Few Features are Enough: Communication-Efficient AI-RAN

Dayoung Choi
Electronic and Electrical Engineering
Ewha Womans University
Seoul 03760, South Korea
dayoung.choi@ewha.ac.kr

Siyoun Park
Electronic and Electrical Engineering
Ewha Womans University
Seoul 03760, South Korea
siyoun0116@ewha.ac.kr

Jungmin Kwon
Electrical and Electronics Engineering
Kangwon National University
Chuncheon 24341, South Korea
jungmin.kwon@kangwon.ac.kr

Hyunggon Park
Electronic and Electrical Engineering
Ewha Womans University
Seoul 03760, South Korea
hyunggon.park@ewha.ac.kr

Abstract

The deployment of Artificial Intelligence and Machine Learning (AI/ML) in AI-Radio Access Networks (AI-RAN) should balance the performance with communication load over the interface and model complexity. We propose a communication-efficient feature selection framework fully compliant with the O-RAN architecture. Key Performance Indicator (KPI) traffic in the non-RT RIC is analyzed to quantify the statistical relevance of each KPI to the task. A compact feature mask is then deployed to the Distributed Unit, enabling real-time filtering so that only the most informative KPIs are transmitted to the near-RT RIC for online inference. Using realistic O-RAN compliant datasets, we evaluate the proposed framework with classifiers for the slice classification use case. Results show that our framework sustains high performance while significantly reducing the number of KPIs, model parameters, and communication overhead, thereby demonstrating its suitability for scalable, low-latency AI-RAN deployments.

1 Introduction

The integration of artificial intelligence (AI) and machine learning (ML) into Radio Access Networks (RAN), often referred to as AI-RAN, has emerged as a pivotal direction to address the increasing complexity and dynamic nature of next-generation networks [Khan and Schmid, 2023]. The O-RAN (Open RAN) Alliance [O-RAN Alliance, 2018] has been instrumental in this transformation, introducing open, disaggregated RAN architectures that foster interoperability and innovation. The core of the O-RAN is the RAN Intelligent Controller (RIC), which hosts modular applications known as xApps operating in the near-real-time (near-RT) domain and rApps in the non-real-time (non-RT) domain to enable network intelligence orchestration [O-RAN Working Group 1, 2025]. However, as networks become increasingly heterogeneous and data-intensive, there arises a critical need for scalable and efficient AI/ML-based solutions that can leverage real-time network data analytics to drive intelligent decision-making within the RIC.

The practical deployment of intelligent control faces critical communication challenges. The increasing number of User Equipments (UEs) and the Key Performance Indicators (KPIs) introduce a high-dimensional data space that must be continuously observed and analyzed. These KPIs, such as

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI and ML for Next-Generation Wireless Communications and Networking (AI4NextG).

throughput, signal quality, resource utilization, etc., are often used to make RAN control decisions, necessitating frequent data exchanges between RIC and gNB (gNodeB) over an interface. This incurs the communication overheads, leading to congestion, increased latency, and potential degradation of control effectiveness. This bottleneck becomes particularly critical in near-RT control scenarios, where the timeliness and accuracy of data-driven decisions directly impact network performance. Consequently, it is essential to develop AI/ML-based analytics frameworks that can efficiently process high-dimensional KPIs while minimizing communication overhead, thereby enabling scalable and responsive intelligent control.

AI/ML approaches have demonstrated significant potential in improving network performance by enabling data-driven, adaptive control mechanisms that surpass traditional rule-based methods [Groen et al., 2024, Bonati et al., 2021a, Huang et al., 2023, Kasuluru et al., 2023]. However, many of these solutions adopt AI/ML models in a *black-box* manner, often without considering trade-offs between data structure complexity, model complexity, data exchange overhead, and real-time operational constraints inherent to RAN. As a result, while performance gains are achieved in isolated scenarios, the scalability and efficiency remain questionable when deployed across large-scale, heterogeneous networks. This gap underscores the necessity for AI/ML frameworks that are not only accurate but also communication-efficient, ensuring that the intelligence embedded within the RAN aligns with the stringent requirements of real-time control and optimization.

