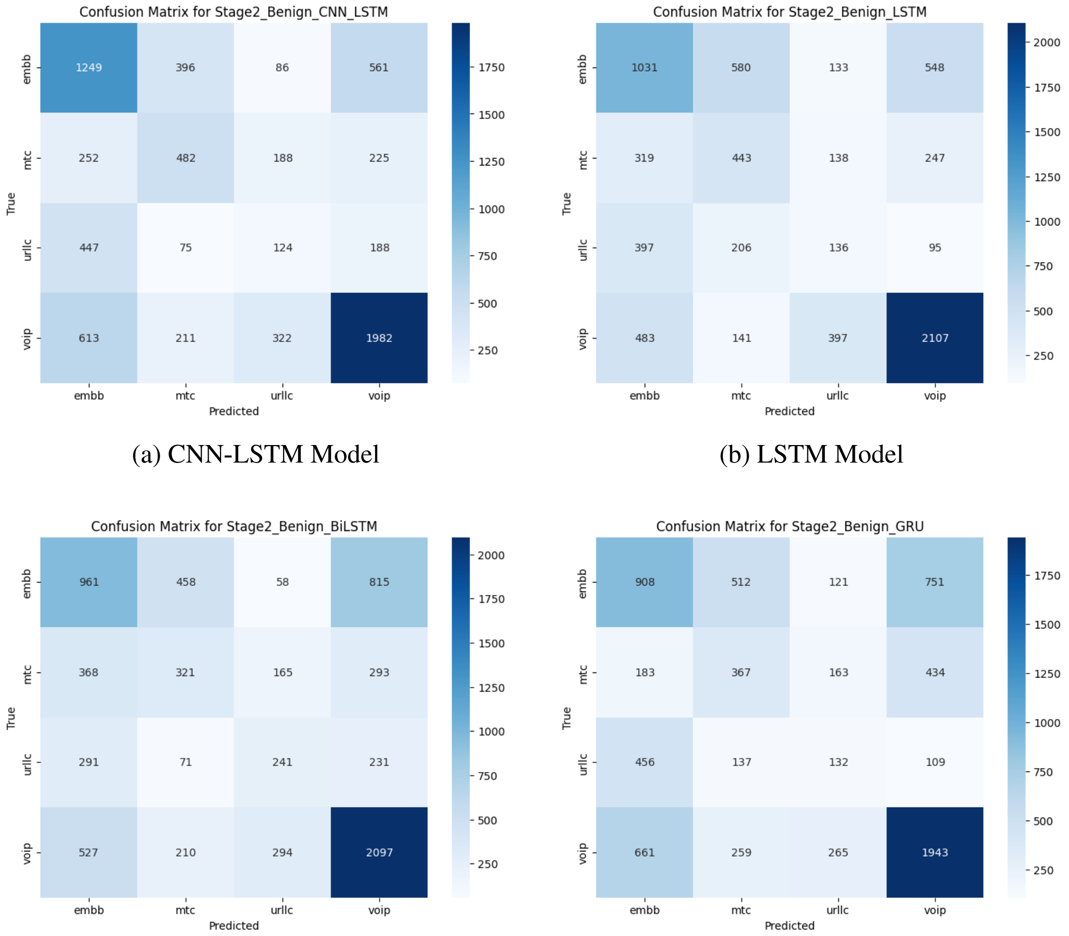
(a) Transformer (Normalized)

(b) MLP (Normalized)

Figure A.6: Normalized confusion matrices for transformer and feedforward architectures in Stage 1.

A.2 Stage 2a: Benign Traffic Classification Results

The following confusion matrices show the performance of different model architectures for classifying benign traffic into service types (eMBB, URLLC, mMTC, VoIP) in Stage 2a of the cascading pipeline.



	embb	mtc	urllc	voip
embb	1249	396	86	561
mtc	252	482	188	225
urllc	447	75	124	188
voip	613	211	322	1982

(a) CNN-LSTM Model

	embb	mtc	urllc	voip
embb	1031	580	133	548
mtc	319	443	138	247
urllc	397	206	136	95
voip	483	141	397	2107

