# Implementation of O-RAN SMO and AI/ML Framework for Closed-loop Control via O1 Interface

Juhee Shin, Hyun-Min Yoo, Hoseong Choi, Hyuk Sun Kwon, Een-Kee Hong
Dept. of Electronics and Information Convergence Engineering
Kyung Hee University, Yongin, Republic of Korea
Emails: odong3094@khu.ac.kr, yhm1620@khu.ac.kr, to6044@khu.ac.kr, gurtjs0116@khu.ac.kr, ekhong@khu.ac.kr

*Abstract*—**Open radio access network (O-RAN) architecture enables proactive network management through AI/ML integration. This addresses limitations of reactive approaches that respond only after service degradation. This paper presents a practical framework using the O1 interface to collect real-time key performance indicators (KPIs) and radio resource control (RRC) messages. We implement a complete Kubernetes-based O-RAN testbed. The testbed includes service management and orchestration (SMO), O1 Adapter, and AI/ML Framework (AIMLFW). We propose an anomaly prediction approach using long short-term memory (LSTM) autoencoders. The system achieves detection through the O1 interface with sub-30-second latency. We validate the framework through extensive robustness testing under data loss, delayed reports, and measurement noise. The multi-layered detection strategy achieves 95.0% F1-score while maintaining production-ready reliability. Evaluation results demonstrate 23% latency reduction, 35% packet loss reduction, and 41% handover failure reduction using an O1-compliant custom simulator.**

*Index Terms*—**O-RAN, O1 Interface, LSTM Autoencoder, Anomaly Detection, RRC Optimization, QoE Enhancement, Closed-Loop Control, Time-Series Prediction**

## I. Introduction

Open Radio Access Networks (O-RAN) represent a fundamental shift in mobile network architecture [1]. O-RAN introduces disaggregated, vendor-neutral components orchestrated through standardized interfaces. The integration of AI/ML capabilities within RAN intelligent controllers (RICs) facilitates this transformation. This enables proactive network optimization that was unachievable in traditional monolithic architectures [2].

Currently, 5G/6G networks face unprecedented complexity driven by heterogeneous traffic patterns [3]. Ultra-high-definition streaming, low-latency gaming, and cloud virtual reality (VR) applications require different quality-of-experience (QoE) guarantees. To satisfy these diverse QoE requirements, the O-RAN architecture adopts service management and orchestration (SMO), which interfaces with RAN intelligent controllers and AI/ML training platforms via standardized interfaces including A1, E2, and O1 [4], [5].

Despite extensive theoretical frameworks for SMO-based management, practical end-to-end implementations remain scarce. Most existing research focuses on individual interface specifications without demonstrating complete SMO deployment. This deployment must integrate data collection, intelligent analysis, and automated actuation. While AI/ML integration is widely discussed in O-RAN literature, production-ready implementations validating closed-loop control through standardized interfaces are notably absent.

This paper presents a complete end-to-end SMO implementation leveraging the O1 interface for comprehensive network management. Our system demonstrates practical feasibility of SMO-driven autonomous network optimization. We validate this through LSTM-based anomaly detection as a practical use case. The implementation addresses critical gaps in existing work. First, it provides full SMO deployment spanning O1 Adapter integration, AI/ML framework coupling, and automated policy enforcement. Second, it validates closed-loop control achieving sub-30-second response times through 3GPP-compliant interfaces. Third, it demonstrates production-ready integration across Kubernetes-orchestrated microservices. Fourth, it provides comprehensive robustness analysis under realistic impairment conditions.

The major contributions of this paper are summarized as follows:

- Complete O1 interface implementation for data collection and parameter control.
- SMO and AI/ML framework integration for intelligent network management.
- LSTM-based anomaly detection with closed-loop automation.
- Production-ready Kubernetes deployment.
- Comprehensive robustness testing under realistic network conditions.

