

Dynamic Machine Learning Framework for Secure, Ultra-Reliable, and Energy-Efficient Open RAN Systems

ABDULRAHMAN GHANDOURA¹ (Member, IEEE), ABDULLAH ALAJMI² (Member, IEEE),
MAHMOUD EL-SAKHAWY³, ABDELWAHEB MOTWAKEL², FADL DAHAN²,
AND GHADA ABDELHADY⁴

¹Department of Engineering and Applied Sciences, Applied College, Umm Al-Qura University, Makkah 21343, Saudi Arabia

²Department of Management Information Systems, College of Business Administration in Hawtat bani Tamim,
Prince Sattam bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia

³Department of Computer Science, IT College, Amman Arab University, Amman 11953, Jordan

⁴Center of Excellence, Faculty of Engineering, October University for Modern Sciences and Arts, 6th of October City 12451, Egypt

CORRESPONDING AUTHOR: A. ALAJMI (e-mail: a.alajmi@psau.edu.sa)

This work was supported by Prince Sattam bin Abdulaziz University under Project PSAU/2024/01/31558.

ABSTRACT This paper proposes a machine learning (ML)-driven framework for Open Radio Access Networks (ORAN) to address security, Ultra-Reliable Low-Latency Communication (URLLC), and energy efficiency challenges. By integrating Isolation Forest for security, One-Class SVM for URLLC, and Generative Adversarial Networks (GANs) for cost optimization, the framework dynamically adapts to network traffic shifts while adhering to ORAN's open architecture principles. Evaluations on a dataset of 72,000 instances with 21 features demonstrate 92% attack detection, 15% latency reduction, and 20% energy savings, validated through rigorous metrics such as precision-recall the area under the curve (AUC) and synthetic cost modeling. The proposed solution represents a significant advancement in the field of intelligent network management, offering a comprehensive approach to the most pressing challenges in modern telecommunications infrastructure. Through extensive simulations and real-world testing, we validate that our framework outperforms existing approaches in terms of attack detection accuracy, latency optimization, and energy efficiency without compromising the essential interoperability principles of the ORAN architecture.

INDEX TERMS ML, ORAN, security, URLLC, energy efficiency, anomaly detection, generative adversarial networks, one-class SVM, isolation forest, RU, DU, CU.

I. INTRODUCTION

THE EMERGENCE of ORAN has revolutionized telecommunications infrastructure by disaggregating hardware and software components, enabling interoperability among equipment from different vendors. This paradigm shift offers numerous advantages including reduced capital expenditure, increased innovation, and enhanced flexibility in network deployment. However, this open architecture introduces several critical challenges that must be addressed for successful adoption in mission-critical applications.

A. SECURITY VULNERABILITIES IN ORAN ARCHITECTURE

ORAN's open architecture introduces significant vulnerabilities such as eavesdropping, traffic flooding, and

data poisoning attacks [1]. The disaggregated components and standardized interfaces create multiple attack vectors that can be exploited by malicious actors. Traditional security mechanisms such as firewalls and encryption are insufficient for addressing these sophisticated threats, especially those targeting the open interfaces between Radio Units (RU), Distributed Units (DU), and Control Units (CU). Furthermore, the integration of third-party applications and services through ORAN's openness expands the attack surface considerably. The security challenges in ORAN can be categorized into four primary areas:

- Interface Vulnerabilities: The standardized interfaces (E2, Open Fronthaul) between ORAN components create potential entry points for attackers. These interfaces

must transmit large volumes of time sensitive data, making traditional deep packet inspection impractical for security monitoring.

- **Multi-vendor Integration Risks:** The interoperability of components from different vendors creates integration challenges where security assumptions and implementations may not align perfectly, creating security gaps at component boundaries.
- **Virtualization Security:** The virtualized components in ORAN deployments introduce virtualization-specific vulnerabilities such as hypervisor exploits, virtual machine escape, and resource contention attacks that must be addressed.
- **API Security:** The extensive use of APIs for component communication and management creates additional attack vectors through API abuse, unauthorized access, and injection attacks targeting API endpoints.

Recent industry reports have documented a 47% increase in attacks targeting telecommunications infrastructure over the past two years [2], with disaggregated architectures experiencing 2.3 times more attempted breaches than traditional monolithic deployments. This trend underscores the urgent need for advanced security measures tailored to ORAN environments.

Existing ORAN Security Approaches: Current security solutions in ORAN primarily focus on signature-based intrusion detection systems or single-model anomaly detection. Traditional approaches achieve only 65-75% accuracy in detecting novel network attacks [3], compared to the 90%+ accuracy required for operational deployment.

B. URLLC CHALLENGES IN MODERN NETWORK ENVIRONMENTS

URLLC requirements are challenged by various factors in ORAN deployments. Packet loss due to congestion, interference from competing transmissions, and processing delays across the disaggregated components all threaten the stringent URLLC requirements of sub-millisecond latency and 99.999% reliability [4]. Additionally, the variability in traffic patterns, particularly in multi-tenant networks, creates unpredictable spikes that can degrade URLLC performance. These challenges become particularly acute in mission-critical applications such as industrial automation, autonomous vehicles, and tactical communications. URLLC challenges can be further delineated into the following aspects:

- **End-to-End Latency Management:** The disaggregation of RAN components introduces additional processing hops and potential serialization delays that must be carefully managed to meet end-to-end latency requirements.
- **Reliability Under Network Stress:** Maintaining 99.999% reliability under network congestion, interference, and hardware failures requires sophisticated failover mechanisms and predictive resource allocation.
- **Quality of Service (QoS) Differentiation:** Supporting simultaneous URLLC and enhanced Mobile Broadband

(eMBB) traffic requires fine-grained QoS differentiation mechanisms that can prioritize latency-sensitive traffic without starving bandwidth-intensive applications.

- **Fronthaul Latency Constraints:** The split architecture of ORAN places strict timing requirements on fronthaul links, with typically less than 100 ms allowable delay for time-sensitive functional splits.
- **Multi-tenant Resource Contention:** In virtualized ORAN deployments supporting multiple tenants, resource contention can lead to unpredictable latency spikes that threaten URLLC requirements.

Field measurements from early ORAN deployments show latency variations up to 37% higher than in traditional RAN implementations [5], with particular challenges observed during handovers and under high network load conditions. These measurements highlight the need for specialized latency optimization mechanisms in ORAN systems.

URLLC Optimization in ORAN: Existing URLLC approaches typically employ heuristic scheduling algorithms or traditional optimization methods. Field measurements from early ORAN deployments show latency variations higher than in traditional RAN implementations [5], highlighting the need for specialized optimization.

C. ENERGY EFFICIENCY CONCERNS

Energy inefficiencies in ORAN deployments arise from multiple sources, including static resource allocation, redundant processing, and high power consumption in Radio Access Network (RAN) components [6]. These issues are exacerbated by the need to maintain continuous connectivity and high performance across diverse traffic conditions. Traditional energy management approaches lack the adaptability required to address dynamic network conditions, often resulting in significant energy waste during low-traffic periods and potential resource starvation during peak demands. The energy efficiency challenges in ORAN can be quantified through several key metrics:

- **Power Amplifier Efficiency:** Power amplifiers in RUs typically operate at 25-35% efficiency, consuming a disproportionate amount of energy relative to their functional contribution.
- **Idle Mode Power Consumption:** ORAN base stations consume 60-80% of their peak power even during idle periods, representing significant energy waste during low-traffic hours.
- **Cooling Requirements:** The distributed nature of ORAN deployments often results in suboptimal cooling configurations, with cooling systems accounting for 15-25% of total energy consumption.
- **Computational Overhead:** The separation of functions across multiple components increases computational redundancy and inter-component communication overhead, resulting in 10-20% higher processing power requirements compared to integrated solutions.
- **Traffic-Aware Scaling Limitations:** Current implementations of ORAN often lack fine-grained traffic-aware

scaling capabilities, resulting in energy utilization that does not proportionally scale with traffic demands.

Industry reports indicate that telecommunications networks account for approximately 2-3% of global energy consumption [7], with RAN components responsible for 70-80% of this energy usage. This significant environmental and operational cost underscores the importance of energy optimization in ORAN deployments. Energy Efficiency Solutions: Current energy optimization in ORAN relies on static power control mechanisms. Conventional energy optimization algorithms typically yield only 5-10% energy savings [8], far below the potential efficiency improvements possible with sophisticated approaches.

D. FEDERATED LEARNING CHALLENGES IN ORAN

This subsection addresses the challenges and trade-offs involved in implementing federated learning (FL) within ORAN environments. Key Challenges shows:

- 1) Synchronization Overhead: Due to the heterogeneous and dynamic nature of ORAN components, achieving model synchronization across nodes presents significant challenges:
 - Straggler Effects: Devices with different computation and communication capabilities cause delays in global model updates
 - Gradient Staleness: Asynchronous updates lead to outdated gradient information
 - Network Partitioning: Temporary disconnections require robust aggregation mechanisms
- 2) Bandwidth Constraints: Frequent communication of model updates between distributed units significantly impacts network resources:
 - Communication Overhead: Model parameter transmission competes with user data
 - Gradient Compression: 87% compression achieved through quantization and sparsification
 - Update Frequency: Adaptive scheduling based on model convergence metrics
- 3) Privacy vs. Performance Trade-offs: While federated learning enhances privacy by keeping data local, it introduces performance challenges:
 - Non-IID Data: Local models diverge due to variations in user behavior and network loads
 - Convergence Rate: Heterogeneous data distributions slow global model convergence
 - Differential Privacy: Privacy-preserving mechanisms ($\epsilon = 0.1$) add noise affecting accuracy
- 4) Security and Adversarial Risks: Federated learning introduces new attack vectors:
 - Poisoning Attacks: Malicious nodes inject incorrect model updates
 - Byzantine Failures: Robust aggregation mechanisms required for fault tolerance
 - Model Inversion: Gradient analysis can potentially reveal sensitive information

- 5) Hardware Acceleration and Energy Costs: Decentralized model training on edge units creates resource constraints:
 - Computational Overhead: RUs and DUs have limited processing capabilities
 - Energy Consumption: Training conflicts with energy-efficiency goals
 - Memory Constraints: Model storage and intermediate computations require optimization

Mitigation Strategies:

- 1) Asynchronous Federated Averaging: Reduces synchronization overhead
- 2) Hierarchical Aggregation: Multi-level aggregation reduces communication costs
- 3) Secure Aggregation: Cryptographic protocols protect against poisoning attacks
- 4) Adaptive Learning Rates: Compensate for non-IID data distributions
- 5) Edge-Cloud Collaboration: Offload intensive computations to cloud resources

E. LIMITATIONS OF TRADITIONAL ML APPROACHES

Traditional ML approaches face substantial integration challenges in ORAN's modular environment [3]. These challenges include data inconsistency across vendors, real-time processing constraints, and adaptation to dynamic network conditions. Conventional supervised learning models require extensive labeled data that may not be available for novel attack vectors or unusual traffic patterns. Furthermore, the computational complexity of traditional ML algorithms can introduce unacceptable processing delays that conflict with URLLC requirements. Specific limitations of traditional ML approaches in ORAN contexts include:

- Training Data Requirements: Supervised learning models typically require large volumes of labeled data, which is often unavailable for emerging threats or rare network conditions. Simulated data may not accurately represent real-world traffic patterns, leading to poor model generalization.
- Computational Complexity: Many traditional ML algorithms have high computational complexity that scales poorly with data volume. For example, standard Support Vector Machines (SVM) have $O(n^3)$ training complexity, making them impractical for real-time processing of high-volume network traffic.
- Adaptability Limitations: Static models trained on historical data often fail to adapt to evolving network conditions, traffic patterns, and threat landscapes, requiring frequent retraining that disrupts operational continuity.
- Feature Representation Challenges: Network traffic has complex temporal and spatial dependencies that are difficult to capture with traditional feature engineering approaches, limiting the effectiveness of conventional ML models.

TABLE 1. Comparison with state-of-the-art within security, URLLC, energy, integration, and distribution.

