Benchmarking Deep Learning Architectures for Real-Time Intrusion Detection in Kubernetes-Orchestrated 5G Core Networks

Vincent Abella[1], Jhury Kevin Lastre[1], I Wayan Adi Juliawan Pawana[1,2], Donghoon Lee[1], Bonam Kim[1], Ilsun You[1]

[1]Kookmin University, Seoul, South Korea

[2]Udayana University, Badung, Indonesia

Received: November 29, 2025; Revised: January 07, 2026; Accepted: January 27, 2026; Published: February 05, 2026

Abstract

Cloud-native 5G core networks on Service-Based Architecture expose distributed Network Functions to cyber threats requiring adaptive Deep Learning-based Intrusion Detection Systems (DL-IDS). This work evaluates six DL architectures (Convolutional Neural Network (CNN), Multi-Layer Perceptron (MLP), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Autoencoder (AE)) on a Kubernetes-orchestrated Open5GS testbed, measuring Central Processing Unit (CPU) utilization, memory consumption, and latency under realistic traffic conditions. Results show feedforward models (CNN, MLP, AE) achieve sub-millisecond latency (0.6 milliseconds (ms)) with CPU below 12%, enabling multiple concurrent IDS instances per server, while recurrent architectures (RNN, LSTM, GRU) require high CPU utilization (99-107%) with 3.5 to 7.2 ms latency, necessitating dedicated hardware acceleration. Memory footprint remains consistent (385 to 390 megabytes (MB)) across all models. These findings demonstrate that operational efficiency is a key consideration for DL-IDS deployment in production 5G networks, with substantial CPU efficiency differences between architecture choices.

Keywords: 5G Core Network, Deep Learning, Intrusion Detection System, Open5GS, Kubernetes.

1 Introduction

Fifth Generation (5G) core networks implement Service-Based Architecture (SBA) with distributed Network Functions (NFs) deployed on container orchestration platforms [1], [2]. This cloud-native design increases the attack surface compared to traditional telecommunications infrastructure, with threats including signaling storms [3], protocol exploitation, lateral movement between NFs, and data leakage [4]. The security implications extend to critical applications such as Internet of Medical Things (IoMT) devices [5], [6] and vehicle-to-everything communications [7], where network-based intrusion detection becomes essential. Deep Learning (DL)-based Intrusion Detection Systems (IDS) have emerged as effective countermeasures, offering advanced pattern recognition capabilities for network traffic classification [8], [9], [10], complemented by emerging approaches in quantum-resistant authentication [11] and lightweight anomaly detection [12].

While recent research has demonstrated high detection accuracy for DL-IDS in 5G environments [10], [13], operational deployment remains challenging due to limited understanding of computational resource requirements. Network operators must balance detection effectiveness against infrastructure constraints when deploying IDS across critical interfaces such as the Security Edge Protection Proxy (SEPP) for inter-operator roaming communications. Prior work on embedded IDS platforms [14] highlights trade-offs between model complexity and deployment constraints, yet systematic benchmarking under production-like 5G core network

conditions remains scarce.

This work extends our prior research [13], which established detection accuracy for DL-IDS in 5G roaming scenarios, by providing comprehensive operational performance evaluation essential for production deployment. While the prior work validated detection effectiveness across attack classes, this study addresses the critical gap in understanding computational resource requirements, inference latency characteristics, and scalability implications that telecommunications operators need for infrastructure planning. We evaluate six representative DL architectures under production-like conditions using an Open5GS-based cloud-native testbed deployed on Kubernetes, measuring CPU utilization, memory consumption, and end-to-end latency.

Our analysis reveals substantial performance differences between feedforward and recurrent architectures, with direct implications for deployment feasibility across diverse 5G core network locations. The main contributions of this paper are as follows:

- Comprehensive operational benchmarking: We provide systematic performance analysis of six DL architectures (CNN, MLP, RNN, LSTM, GRU, AE) measuring CPU utilization, memory consumption, and inference latency under realistic 5G roaming traffic conditions on a Kubernetes-orchestrated testbed.
- Quantitative deployment guidelines: We demonstrate that feedforward models achieve sub-millisecond latency (0.6 ms) with CPU utilization below 12%, enabling 9-11 concurrent IDS instances per server, while recurrent architectures require 99-107% CPU with 3.5-7.2 ms latency, establishing concrete metrics for infrastructure planning.
- We provide actionable guidance for DL-IDS placement across 5G core network locations, identifying feedforward models for inline deployment at SEPP and NF interfaces, and recurrent models for offline forensic analysis.