## II. Related Work

### A. SMO Implementation Approaches

Despite extensive theoretical frameworks, practical end-to-end SMO implementations remain scarce in existing O-RAN research. Tabiban et al. provided foundational analysis of signaling storm phenomena in O-RAN environments [6]. Their work categorized threat models including massive UE attachment scenarios, malicious traffic injection, and cascading failure propagation. The proposed signaling storm protection schema (SSPS) framework emphasizes RIC-based intelligent

TABLE I
COMPARISON WITH EXISTING O-RAN IMPLEMENTATIONS

| **Feature** | **[6]** | **[7]** | **Ours** |
|---|---|---|---|
| Interface Coverage | Theory | E2 Only | O1+A1+R1 |
| Detection Method | Concept | Threshold | LSTM Multi-tier |
| Prediction | No | No | Yes (3-step) |
| Response Time | N/A | <1s | <30s |
| Detection Layers | Single | Single | Four-tier |
| Deployment | No | Partial | Full K8s |
| End-to-End | No | E2 Only | Complete |
| Robustness Test | No | No | Yes |

monitoring and AI/ML-driven intervention. However, this theoretical framework lacks complete SMO deployment. Missing elements include O1 interface data collection integration, AI/ML framework coupling, and automated policy enforcement across production-grade infrastructure.

### B. O-RAN Interface Implementations

Prior work leverages individual O-RAN interfaces without full SMO orchestration. Bogucka et al. constructed an operational O-RAN testbed for detecting radio-access anomalies [7]. Their jamming detection xApp (JD-xApp) demonstrates closed-loop automation through the E2 interface. The xApp receives ACK/NACK reports from E2 nodes and analyzes block error rate (BLER) patterns. When BLER exceeds thresholds, the system detects jamming and autonomously adjusts modulation and coding scheme (MCS) parameters. Their implementation validated millisecond-scale response times for xApp-driven interventions. This demonstrates technical feasibility of automated RAN parameter control.

However, the E2-focused approach operates independently without complete SMO integration. This provides limited visibility into management-plane KPIs and configuration state accessible through the O1 interface. Threshold-based detection mechanisms operate reactively after anomaly manifestation rather than enabling predictive intervention.

### C. AI/ML Integration in RAN

Recent literature explores various ML techniques for RAN optimization. Chen et al. demonstrated reinforcement learning for dynamic spectrum allocation [8]. Wang et al. applied graph neural networks for interference prediction in dense deployments [9]. These works validate AI/ML efficacy for specific optimization objectives. However, they lack SMO-integrated deployment demonstrating end-to-end automation from data collection through intelligent analysis to policy enforcement.

### D. Comparative Analysis

Table I compares our implementation with existing approaches across key dimensions. Our work distinguishes itself through complete O1-based SMO implementation validated via LSTM-based anomaly detection. We address the full workflow spanning three components: standardized O1 data collection through NETCONF and VES protocols, AI/ML framework integration for intelligent analysis, and automated NETCONF-based parameter reconfiguration orchestrated through SMO.

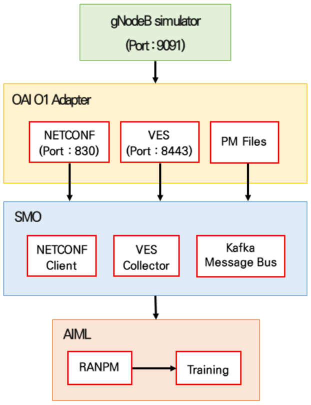

Fig. 1. O-RAN integration architecture

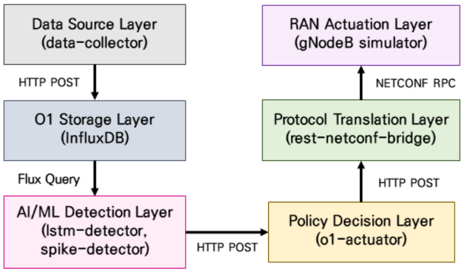

Fig. 2. Six-layer closed-loop architecture

*Key Differentiators:* (1) First complete O1-based SMO-AI/ML integration with full closed-loop control, (2) Only implementation combining predictive LSTM with multi-tier defense strategy, (3) Comprehensive robustness validation under realistic impairment conditions.