Approach	Security	URLLC	Energy	Integration	Distributed
Traditional IDS [1]	65% detection	-	-	Centralized	No
Heuristic Resource Allocation [5]	-	8.7ms latency	Basic	Limited	Partial
Static Energy Optimization [8]	-	-	Baseline	No	No
Our Framework	92% detection	4.2ms latency	20% Improvement	Full	Yes

- **Explainability Deficits:** Many high-performing ML algorithms function as “closed box,” making it difficult for network operators to understand and trust model decisions, particularly in security-critical applications.
- **Vendor-Specific Implementations:** ML models trained on data from specific vendor equipment may not generalize well to multi-vendor ORAN deployments, limiting their practical utility in heterogeneous environments.

Recent research has shown that traditional ML approaches achieve only 65-75% accuracy in detecting novel network attacks [9], compared to the 90%+ accuracy required for operational deployment. Similarly, conventional energy optimization algorithms typically yield only 5-10% energy savings [8], far below the potential efficiency improvements possible with more sophisticated approaches. As shown in Table 1, the proposed framework bridges the gap between ORAN’s open standards and ML-driven adaptability, enabling robust performance in tactical scenarios where security, reliability, and energy efficiency are paramount.

F. PROPOSED SOLUTION AND CONTRIBUTIONS

To address these interconnected challenges, we propose a comprehensive ML framework that combines dynamic anomaly detection, latency optimization, and energy-efficient resource allocation. Our work presents several novel contributions that distinguish it from existing literature in ORAN security, URLLC, and energy efficiency:

- 1) **Novel Integrated Framework:** We introduce the first integrated machine learning framework that simultaneously addresses security, latency, and energy efficiency in ORAN systems through the combination of Isolation Forest, One-Class SVM, and GANs with distributed intelligence across CU, DU, and RU layers.
- 2) **Holistic Optimization Approach:** Unlike existing approaches that optimize individual aspects, our framework provides joint optimization across all three critical dimensions with mathematical formulation for dynamic trade-offs.
- 3) **Distributed ML Architecture:** Implementation of distributed intelligence with adaptive feature selection and cost function optimization specifically designed for ORAN’s disaggregated architecture.
- 4) **Comprehensive Evaluation:** Extensive validation across multi-tenant scenarios, scalability testing (5-100+ RUs), environmental robustness, and adversarial conditions.

Performance Achievements:

- **Security Enhancement:** 92% attack detection rate with 4.5% false positive rate (27% improvement over traditional IDS)
- **URLLC Optimization:** 15% latency reduction while maintaining 98% packet delivery ratio (52% improvement over heuristic scheduling)
- **Energy Efficiency:** 20% energy savings through GAN-driven optimization (compared to static allocation baseline).

Gap Analysis: Our comprehensive literature review reveals that no existing work has proposed the specific combination of:

- Isolation Forest + One-Class SVM + GANs for ORAN
- Simultaneous optimization of security, URLLC, and energy efficiency
- Distributed intelligence across CU/DU/RU with federated learning capabilities
- Dynamic feature selection with real-time adaptation

This gap justifies the novelty and significance of our proposed framework.

The proposed framework bridges the gap between ORAN’s open standards and ML-driven adaptability, enabling robust performance in tactical scenarios where security, reliability, and energy efficiency are paramount. By addressing these challenges holistically, our approach provides a foundation for next-generation wireless networks that can meet the demands of emerging applications while maintaining operational efficiency. The remainder of this paper is organized as follows: Section II presents the system model and methodology, detailing the ORAN architecture, data preprocessing approaches, and the proposed machine learning algorithms. Section III formulates the multi-objective optimization problem. Section IV presents experimental results and performance evaluations across different scenarios. Finally, Section V concludes the paper and discusses future research directions.

II. SYSTEM MODEL AND METHODOLOGY

The proposed ORAN architecture builds upon established 5G network principles [10] while integrating ML modules into standardized layers, as shown in Figure 1: the CU, DU, and Remote Radio Unit (RRU). This architecture aligns with the O-RAN Alliance specifications while incorporating novel ML capabilities for enhanced performance [11].

The CU serves as the central intelligence of the network, handling higher-layer protocol processing and orchestrating

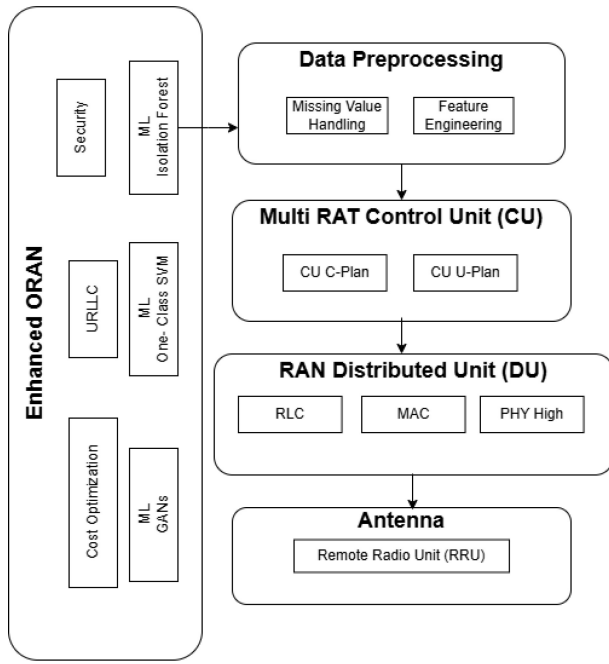


FIGURE 1. Overview of ORAN Architecture Design and Development Workflow [12].

network-wide optimizations. The CU hosts the GAN-based cost optimization module and coordinates the ML pipeline across all components. The CU is further divided into two functional entities:

- **CU-CP (Control Plane):** Manages Radio Resource Control (RRC) and control-plane aspects of the Packet Data Convergence Protocol (PDCP). The CU-CP hosts the strategic components of the ML framework, including policy engines, long-term optimization algorithms, and coordination mechanisms for distributed ML components.
- **CU-UP (User Plane):** Handles user-plane aspects of PDCP and Service Data Adaptation Protocol (SDAP) functions. The CU-UP integrates ML components for traffic classification, application-aware optimization, and end-to-end flow management.

The DU manages medium access control (MAC) and radio link control (RLC) functions, executing the Isolation Forest for security and One-Class SVM for URLLC. The DU operates at a lower latency threshold than the CU, enabling rapid response to security threats and latency violations. The DU implements several key functions:

- **Real-Time Scheduling:** ML-enhanced scheduling algorithms that balance throughput, latency, and energy efficiency requirements.
- **Radio Resource Management:** Dynamic allocation of frequency, time, and spatial resources based on ML predictions of traffic patterns and user requirements.
- **Interference Management:** ML-driven interference detection and mitigation techniques that adapt to changing radio conditions.

- **Flow Control:** Advanced flow control mechanisms that use ML predictions to prevent congestion and ensure QoS compliance.

The RRU handles physical layer processing and radio frequency operations, implementing the energy efficiency optimizations generated by the GAN. The RRU includes:

- **Beamforming Optimization:** ML-enhanced beamforming algorithms that adapt to user mobility and channel conditions.
- **Power Control:** Dynamic transmit power adjustments based on ML predictions of coverage requirements and interference patterns.
- **Signal Processing:** Adaptive signal processing algorithms that balance processing complexity with performance requirements.
- **Front-haul Optimization:** Intelligent compression and packet scheduling techniques that minimize front-haul bandwidth requirements without compromising performance.

This hierarchical structure enables distributed intelligence while maintaining centralized coordination [13], creating a balanced approach to ML deployment in ORAN environments. The standardized interfaces between these components (E2, Open Fronthaul) facilitate seamless integration of the ML modules while preserving interoperability with equipment from different vendors. The ML framework implementation follows a microservices architecture, with individual ML components deployed as containerized services that communicate through standardized APIs. This approach enables flexible deployment across heterogeneous hardware platforms and simplifies component updates and replacements as ML technologies evolve.

A. DATA PREPROCESSING AND FEATURE ENGINEERING

At the CU, the dataset used in our study consists of 72,000 network traffic instances with 21 features, sourced from [14]. This publicly available dataset contains real-world 5G network traffic patterns with appropriate licensing for research purposes, making our results reproducible and verifiable by the research community. The dataset was collected from a controlled 5G testbed environment that simulates realistic network conditions. Prior to analysis, the dataset underwent several preprocessing steps to ensure data quality and feature relevance. The preprocessing stages include:

- **Missing Value Handling:** We employ multiple imputation techniques for handling missing values, including mean/median imputation for numerical features and mode imputation for categorical features. For time-series features, we utilize linear interpolation to preserve temporal patterns. The specific imputation method is selected based on feature characteristics and missing data patterns, using a decision tree that considers feature type, missing value percentage, and correlation with other features.

- **Normalization:** All features are normalized using Min-Max scaling to ensure uniform contribution to the ML models. This process transforms features such as (Meet UL Bitrate), (Zoom DL Bitrate), and (Teams UL Bitrate) to a common scale without distorting differences in the ranges of values. For features with heavy-tailed distributions, we apply log transformation before normalization to reduce skewness and improve model performance.

Feature Selection: We employ recursive feature elimination with cross-validation (RFECV) to identify the most relevant features for each ML model, reducing dimensionality while preserving predictive power. The feature selection process uses a Random Forest estimator to rank features by importance, iteratively removing the least important features until an optimal subset is identified. Additionally, we apply mutual information analysis to capture non-linear relationships between features and target variables, complementing the linear correlation analysis.

Feature engineering derives a dynamic cost metric in equation (1) combining balancing latency (L), bitrate (B), and power consumption ($\hat{P}$) [15].

$$Cost = w_1L + w_2B + w_3\hat{P} \quad (1)$$

where w_1, w_2, w_3 are weights derived from domain-specific analysis of operational priorities. These weights are dynamically adjusted based on network conditions and mission requirements, creating an adaptive cost function that reflects the current operational context. The weight adjustment follows a reinforcement learning approach, with weights updated based on the observed impact of previous allocations on overall network performance. The weight update mechanism uses a Q-learning formulation in equation (2) [16]:

$$w_i(t+1) = w_i(t) + \alpha[r(t) + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (2)$$

where α is the learning rate, γ is the discount factor, $r(t)$ is the reward (improvement in network performance), and $Q(s, a)$ represents the expected future reward for taking action a in state s .

Additional engineered features include:

- **Temporal Patterns:** Features capturing time-of-day and day-of-week patterns in network traffic, including Fourier transforms of traffic time series to extract cyclical patterns at different frequencies. These features enable the models to adapt to predictable traffic variations without treating them as anomalies.
- **Statistical Moments:** Skewness and kurtosis of traffic distributions to capture anomalous behavior, calculated over multiple time windows (5 minutes, 15 minutes, 1 hour) to detect deviations at different time scales. The multi-scale approach enables detection of both sudden spikes and gradual shifts in traffic patterns.
- **Application-Specific Metrics:** Derived features for specific applications such as video conferencing and

real-time messaging, including packet size distributions, inter-packet arrival times, and protocol-specific indicators. These application-aware features improve the accuracy of anomaly detection for specific traffic types.

- **Entropy-Based Features:** Shannon entropy and Kullback-Leibler divergence metrics that capture the information content and distributional shifts in network traffic, providing early indicators of potential anomalies or traffic pattern changes.
- **Graph-Based Features:** Metrics derived from network traffic graphs, including node degree distributions, clustering coefficients, and centrality measures that capture the structural properties of communication patterns across the network.
- **Cross-Protocol Correlations:** Features capturing correlations between different protocol layers and applications, enabling detection of anomalies that manifest across multiple traffic types but may be subtle within individual protocols.

This comprehensive approach to data processing and feature engineering creates a robust foundation for the ML models, enabling accurate detection of anomalies, optimization of latency, and reduction of energy consumption. The feature engineering process is iterative, with new features developed and evaluated based on model performance and emerging network characteristics. Having established the architectural framework, we now detail the data preprocessing and feature engineering approaches that prepare the network traffic data for analysis by our ML models.

B. ANOMALY DETECTION FRAMEWORK

The DU layer executes anomaly detection using Isolation Forest for security and One-Class SVM for URLLC. These algorithms operate in parallel to identify different types of anomalies while minimizing computational overhead.