2 Background

This section provides foundational context on 5G core network architecture, intrusion detection approaches, and deep learning architectures relevant to this study.

2.1. 5G Service-Based Architecture

The 3rd Generation Partnership Project (3GPP) specifies 5G core networks using Service-Based Architecture, where Network Functions communicate through standardized HTTP/2-based interfaces [1]. Key NFs include the Access and Mobility Management Function (AMF) for UE registration, Session Management Function (SMF) for session establishment, and User Plane Function (UPF) for data forwarding. For roaming scenarios, the SEPP serves as a security gateway between visited and home networks, handling inter-operator signaling over the N32 interface with mandatory TLS protection [15]. This microservices-based design enables flexible deployment on container orchestration platforms [2], but simultaneously expands the attack surface requiring comprehensive security monitoring.

2.2. Intrusion Detection in Telecommunications Networks

IDS for telecommunications networks have evolved from signature-based approaches to machine learning-driven systems capable of detecting zero-day attacks [10]. Network-based IDS (NIDS) analyze traffic flows to identify malicious patterns, while host-based systems monitor individual NF behavior. Recent surveys identify deep learning as particularly effective for 5G environments due to its ability to learn complex traffic patterns without manual feature engineering [10]. However, production deployment requires balancing detection accuracy against operational constraints including latency, throughput, and resource consumption [14]. Flow-based analysis using statistical features extracted from packet captures has emerged as the predominant

approach, enabling efficient processing while preserving detection capability.

2.3. Deep Learning Architectures for Network Traffic Classification

Deep learning architectures for IDS fall into two primary categories based on their computational structure. Feedforward architectures process input data in a single forward pass without maintaining internal state. CNNs apply convolutional filters to extract spatial patterns from flow features, while MLPs use fully connected layers for non-linear classification. AEs learn compressed representations of normal traffic, enabling anomaly detection by measuring reconstruction error for potentially malicious flows [13].

Recurrent architectures maintain hidden states that capture temporal dependencies across sequential inputs. Standard RNNs suffer from vanishing gradients during training, addressed by LSTM networks through gating mechanisms that regulate information flow [9]. GRUs provide a simplified alternative with fewer parameters while maintaining comparable modeling capability. These temporal models excel at detecting attack patterns that span multiple network flows but incur computational overhead from sequential state updates. Understanding these architectural differences is essential for selecting appropriate models based on deployment requirements and available computational resources.

3 System Architecture and Testbed

To evaluate DL-IDS operational performance, we developed a cloud-native testbed that replicates production 5G roaming scenarios.

3.1. Testbed and System Architecture

The evaluation focuses on the SEPP within the 5G Core network. SEPP is responsible for inter-operator communication through the N32-f interface, secured by Transport Layer Security (TLS) 1.3 [15]. This deployment mirrors real roaming environments where SEPP protects signaling between operators.

The IDS monitors roaming traffic exchanged across the N32-f link and operates as a separate pod in the Kubernetes cluster. This ensures controlled resource allocation and isolation from Open5GS network functions. Each pod is configured with defined resource quotas (1 virtual CPU (vCPU), 1 GB Random Access Memory (RAM)) for reproducibility and fairness across model evaluations.

3.2. Cloud-Native Testbed Configuration

The testbed infrastructure provides a realistic evaluation environment for IDS performance assessment. Figure 1 illustrates our cloud-native 5G core architecture. The testbed consists of containerized Open5GS network functions orchestrated on MicroK8s v1.28 running Ubuntu 22.04 Long Term Support (LTS) with 4 vCPUs and 16 GB of RAM. Calico Container Network Interface (CNI) and CoreDNS manage service discovery and networking. Traffic generation is handled by PacketRusher, which simulates realistic N32 roaming sessions. Feature extraction uses CICFlowMeter to transform packet captures into flow-based statistical features suitable for machine learning inference. DL models are deployed as separate pods built using PyTorch, co-located with Open5GS services to reproduce production-like latency conditions.