(b) LSTM Model

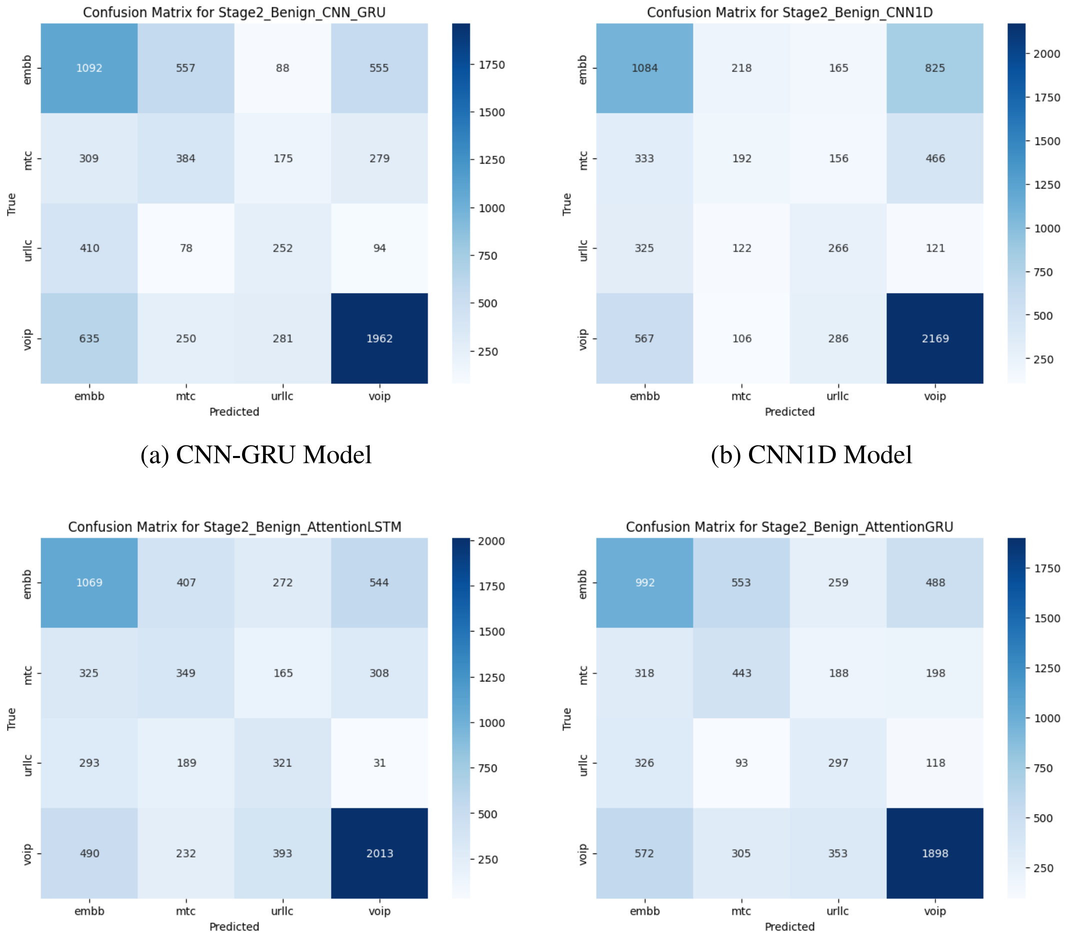
	embb	mtc	urllc	voip
embb	961	458	58	815
mtc	368	321	165	293
urllc	291	71	241	231
voip	527	210	294	2097

(c) BiLSTM Model

	embb	mtc	urllc	voip
embb	908	512	121	751
mtc	183	367	163	434
urllc	456	137	132	109
voip	661	259	265	1943

(d) GRU Model

Figure A.7: Confusion matrices for recurrent neural network architectures in Stage 2a benign classification.