To address the challenges of scalability and efficiency in AI-RAN, we design a framework that utilizes feature selection. Our design leverages offline statistics of the network data in the non-RT RIC to identify a compact KPI set that exhibits both strong and explainable correlation with the task. This approach reduces the model's input dimensionality, resulting in a smaller parameter count, while also preserving interpretability by explicitly linking each retained feature to measurable statistical relevance. Hence, we demonstrate that a few, well-chosen KPIs are sufficient to sustain high classification performance while significantly lowering communication and computational overhead in an O-RAN deployment. The key contributions of this paper are summarized as follows. We propose a communication-efficient feature selection framework that complies with the O-RAN architecture, providing a specific AI/ML deployment path. We analyze the statistical correlation between each KPI feature and the target variable using realistic KPI network traffic, deriving interpretable insights in the context of model performance and network data analytics. We confirm that a few features are sufficient by evaluating model performance, model complexity, and communication overhead of the E2 and O1 interface.

2 Related Work

AI/ML can significantly improve RAN control and optimization in the O-RAN architecture, primarily through xApp and rApp design for intelligent decision-making. The data-driven closed-control loops integration has been implemented using Deep Reinforcement Learning (DRL) as xApps in the Colosseum, dynamically selecting scheduling policies for network slicing [Bonati et al., 2021a]. ColO-RAN [Polese et al., 2023], a framework for large-scale datasets in near-RT control, where a policy scheduler in xApps using DRL, Autoencoder, and Transformer has been proposed, demonstrating standard-compliant learning and orchestration capabilities in practice. TRACTOR [Groen et al., 2024] has built an O-RAN compliant KPI dataset and deployed a Convolutional Neural Network (CNN)-based classifier as xApps for slice classification in the Colosseum. The Colosseum is a large-scale, O-RAN compliant wireless network emulator that strictly adheres to O-RAN architectural specifications [Bonati et al., 2021b]. Note that these works primarily focus on optimizing network metrics rather than considering how network operations to be more efficient in terms of communication and deployment.

To reduce the dimensionality of network data while maintaining accuracy, feature selection has been widely employed, as it can identify a compact subset of input variables that retains the predictive power and eliminates irrelevance [Kira and Rendell, 1992]. In AI-RAN, fewer KPIs can reduce telemetry over the E2, lower inference complexity in xApps, and decrease model payloads over O1. In the 5G core network, a comparative analysis of various feature selection methods for anomaly detection has been demonstrated [Oliveira et al., 2025]. Similarly, an ML framework for Internet of Things traffic classification in 5G network slicing has been proposed [Min et al., 2023]. In O-RAN, an Explainable AI approach has been introduced using Shapley additive explanation values to rank KPIs by importance for slice classification, demonstrating the potential for lowering control-plane overhead

without critical performance degradation [Tassie et al., 2024]. However, most of these studies do not address the interpretability of KPIs and runtime deployment constraints such as near-RT inference latency or O1 model transfer costs.

3 Proposed Framework

3.1 System Architecture Overview

Data collection in the O-RAN begins at the Radio Unit (RU), which interfaces with multiple UEs via the air interface. The physical-layer radio signals are forwarded to the Distributed Unit (DU) over the Fronthaul. DU derives various KPIs, such as throughput, and physical resource block (PRB) usages, etc. These KPIs reflect the current status of wireless traffic and radio link conditions. KPI vectors are then transmitted to rApps in non-RT RIC through the O1, where they are aggregated and analyzed offline. Then, the trained models are deployed to xApps over the O1 from the non-RT RIC.

In order to design a communication-efficient system that exchanges only essential KPIs in a slice classification use case, the proposed framework comprises a classifier and a feature selection function. Figure 1 illustrates the data flow in the O-RAN and runtime deployment path of the selected KPIs. The feature selection identifies the most relevant KPIs that significantly contribute to network slice discrimination, thereby reducing the data dimensionality and enhancing the interpretability of the system. The trained classifier is deployed to xApps, and the selected feature mask from the feature selection function is installed at the DU. During online operation, the DU applies a feature mask to filter the full KPI vector and transmits only the selected KPIs to the xApps via the E2. The xApp then performs slice classification in near real-time using the incoming reduced-dimensional KPI vector.

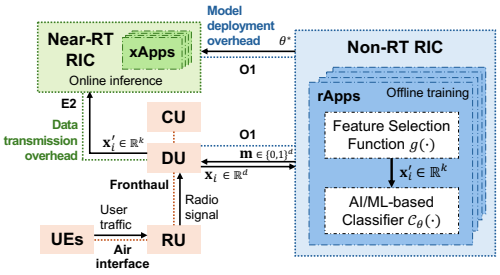


Figure 1: Data flow in the O-RAN architecture and runtime deployment path.