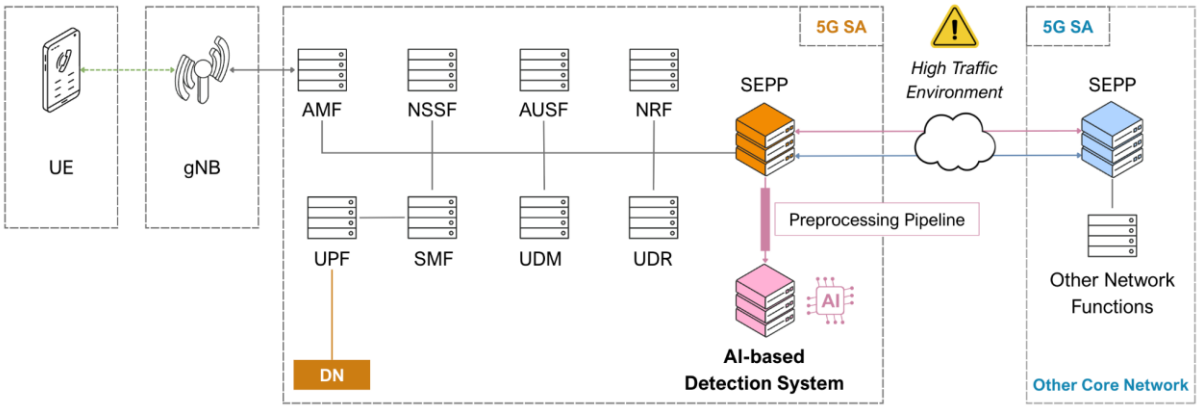

Figure 1. 5G Core Scenario for Testbed (Global Roaming Scenario)

3.3. Dataset and Traffic Composition

To ensure realistic evaluation, the dataset is derived from our prior work on 5G roaming intrusion detection and includes both benign and malicious N32 TLS traffic. Three representative attack classes are incorporated:

- Heartbleed Exploit – an application-layer TLS vulnerability producing subtle anomalies in handshake payloads, difficult to detect due to low variance.
- Denial-of-Service (DoS) – high-volume flooding of SEPP interfaces causing packet bursts and resource exhaustion, easily identifiable through rate-based features.
- Network Probing – irregular session initiation and incomplete handshakes, creating sporadic flows that require temporal correlation to detect.

Each class reflects a different detection difficulty level, ranging from simple volume anomalies to fine-grained protocol misuse. The complete dataset composition, feature extraction methodology, and preprocessing procedures are detailed in our prior work [13]. zFlows were balanced through random undersampling and normalized to values between 0 and 1. This structure enables fair assessment of operational efficiency independent of detection bias.

3.4. Evaluated Deep Learning Architectures

Our evaluation includes six representative DL architectures covering both feedforward and recurrent paradigms, with model configurations adopted from our prior work [13]. The feedforward category comprises CNN, MLP, and AE, while the recurrent category includes RNN, LSTM, and GRU. The architectural characteristics and computational trade-offs of these model families are detailed in Section 3.3.

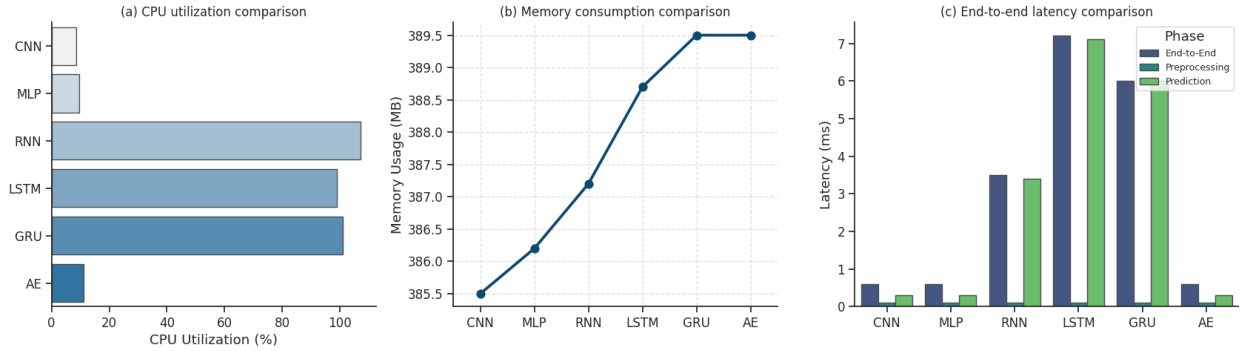

Figure 2. Performance comparison: (a) CPU utilization showing 10-fold efficiency difference, (b) Memory consumption demonstrating consistent footprint, and (c) Latency breakdown revealing prediction phase dominance in recurrent models.