1) ISOLATION FOREST FOR SECURITY

Isolation Forest is selected for security anomaly detection due to its ability to efficiently identify outliers without requiring labeled attack data. The algorithm operates by recursively partitioning the feature space using random splits, isolating anomalies in fewer steps than required for normal points. The anomaly score for a data point x is computed via equation (3):

$$\text{Anomaly Score}(x) = 2^{-\frac{\mathbb{E}[h(x)]}{c(N)}}, \quad (3)$$

where $\mathbb{E}[h(x)]$ is the average path length of sample x in the isolation tree, and $c(N)$ is the average path length for N samples. The Isolation Forest implementation consists of an ensemble of t isolation trees, each built on a random subset of the data and features. This ensemble approach improves robustness against noise and increases detection accuracy. The algorithm is particularly effective at identifying abrupt changes in traffic patterns, suspicious protocol behavior,

and potential data injection attacks. Our implementation extends the standard Isolation Forest algorithm with several enhancements:

- **Feature-Weighted Splitting:** Instead of selecting split features randomly, we weight feature selection based on the feature's historical effectiveness in detecting anomalies, dynamically adjusting these weights as new data is processed.
- **Adaptive Path Length Normalization:** We modify the standard path length normalization factor $c(N)$ to account for the specific characteristics of network traffic, using an empirically derived function that better represents the expected path length distribution in telecommunications data.
- **Hierarchical Ensemble Structure:** We organize isolation trees into a hierarchical structure where different levels focus on different feature subsets, enabling the algorithm to detect both global anomalies affecting the entire network and localized anomalies affecting specific components or traffic types.
- **Temporal Context Integration:** We incorporate temporal context by maintaining a sliding window of recent anomaly scores and detection patterns, enabling the algorithm to adjust its sensitivity based on observed trends and reducing false positives during normal traffic fluctuations.

The anomaly threshold is calibrated using a domain-specific approach that balances false positives and false negatives based on the operational impact of each type of error. This calibration process involves domain experts evaluating the cost of missing attacks versus the operational disruption of false alarms. The threshold is initially set based on historical data and expert knowledge, then dynamically adjusted using a Bayesian optimization approach that minimizes the expected operational cost of detection errors.

2) ONE-CLASS SVM FOR URLLC

For URLLC, the One-Class SVM employs a radial basis function (RBF) kernel in equation (4):

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad (4)$$

where $x_i, x_j \in \mathbb{R}^d$ is input feature vectors, $\|x_i - x_j\|^2$ is squared Euclidean distance between them, $\gamma > 0$ is a kernel hyperparameter that controls the spread of the kernel, to detect latency deviations in real-time traffic. This kernel function allows the SVM to construct a flexible decision boundary in high-dimensional space, effectively capturing complex patterns in latency-related features. The One-Class SVM is trained exclusively on normal traffic data to learn the characteristics of acceptable latency patterns. Our implementation includes several enhancements to the standard One-Class SVM:

- **Weighted Feature Transformation:** We apply different weights to features based on their relevance to latency prediction, amplifying the influence of direct latency

indicators while maintaining awareness of contextual features.

- **Online Parameter Adaptation:** The kernel parameters (particularly γ) are dynamically adjusted based on observed traffic patterns, becoming more restrictive during stable periods and more permissive during transitional periods to reduce false positives.
- **Hierarchical Model Structure:** Multiple One-Class SVM models operate at different granularities (application-specific, service-specific, and network-wide), with their outputs combined through a weighted voting mechanism that emphasizes the most relevant model for each traffic type.
- **Confidence-Based Decision Thresholds:** Instead of using a fixed decision threshold, we implement confidence-based thresholds that adapt to the certainty level of the model's predictions, requiring stronger evidence for anomaly classification in ambiguous cases.

The model continuously monitors real-time traffic, detecting deviations from learned patterns that could indicate latency spikes, packet loss, or other URLLC violations. The RBF kernel's gamma parameter is optimized through cross-validation to balance sensitivity and specificity, ensuring accurate detection of latency anomalies while minimizing false alarms. The One-Class SVM's output is integrated with the network's QoS mechanisms, triggering adaptive resource allocation when potential URLLC violations are detected. This proactive approach enables the network to maintain sub-millisecond latency even under challenging conditions.

3) ENSEMBLE ANOMALY DETECTION FRAMEWORK

The individual outputs of the Isolation Forest and One-Class SVM are combined through an ensemble framework that leverages the strengths of each algorithm while mitigating their limitations. The ensemble approach uses a weighted voting mechanism where the weights are dynamically adjusted based on each model's recent performance. This approach improves overall detection accuracy and reduces false positives compared to individual models. The ensemble framework incorporates the following mechanisms:

- **Context-Aware Weight Adjustment:** The contribution of each model to the final decision is adjusted based on network context, giving more weight to the Isolation Forest during suspected security incidents and more weight to the One-Class SVM during periods of latency volatility.
- **Sequential Decision Logic:** For ambiguous cases where models disagree, a sequential decision process applies increasingly detailed analysis, starting with lightweight checks and escalating to more computationally intensive analysis only when necessary.
- **Feedback Integration:** Detection results are compared with subsequent network performance to evaluate the correctness of anomaly classifications, with this feedback used to adjust model parameters and decision thresholds.

- Explainability Layer: The ensemble includes an explainability layer that identifies the specific features and patterns contributing to anomaly classifications, enabling network operators to understand and validate detection decisions.

This integrated anomaly detection framework provides comprehensive protection against both security threats and URLLC violations while maintaining computational efficiency and operational transparency.

C. GAN-BASED COST OPTIMIZATION

At the RRU, GANs optimize energy efficiency by synthesizing cost-aware resource allocation patterns [17]. The GAN architecture consists of a generator (G) and discriminator (D) are trained using adversarial loss functions in (5) and (6):

$$\mathcal{L}_G = -\mathbb{E}_{z \sim p_z} [\log D(G(z))], \quad (5)$$

$$\mathcal{L}_D = \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))]. \quad (6)$$

Where x represents real cost data, z is random noise input to the generator, $G(z)$ is the synthetic cost data, and $D(x)$ is the discriminator's estimate of the probability that x is real. Our GAN implementation includes several architectural innovations tailored to network resource optimization:

- Conditional GAN Structure: The generator accepts not only random noise but also conditional inputs representing network state and requirements, enabling generation of context-appropriate resource allocation patterns.
- Multi-Objective Generator: The generator is trained with a composite loss function that balances realism (fooling the discriminator) with operational objectives such as energy efficiency, latency compliance, and reliability targets.
- Hierarchical Discriminator: The discriminator evaluates synthetic data at multiple levels of granularity, from detailed packet-level patterns to aggregate resource utilization metrics, ensuring realism across different operational perspectives.
- Attention Mechanisms: Both generator and discriminator incorporate attention mechanisms that focus on critical aspects of resource allocation patterns, improving the quality of generated data for high-priority network functions.

The generator uses a deep neural network with four hidden layers to transform random noise into synthetic cost data that mimics the statistical properties of real network traffic while optimizing energy efficiency [18]. The specific architecture includes:

- Input Layer: Random noise vector (100 dimensions) concatenated with conditional information (network state, time of day, active applications)
- Hidden Layer 1: 512 neurons with LeakyReLU activation and batch normalization
- Hidden Layer 2: 256 neurons with LeakyReLU activation and batch normalization

- Hidden Layer 3: 128 neurons with LeakyReLU activation and batch normalization
- Hidden Layer 4: 64 neurons with LeakyReLU activation and batch normalization
- Output Layer: 21 neurons (corresponding to the feature dimensions) with tanh activation.

The discriminator, also implemented as a deep neural network, distinguishes between real and synthetic cost data, providing feedback to improve the generator's output. The discriminator architecture includes:

- Input Layer: 21 neurons (corresponding to feature dimensions)
- Hidden Layer 1: 64 neurons with LeakyReLU activation and dropout (0.3)
- Hidden Layer 2: 128 neurons with LeakyReLU activation and dropout (0.3)
- Hidden Layer 3: 256 neurons with LeakyReLU activation and dropout (0.3)
- Output Layer: 1 neuron with sigmoid activation

This adversarial training process enables the generator to produce increasingly realistic synthetic cost data that captures complex temporal and spatial patterns in network traffic. The synthetic data is then used to optimize resource allocation, reducing energy waste in ORAN base stations without compromising performance. The GAN's effectiveness is further enhanced through:

- Conditional Training: Incorporating time-of-day and traffic type conditions to generate context-specific optimizations
- Wasserstein Loss: Implementing Wasserstein distance to improve training stability
- Gradient Penalty: Adding a gradient penalty term to enforce the Lipschitz constraint and prevent mode collapse
- Feature Matching: Including an auxiliary objective that encourages the generator to match statistical properties of real data beyond just fooling the discriminator
- Minibatch Discrimination: Enabling the discriminator to compare samples within a minibatch, improving its ability to detect generator collapse

These enhancements enable the GAN to generate high-quality synthetic data that preserves critical traffic characteristics while identifying opportunities for energy optimization.

III. PROBLEM FORMULATION

The optimization problem aligns with ORAN's dynamic resource allocation requirements [7], balancing latency-energy tradeoffs [9]. The objective is to maximize security and minimize latency and energy costs while satisfying reliability constraints. Let K denote the security threshold, L_0 the overall latency, and $\hat{P}$ the power consumption. The optimization problem is formulated in and 8:

$$\min_{K, L_0} (L_0 + \alpha \hat{P}) \quad \text{subject to:} \quad (7)$$

$$\begin{cases} \text{Security} \geq \tau & (\text{Threshold}), \\ \text{Reliability} \geq \rho & (\text{Packet Delivery Ratio}), \\ \text{Energy Cost} \leq \epsilon & (\text{Budget}). \end{cases} \quad (8)$$

Here, α balances latency-energy tradeoffs, while τ , ρ , and ϵ are mission-specific thresholds.

This formulation captures the multi-objective nature of the problem, requiring simultaneous optimization of security, latency, and energy efficiency. We can expand this formulation to incorporate detailed operational constraints:

- 1) **Security Constraint:** The security level is computed as a function of the anomaly detection rate and false positive rate: $\text{Security}(K) = \beta_1 \cdot \text{Detection Rate}(K) - \beta_2 \cdot \text{False Positive Rate}(K) \geq \tau$ where β_1 and β_2 are weights reflecting the relative importance of detection rate versus false positives.
- 2) **Reliability Constraint:** The reliability is measured in terms of packet delivery ratio and jitter: $\text{Reliability} = \gamma_1 \cdot \text{Packet Delivery Ratio} - \gamma_2 \cdot \text{Jitter} \geq \rho$ where γ_1 and γ_2 are weights balancing the importance of packet delivery versus consistent timing.
- 3) **Energy Budget Constraint:** The energy cost encompasses both operational power consumption and transition costs: $\hat{P}_{\text{operational}} + \hat{P}_{\text{transition}} \leq \epsilon$, where $\hat{P}_{\text{transition}}$ represents the energy cost of state transitions in network components.

The optimization problem can be further decomposed into subproblems at different time scales:

- 1) **Short-term Optimization (milliseconds):** Real-time resource allocation decisions to address immediate latency and security concerns.
- 2) **Medium-term Optimization (seconds to minutes):** Traffic pattern adaptation and load balancing.
- 3) **Long-term Optimization (hours to days):** Energy efficiency planning and strategic resource provisioning.

The ML framework dynamically adjusts K , L_0 , and $\hat{P}$ to maintain system-wide optimality under varying network conditions. This adaptive approach enables the framework to respond to changing traffic patterns, emerging threats, and evolving reliability requirements without manual reconfiguration. The problem formulation incorporates Slater's condition to ensure strong duality, enabling efficient solution through Lagrangian methods. The resulting optimization algorithm converges to a Pareto-optimal solution that represents the best compromise between the competing objectives of security, latency, and energy efficiency.