3.2 Reduced KPI Set

Let $\mathbf{x}_i \in \mathbb{R}^d$ $(i = 1, 2, \cdots, N)$ denote the i-th full KPI instance vector captured at the DU, where d is the total number of KPIs and N is the total number of instances. $\mathbf{f}_j \in \mathbb{R}^N$ $(j = 1, 2, \cdots, d)$ represent the j-th KPI feature vector across all instances. The goal is to construct a reduced KPI vector $\mathbf{x}'_i \in \mathbb{R}^k$ such that $k \leq d$, while preserving the classification performance. We define a feature selection function at rApps as a mapping, $g : \mathbb{R}^d \to \mathbb{R}^k$, where $g(\cdot)$ selects a subset of KPIs from $\mathbf{x}_i$.

As a feature selection criterion, we adopt the Pearson Correlation Coefficient (PCC) with the class label $\mathbf{y}$ in offline data. The PCC, a relevance score s_j, between $\mathbf{f}_j$ and $\mathbf{y}$, is computed as

$$s_j = \frac{\sum_{i=1}^{N}(f_{j,i} - \bar{f}_j)(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(f_{j,i} - \bar{f}_j)^2}\sqrt{\sum_{i=1}^{N}(y_i - \bar{y})^2}}, \tag{1}$$

where $\bar{f}_j$ and $\bar{y}$ denote the sample means of the $\mathbf{f}_j$ and $\mathbf{y}$, respectively. Since strong statistical dependence on the label is relevant to slice classification, the KPIs with higher values of s_j are considered to be assigned higher priority during the feature selection process.

To determine the set of KPIs to be transmitted to xApps based on s_j, we use the Kneedle algorithm [Satopaa et al., 2011] as a thresholding, to adaptively determine the number of selected KPIs k for each $\mathbf{f}_j$. The Kneedle algorithm identifies the elbow point τ in the sorted score curve in descending order, which corresponds to the maximum deviation from a reference line. Then, the binary selection mask $\mathbf{m} = [m_1, m_2, \cdots, m_d] \in \{0, 1\}^d$ is constructed as

$$m_j = \begin{cases} 1 & \text{if } s_j \geq \tau, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The resulting $\mathbf{m}$ is deployed to the DU, where it is applied at runtime to process incoming KPIs. Finally, DU constructs the reduced KPI $\mathbf{x}'_i$ by applying an masking operation $\mathbf{x}'_i = \mathbf{x}_i \odot \mathbf{m}$, where $\odot$

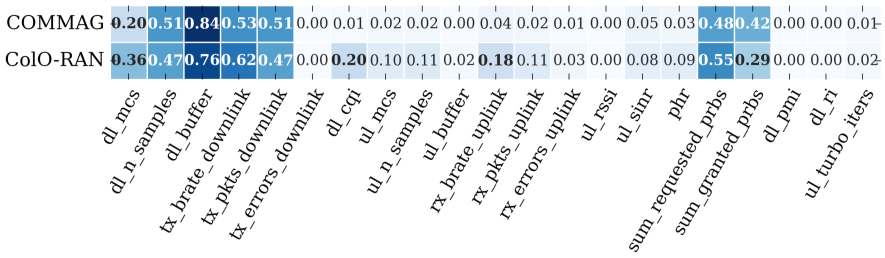


Figure 2: KPI feature ranking results.

denotes the Hadamard product. This allows only the most informative KPIs to be transmitted, thereby reducing communication overhead while preserving the classification capability.

3.3 Deployment in xApp and rApp with Reduced KPI

In the O-RAN specification, rApps are responsible for offline training and evaluation of candidate AI/ML models. Once trained, rApp determines which model should be deployed to a xApp [D'Oro et al., 2022]. The slice classifier $\mathcal{C}_{\theta^*}(\cdot)$ with optimal parameters θ^* maps the $\mathbf{x}'_i$ to y_i, i.e., $y_i = \mathcal{C}_{\theta^*}(\mathbf{x}'_i)$, where

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \ \mathcal{L}_{\text{CE}}(\theta; \{\mathbf{x}'_i, y_i\}), \tag{3}$$

with the cross-entropy $\mathcal{L}_{\text{CE}}$ over the reduced feature space. In our framework, once θ^* is obtained, the resulting model is not only assessed in terms of classification performance but also benchmarked for feature dimensionality, computational cost, and deployment efficiency. We adopt both classical ML and deep learning (DL) models, such as eXtreme Gradient Boosting (XGBoost) [Chen and Guestrin, 2016], Support Vector Machine (SVM) [Cortes and Vapnik, 1995], CNN [Fukushima, 1980], Gated Recurrent Unit (GRU) [Cho et al., 2014], and Transformer [Vaswani et al., 2017].