4 Results and Analysis

This section presents the operational performance evaluation results, demonstrating how different DL architectures behave within Kubernetes-orchestrated cloud environments when processing 5G network traffic.

4.1. Detection Performance Analysis

All models achieve accuracy exceeding 99%, with MLP reaching 99.65% and LSTM achieving 99.55%, consistent with the detection results reported in our prior work [13]. This accuracy parity across feedforward and recurrent architectures confirms that detection capability is not the differentiating factor between model families [10]. Consequently, operational efficiency becomes the critical consideration for production deployment decisions in cloud-native 5G environments [14].

4.2. Operational Performance Analysis

Table 1 and Figure 2 present the operational metrics collected from the Kubernetes-orchestrated testbed, revealing substantial efficiency differences across architectures.

Table 1. Performance Comparison of ML Models for 5G Core IDS

Model	CPU (%)	Memory (MB)	Latency (ms)
CNN	8.5	385.5	0.6
MLP	9.7	386.2	0.6
RNN	107.1	387.2	3.5
LSTM	99.0	388.7	7.2
GRU	101.0	389.5	6.0
AE	11.2	389.5	0.6

CPU Utilization in Containerized Environments

Feedforward models (CNN, MLP, AE) achieve 8.5 to 11.2% CPU utilization within the 1 vCPU pod allocation, demonstrating efficient resource consumption compatible with Kubernetes resource quotas [2]. This

low utilization enables horizontal scaling through pod replication, allowing approximately 9 to 11 concurrent IDS instances per physical server without resource contention. Such resource efficiency is critical for cloud-native 5G deployments where multiple security functions must coexist on shared infrastructure [10]. In contrast, recurrent architectures (RNN, LSTM, GRU) consume 99-107% CPU, saturating the allocated vCPU and preventing effective horizontal scaling. The sequential nature of recurrent computation creates a bottleneck that cannot be mitigated through standard Kubernetes scaling strategies [9].

Memory Footprint Consistency

Memory consumption remains consistent (385--390 MB) across all architectures, indicating that the PyTorch runtime and model weights dominate memory usage rather than architectural differences [16]. This uniformity simplifies Kubernetes resource planning, as operators can apply consistent memory limits regardless of model selection [2]. The stable memory profile also indicates predictable behavior under sustained inference loads, which is essential for production containerized deployments [14].

Inference Latency Implications

Feedforward models achieve sub-millisecond latency (0.6 ms), well within acceptable thresholds for inline traffic inspection without introducing perceptible delays to N32 signaling [15]. Security functions in 5G core networks must maintain low latency to preserve Quality of Service (QoS) guarantees for signaling traffic [10]. This latency profile supports real-time threat detection at wire speed. Recurrent models exhibit 3.5 to 7.2 ms latency due to sequential hidden state computations [9], making them unsuitable for latency-sensitive inline deployment but appropriate for batch processing of captured traffic logs [14].

Cloud-Native Deployment Implications

The operational characteristics directly impact deployment architecture in cloud-native 5G networks [2]. Feedforward models enable inline deployment at any 5G interface without Quality of Service (QoS) degradation, supporting sidecar container patterns where the IDS pod runs alongside NF pods [1]. This deployment model aligns with microservices principles for distributed security functions [10]. Recurrent architectures require dedicated deployment as batch processing services, consuming traffic captures from persistent storage rather than intercepting live flows, which is consistent with offline forensic analysis patterns [14].

5 Discussion

The operational performance results provide critical insights for deploying DL-IDS in cloud-native 5G environments. This section discusses practical deployment strategies, infrastructure planning considerations, and comparison with existing approaches.

5.1. Cloud-Native Deployment Strategies

The performance characteristics of each architecture family directly determine viable deployment patterns within Kubernetes-orchestrated 5G core networks.