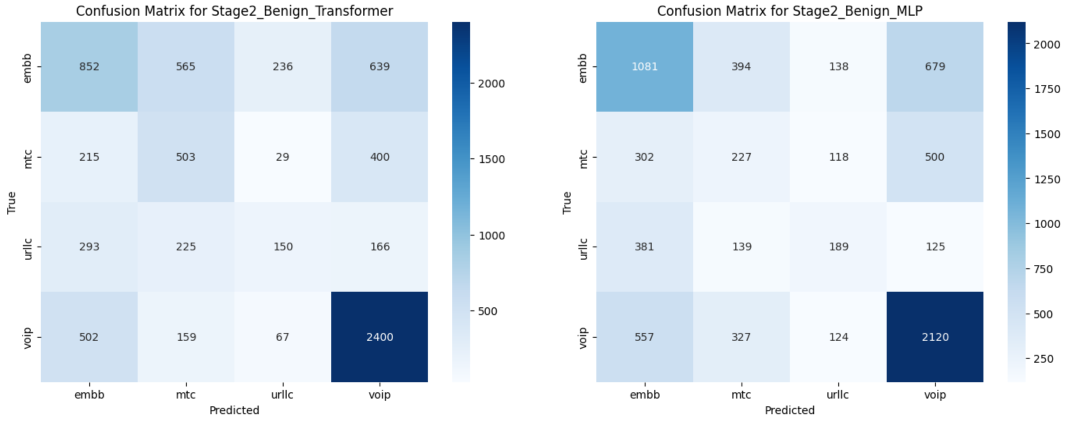
(a) CNN-GRU Model

(b) CNN1D Model

(c) Attention-LSTM Model

(d) Attention-GRU Model

Figure A.8: Confusion matrices for hybrid and attention-based architectures in Stage 2a benign classification.