4 Experiments

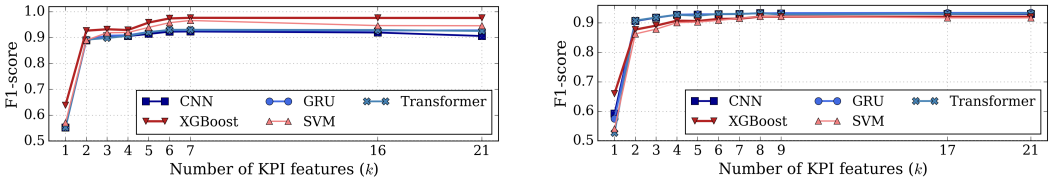
4.1 Network Traffic Datasets and Slice Classifiers

We use two publicly available datasets, Colosseum O-RAN COMMAG Dataset (COMMAG) [Bonati et al., 2021a] and Colosseum ColO-RAN Dataset (ColO-RAN) [Polese et al., 2023], both collected from the Colosseum. Both datasets include three types of network slices, enhanced Mobile Broadband (eMBB), Ultra Reliable Low Latency Communications (URLLC), and massive Machine Type Communications (mMTC), generated in a realistic 5G over-the-air environment. The captured KPIs correspond to $d = 21$.

The COMMAG and the ColO-RAN datasets contain a total of 68,163 and 78,702 data instances, respectively. Both are split into 60% training, 15% validation, and 25% testing sets. Since feature selection is performed offline, only the training data is used. XGBoost and SVM are trained using default parameters, with the radial basis function kernel for SVM. GRU and Transformer consist of 3 layers with learning rates of 0.001 and 0.0001, respectively. CNN contains one layer with a learning rate of 0.001. For GRU and Transformer, KPIs are processed as time-series data using a sliding window of length 10, with positional encoding in the Transformer. All DLs are trained with the Adam optimizer and with a stride of 1 for each instance in online inference. A learning rate scheduler reduces the rate by a factor of 10 upon a validation-loss plateau, acting as early stopping.

4.2 Interpretability of Selected KPIs

Figure 2 shows the relevance score s_j for all KPI features. In the heatmap, boldface values denote the selected KPIs, yielding 7 for COMMAG and 9 for ColO-RAN. Interestingly, despite being collected under different configurations, the top-7 KPIs are identical across both datasets, suggesting that these KPIs consistently show the strongest correlation with slice labels. Note that these are associated with downlink throughput and PRB allocation, while the rest exhibit near-zero s_j, indicating sparse



(a) COMMAG dataset (b) ColO-RAN dataset

Figure 3: Classification performance variation to the number of selected KPIs (k).

correlation to the classification. This suggests that the discriminative KPIs are highly stable and configuration-invariant, which is a desirable property for robust AI-RAN deployments.

The dominance of downlink throughput and PRB allocation aligns with the fact that 5G/6G service types, eMBB, URLLC, and mMTC, differ significantly in their downlink requirements. PRB-related KPIs capture how the scheduler allocates radio resources to UEs and slices. Since PRB scheduling policies are tightly coupled with slice-specific Quality of Service guarantees, these KPIs exhibit high discriminative power. The stronger relevance of downlink compared to uplink can be attributed to typical traffic asymmetry in mobile networks. Most slice services are downlink-heavy, with the scheduler adapting downlink PRB assignments more aggressively to meet slice service level agreements, while uplink allocation tends to be less differentiated across slices [ElBamby et al., 2014].