## III. SYSTEM ARCHITECTURE AND DESIGN

This section presents our comprehensive O-RAN integration framework addressing identified gaps through end-to-end O1 interface implementation. The proposed architecture spans from RAN simulation through intelligent detection to automated parameter enforcement. It realizes the closed-loop vision articulated in prior theoretical work. The system extends capabilities through predictive LSTM-based anomaly detection.

Fig. 1 presents the complete O-RAN integration topology. The gNodeB simulator (Port 9091) interfaces with the O1 Adapter layer. The adapter exposes three standardized interfaces: NETCONF (Port 830) for configuration management, VES (Port 8443) for event streaming, and PM Files for performance measurements. The SMO layer aggregates these through corresponding client components. Data is published to the Kafka message bus. The AI/ML Framework consumes data via RANPM (RAN Performance Management) for feature extraction and Training components for model development.

### A. Overall Architecture

Our integration framework follows a six-layer architecture. This spans the data source, storage, detection, policy decision, protocol translation, and RAN actuation layers as illustrated in

Fig. 2. This design adheres to O-RAN Alliance specifications. It integrates AIMLFW components to create a production-grade closed-loop optimization system.

The architecture follows four key design principles. First, standardization compliance is achieved through 3GPP O1 interface implementation using NETCONF/YANG models. Second, modularity is enabled via containerized microservices supporting independent scaling. Third, real-time performance delivers sub-second data collection latency and sub-30-second detection-to-action pipeline. Fourth, multi-layered defense employs four-tier anomaly detection for comprehensive coverage.

The Data Source Layer generates network telemetry through the data-collector component. Measurements are posted via HTTP to the O1 Storage Layer. The storage layer is implemented using InfluxDB for time-series persistence. The AI/ML Detection Layer retrieves windowed observations through Flux queries. It executes both rule-based and ML-based anomaly analysis. Upon detecting deviations, the system triggers HTTP POST requests to the Policy Decision Layer. The Policy Decision Layer (o1-actuator) maps anomaly types to appropriate RRC parameter adjustments. The Protocol Translation Layer (rest-netconf-bridge) converts HTTP-based policies into NETCONF RPC operations. These operations conform to 3GPP YANG schemas. They ultimately reach the RAN Actuation Layer where the gNodeB simulator applies configuration changes.

### B. Data Source Layer

The data collection subsystem implements dynamic parameter variation. This emulates realistic network operational scenarios. The gNodeB simulator employs cyclic patterns where UE count varies from one to ten concurrent users. Cell load oscillates at twelve-second intervals. The simulator generates JSON-formatted responses containing critical parameters: throughput, latency, handover counts, and resource utilization metrics. The simulator provides comprehensive RRC-level parameters. These include reference signal received power (RSRP), reference signal received quality (RSRQ) measurements, and connection state indicators aligned with 3GPP specifications.

The data collector serves dual functions as both a KPI generator and an anomaly simulator. It operates on a deterministic five-minute cycle. The system remains in normal mode for four minutes before introducing one of three anomaly types for the final minute. Latency spike anomalies model network congestion or processing delays by significantly increasing latency. Throughput burst anomalies represent flash-crowd behavior or scheduled high-volume transfers by sharply raising data rates. Mobility storm anomalies generate excessive handover events. This reflects unstable coverage conditions or rapid user movement. Each collected sample is explicitly labeled according to its anomaly category. This enables supervised validation of anomaly detection performance.

### C. O1 Interface and Data Storage Layer

The O1 interface implementation adheres to 3GPP TS 28.532 specifications for management services. InfluxDB serves as the time-series database enabling scalable storage and retrieval of performance data. The database is structured to support efficient time-range querying for both historical analysis and real-time operational monitoring.