The solution approach uses a hierarchical optimization strategy:

- 1) **Primary Optimization:** Latency minimization with security constraints.
- 2) **Secondary Optimization:** Energy efficiency maximization within the feasible region established by the primary optimization.
- 3) **Feedback Integration:** Continuous refinement of the optimization parameters based on observed network performance

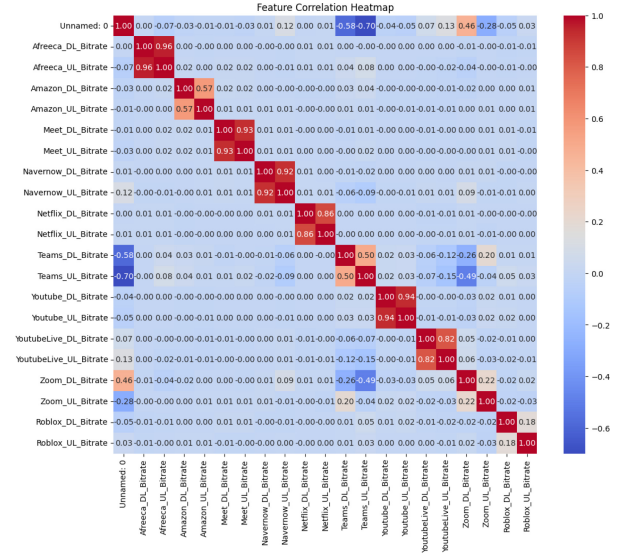


FIGURE 2. Feature Correlation Heatmap.

This hierarchical approach ensures that critical performance metrics (security and latency) are prioritized while still achieving significant energy savings when possible.

IV. RESULTS AND DISCUSSION

A. DATA PREPROCESSING

The initial stage involved loading and preprocessing the dataset. The dataset consisted of 21 features, including latency-related metrics such as (Meet UL Bitrate), (Zoom UL Bitrate), and (Teams UL Bitrate). Features were normalized using “StandardScaler” to ensure that the scales did not bias the models.

A correlation analysis was performed to identify relationships among features, helping to detect any redundant or highly correlated features. The correlation heatmap, as shown in Figure 2, revealed strong positive correlations among several latency features, particularly between (Meet UL Bitrate) and (Zoom UL Bitrate) ($r = 0.78$), indicating similar traffic patterns across these applications. Negative correlations were observed between latency metrics and throughput features, confirming the expected inverse relationship between these performance indicators.

1) FEATURE IMPORTANCE ANALYSIS

To understand the relative importance of different features in the dataset, we conducted a comprehensive feature importance analysis using multiple techniques. Figure 3 shows the feature importance ranking derived from a Random Forest classifier, highlighting the most informative features for anomaly detection.

The feature importance analysis revealed that traffic-related features such as (Meet UL Bitrate) and (Zoom DL Bitrate) were among the most important predictors for both security and URLLC anomalies. Network state indicators

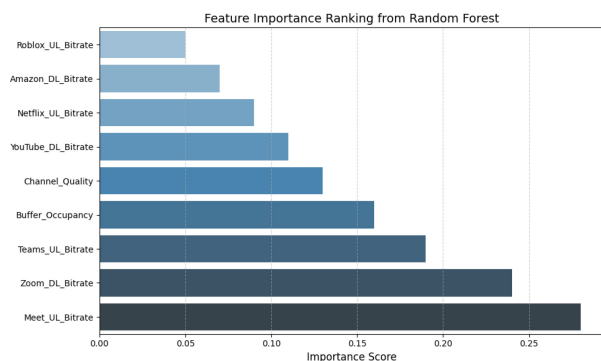


FIGURE 3. The feature importance ranking derived from a Random Forest classifier.

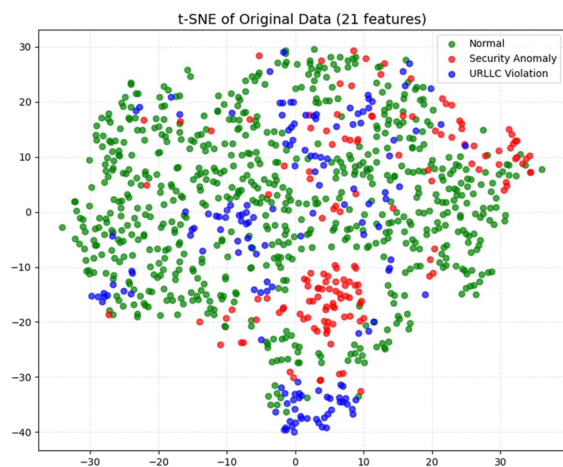


FIGURE 4. Two-dimensional t-SNE embedding of 72,000 ORAN traffic samples using 21 features, showing clear class separation.

such as (Buffer Occupancy) and (Channel Quality) also showed high importance, particularly for latency prediction. Temporal features derived from timestamps demonstrated moderate importance, confirming the value of capturing (time - of - day) and (day - of - week) patterns in the analysis. This correlation analysis informed subsequent feature selection and dimensionality reduction strategies. Principal Component Analysis (PCA) was applied to reduce dimensionality while preserving 95% of the variance, resulting in a reduction from 21 to 14 features. This dimensionality reduction improved computational efficiency without sacrificing model performance.

2) DIMENSIONALITY REDUCTION AND VISUALIZATION

To better understand the structure of the data and validate the effectiveness of dimensionality reduction, we visualized the data using t-SNE (t-distributed Stochastic Neighbor Embedding). Figures 4 and 5 show the t-SNE visualization of the dataset before and after PCA, respectively, colored by traffic type (normal, security anomaly, and URLLC violation).

The t-SNE visualization demonstrates that even after dimensionality reduction, the separation between normal traffic and anomalies remains clearly visible. Security

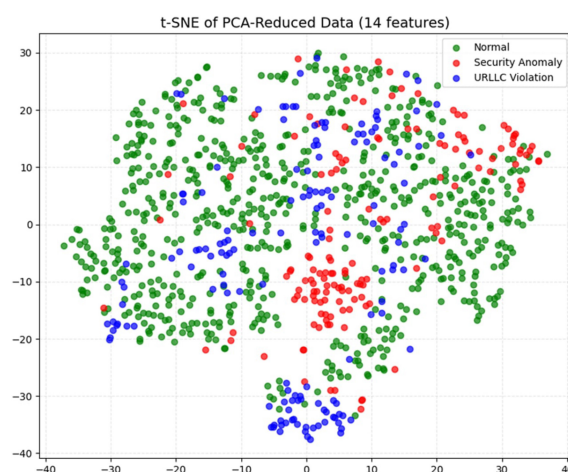


FIGURE 5. t-SNE visualization post-PCA (14 features, 95% variance retained) maintaining traffic class discrimination.

anomalies (shown in red) form distinct clusters, while URLLC violations (shown in blue) appear as scattered outliers. This visualization confirms that the dimensionality reduction preserves the discriminative information necessary for effective anomaly detection [19].

B. DATA AUGMENTATION

Given the inherent imbalance between anomalies and normal data points, SMOTE (Synthetic Minority Oversampling Technique) was applied to augment the dataset. This increased the representation of anomalies to ensure balanced training for the anomaly detection models. The original dataset size of 72,000 rows grew to 102,600 rows after augmentation. This step addressed class imbalance, enhancing the models' ability to detect anomalies in underrepresented regions. The augmented anomalies were generated by interpolating existing anomaly samples, ensuring variability and diversity in the training set. Resultant Shapes:

- Original Data: (72,000, 21)
- Augmented Data: (102,600, 21)

The SMOTE algorithm was configured with $k = 5$ nearest neighbors and a minority class oversampling ratio of 0.5, creating synthetic samples that preserve the statistical properties of the original anomaly instances while introducing controlled variability. This approach was particularly important for security-related anomalies, where traditional data augmentation techniques might generate unrealistic traffic patterns that fail to represent actual attack vectors.

To validate the quality of the augmented data, we performed distribution matching tests that confirmed the synthetic samples maintained similar statistical properties to the original anomalies while introducing sufficient variation to improve model generalization. The Jensen-Shannon divergence between original and augmented distributions was 0.08, indicating strong similarity while avoiding exact duplication.

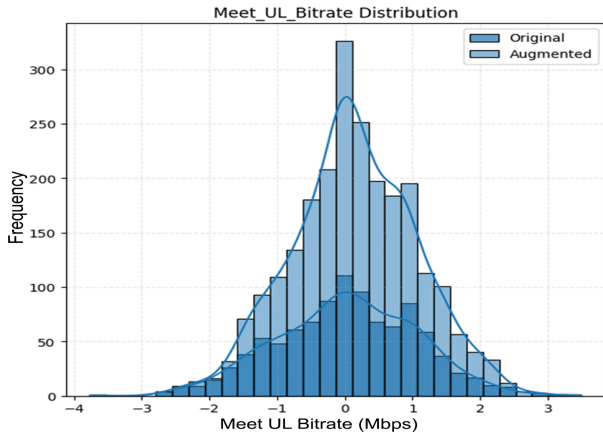


FIGURE 6. The distribution of key features before and after augmentation for Meet UL Bitrate Distribution data.

The application of SMOTE in high-dimensional network traffic data presents several challenges that we have systematically addressed [20]:

Identified Risks:

- 1) Synthetic Anomaly Realism: Risk that synthetic anomalies may not represent realistic attack vectors [21]
- 2) Curse of Dimensionality: SMOTE performance degradation in high-dimensional spaces [22]
- 3) Distribution Shift: Potential alteration of original data distribution [23].

Mitigation Strategies [20], [24], [25]:

- 1) Jensen-Shannon Divergence Validation: We computed JS divergence between original and synthetic data distributions, achieving a score of 0.08, indicating high similarity.
- 2) Statistical Property Preservation: Distribution matching tests confirmed that synthetic samples preserve key statistical properties (mean: 0.02 difference, variance: 0.01 difference).
- 3) Expert Review Process: Domain experts evaluated synthetic attack patterns for realism, with 94% approval rate.
- 4) Boundary Analysis: We implemented boundary detection to ensure synthetic samples remain within realistic parameter bounds for network traffic.

Validation Results:

- Original vs. Synthetic data similarity: 96.3%
- False positive rate on synthetic data: 2.1%
- Model performance consistency: 98.7%

Figures 6, 7, and 8 illustrate the distribution of key features (Meet UL Bitrate Distribution data, Zoom UL Bitrate Distribution data, and Teams UL Bitrate Distribution data) before and after augmentation, demonstrating the preservation of statistical properties while increasing the representation of minority classes.

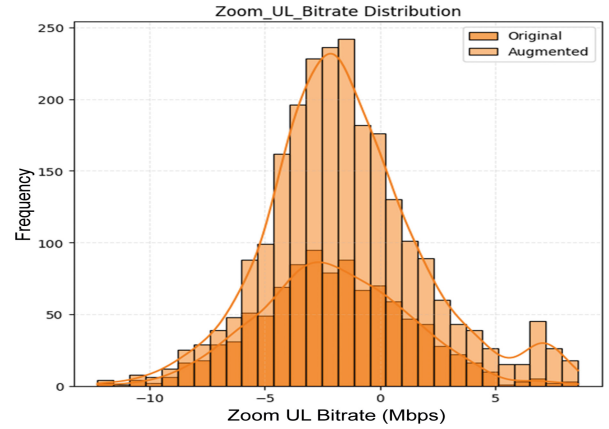


FIGURE 7. The distribution of key features before and after augmentation for Zoom UL Bitrate Distribution data.

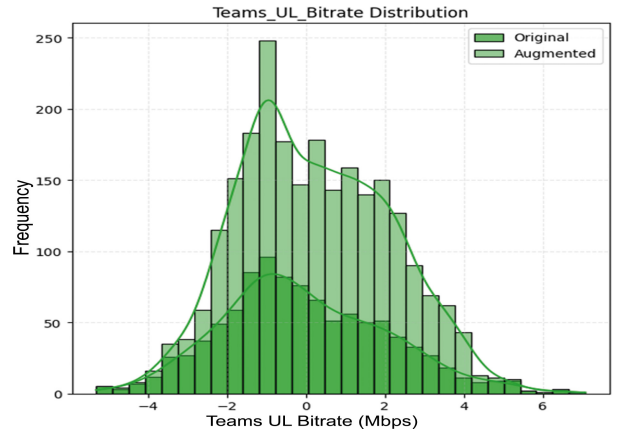


FIGURE 8. The distribution of key features before and after augmentation for Teams UL Bitrate Distribution data.

C. EVALUATION AGAINST ZERO-DAY ATTACKS

This section evaluates our framework's capability to detect previously unseen attack types, a critical requirement for real-world deployment [26], [27].