4.3 Slice Classification Performance with Reduced KPI

Figure 3 shows the macro-averaged F1-score with k. Across all models, performance remains largely stable from the full set of 21 KPIs to the reduced set with zero relevance KPIs removed, and further to the selected set. This stability confirms that KPIs with negligible relevance scores contribute little to the classification decision boundary. Interestingly, when the k exceeds that of the selected set, certain models (e.g., SVM and CNN in COMMAG, and SVM in ColO-RAN) exhibit a slight performance drop, suggesting that the inclusion of low-relevance KPIs may introduce noise. When starting from the selected set and progressively removing KPIs one at a time, performance decreases gradually but at $k = 2$, remains above 0.89 in COMMAG, and above 0.86 in ColO-RAN for all models. This degradation reflects a trade-off that, while reducing the KPIs can further lower computational and communication costs, it may also reduce inter-class separability for certain slice types.

ML and DL models exhibit difference across datasets. In the COMMAG, MLs outperform DLs, whereas in the ColO-RAN, MLs perform slightly below but remain comparable to DLs. The principal component analysis can explain this trend. In COMMAG, the variance ratio reaches 0.98 with only 7 KPIs, indicating a largely linear structure that favors tree and kernel-based methods. In contrast, in ColO-RAN, the 9-KPI set yields a lower variance ratio of 0.82, implying more complex boundaries where DL gains a small edge while ML remains competitive.

5 Deployment Efficiency

5.1 Inference Latency in xApp

To quantitatively assess inference latency, we evaluate the time complexity of each trained classifier. The asymptotic inference time complexity of an ensemble tree, such as XGBoost, is given by $\mathcal{O}(T \cdot D)$, where T is the number of trees and D is the maximum tree depth [Chen and Guestrin, 2016]. For SVM, the time complexity can be expressed as $\mathcal{O}(n_{SV} \cdot k)$, where n_{SV} and k denote the number of support vectors and KPIs, respectively [Maji et al., 2013]. For the quantitative comparison, we present $T \cdot D$ and $n_{SV} \cdot k$. For DL models, the inference latency is quantified by the floating-point operations (FLOPs) per forward pass, representing the arithmetic operations (multiplications and additions) required to generate one prediction.

		XGBoost	SVM
COMMAG	baseline	11.29K	254.86K
	$k = 16$	11.29K	191.98K
	$k = 7$	5.72K	52.14K
	$k = 2$	**5.16K**	**27.40K**
ColO-RAN	baseline	6.17K	208.257K
	$k = 17$	6.17K	168.60K
	$k = 9$	6.15K	78.17K
	$k = 7$	**2.07K**	59.64K
	$k = 2$	4.83K	**41.65K**

Table 1: Time complexity of MLs.

Tables 1 and 2 show the inference complexity of classifiers. Note that DL models have the same structure regardless of datasets. $k = 7$ yields substantial latency gains for ML models, with XGBoost and SVM achieving reductions, compared to that of the full KPIs ($k = 21$), of up to 66.47% in ColO-RAN and 79.54% in COMMAG, respectively. CNN also benefits greatly, up to 66.53% FLOPs reduction due to its dependence on input channels, while GRU and Transformer show under 3% change because their computation is dominated by hidden-state and attention operations. Importantly, even when the KPIs are reduced to $k = 2$, MLs maintain performance within 5% of the full KPIs, and CNN with 2%. This means that xApps can operate with far lower inference complexity without sacrificing much performance, enabling faster responses and more reliable decision-making in latency-constrained environments.

	CNN	GRU	Transformer
baseline	1.53M	2.56M	17.24M
$k = 16$	1.17M	2.54M	17.23M
$k = 9$	0.66M	2.52M	17.22M
$k = 7$	0.51M	2.51M	17.22M
$k = 2$	**0.15M**	**2.49M**	**17.21M**

Table 2: FLOPs of DLs.

5.2 Model Deployment Overhead in O1 Interface

When deploying a trained model over the O1, the data containing learned model parameters and metadata is transmitted. Table 3 shows the data size reduction of trained models for different k, compared to that of the full KPIs ($k = 21$). XGBoost and SVM benefit significantly with file size reductions up to 74.00% in Colo-RAN and 72.27% in COMMAG, respectively. CNN shows the largest gain among neural networks up to 90.30% in both datasets, while GRU and Transformer remain largely unaffected. Considering that these size reductions are achieved with only minor classification performance loss, the proposed framework allows models to be distributed over the O1 with much smaller payloads while retaining most of their computing power. This not only shortens deployment time but also reduces control-plane bandwidth usage, allowing the O1 interface to handle policy and enrichment information more efficiently and supporting more frequent or large-scale model updates without overloading the network.