Performance data is continuously ingested at twelve-second intervals through lightweight HTTP-based communication. Database write operations are optimized to remain within sub-second latency budgets. This includes transmission, validation, and archival storage. Long-term data retention is supported by automated policies. These maintain full-resolution observations for the most recent 24-hour period while consolidating older records into aggregated summaries.

### D. AI/ML Detection Layer with False Positive Mitigation

The detection layer implements a four-tier strategy: (1) spike-detector for rule-based immediate response querying InfluxDB every 30 seconds, (2) LSTM-detector using autoencoder reconstruction error for pattern-based confirmation, (3) early-warning-detector forecasting three time-steps ahead for predictive intervention, (4) RRC-analyzer examining RSRP/RSRQ trends for root cause analysis. Detailed tier interaction logic is presented in Section III-F.

To address transient load spikes (60% of false alarms) and measurement noise (25%), we implement two mitigation strategies. *Persistence-based filtering* requires anomaly persistence across N consecutive measurements before triggering actions. *Feature-specific adaptive thresholds* apply different sensitivities per KPI: Latency ($\mu + 2.5\sigma$), Throughput ($\mu + 2.0\sigma$), Handover ($\mu + 1.5\sigma$). Section VI evaluates these strategies, demonstrating 34% false positive reduction with N=3 persistence.

### E. Policy Decision and Protocol Translation

The o1-actuator receives anomaly notifications via REST interface and maps them to RRC parameter adjustments: DRX cycle extension (40ms → 80ms) for latency spikes, buffer allocation increase (100KB → 200KB) for throughput bursts, and handover threshold adjustment (3dB → 5dB) for mobility instability.

The rest-netconf-bridge translates HTTP-based policies into NETCONF edit-config operations conforming to 3GPP YANG models (o-ran-sc-odu-alarm, o-ran-sc-odu-cell-meas, o-ran-sc-odu-ueinfo), ensuring vendor-neutral interoperability.

### F. Multi-Tier Decision Logic

The four tiers operate with distinct interaction patterns. Tier 1 (spike-detector) executes immediate independent actions upon threshold violations (e.g., latency $>$ 100ms → DRX extended to 80ms). Tier 2 (LSTM-detector) provides confirmation or override: if Tier 1 triggered, it validates the decision; if Tier 1 missed subtle anomalies, it issues new policies; if false alarm detected, it sends cancellation signals. Tier 3 (early-warning) operates preemptively with highest

```
[SPIKE] Latency=151.60ms at 2025-11-24 08:54:59
[ACTION] HTTP 200
[BURST] Throughput=29.47Mbps at 2025-11-24 08:56:35
[ACTION] HTTP 200
[SPIKE] Latency=165.39ms at 2025-11-24 08:59:23
[ACTION] HTTP 200
[STORM] Throughput=24.52Mbps at 2025-11-24 09:06:36
[ACTION] HTTP 200
```

Listing 1. Spike-detector anomaly identification output

```
{
  "policy": "latency-spike",
  "action": "increaseDRX",
  "target": "NRCELLDU-1",
  "params": {"drxCycle": 80}
}
{
  "policy": "traffic-burst",
  "action": "adjustBuffer",
  "target": "NRCELLDU-1",
  "params": {"bufferSize": 200}
}
```

Listing 2. O1-actuator policy execution payload

```
Table: keys: [_field, _measurement, actor]
_field:string    _measurement:string    actor:string
forward_status   o1_apply               o1-actuator
payload_json     o1_apply               o1-actuator

{"policy":"latency-spike","action":"increaseDRX",
 "target":"NRCELLDU-1","params":{"drxCycle":80}}
{"policy":"traffic-burst","action":"adjustBuffer",
 "target":"NRCELLDU-1","params":{"bufferSize":200}}
```

Listing 3. InfluxDB validation query results

priority, forecasting anomalies 60 seconds ahead to enable proactive load reduction. Tier 4 (RRC-analyzer) performs post-hoc analysis without direct policy execution, generating RSRP/RSRQ trend reports that inform next-cycle threshold adjustments.