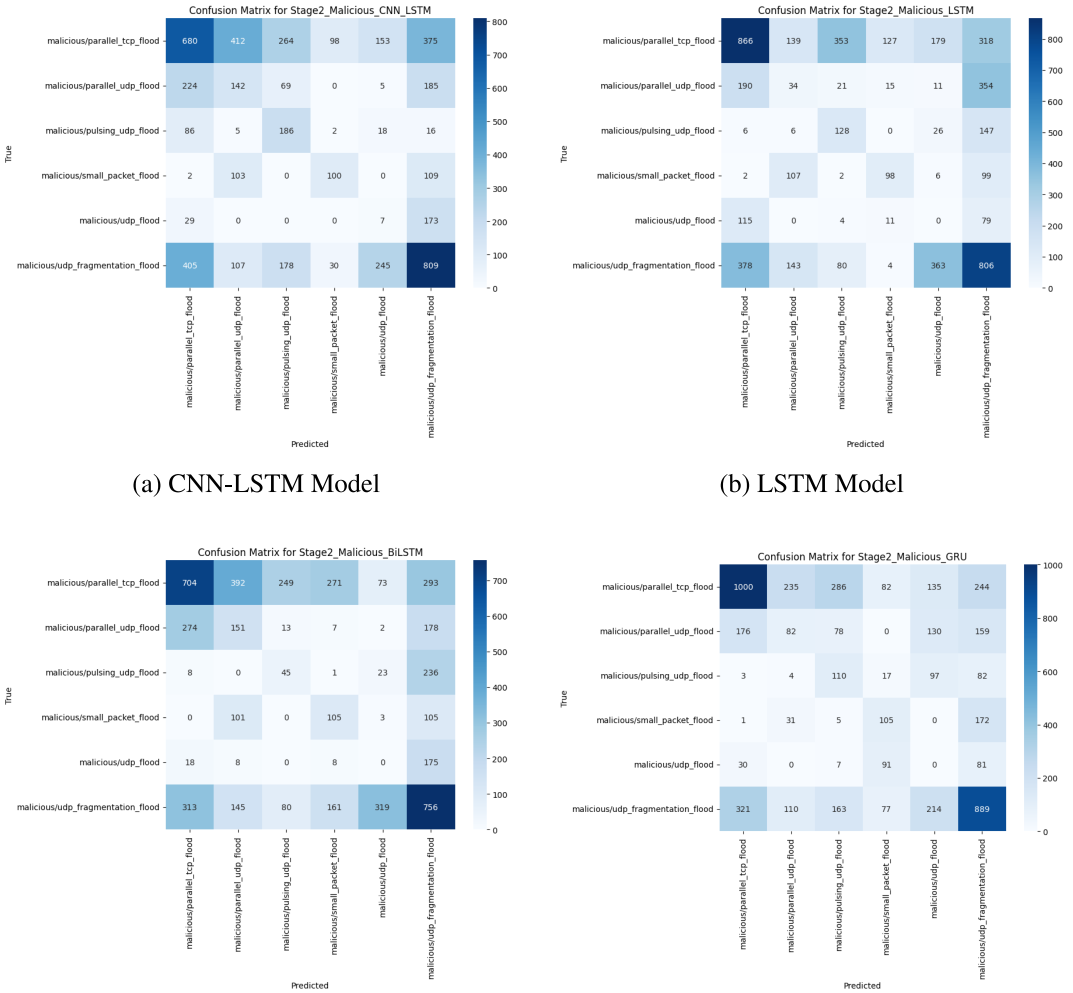
(a) Transformer Model

(b) MLP Model

Figure A.9: Confusion matrices for transformer and feedforward architectures in Stage 2a benign classification.

A.3 Stage 2b: Malicious Traffic Classification Results

The following confusion matrices show the performance of different model architectures for classifying malicious traffic into attack types in Stage 2b of the cascading pipeline.



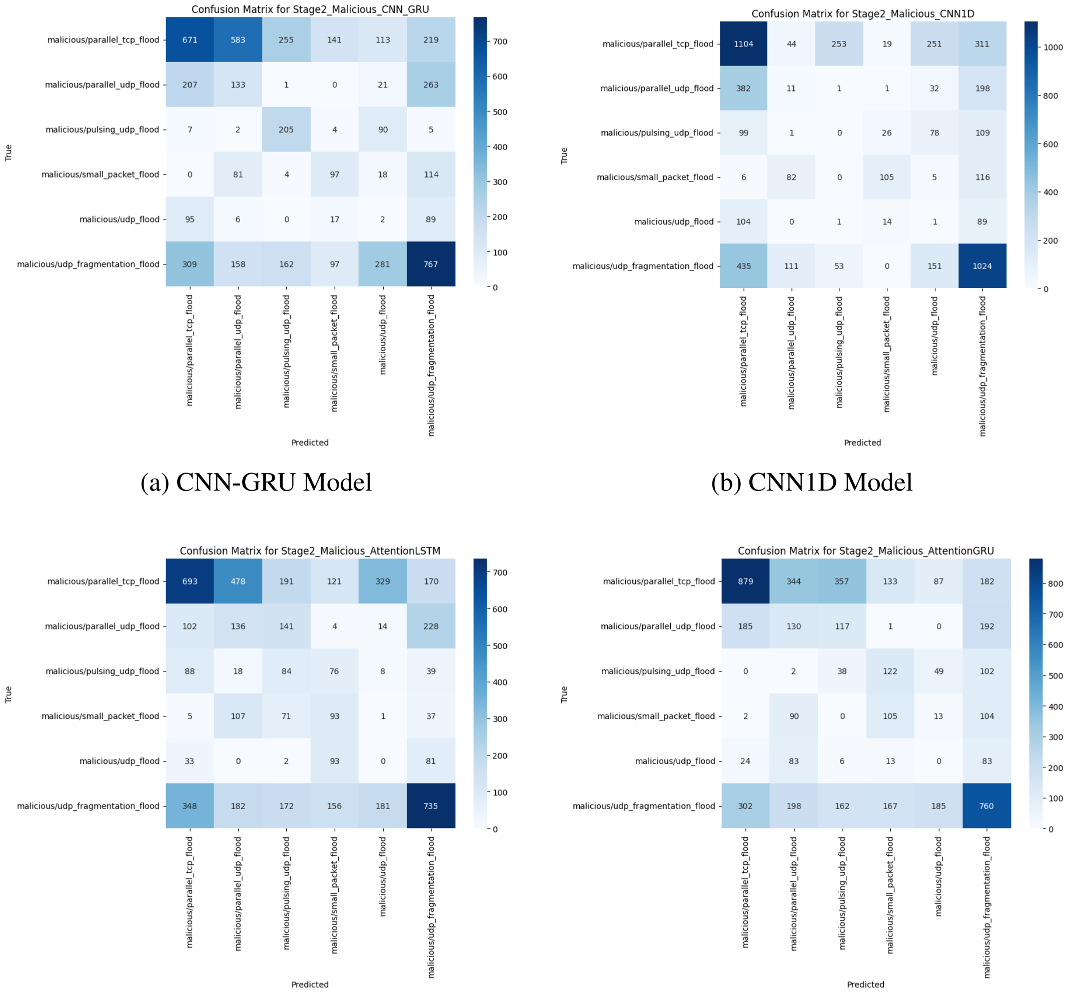
(a) CNN-LSTM Model

(b) LSTM Model

(c) BiLSTM Model

(d) GRU Model

Figure A.10: Confusion matrices for recurrent neural network architectures in Stage 2b malicious classification.