		XGBoost	SVM	CNN	GRU	Transformer
COMMAG	baseline	3.71MB	2.36MB	5.75MB	0.98MB	15.69MB
	$k = 16$	0.002%	20.50%	23.76%	0.76%	0.008%
	$k = 7$	35.18%	**72.27%**	66.53%	2.09%	0.023%
	$k = 2$	**41.07%**	71.17%	**90.30%**	**72.83%**	**0.030%**
ColO-RAN	baseline	1.99MB	1.93MB	5.75MB	0.98MB	15.69MB
	$k = 17$	0.003%	15.67%	19.01%	0.600%	0.006%
	$k = 9$	1.23%	53.60%	57.03%	1.79%	0.020%
	$k = 7$	**74.00%**	**61.20%**	61.79%	2.09%	0.023%
	$k = 2$	34.82%	46.43%	**90.30%**	**2.84%**	**0.030%**

Table 3: Trained model file size reduction (%). Reduction indicates the relative decrease. (Larger values correspond to greater efficiency.)

5.3 Data Transmission Overhead in E2 Interface

To quantify the overhead in data transmission, we calculate the telemetry data volume transmitted from the DU to the xApp via the E2 interface. Since KPI values are encoded in variable length ASCII format in the Colosseum testbed, we compute the actual byte size of each KPI in test data and sum them over the selected KPIs to obtain the per-second data rate in Kbps.

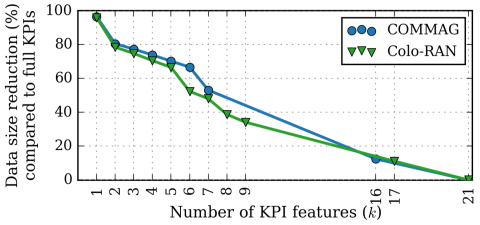


Figure 4: Data size (Kbps) reduction (%).

Figure 4 shows the telemetry data reduction compared to the full KPIs ($k = 21$). Using only $k = 2$ KPIs reduces the data size by 80.74% in COMMAG and 78.43% in ColO-RAN, with at most a 5% drop in classification performance across all models. These drastic reductions directly alleviate the E2 interface overhead by lowering bandwidth usage and reducing the processing required for transmitting and parsing KPI messages.

6 Conclusion

In this paper, we propose a communication-efficient feature selection framework for practical AI/ML deployment, which is fully compliant with the O-RAN architecture. A small, well-chosen subset of KPI features is enough to maintain high slice classification performance in AI-RAN, while achieving significant improvements in communication and deployment efficiency and preserving interpretability by directly linking selected KPIs to their statistical relevance. By selecting only the most informative KPIs in rApps and filtering them at DU, our framework reduces E2 telemetry traffic, decreases online inference complexity in xApps, and minimizes O1 overhead without noticeable performance degradation. Even with only two carefully selected KPIs, lightweight models such as XGBoost, SVM, and CNN sustained performance within 2-5% of the full KPI set, yet achieved harsh reductions in inference complexity and trained model file size. These results suggest that a few features and lightweight models can more effectively meet near-RT latency constraints, enable efficient resource utilization, and satisfy scalability requirements, offering a standards-compliant path for practical AI/ML deployment in AI-RAN.

References

Leonardo Bonati, Salvatore D'Oro, Michele Polese, Stefano Basagni, and Tommaso Melodia. Intelligence and Learning in O-RAN for Data-driven NextG Cellular Networks. *IEEE Communications Magazine*, 59(10):21–27, October 2021a.

Leonardo Bonati, Pedram Johari, Michele Polese, Salvatore D'Oro, Subhramoy Mohanti, Miead Tehrani-Moayyed, Davide Villa, Shweta Shrivastava, Chinenye Tassie, Kurt Yoder, Ajeet Bagga, Paresh Patel, Ventz Petkov, Michael Seltser, Francesco Restuccia, Abhimanyu Gosain, Kaushik R. Chowdhury, Stefano Basagni, and Tommaso Melodia. Colosseum: Large-Scale Wireless Experimentation Through Hardware-in-the-Loop Network Emulation. In *Proceedings of 2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 105–113, Los Angeles, CA, USA, 2021b. IEEE.

Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 785–794, New York, NY, USA, 2016. Association for Computing Machinery (ACM).

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar, 2014. Association for Computational Linguistics (ACL).