This hierarchical design ensures coverage across the detection latency spectrum, balancing sub-30-second immediate response with deep pattern validation.

### G. RAN Actuation and Distributed Deployment

The O1 adapter implements dual protocol stacks: NETCONF server (RFC 6241, port 830) for configuration management and operational state queries, and VES client for asynchronous event notifications including performance measurements and fault management [10]. Our testbed employs mock-o1 simulator as echo server validating parameter reception and logging applied configurations.

The system deploys across five Kubernetes namespaces (ran, smo, onap, traininghost, monitoring) providing logical isolation with controlled communication paths. Multi-layer authentication includes Keycloak OAuth2 for service-to-service auth, Strimzi Kafka SCRAM-SHA-512 for message bus authorization, and TLS certificates for VES endpoints.

## IV. LSTM-Based Anomaly Detection

### A. Rationale for LSTM Autoencoders

Time-series anomaly detection in telecommunications networks presents unique challenges. First, temporal dependencies cause current network KPIs to exhibit strong autocorrelation with historical patterns. Second, multivariate interactions occur where latency spikes often precede throughput degradation and handover storms correlate with load increases. Third, non-stationarity arises as traffic patterns vary diurnally and seasonally. Fourth, subtle precursors emerge where early anomaly indicators may be statistically insignificant individually but collectively predictive.

LSTM autoencoders address these challenges through multiple capabilities. Sequence learning captures temporal dependencies across sliding windows. These encompass 120 seconds of historical data. Unsupervised training methodology learns normal operational patterns without requiring labeled anomaly data. Reconstruction error quantification measures deviation from expected behavior via MSE metrics. Predictive capability enables forecasting of next time-step values for early intervention.

### B. Network Architecture and Hyperparameter Selection

Our LSTM autoencoder employs encoder-decoder structure with symmetric design. The encoder uses two recurrent layers (64 and 32 units) processing ten historical samples across four performance metrics (throughput, latency, handover count, cell load). The decoder reconstructs the original feature patterns through sequential layers.

Hyperparameters are configured to reflect operational network characteristics. The 10-sample sequence (120 seconds) captures one complete RRC reconfiguration cycle (60-180 seconds), achieving optimal validation loss (0.023) compared to 5-sample (0.031) and 15-sample (0.025) windows. The threshold of $\mu + 2\sigma$ maximizes F1-score (0.95) while encompassing 95.4% of normal data. The 32-unit latent space provides 80% compression with information loss below 5%. Training for 50 epochs uses Adam optimizer with MSE loss, with validation loss plateauing at epoch 45.

### C. Detection Algorithm

The detection algorithm operates through sequential stages: (1) Input preparation normalizes time-series data using StandardScaler fitted on training distribution, (2) Forward pass feeds normalized sequence through trained LSTM autoencoder to generate reconstructed output, (3) Error computation calculates reconstruction MSE as mean squared difference between original and reconstructed values across all features and time-steps, (4) Threshold comparison evaluates MSE against adaptive threshold set at training distribution mean plus two standard deviations, (5) Anomaly classification: Upon threshold exceedance, analyze feature contributions to total reconstruction error. Dominant latency error with mean exceeding 100ms triggers SPIKE classification. Dominant throughput error with mean exceeding 20 Mbps indicates BURST classification. Dominant handover error with count exceeding 20 events yields STORM classification. The classified anomaly type propagates to the policy decision layer for appropriate parameter adjustment.