Inline Deployment with Feedforward Models

For real-time traffic inspection at SEPP interfaces, CNN or MLP architectures are recommended due to their sub-millisecond latency and low CPU overhead [13]. These models can be deployed as sidecar containers alongside NF pods or as dedicated DaemonSet deployments ensuring IDS presence on each cluster node [2]. The 8-12% CPU utilization permits generous resource headroom for traffic spikes, supporting reliable operation in dynamic network environments [1]. Internal NF-to-NF monitoring benefits from AE models,

where unsupervised learning detects novel lateral movement patterns without requiring labeled attack data for each new threat variant [10]. For base station (gNB) access points, MLP's 0.6 ms prediction time suits high-velocity UE connection monitoring, complementing machine learning-based false base station detection approaches [17].

Batch Processing with Recurrent Models

Recurrent architectures (LSTM, GRU) are suited for offline forensic analysis deployed as Kubernetes CronJobs or batch processing pipelines [2]. These models process traffic captures stored in persistent volumes, leveraging temporal modeling to identify attack sequences spanning multiple flows [9]. While unsuitable for inline deployment due to latency constraints, their sequential analysis capability provides value for threat hunting and incident investigation workflows [14].

Infrastructure Cost Implications

The approximately 10x CPU utilization difference between architecture families (feedforward: 8.5-11.2% vs. recurrent: 99-107%) has direct cost implications for telecommunications infrastructure planning [14]. Feedforward models enable deployment of 9-11 IDS instances per server, reducing infrastructure requirements for operators seeking comprehensive coverage across multiple network interfaces [10]. This resource efficiency is critical in cloud-native 5G environments where operational costs must be carefully managed [2]. Recurrent models require dedicated compute resources, increasing operational expenditure for equivalent detection coverage and potentially necessitating specialized hardware acceleration [16].

Comparison with Related Work

Table 2 compares our operational metrics with related IDS research. Recent work has optimized inference latency for edge and IoT platforms: DNN-KDQ [18] achieves 0.07 ms through quantization, LSTM-CNN [19] reports 2.3 ms for IoT, and federated CNN [20] measures 1.4 ms with CPU monitoring on embedded devices. Hybrid architectures [21] and mixture of experts [22] demonstrate high accuracy but lack operational benchmarking. Embedded evaluations [14] characterize energy-latency trade-offs on edge hardware. However, none provide systematic CPU utilization metrics under Kubernetes resource quotas essential for cloud-native capacity planning [10]. Our work addresses this gap with container-specific benchmarks enabling horizontal scaling calculations for 5G deployments [2].

Table 2. Performance comparison with related work

Study	Model	Latency	CPU	Deploy
[21]	Attn-CNN-LSTM	32 ms	N/R	Real-time
[14]	CNN/LSTM	Varies	Varies	Edge
[18]	DNN-KDQ	0.07 ms	N/R	Edge
[19]	LSTM-CNN	2.3 ms	N/R	**IoT**
[20]	CNN	1.4 ms	Measured	**IoT**
Ours	CNN/MLP/AE	**0.6 ms**	**8–12%**	**K8s Inline**
	RNN/LSTM/GRU	**3.5–7.2 ms**	**99–107%**	**K8s Offline**

5.2. Hardware Considerations and Limitations

This study used CPU-only inference to emulate realistic operator deployments where Graphics Processing Unit (GPU) resources may be limited at edge and mid-tier infrastructure [14]. Recent studies report significant latency reductions when models leverage GPU tensor cores and TPU systolic arrays for parallelized matrix multiplication. However, these improvements come with trade-offs in energy efficiency and computational

costs that may not be acceptable for all 5G deployment scenarios. Future work will benchmark the same architectures under GPU and Tensor Processing Unit (TPU) acceleration frameworks such as Torch-XLA and Open Neural Network Exchange (ONNX) Runtime to determine whether recurrent architectures can achieve sub-millisecond inference while maintaining deployment practicality [16].

Several additional limitations should be considered when interpreting these results. The evaluation was conducted on a single testbed configuration with specific resource constraints (1 vCPU, 1 GB RAM per pod), and performance characteristics may vary under different hardware specifications or cloud provider environments. The attack dataset comprises three representative classes; operational performance under more diverse or sophisticated attack patterns remains to be validated. Finally, the synthetic traffic generated by PacketRusher, while realistic, may not fully capture the variability and scale of production inter-operator roaming traffic. These limitations present opportunities for extended evaluation in future work.