Corinna Cortes and Vladimir Vapnik. Support–Vector Networks. *Machine Learning*, 20(3):273–297, 1995.

Salvatore D'Oro, Leonardo Bonati, Michele Polese, and Tommaso Melodia. OrchestRAN: Network Automation through Orchestrated Intelligence in the Open RAN. In *Proceedings of IEEE INFOCOM 2022–IEEE Conference on Computer Communications*, pages 270–279, London, UK, 2022. IEEE Press.

Mohammed S ElBamby, Mehdi Bennis, Walid Saad, and Matti Latva-Aho. Dynamic Uplink-Downlink Optimization in TDD-based Small Cell Networks. In *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*, pages 939–944. IEEE, 2014.

Kunihiko Fukushima. Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*, 36(4):193–202, 1980.

Joshua Groen, Mauro Belgiovine, Utku Demir, Brian Kim, and Kaushik Chowdhury. TRACTOR: Traffic Analysis and Classification Tool for Open RAN. In *Proceedings of ICC 2024–IEEE International Conference on Communications*, pages 4894–4899, Denver, CO, USA, 2024. IEEE.

Jun-Hong Huang, Shin-Ming Cheng, Rafael Kaliski, and Cheng-Feng Hung. Developing xApps for Rogue Base Station Detection in SDR-Enabled O-RAN. In *Proceedings of IEEE INFOCOM 2023–IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 1–6, Hoboken, NJ, USA, 2023. IEEE.

Vaishnavi Kasuluru, Luis Blanco, and Engin Zeydan. On the use of Probabilistic Forecasting for Network Analysis in Open RAN. In *Proceedings of 2023 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, pages 258–263, Dubrovnik, Croatia, 2023. IEEE.

Naveed Ali Khan and Stefan Schmid. AI-RAN in 6G Networks: State-of-the-Art and Challenges. *IEEE Open Journal of the Communications Society*, 5:294–311, 2023.

Kenji Kira and Larry A Rendell. The Feature Selection Problem: Traditional Methods and a New Algorithm. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 129–134, San Jose, California, 1992. AAAI Press.

Subhransu Maji, Alexander C Berg, and Jitendra Malik. Efficient Classification for Additive Kernel SVMs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):66–77, 2013.

Ziran Min, Swapna Gokhale, Shashank Shekhar, Charif Mahmoudi, Zhuangwei Kang, Yogesh Barve, and Aniruddha Gokhale. A Classification Framework for IoT Network Traffic Data for Provisioning 5G Network Slices in Smart Computing Applications. In *Proceedings of 2023 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 133–140, Nashville, TN, USA, 2023. IEEE.

O-RAN Alliance. Towards an Open and Smart RAN. White paper, O-RAN Alliance, Alfter, Germany, oct 2018. URL `https://www.o-ran.org/resources`.

O-RAN Working Group 1. O-RAN Architecture Description 14.0. Technical Specification O-RAN.WG1.OAD-R004-v14.00, O-RAN Alliance, Alfter, Germany, jun 2025. URL `https://www.o-ran.org/specifications`.

Júnia Maísa Oliveira, César Morais, Daniel Macedo, and José Marcos Nogueira. A Comparative Analysis of Feature Selection and Machine Learning Algorithms for Enhanced Anomaly Detection in 5G Core Networks. In *Proceedings of 2025 Global Information Infrastructure and Networking Symposium (GIIS)*, pages 1–6, Dubai, UAE, 2025. IEEE.

Michele Polese, Leonardo Bonati, Salvatore D'Oro, Stefano Basagni, and Tommaso Melodia. ColO-RAN: Developing Machine Learning-Based xApps for Open RAN Closed-Loop Control on Programmable Experimental Platforms. *IEEE Transactions on Mobile Computing*, 22(10):5787–5800, 2023.

Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior. In *Proceedings of 2011 31st International Conference on Distributed Computing Systems Workshops*, pages 166–171, Minneapolis, MN, USA, 2011. IEEE.

Chinenye Tassie, Brian Kim, Joshua Groen, Mauro Belgiovine, and Kaushik R Chowdhury. Leveraging Explainable AI for Reducing Queries of Performance Indicators in Open RAN. In *Proceedings of ICC 2024–IEEE International Conference on Communications*, pages 5413–5418, Denver, CO, USA, 2024. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*, pages 6000–6010, 2017.