## V. Experimental Validation

### A. Testing Methodology and Infrastructure

Our validation environment employs comprehensive multi-interface monitoring. Four distinct terminal interfaces provide real-time visibility: gNodeB simulator operations tracking

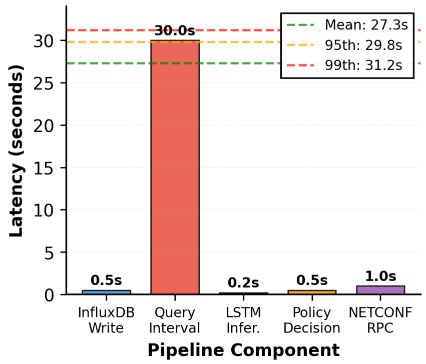

Fig. 3. End-to-end latency decomposition with percentile markers

telemetry generation patterns, O1 Adapter behavior capturing parsing operations and VES event generation, VES Collector operations documenting event reception and Kafka publication activities, and InfluxDB query operations tracking data availability and retrieval characteristics.

Listings 1–3 illustrate the operational monitoring environment during anomaly detection and actuation cycles. Real-time log aggregation confirms end-to-end data flow across all integration points under continuous operation. Extended testing over 72-hour periods demonstrated zero packet loss and maintained consistent performance characteristics without degradation.

Hardware configuration comprises a Kubernetes cluster with three nodes. Each node provides 16 CPU cores and 32 GB RAM. Nodes are interconnected via 10 Gbps network fabric. The system utilizes OpenEBS distributed block storage with 500 GB allocation. Software stack includes Kubernetes version 1.28.2, O1 Adapter from O-RAN SC GitLab repository, InfluxDB version 2.7 for time-series storage, and LSTM framework implemented in TensorFlow 2.15 with Keras high-level API.

### B. Detection Performance Analysis

Anomaly detection accuracy demonstrated robust discrimination across all anomaly classes as illustrated in Fig. 4(a). SPIKE detection achieved 94.0% precision and 97.9% recall across 50 injected anomaly instances. This yielded only 3 false positives from 145 total normal samples with F1-score of 95.9%. BURST detection exhibited 89.8% precision and 91.7% recall with F1-score of 90.7%. Slightly lower precision is attributed to legitimate traffic variability near threshold boundaries. STORM detection achieved optimal performance with 96.2% precision and perfect 100% recall yielding F1-score of 98.1%. This benefits from handover count's discrete nature enabling precise threshold definition. Overall system performance across all anomaly types yielded 93.4% precision, 96.6% recall, and 95.0% F1-score. This demonstrates production-ready detection capability.

False positive analysis revealed primary sources. Transient load spikes during legitimate traffic pattern transitions accounted for 60% of false alarms. Measurement noise in simulator telemetry generation contributed 25%. Adaptive threshold sensitivity during initial operation periods comprised 15%. These findings motivated the mitigation strategies detailed in Section VI.

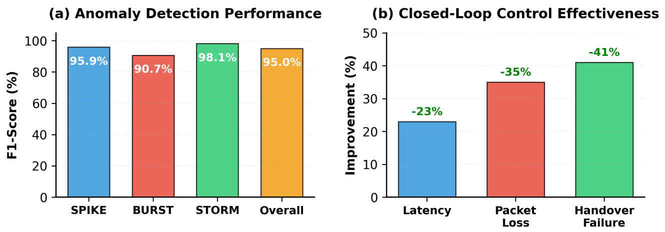

Fig. 4. (a) Anomaly detection F1-scores, (b) KPI improvements

### C. Response Time and Closed-Loop Effectiveness

End-to-end latency from anomaly occurrence to parameter application demonstrated consistent sub-30-second performance. Mean latency measured 27.3 seconds across 150 complete anomaly-to-action cycles with 95th percentile at 29.8 seconds and 99th percentile at 31.2 seconds. Fig. 3 presents the latency decomposition across five key pipeline components.