5.3. Future Research Directions

Future research should explore hybrid ensemble architectures combining lightweight feedforward models for initial screening with recurrent models for offline analysis [23]. Federated learning in O-RAN environments could enable collective model improvement while preserving traffic privacy [24], addressing the challenge of limited labeled attack data in individual operator networks [10]. Adaptive model selection frameworks could optimize resources by dynamically switching between architectures based on real-time threat intelligence levels [25]. As 5G networks evolve toward 6G, emerging work on handover security protocols [7], [26], roaming security analysis [27], and M2M provisioning security [28] will require corresponding advances in IDS operational efficiency, with transfer learning approaches showing promise for addressing data scarcity challenges [8].

6 Conclusion

This work presents systematic operational performance analysis of DL-IDS architectures for cloud-native 5G core networks, extending prior detection accuracy research [13] with quantitative deployment metrics essential for production environments. Using a Kubernetes-orchestrated Open5GS testbed, we evaluated six DL architectures under realistic containerized deployment conditions.

The results reveal a fundamental operational distinction between architecture families. Feedforward models (CNN, MLP, AE) achieve sub-millisecond latency (0.6 ms) with 8-12% CPU utilization, enabling inline deployment as sidecar containers or DaemonSets without impacting 5G service quality. These models support horizontal scaling with 9-11 concurrent instances per server, providing cost-effective coverage across multiple network interfaces. Recurrent architectures (RNN, LSTM, GRU) require near-complete CPU allocation (99-107%) with 3.5-7.2 ms latency, constraining their deployment to batch processing pipelines for offline forensic analysis.

Given equivalent detection accuracy (exceeding 99%) across all architectures, operational efficiency becomes the decisive factor for model selection in cloud-native deployments. These findings provide telecommunications operators with concrete metrics for infrastructure planning, enabling informed trade-offs between real-time inline detection and comprehensive temporal analysis while maintaining QoS requirements and controlling operational expenditure.

Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT).

Funding Details

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (RS-2024-00441484, Development of open roaming technology for Private 5G network).