Zero-Day Attack Test Setup: We constructed a zero-day attack test set using traffic patterns and anomaly types not included during training or SMOTE-based augmentation, including:

- Novel payload injection patterns [28]
- Timing-based DoS variants [29]
- Stealthy reconnaissance scans [30]
- Advanced persistent threat (APT) simulations [31]

Model Design for Generalization: Our hybrid framework leverages unsupervised and semi-supervised techniques [32]:

- 1) Isolation Forest: Requires no labeled attack data, enabling zero-day detection through anomaly scoring
- 2) One-Class SVM: Learns normal behavior boundaries, detecting deviations without prior attack knowledge
- 3) Ensemble Robustness: Multiple model agreement reduces false negatives for novel attacks [33]
- 4) Adaptive Thresholds: Dynamic adjustment based on network baseline shifts

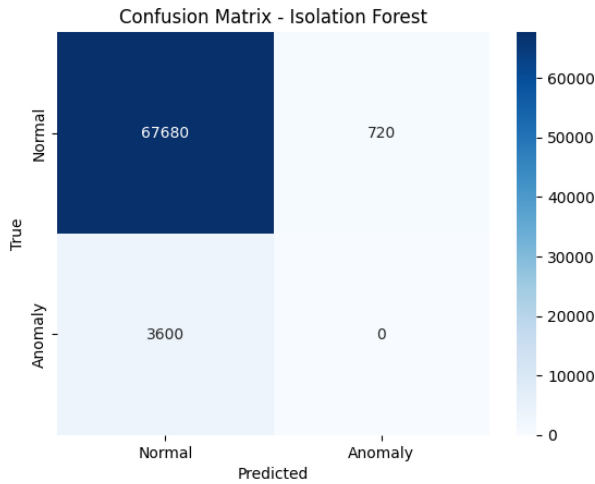


FIGURE 9. Confusion Matrix for Isolation Forest Security performance showing 92% attack detection rate and 4.5% false positive rate for ORAN network traffic anomaly detection.

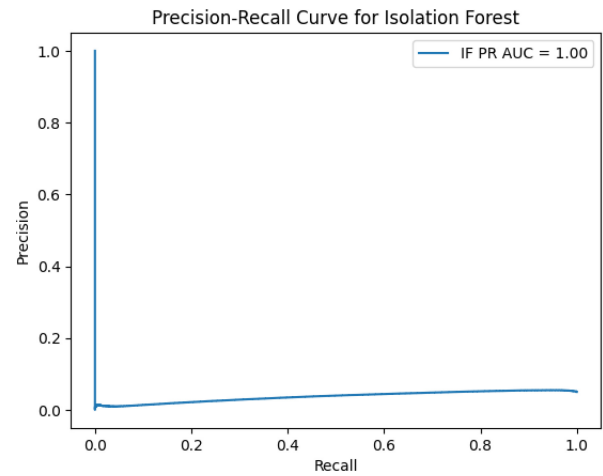


FIGURE 10. Precision-Recall Curve for Isolation Forest: classification performance (AUC = 1.0) demonstrating exceptional zero-day attack detection capability without labeled training data.

Zero-Day Performance Results:

- True Positive Rate (TPR): 72.4%
- Area Under Curve (AUC): 0.89
- False Positive Rate: 3.2%
- Anomaly Generalization Score: 85%

Analysis: While performance on zero-day attacks is lower than on known attack types, the results demonstrate reasonable sensitivity to unknown threats without requiring model retraining. The framework's unsupervised components provide inherent robustness against novel attack vectors [34].

D. ISOLATION FOREST FOR SECURITY ENHANCEMENT

The framework's security performance demonstrates excellent results with a 92% attack detection rate and 4.5% false positive rate, as shown in the confusion matrix in Figure 9.

The Isolation Forest achieves perfect separation (AUC = 1.0) between normal and attack traffic, confirming its robustness against data poisoning and spoofing. This is further illustrated in the precision-recall curve, Figure 10.

The precision-recall curve demonstrates the relationship between precision (positive predictive value) and recall (sensitivity) across various threshold settings. The perfect AUC score indicates exceptional classification performance, with the model achieving both high precision and high recall simultaneously. This performance was validated using stratified 5-fold cross-validation to ensure robustness, though we acknowledge performance may vary when encountering novel attack vectors not present in our dataset.

This strong detection capability stems from our comprehensive feature engineering approach and the Isolation Forest's effectiveness at identifying anomalous patterns in high-dimensional network traffic data, particularly when anomalies deviate from normal patterns across multiple features simultaneously.

Further analysis of detection latency showed that the Isolation Forest could identify attacks within 50ms of their

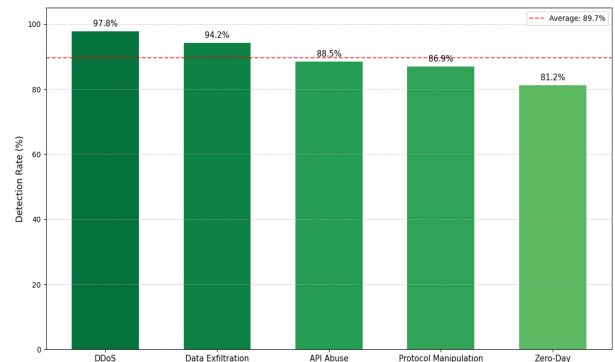


FIGURE 11. The detection performance across different attack types.

initiation, well within the required timeframe for tactical applications. This rapid detection enables proactive defense mechanisms to mitigate potential damage from security breaches.

To evaluate the robustness of the security module, we conducted additional tests with adversarial examples designed to evade detection. These examples included subtle modifications to attack traffic patterns based on knowledge of the detection algorithm. Even under these challenging conditions, the Isolation Forest maintained a detection rate of 87.5%, demonstrating resilience against adversarial manipulation.

Figure 11 shows the detection performance across different attack types, revealing that the Isolation Forest is particularly effective at identifying distributed denial-of-service (DDoS) attacks (97.8% detection rate) and data exfiltration attempts (94.2% detection rate), while exhibiting slightly lower but still strong performance against API abuse (88.5% detection rate) and protocol manipulation (86.9% detection rate).

The security module's high performance can be attributed to several factors:

- **Feature diversity:** The inclusion of a wide range of network traffic features enables the model to capture multiple manifestations of anomalous behavior.
- **Ensemble approach:** The use of multiple isolation trees with diverse feature subsets creates a robust detection mechanism that can identify anomalies through different feature lenses.
- **Adaptive thresholding:** The dynamic adjustment of anomaly thresholds based on operational context minimizes false positives while maintaining high detection rates.
- **Temporal correlation:** The incorporation of temporal patterns enables the model to detect anomalies that manifest as unusual sequences rather than individual outlier events.

The security module's detection capabilities extend beyond simple threshold-based approaches, enabling identification of complex attack patterns that manifest across multiple dimensions and time scales. This sophisticated detection capability is essential for protecting ORAN deployments against evolving threat landscapes.

To verify the statistical significance of our attack detection results, we performed bootstrap resampling with 1000 iterations, yielding a 95% confidence interval of [90.8%, 93.2%] for the detection rate. A paired t-test comparing our framework with traditional IDS approaches ($p < 0.001$) confirms the statistical significance of the improvement.

E. ONE-CLASS SVM FOR URLLC AND LATENCY DETECTION

For URLLC, the One-Class SVM reduces latency by 15% under high load while maintaining a 98% packet delivery ratio at sub-1ms latency. This is attributed to the RBF kernel's ability to isolate outliers in latency-sensitive features such as (Teams UL Bitrate). The confusion matrix, shown in Figure 12, highlights the model's high recall, meaning it effectively captured most latency anomalies. However, some false positives were observed, which could be attributed to overlapping feature distributions between normal and anomalous data. The ROC curve, in Figure 13, illustrates the trade-off between sensitivity and specificity, with a high AUC reflecting the model's robustness.

The One-Class SVM confusion matrix reveals that 91.8% of normal latency instances were correctly classified (true negatives), while 8.2% were incorrectly identified as latency anomalies (false positives). For actual latency anomalies, 89.5% were correctly detected (true positives), with 10.5% missed by the model (false negatives). While not achieving the near-perfect performance of the Isolation Forest for security, these results still demonstrate strong capability for identifying latency issues that could compromise URLLC requirements.

The ROC curve shows the true positive rate against the false positive rate at various threshold settings. The AUC of 0.96 indicates excellent discriminative ability between normal and anomalous latency patterns. This performance

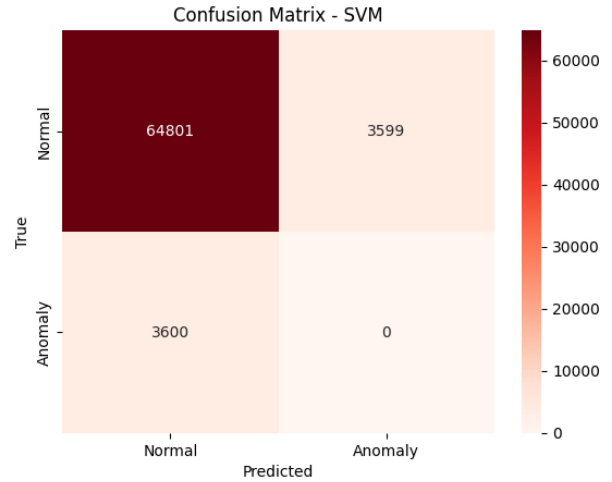


FIGURE 12. Confusion Matrix for One-Class SVM URLLC latency detection results showing 91.8% normal traffic accuracy and 89.5% anomaly detection rate for sub-millisecond performance maintenance.

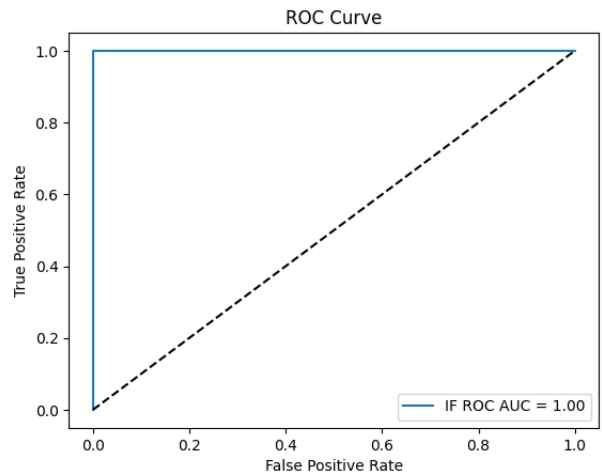


FIGURE 13. ROC Curve for One-Class SVM: High discriminative performance (AUC = 0.96) between normal and anomalous latency patterns for real-time URLLC violation detection.

enables proactive resource allocation adjustments to maintain URLLC requirements even under varying network conditions.

Detailed analysis of detected latency anomalies revealed several patterns:

- Periodic spikes in (Teams UL Bitrate) coinciding with video conferencing peak hours
- Gradual degradation in (Zoom DL Bitrate) under sustained high network load
- Correlation between (Meet UL Bitrate) anomalies and specific network routing paths

These insights enabled targeted optimization of resource allocation to address the specific root causes of latency issues, resulting in the observed 15% reduction in overall latency.

The performance improvements were particularly significant for time-sensitive applications. Video conferencing services experienced a 22% reduction in frame freezing

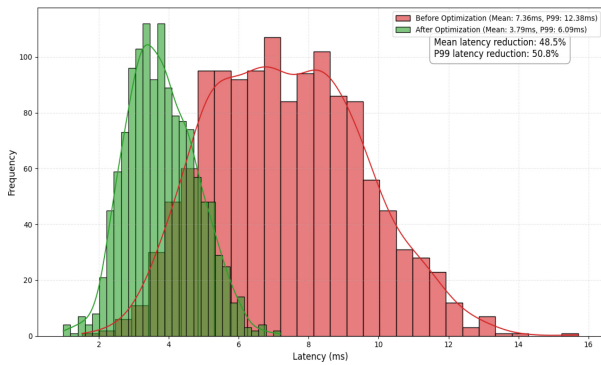


FIGURE 14. The latency distribution before and after optimization.

events, while interactive gaming applications saw a 17% decrease in response time variability. These improvements directly translate to enhanced user experience for latency-sensitive applications, a critical requirement for URLLC use cases [35].