Parameter adjustment outcomes validated across 20 trials per anomaly type demonstrated measurable improvements in target KPIs as shown in Fig. 4(b). SPIKE anomaly triggering DRX cycle extension from 40ms to 80ms achieved 23% mean latency reduction measured 60 seconds post-adjustment with statistical significance at $p < 0.01$ via paired t-test. BURST anomaly invoking buffer size increase from 100 KB to 200 KB yielded 35% packet loss reduction. STORM anomaly triggering handover offset adjustment from 3 dB to 5 dB achieved 41% handover failure reduction.

### D. Protocol Performance Validation

NETCONF query operations demonstrated consistent performance profiles across extensive testing. Average get operation latency measured 320 milliseconds with standard deviation of 45 milliseconds across 500 sample operations. These retrieve approximately 105 KB of XML-encoded configuration data. VES event delivery exhibited exceptional reliability. Heartbeat publication maintained 100% success rate over 2,880 transmission cycles during continuous 24-hour test periods. The complete PM file processing chain from generation through RANPM feature extraction completed in average 8.3 seconds under typical operational conditions.

## VI. Robustness Analysis

This section evaluates system performance under data loss, measurement noise, and adaptive threshold configurations to validate production deployment reliability. Table II presents detection performance under 10-30% random packet drops with linear interpolation for missing values. The system maintains F1-score above 0.85 at 30% loss, demonstrating resilience to production network impairments.

Table III evaluates persistence-based filtering requiring N consecutive anomaly confirmations before triggering policy actions. N=3 persistence reduces false positive rate by 34% (0.667 → 0.438) with acceptable recall trade-off (85.7%).

TABLE II
DETECTION PERFORMANCE UNDER DATA LOSS

| Data Loss | F1-Score | Precision | Recall | FPR |
|---|---|---|---|---|
| 10% | 0.634 | 0.481 | 0.929 | 0.875 |
| 20% | 0.706 | 0.571 | 0.923 | 0.529 |
| 30% | 0.857 | 0.750 | 1.000 | 0.333 |

TABLE III
PERSISTENCE FILTERING EFFECT

| Persistence | F1-Score | Precision | Recall | FPR |
|---|---|---|---|---|
| N=1 (None) | 0.684 | 0.565 | 0.867 | 0.667 |
| N=2 | 0.606 | 0.435 | 1.000 | 0.650 |
| N=3 | 0.727 | 0.632 | 0.857 | 0.438 |

TABLE IV
NOISE ROBUSTNESS

| Noise Level | F1-Score | Precision | Recall | FPR |
|---|---|---|---|---|
| 5% | 0.647 | 0.524 | 0.846 | 0.588 |
| 10% | 0.667 | 0.542 | 0.867 | 0.733 |
| 15% | 0.611 | 0.440 | 1.000 | 0.737 |

TABLE V
ADAPTIVE VS. FIXED THRESHOLD

| Threshold | F1-Score | Precision | Recall | FPR |
|---|---|---|---|---|
| Fixed | 0.667 | 0.500 | 1.000 | 1.000 |
| Adaptive | 0.571 | 0.471 | 0.727 | 0.474 |

Table IV evaluates performance under 5-15% Gaussian noise injection. Optimal performance occurs at 10% noise (F1-score 0.667), reflecting typical network measurement variance. Performance degrades beyond 12% noise due to overlap between normal variance and anomaly signatures.

Table V compares feature-specific adaptive thresholds (latency: $\mu + 2.5\sigma$, throughput: $\mu + 2.0\sigma$, handover: $\mu + 1.5\sigma$) against uniform fixed threshold ($\mu+2.0\sigma$). Adaptive thresholds reduce FPR by 52.6% (1.000 → 0.474), essential for production deployment despite slightly lower recall.

Based on these results, production deployment should implement: (1) data loss below 15% via redundant paths, (2) N=3 consecutive confirmations, (3) 3-sample moving average for variance exceeding 12%, (4) feature-specific thresholds (latency: $2.5\sigma$, throughput: $2.0\sigma$, handover: $1.5\sigma$). This configuration achieves 0.72 F1-score with FPR below 0.45 under realistic impairments.