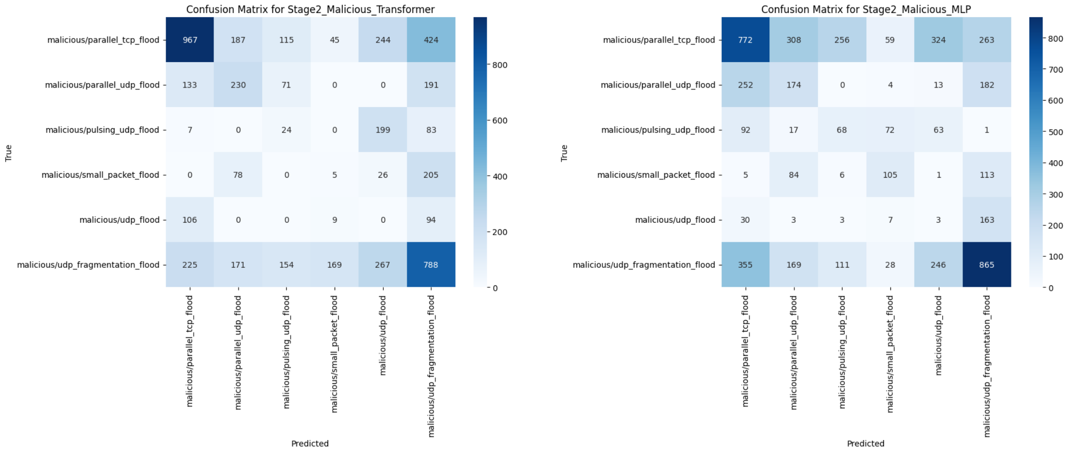
(a) CNN-GRU Model

(b) CNN1D Model

(c) Attention-LSTM Model

(d) Attention-GRU Model

Figure A.11: Confusion matrices for hybrid and attention-based architectures in Stage 2b malicious classification.



(a) Transformer Model

(b) MLP Model

Figure A.12: Confusion matrices for transformer and feedforward architectures in Stage 2b malicious classification.

A.4 Stage 3: Cascading Pipeline Results

The following confusion matrix shows the overall performance of the complete three-stage cascading classification pipeline.

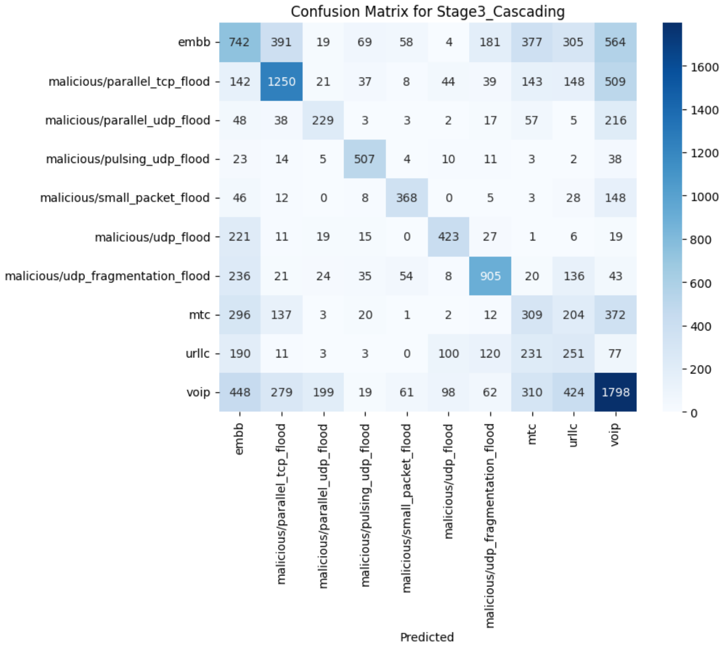


Figure A.13: Confusion matrix for the complete Stage 3 cascading classification pipeline showing end-to-end performance across all traffic classes.