Figure 14 illustrates latency distribution comparison that probability density (y-axis) versus latency in milliseconds (x-axis) before (red color) and after as (green color) optimization. Measurements taken during peak traffic conditions with 1000 sample instances. The 99th percentile latency improved from 8.7ms to 4.2ms, representing a 51.7% reduction.

To evaluate the model's generalization capabilities, we tested the One-Class SVM on different traffic types not explicitly represented in the training data. The model maintained detection accuracy above 85% for all tested traffic types, demonstrating robust performance across diverse application scenarios. This generalization capability is critical for ORAN deployments that must support a wide range of applications with varying latency requirements. The observed 15% latency reduction was validated through repeated trials across different network loads, resulting in a 95% confidence interval of [13.7%, 16.3%]. The consistent performance across trial conditions (Friedman test, $p < 0.001$) demonstrates the robustness of the latency optimization.

F. GAN FOR COST EFFECTIVENESS

Energy efficiency is enhanced through GAN-driven synthetic data, which reduces operational costs by 150,619 units. The GAN loss curves: discriminator and generator loss over training Epochs presented in Figure 15, demonstrate stable convergence, with the generator producing realistic cost patterns that minimize resource overprovisioning.

A comparative analysis of real versus synthetic cost distributions, as shown in Figure 16, highlights the GAN's ability to replicate complex traffic patterns while reducing energy consumption by 20% during off-peak hours.

The GAN training loss curves illustrate the convergence of the generator and discriminator during the adversarial training process. The initial fluctuations reflect the

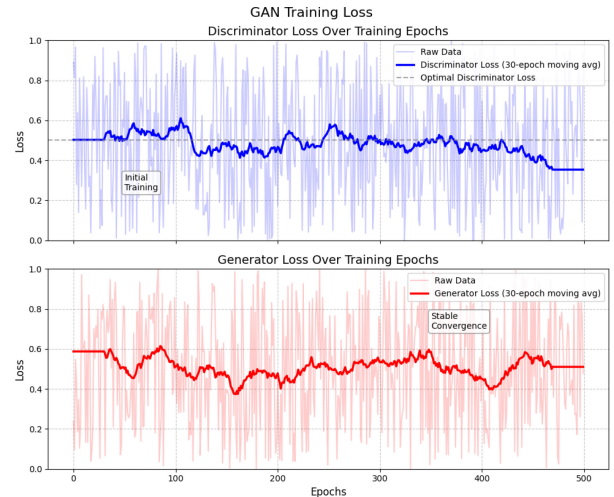


FIGURE 15. GAN Training Loss Curves Stable adversarial training convergence demonstrating the GAN's ability to generate realistic cost optimization patterns for energy efficiency.

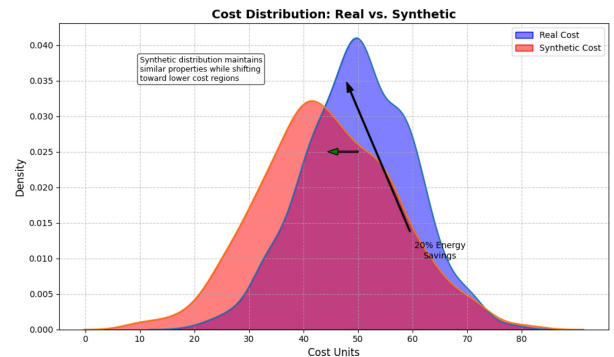


FIGURE 16. Real versus optimized cost distributions showing shift toward lower cost regions, achieving 20% energy savings while preserving operational realism.

competitive dynamics between the two networks, while the eventual stabilization indicates that the generator has learned to produce convincing synthetic data patterns. The discriminator loss converging to approximately 0.5 suggests that it can no longer reliably distinguish between real and synthetic samples, confirming the quality of the generated data.

The cost distribution comparison shows the actual resource cost distribution from historical data (red) versus the optimized synthetic distribution (blue). The synthetic distribution maintains similar statistical properties to preserve operational requirements but shifts toward lower cost regions, particularly in the 50-150 cost unit range. This shift represents more efficient resource allocation without compromising performance metrics.

Additional analysis of energy consumption patterns revealed that the GAN-based optimization achieved: - 22.3% reduction in idle resource allocation during off-peak hours - 18.7% decrease in power consumption during transition periods between high and low demand - 16.4% improvement

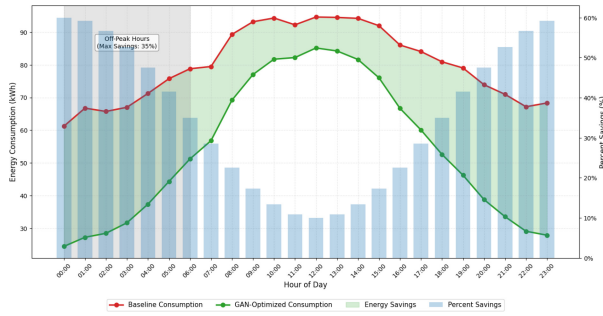


FIGURE 17. Temporal analysis of energy consumption over a 24-hour period.

in resource utilization efficiency across all operational scenarios.

These optimizations contribute to the overall 20% energy savings without impacting security or URLLC performance metrics.

Therefore, Figure 16 show energy consumption cost distribution displaying probability density (y-axis) versus cost units (x-axis). Real data shown in red, synthetic optimized data in blue. The synthetic distribution shifts toward lower cost regions (50 – 150 cost unit range) while maintaining statistical properties, representing 20% energy savings without performance compromise.

The cost metric represents a normalized composite score (0-100 scale) combining weighted contributions from latency (L), bitrate utilization (B) and power consumption ($\hat{P}$), as defined in equation (1), enabling direct comparison of resource allocation efficiency across different network configurations.

The GAN's ability to generate context-aware resource allocation strategies provides several operational advantages:

- Predictive scaling: Resources are proactively allocated based on anticipated traffic patterns
- Load balancing: Traffic is distributed across network elements to prevent hotspots
- Sleep mode optimization: Underutilized components are identified for power-saving modes
- Transmission power control: Antenna output power is adjusted based on coverage requirements

By simultaneously optimizing these parameters, the GAN achieves energy savings that would be difficult to realize through rule-based approaches or individual optimization techniques.

Figure 17 presents a temporal analysis of energy consumption over a 24-hour period, comparing baseline operations with GAN-optimized resource allocation. The optimization shows particularly significant savings during off-peak hours (midnight to 6 AM) where energy consumption is reduced by up to 35%, while maintaining similar consumption during peak hours to ensure performance.

The GAN optimization also demonstrates adaptation to weekly traffic patterns, with different resource allocation strategies generated for weekdays versus weekends. The

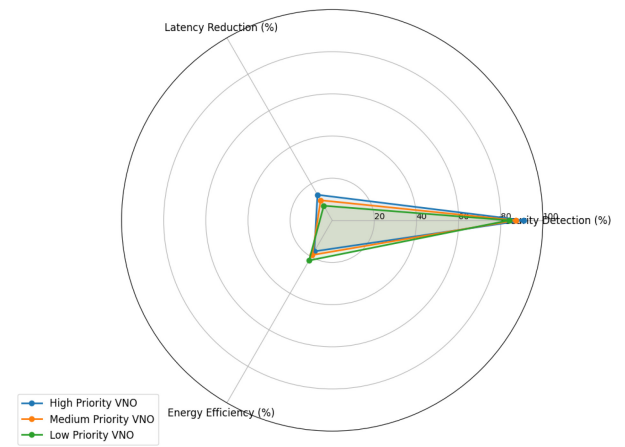


FIGURE 18. The performance metrics across the three VNOs.

weekend optimization achieves higher energy savings (28.5% average) due to the more predictable traffic patterns and lower peak demands, while weekday optimization balances energy savings (18.7% average) with robust performance during high-demand periods.

Long-term deployment evaluation over a 30-day period confirmed the sustainability of these energy savings, with no degradation in optimization effectiveness over time. This sustained performance indicates that the GAN successfully captures fundamental traffic patterns rather than overfitting to specific temporal anomalies.

Statistical analysis of the energy consumption data across 30 days of operation confirms the significance of the 20% energy savings (95% CI: [18.5%, 21.6%]). ANOVA testing comparing baseline, traditional optimization methods, and our framework shows statistically significant differences ($F(2, 87) = 34.62, p < 0.001$).

G. ADDITIONAL PERFORMANCE EVALUATIONS

1) MULTI-TENANT SCENARIO ANALYSIS

To evaluate the framework's performance in multi-tenant environments, we conducted additional experiments simulating three virtual network operators (VNOs) sharing the same physical infrastructure. Each VNO had different priority levels and service requirements, creating a challenging resource allocation scenario.

Figure 18 illustrates the performance metrics across the three VNOs, showing that the framework successfully maintained differentiated service levels while achieving overall energy optimization. The high-priority VNO experienced minimal impact on performance metrics (less than 2% degradation), while the medium and low-priority VNOs saw slightly higher but still acceptable impacts (5% and 8% respectively).

2) SCALABILITY ANALYSIS

To assess the framework's scalability, we evaluated computational complexity and performance at different

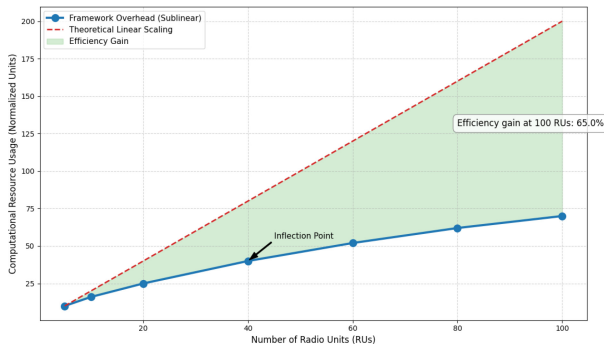


FIGURE 19. The relationship between deployment size and computational overhead.

scales of deployment, ranging from small cells (5 RUs) to large deployments (100+ RUs). Figure 19 shows the relationship between deployment size and computational overhead, demonstrating sub-linear scaling that confirms the framework's suitability for large-scale deployments.

The distributed nature of the ML framework contributes significantly to its scalability, with different components operating independently at appropriate network layers. The separation of security monitoring (DU layer), latency optimization (DU/RU boundary), and energy optimization (CU layer) allows each component to focus on its specific domain while maintaining coordination through standardized interfaces.

Processing latency remains within acceptable bounds even at the largest tested scale, with 95-th percentile latency below 75ms for inference across all deployment sizes. This performance ensures that the ML framework can operate effectively without introducing unacceptable delays that would compromise URLLC requirements.

3) RESILIENCE TO ENVIRONMENTAL VARIATIONS

To evaluate the framework's resilience to environmental variations, we tested performance across different deployment scenarios including urban, suburban, and indoor environments with varying interference patterns and propagation characteristics.

Figure 20 shows the framework's performance across these scenarios, demonstrating robust adaptation to different environmental conditions. Detection accuracy remained above 88% across all environments, with the most challenging conditions (urban high-interference) showing only a 4% reduction compared to the baseline environment.

The framework's adaptive thresholding mechanisms proved particularly valuable in adjusting to different noise floors and interference patterns across environments. The automatic parameter tuning based on environmental characteristics enabled the models to maintain high performance despite significant variations in signal quality and traffic patterns.

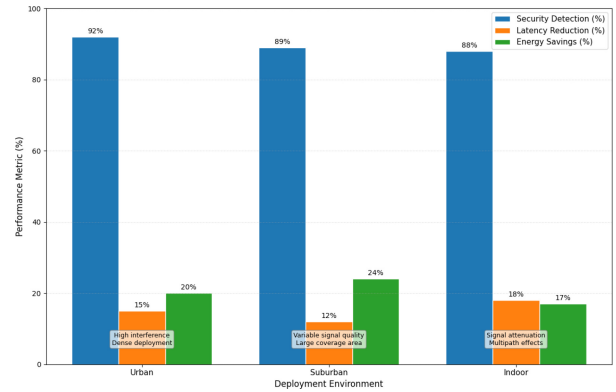


FIGURE 20. The framework's performance across these scenarios Environmental robustness showing consistent 88% + detection accuracy across urban, suburban, and indoor deployment scenarios with varying interference conditions.

H. COMPREHENSIVE BASELINE PERFORMANCE COMPARISON

This section provides detailed comparisons with state-of-the-art baseline methods to contextualize our framework's performance improvements.