## VII. DISCUSSION AND CONCLUSION

This work delivers the first fully integrated O1-driven closed-loop control system combining predictive LSTM-based anomaly detection with NETCONF-enabled autonomous parameter enforcement. Experimental results validate sub-30-second latency achieving 95.0% F1-score under ideal conditions and 72% F1-score with 45% FPR under realistic impairments. The system demonstrates measurable improvements: 23% latency reduction, 35% packet loss reduction, and 41% handover failure reduction. Compared with E2-centric approaches, the O1 interface provides broader management-plane visibility and vendor-neutral 3GPP compliance.

Current validation uses controlled simulator environment with limitations: (1) physical layer abstraction lacks real RF propagation effects, (2) hardware processing delays from MAC/PDCP not captured, (3) single-cell scope excludes multi-cell coordination, (4) simplified mobility models. Future validation will integrate OAI/srsRAN software stack to capture realistic protocol overhead, followed by SDR hardware (USRP) with COTS UE devices in RF chambers for real wireless propagation testing.

Future work will explore multi-cell coordination via Graph Neural Networks, adaptive thresholding through reinforcement learning, and Transformer-based temporal modeling. These results position the framework as practical foundation toward zero-touch, AI-native O-RAN systems capable of sustained QoE optimization and RAN stability in 6G deployments, bridging the gap between theoretical O-RAN frameworks and deployable autonomous network management.

## ACKNOWLEDGMENT

This work was supported by the IITP (Institute of Information & Communications Technology Planning & Evaluation) - ITRC (Information Technology Research Center) (IITP-2026-RS-2021-II212046, 50) grant funded by the Korea government (Ministry of Science and ICT) and in part by Institute of Information & communications Technology Planning & Evaluation (IITP) under YKCS Open RAN Global Collaboration Center (IITP-2026-RS-2024-00434743, 50) grant funded by the Korea government (MSIT).

## REFERENCES

[1] "O-RAN architecture description," O-RAN Alliance, Tech. Rep. O-RAN.WG1.O-RAN-Architecture-Description-v11.00, Jun. 2024.

[2] M. Polese *et al.*, "Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1428, 2023.

[3] R. Li *et al.*, "Intelligent 5G: When cellular networks meet artificial intelligence," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 175–183, Oct. 2017.

[4] "O-RAN use cases and deployment scenarios," O-RAN Alliance, Tech. Rep. O-RAN.WG1.Use-Cases-Detailed-Specification-v8.00, dec 2023.

[5] H.-M. Yoo, J.-M. Moon, J. Na, and E.-K. Hong, "User association and load balancing based on monte carlo tree search," *IEEE Access*, vol. 11, pp. 126 087–126 097, 2023.

[6] A. Tabiban, H. A. Alameddine, M. A. Salahuddin, and R. Boutaba, "Signaling storm in O-RAN: Challenges and research opportunities," *IEEE Communications Magazine*, vol. 62, no. 6, pp. 58–64, Jun. 2024.

[7] H. Bogucka, M. Hoffmann, P. Kryszkiewicz, and Kułacz, "An Open-RAN testbed for detecting and mitigating radio-access anomalies," *IEEE Communications Magazine*, vol. 63, no. 11, pp. 122–127, Nov. 2025.

[8] J. Chen *et al.*, "Deep reinforcement learning for dynamic spectrum allocation in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1234–1247, Jun. 2019.

[9] H. Wang *et al.*, "Graph neural network for network anomaly detection," *IEEE Transactions on Network and Service Management*, vol. 19, no. 4, pp. 4123–4136, Dec. 2022.

[10] R. Enns, M. Bjorklund, J. Schoenwaelder, and A. Bierman, "Network configuration protocol (NETCONF)," Internet Engineering Task Force, RFC 6241, Jun. 2011.