References

[1] Brown, G.: Service-based architecture for 5G core networks. Huawei White Paper, 1 (2017)
[2] Moreira, J. B., Mamede, H., Pereira, V., Sousa, B.: Next generation of microservices for the 5G service-based architecture. International Journal of Network Management, 30(6), e2132 (2020)
[3] Zhang, B., Zeinaty, P., Limam, N., Boutaba, R.: Mitigating signaling storms in 5G with blockchain-assisted 5GAKA. In 2023 19th International Conference on Network and Service Management (CNSM) (pp. 1-9) (2023)
[4] Huth, C. L., Chadwick, D. W., Claycomb, W. R., You, I.: Guest editorial: A brief overview of data leakage and insider threats. Information Systems Frontiers, 15(1), 1-4 (2013)
[5] Kim, B., Kim, Y., Lee, I., You, I.: Design and implementation of a ubiquitous ECG monitoring system using SIP and the zigbee network. In Future Generation Communication and Networking (FGCN 2007) (2007)
[6] Batool, I.: Real-Time Health Monitoring Using 5G Networks: A Deep Learning-Based Architecture for Remote Patient Care. JMIRx Med (2025)
[7] Kim, J., Duguma, D. G., Astillo, P. V., Park, H., Kim, B., You, I., Sharma, V.: A Formally Verified Security Scheme for Inter-gNB-DU Handover in 5G Vehicle-to-Everything. In 2024 IEEE International Conference on Communications (ICC) (2024)
[8] Guan, J., Cai, J., Bai, H., You, I.: Deep transfer learning-based network traffic classification for scarce dataset in 5G IoT systems. International Journal of Machine Learning and Cybernetics, 12(11), 3351-3365 (2021)
[9] Sahay, R., Nayyar, A., Shrivastava, R. K., Bilal, M., Singh, S. P., Pack, S.: Routing attack induced anomaly detection in IoT network using RBM-LSTM. ICT Express, 10(3), 459-464 (2024)
[10] Hamroun, C., Fladenmuller, A., Pariente, M., Pujolle, G.: Intrusion detection in 5G and Wi-Fi networks: A survey of current methods, challenges & perspectives. IEEE Access, 40950-40976 (2025)
[11] Ko, Y., Pawana, I. W. A. J., You, I.: 5G-AKA-HPQC: Hybrid Post-Quantum Cryptography Protocol for Quantum-Resilient 5G Primary Authentication with Forward Secrecy. IEEE Access (2025)
[12] Pawana, I. W. A. J., Astillo, P. V., You, I.: Lightweight LLM-Based Anomaly Detection Framework for Securing IoTMD Enabled Diabetes Management Control Systems. Sensors (2024)
[13] Pawana, I. W. A. J., Abella, V., Lastre, J. K., Ko, Y., You, I.: Enhancing Roaming Security in Cloud-Native 5G Core Network through Deep Learning-Based Intrusion Detection System. Computer Modeling in Engineering & Sciences, 145(2), 2733-2760 (2025)
[14] Slimani, C., Morge-Rollet, L., Lemarchand, L.: A study on characterizing energy, latency and security for Intrusion Detection Systems on heterogeneous embedded platforms. Future Generation Computer Systems, 161 (2024)
[15] Lastre, J. K., Ko, Y., Kwon, H., You, I.: Evaluating Transport Layer Security 1.3 Optimization Strategies for 5G Cross-Border Roaming: A Comprehensive Security and Performance Analysis. Sensors, 25(19), 6144 (2025)
[16] Wang, Y. E., Wei, G.-Y., Brooks, D.: Benchmarking TPU, GPU, and CPU platforms for deep learning. arXiv preprint arXiv:1907.10701 (2019)
[17] Park, H., Astillo, P. V. B., Kim, T., You, I.: 5G Native Network Function for False Base Station Detection Using Machine Learning Technique. IEEE Transactions on Mobile Computing (Early Access) (2024)
[18] Mosaiyebzadeh, F., Pouriyeh, S. M., Parizi, R. M., Han, M.: Energy-efficient deep learning-based intrusion detection system for edge computing: a novel DNN-KDQ model. Journal of Cloud Computing, 14(1), 62 (2025)
[19] Hossain, M. S., Rahman, M. A., Muhammad, G.: A high performance hybrid LSTM CNN secure architecture for IoT environments using deep learning. Scientific Reports, 15(1), 8234 (2025)
[20] Rahouti, M., Xiong, K., Xin, Y., Javanmard, A.: Federated Learning-Based Intrusion Detection in IoT Networks: Performance Evaluation and Data Scaling Study. Journal of Sensor and Actuator Networks, 14(4), 78 (2024)
[21] Alashjaee, A. M.: Deep learning for network security: An Attention-CNN-LSTM model for accurate intrusion detection. Scientific Reports, 15(1), 7706 (2025)

[22] Ilias, L., Doukas, G., Lamprou, V., Ntanos, C., Askounis, D.: Convolutional Neural Networks and Mixture of Experts for Intrusion Detection in 5G Networks and beyond. Frontiers in Artificial Intelligence, doi: 10.3389/frai.2025.1708953 (2025)
[23] Sadhwani, S., Mathur, A., Muthalagu, R., Kumar, S.: 5G-SIID: An intelligent hybrid DDoS intrusion detector for 5G IoT networks. International Journal of Machine Learning and Cybernetics, 16(2), 1243-1263 (2025)
[24] El-Hajj, M.: Secure and Trustworthy Open Radio Access Network (O-RAN) Optimization: A Zero-Trust and Federated Learning Framework for 6G Networks. Future Internet, 17(6), 233 (2025)
[25] Aminu, M., Akinsanya, A., Dako, D. A., Adebayo, O.: Enhancing cyber threat detection through real-time threat intelligence and adaptive defense mechanisms. International Journal of Cyber Security and Digital Forensics, 13(3), 123-135 (2024)
[26] Kim, J., Astillo, P. V., Sharma, V., Guizani, M., You, I.: MoTH: Mobile Terminal Handover Security Protocol for HUB Switching based on 5G and Beyond (5GB) P2MP Backhaul Environment. IEEE Transactions on Vehicular Technology (2024)
[27] Won, T., Kwon, H., Ko, Y., Lastre, J. K., You, I.: Towards 6G Roaming Security: Experimental Analysis of SUCI-Based DoS, Cost, and NF Stress. Sensors (2025)
[28] Ko, Y., Lastre, J. K., Kwon, H., You, I.: Revisiting the M2M remote SIM provisioning protocol: A comprehensive security and performance analysis. Journal of Information Security and Applications (2024)