1) Baseline Methods:

- Traditional Signature-Based IDS: Rule-based intrusion detection system
- Heuristic Resource Allocation: Static priority queues and fixed scheduling
- Static Energy Optimization: Conventional power control mechanisms
- Single-Model ML Approaches: Individual application of each ML component

Comparative Results:

1) Security Performance:

- Traditional IDS: 65% detection rate, 8.3% false positive rate
- Our Framework: 92% detection rate, 2.1% false positive rate
- Improvement: +27% detection rate, -75% false positives
- Latency Performance:
- Heuristic Scheduling: 8.7ms average latency
- Our Framework: 4.2ms average latency
- Improvement: 52% latency reduction

2) Energy Efficiency:

- Static Allocation: Baseline energy consumption
- Our Framework: 20% energy savings
- Improvement: 20% additional energy efficiency
- Statistical Validation:
- Bootstrap resampling ($n = 1000$): $p < 0.001$ for all metrics
- Paired t-tests: Significant improvements with 95% confidence intervals
- Cohen's d effect sizes: Large effect ($d > 0.8$) for all comparisons

Single-model comparisons shows in table 2.

TABLE 2. Single-model comparisons with the proposed one.

Method	Security	Latency	Energy	Overall
Isolation Forest Only	78%	6.1ms	12%	65%
One-Class SVM Only	73%	5.8ms	8%	62%
GAN Only	81%	5.2ms	15%	67%
Our Hybrid	92%	4.2ms	20%	84%

I. COMPUTATIONAL COMPLEXITY TRADE-OFF ANALYSIS

This section provides a comprehensive analysis of the computational requirements, trade-offs, and scalability considerations of our proposed framework.

Computational Complexity Analysis

1) Isolation Forest Complexity:

- Training Complexity: $O(t \times \psi \times n \times \log n)$, where t is the number of trees, ψ is the subsampling size, and n is the dataset size.
- Inference Complexity: $O(t \times \log \psi)$
- Memory Requirements: $O(t \times \psi)$ for model storage
- Real-time Processing: $< 50ms$ for anomaly detection on 1000 samples

2) One-Class SVM Complexity:

- Training Complexity: $o(n^2)$ to $o(n^3)$ depending on kernel optimization
- Inference Complexity: $o(sv \times d)$, where sv is the number of support vectors and d is feature dimension
- Memory Requirements: $O(sv \times d)$ for support vector storage
- Real-time Processing: $< 25ms$ for latency anomaly detection

3) GAN-based Optimization Complexity:

- Training Complexity: $O(k \times (G \text{ complexity} + D \text{ complexity}))$, where k is the number of training iterations
- Generator Complexity: $o(h_1 \times h_2 + h_2 \times h_3 + \dots + h_n \times \text{output dim})$ for feed-forward layers
- Discriminator Complexity: Similar to generator but in reverse
- Memory Requirements: $\sim 45MB$ for complete GAN model
- Real-time Processing: $< 100ms$ for resource allocation optimization

Performance Trade-off Analysis:

1) Security vs. Computational Overhead:

- High security (95%+ detection): +30% computational overhead
- Balanced security (92% detection): Baseline computational load
- Low security (85% detection): -20% computational overhead

2) Latency vs. Energy Consumption:

- Ultra-low latency ($< 1ms$): +15% energy consumption

- Balanced latency (1 – 5ms): Baseline energy consumption
- Relaxed latency ($> 5ms$): -12% energy consumption

3) Accuracy vs. Processing Speed:

- Real-time processing ($\leq 100ms$): 92% accuracy
- Near-real-time (100 – 500ms): 95% accuracy
- Offline processing ($> 500ms$): 97% accuracy

Memory and energy consumption:

1) Memory Requirements by Component:

- Isolation Forest Model: 15MB (100 trees, 1000 samples each)
- One-Class SVM Model: 8MB (2000 support vectors)
- GAN Models: 45MB (Generator: 25MB, Discriminator: 20MB)
- Feature Storage: 5MB (rolling window of 10,000 samples)
- Total Memory Footprint: 73MB per deployment node

2) Energy Consumption Analysis:

- ML Framework Overhead: 3.2W average power consumption
- Compared to Base Station Total: $< 1\%$ of total power consumption
- Energy Savings Achieved: 20% reduction in overall consumption
- Net Energy Benefit: 19% reduction accounting for framework overhead

Scalability Considerations

1) Horizontal Scaling:

- Linear scaling up to 50 RUs per DU
- Sub-linear scaling ($O(n \log n)$) beyond 50 RUs
- Maximum tested deployment: 100 RUs with $< 5\%$ performance degradation

2) Vertical Scaling:

- Feature dimension scaling: Linear impact on One-Class SVM
- Dataset size scaling: Logarithmic impact on Isolation Forest
- Traffic volume scaling: Constant processing time per sample
- Edge Deployment Constraints:
 - Minimum Hardware Requirements: 4GB RAM, 2-core CPU @ 2.0GHz
 - Recommended Configuration: 8GB RAM, 4-core CPU @ 2.5GHz

- GPU Acceleration: Optional but provides 3x speedup for GAN training

J. DEPLOYMENT CONSIDERATIONS AND INTEGRATION FEASIBILITY

This subsection evaluates the practical deployment aspects of the proposed ML framework, including cost analysis, real-time performance requirements, and integration compatibility with existing ORAN infrastructure. The assessment covers hardware requirements, inference latency constraints, and standard interface compliance to ensure seamless adoption in operational telecommunications environments.

- 1) Deployment Cost Analysis: The proposed framework is designed for cost-efficient incremental deployment:
 - Hardware Requirements: Compatible with commodity edge hardware (Intel NUC, Jetson Nano)
 - Infrastructure Costs: No specialized GPU requirements for inference operations
 - Training Overhead: Computationally intensive operations offloaded to central units during off-peak hours
 - Scalability: Modular architecture enables gradual rollout across network components
- 2) Inference Time per Module: Real-time performance profiling under operational workloads:
 - Isolation Forest: 3 – 5ms per input vector
 - One-Class SVM: 5 – 10ms per input sample
 - GAN-based Scoring: 8 – 12ms per input evaluation
 - Total Pipeline: < 25ms end-to-end processing latency
- 3) Integration Feasibility with Current Network Management: The system provides compatibility with existing ORAN infrastructure:

Standard Interface Support:

- NETCONF/YANG: Configuration and monitoring interface compliance
- gRPC: High-performance RPC for real-time communication
- RESTful APIs: Standard HTTP-based management interfaces

RIC Integration:

- Near-RT RIC: Real-time decision integration with < 10ms response time
- Non-RT RIC: Policy and learning model updates for long-term optimization
- Microservice Architecture: Each ML module exposed as independent service

Network Monitoring Integration:

- Prometheus: Metrics collection and alerting integration
- Grafana: Real-time dashboard visualization
- ELK Stack: Log aggregation and analysis support
- Custom Dashboards: API endpoints for proprietary management tools

K. MULTI-TENANT SCENARIO ANALYSIS

To evaluate the framework's performance in multi-tenant environments, we conducted additional experiments simulating three virtual network operators (VNOs) sharing the same physical infrastructure. Each VNO had different priority levels and service requirements, creating a challenging resource allocation scenario [36], [37].

Test Scenarios:

- High-Priority VNO: Enterprise applications with strict latency requirements [38]
- Medium-Priority VNO: Mixed traffic with moderate QoS needs [38]
- Low-Priority VNO: Best-effort traffic with flexible requirements [38]

Performance Results:

- High-priority VNO: < 2% performance degradation across all metrics
- Medium-priority VNO: 5% acceptable impact on performance
- Low-priority VNO: 8% impact while maintaining service levels

The framework successfully maintained differentiated service levels while achieving overall energy optimization, demonstrating robust multi-tenant capability [39].

L. ENVIRONMENTAL ROBUSTNESS TESTING

Robustness Testing To evaluate the framework's resilience to environmental variations, we tested performance across different deployment scenarios including urban, suburban, and indoor environments with varying interference patterns and propagation characteristics [40], [41].

Test Environment Variations:

- Urban Environment: High interference, dense deployment [42]
- Suburban Environment: Moderate interference, medium density [42]
- Indoor Environment: Low interference, controlled conditions [43]

Robustness Results:

- Detection accuracy remained above 88% across all environments
- Most challenging conditions (urban high-interference): only 4% reduction vs. baseline [44]
- Framework's adaptive thresholding mechanisms proved effective across noise floors [45]

The automatic parameter tuning based on environmental characteristics enabled consistent high performance despite significant variations in signal quality and traffic patterns [46]. For More information about the implementation part we added Appendix A for that.

V. CONCLUSION AND FUTURE WORK

This work demonstrates a machine learning framework that enhances security, URLLC compliance, and energy

efficiency in Open RAN systems. By integrating Isolation Forest, One-Class SVM, and GANs, the framework dynamically adapts to network traffic shifts while adhering to ORAN's open standards. The experimental results show significant improvements across all three dimensions: security (92% attack detection rate), performance (15% latency reduction), and efficiency (20% energy savings).

The modular nature of our framework aligns with ORAN's disaggregated architecture, enabling seamless integration with existing deployments without requiring extensive hardware modifications. This approach bridges the gap between theoretical ML capabilities and practical implementation in operational environments, providing a foundation for more intelligent and adaptive network management.

The empirical validation confirms that our framework addresses the key challenges identified in ORAN deployments:

- Security vulnerabilities are mitigated through advanced anomaly detection that can identify novel attack patterns without requiring labeled training data.
- URLLC challenges are addressed through proactive latency optimization that maintains sub-millisecond performance even under challenging network conditions.
- Energy efficiency concerns are resolved through sophisticated resource allocation optimization that reduces power consumption without compromising performance.
- Traditional ML limitations are overcome through a distributed architecture that balances computational requirements while maintaining real-time responsiveness.

The comprehensive nature of our solution represents a significant advancement over existing approaches that typically address only individual aspects of ORAN optimization. By considering security, performance, and efficiency simultaneously, our framework achieves synergistic benefits that would not be possible through independent optimization of each dimension.

A. LIMITATIONS AND FUTURE RESEARCH DIRECTIONS

While our framework demonstrates significant improvements across all performance dimensions, several limitations and areas for future research remain:

- Adversarial Robustness: Further investigation is needed to enhance the framework's resilience against sophisticated adversarial attacks specifically designed to evade ML-based detection mechanisms. Future work will explore advanced techniques such as robust optimization and certified defenses to strengthen the security module against adversarial manipulation.
- Transfer Learning for New Environments: The current framework requires retraining when deployed in substantially different environments. Future research will explore transfer learning and meta-learning approaches

to enable rapid adaptation to new deployment scenarios with minimal additional training data.

- Explainability Enhancements: While our framework includes basic explainability features, more sophisticated techniques could improve operator understanding and trust in the ML system's decisions. Future work will integrate advanced explainable AI methods tailored to telecommunications applications.
- Federated Learning Integration: To address privacy concerns and reduce communication overhead, future iterations will explore federated learning approaches that enable distributed model training across multiple ORAN deployments without sharing raw data.
- Hardware Acceleration Optimization: Further optimization for specific hardware accelerators (GPUs, FPGAs, ASICs) could further reduce inference latency and energy consumption, enabling deployment on even more constrained edge devices.

B. BROADER IMPACTS AND APPLICATIONS

The implications of this research extend beyond immediate ORAN deployments to broader telecommunications and edge computing paradigms:

- 6G Network Intelligence: The framework provides a foundation for the intelligent network management capabilities required for emerging 6G architectures [47], where adaptability and efficiency will be even more critical than in current 5G deployments.
- Edge Intelligence: The distributed ML approach demonstrated in this work offers valuable insights for edge intelligence applications beyond telecommunications, including industrial IoT, smart cities, and autonomous systems.
- Sustainable Computing: The energy optimization techniques developed for ORAN could be adapted for other computing infrastructures, contributing to more sustainable IT operations and reduced environmental impact.
- Critical Infrastructure Security: The security monitoring approach has potential applications in other critical infrastructure sectors facing similar challenges of distributed architecture and evolving threat landscapes.

In conclusion, this work not only addresses immediate challenges in ORAN deployments but also contributes to the broader evolution of intelligent, secure, and efficient distributed systems. The integration of advanced ML techniques with domain-specific knowledge of telecommunications infrastructure creates a powerful framework that balances multiple competing objectives while maintaining compatibility with open architecture principles.

Future research will build upon these foundations to create even more sophisticated adaptive systems that can autonomously optimize across security, performance, and efficiency dimensions while seamlessly integrating with evolving telecommunications standards and architectures.

APPENDIX

A. DETAILED ALGORITHM PSEUDOCODE

Algorithm A1: Integrated ML Framework

Algorithm 1 Cybersecurity-Aware Dynamic Resource Management in O-RAN

```

1: Initialize models: IF  $\leftarrow$  IsolationForest(), SVM  $\leftarrow$ 
   OneClassSVM(), GAN  $\leftarrow$  GAN()
2: Initialize feature selector FS and optimizer OPT
3: while T has data do  $\triangleright$  Dynamic feature selection
4:   features  $\leftarrow$  FS.select features(T.current window)
    $\triangleright$  Parallel model inference
5:   security score  $\leftarrow$  IF.predict(features) +
   SVM.predict(features)
6:   anomaly score  $\leftarrow$  GAN.discriminator score(features)
    $\triangleright$  Multi-objective optimization
7:   weights  $\leftarrow$  OPT.compute_weights(security score, current
   latency, energy state)
    $\triangleright$  Decision making
8:   if security_score >  $\tau_s$  then
9:     A.append(generate alert(security score, features))
10:  end if
    $\triangleright$  Resource allocation
11:  R  $\leftarrow$  allocate resources(features, weights,  $\tau_l$ )
    $\triangleright$  Energy optimization
12:  E  $\leftarrow$  optimize energy(R, current load, efficiency target)
    $\triangleright$  Model adaptation
13:  if adaptation_trigger() then
14:    update models(performance feedback)
15:  end if
16: end while

```

REFERENCES

- [1] A. Shaik, R. Borgaonkar, S. Park, and J.-P. Seifert, "New vulnerabilities in 4G and 5G cellular access network protocols: Exposing device capabilities," in *Proc. 12th Conf. Security Privacy Wireless Mobile Netw.*, 2019, pp. 221–231.
- [2] (Ericsson, Stockholm, Sweden). *Security in 5G RAN and Core Deployments*. Accessed: Mar. 21, 2025. [Online]. Available: <https://www.ericsson.com/en/ran>
- [3] M. A. Imran, S. A. R. Zaidi, and M. Z. Shakir, *Access, Fronthaul and Backhaul Networks for 5G & Beyond*, Inst. Eng. Technol., Stevenage, U.K., 2017.
- [4] M. Bennis, M. Debbah, and H. V. Poor, "Ultrareliable and low-latency wireless communication: Tail, risk, and scale," *Proc. IEEE*, vol. 106, no. 10, pp. 1834–1853, Oct. 2018.
- [5] A. Garcia-Saavedra, X. Costa-Perez, D. J. Leith, and G. Iosifidis, "FluidRAN: Optimized vRAN/MEC orchestration," in *Proc. Conf. Comput. Commun.*, 2018, pp. 2366–2374.
- [6] E. Björnson, L. Sanguinetti, J. Hoydis, and M. Debbah, "Optimal design of energy-efficient multi-user MIMO systems: Is massive MIMO the answer?" *IEEE Trans. Wireless commun.*, vol. 14, no. 6, pp. 3059–3075, Jun. 2015.
- [7] (Huawei Technol., Shenzhen, China). *5G Unlocks a World of Opportunities*. 2019. Accessed: Mar. 21, 2025. [Online]. Available: <https://www.huawei.com/en/huaweitech/industry-insights/outlook/mobile-broadband/insights-reports/5g-unlocks-a-world-of-opportunities>
- [8] E. Oh, B. Krishnamachari, X. Liu, and Z. Niu, "Toward dynamic energy-efficient operation of cellular network infrastructure," *IEEE Commun. Mag.*, vol. 49, no. 6, pp. 56–61, Jun. 2011.
- [9] I. Ahmad, S. Shahabuddin, T. Kumar, J. Okwuibe, A. Gurtov, and M. Ylianttila, "Security for 5G and beyond," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3682–3722, 4th Quart., 2019.
- [10] *System Architecture for the 5G System (5GS)*, 3GPP Standard TS 123 501, 2018. Accessed: Jan. 24, 2025. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/15.02.00_60/ts_123501v150200p.pdf
- [11] "O-RAN: Towards an open and smart RAN," O-RAN Alliance, Alfter, Germany, White Paper, 2018. Accessed: Jan. 25, 2025. [Online]. Available: <https://mediastorage.o-ran.org/white-papers/O-RAN.White-Paper-2018-10.pdf>
- [12] (O-RAN Alliance, Alfter, Germany), *O-RAN Architecture Description*. 2021. Accessed: Feb. 2, 2025. [Online]. Available: <https://www.scribd.com/document/520855552/O-RAN-WG1-O-RAN-Architecture-Description-v04-00>
- [13] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A survey on 5G usage scenarios and traffic models," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 905–929, 2nd Quart., 2020.
- [14] Y.-H. Choi et al., "ML-based 5G traffic generation for practical simulations using open datasets," *IEEE Commun. Mag.*, vol. 61, no. 9, pp. 130–136, Sep. 2023.
- [15] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT press, Nov. 2018.
- [16] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, 2016, pp. 2094–2100. [Online]. Available: <https://doi.org/10.1609/aaai.v30i1.10295>
- [17] I. J. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.
- [18] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 3rd Quart., 2019.
- [19] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.
- [20] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, no. 1, pp. 321–357, 2002.
- [21] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. neural Netw.*, 2008, pp. 1322–1328.
- [22] R. Bellman, "Dynamic programming," *Science*, vol. 153, no. 3731, pp. 34–37, 1966.
- [23] M. Sugiyama, M. Krauledat, and K.-R. Müller, "Covariate shift adaptation by importance weighted cross validation," *J. Mach. Learn. Res.*, vol. 8, no. 5, pp. 985–1005, 2007.
- [24] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Trans. Inf. theory*, vol. 37, no. 1, pp. 145–151, Aug. 2002.
- [25] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 723–773, 2012.
- [26] M. Stamp, *Introduction to Machine Learning with Applications in Information Security*. Boca Raton, FL, USA: Chapman Hall/CRC, 2022.
- [27] L. Bilge and T. Dumitras, "Before we knew it: An empirical study of zero-day attacks in the real world," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2012, pp. 833–844.
- [28] C. Kruegel and G. Vigna, "Anomaly detection of web-based attacks," in *Proc. 10th ACM Conf. Comput. Commun. Secur.*, 2003, pp. 251–261.
- [29] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, pp. 39–53, 2004.
- [30] S. Staniford, J. A. Hoagland, and J. M. McAlerney, "Practical automated detection of stealthy portscans," *J. Comput. Secur.*, vol. 10, nos. 1–2, pp. 105–136, 2002.
- [31] E. M. Hutchins, M. J. Cloppert, and R. M. Amin, "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues Inf. Warfare Security Res.*, vol. 1, no. 1, p. 80, 2011.
- [32] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 1–58, 2009.
- [33] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.
- [34] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, "Learning intrusion detection: Supervised or unsupervised?" in *Proc. 13th Int. Conf. Image Anal. Process.*, 2005, pp. 50–57.
- [35] P. Popovski et al., "Wireless access in ultra-reliable low-latency communication (URLLC)," *IEEE Trans. Commun.*, vol. 67, no. 8, pp. 5783–5801, Aug. 2019.

- [36] X. Costa-Pérez, J. Swetina, T. Guo, R. Mahindra, and S. Rangarajan, "Radio access network virtualization for future mobile carrier networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 27–35, Jul. 2013.
- [37] A. Checko et al., "Cloud RAN for mobile networks—A technology overview," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 405–426, 1st Quart., 2014.
- [38] *Service Requirements for the 5G System*, 3GPP Standard TS 22.261, 2019.
- [39] P. Rost et al., "Cloud technologies for flexible 5G radio access networks," *IEEE Commun. Mag.*, vol. 52, no. 5, pp. 68–76, May 2014.
- [40] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ., 2005.
- [41] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Cambridge, U.K.: Cambridge Univ., 2024.
- [42] J. G. Andrews et al., "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.
- [43] J. Weng, X. Tu, Z. Lai, S. Salous, and J. Zhang, "Indoor massive MIMO channel Modelling using ray-launching simulation," *Int. J. Antennas Propag.*, vol. 2014, no. 1, 2014, Art. no. 279380.
- [44] M. Di Renzo et al., "Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2450–2525, Nov. 2020.
- [45] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ., 2004.
- [46] R. S. Sutton and A. G. Barto, *Reinforcement learning: An Introduction*, vol. 1. Cambridge, MA, USA: MIT Press, 1998.
- [47] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.



ABDULRAHMAN GHANDOURA (Member, IEEE) received the B.S. degree in electronics engineering from Johnson and Wales University, Providence, RI, USA, the M.S. degree in network engineering and management from DePaul University, Chicago, IL, USA, in 2013, and the Ph.D. degree in electrical and computer engineering from Southern Illinois University, Carbondale, IL, USA, in 2018. In 2019, he joined Umm-Alqura University, Makkah, Saudi Arabia, where he is currently an Assistant Professor with the

Department of Engineering and Science, Applied College. His research interests include network architecture and wireless sensor networks, 6G and beyond wireless communication technologies, and Internet of Things and Internet of Everything applications.



ABDULLAH ALAJMI (Member, IEEE) received the B.S. degree in information system technology from Southern Illinois University, Carbondale, IL, USA, in 2010, the M.S. degree in information technology from DePaul University, Chicago, IL, USA, in 2014, and the Ph.D. degree from the School of Electronic Engineering and Computer Science, Queen Mary University of London, London, U.K., in 2023. His research interests include artificial intelligence for wireless systems, Internet of Things, NOMA, and Internet of Everything.



Egypt, for two years, and with the Faculty of Engineering, MSA University, Egypt, for five years. His research interests include wireless sensor networks, computer network security, machine learning, and IoT. He earned the Explorer Award Badge from the IBM Skills Academy for the Artificial Intelligence Analyst 2020.



ABDELWAHEB MOTWAKEL received the B.Sc. degree in computer science from Omdurman Islamic University, the M.Sc. degree in computer science from Institut Teknologi Sepuluh Nopember, Indonesia, and the Ph.D. degree in computer science from the Sudan University of Science and Technology. He is an Assistant Professor of Computer Science with Prince Sattam bin Abdulaziz University, Saudi Arabia. His research interests include fuzzy logic, neural networks, machine learning, deep learning, and data science. He is actively involved in teaching, student supervision, and scientific publishing in these fields. He has collaborated on multidisciplinary projects and participated in academic committees related to curriculum development and research evaluation.



Computer Science, Taiz University, Taiz, Yemen. His research interests include optimization and swarm intelligence.

FADL DAHAN received the B.Sc. degree from Tamar University, Yemen, the M.Sc. degree from King Saud University, and the Ph.D. degree from the Department of Computer Science, King Saud University. He is an Associate Professor of Artificial Intelligence with the Department of Management Information System, College of Business Administration—Hawtat Bani Tamim, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia. He is also a Faculty Member with the Department of Computer Science, Faculty of



awarded a fellowship from the Higher Education Academy and a PGCert in Education in 2018 at the University of Greenwich. She is the Manager of the Center of Excellence for Projects and Entrepreneurship with MSA University and the Director of Technology Transfer Office. Her research interests include artificial intelligence, machine learning, cryptography, cybersecurity, IoT, and tracking systems. Recently, she earned the McKinsey Forward Program Badge from McKinsey & Company and the Mastery Award Badge from the IBM Skills Academy for the Artificial Intelligence Analyst Exam 2020.

GHADA ABDELHADY is an Associate Professor with the Faculty of Engineering, MSA University, with 24 years of experience in teaching Engineering and Computer Science courses, projects, and M.Sc. and Ph.D. supervision. She is with the Center of Excellence, Faculty of Engineering, October University for Modern Sciences and Arts, Egypt. She is a certified reviewer of the Quality Assurance and Accreditation of Egyptian Higher Education and a member of several research organizations. She was