Received 15 November 2024; revised 16 December 2024; accepted 24 December 2024. Date of publication 30 December 2024; date of current version 18 April 2025.

Digital Object Identifier 10.1109/OJCOMS.2024.3523468

Secure Distributed Federated Learning for Cyberattacks Detection in B5G Open Radio Access Networks

FAHDAH ALALYAN [1] (Student Member, IEEE), MIRNA AWAD [1], WAEL JAAFAR [1] (Senior Member, IEEE), AND RAMI LANGAR [1,2] (Member, IEEE)

(Selected Best Papers of IEEE International Conference on Communications (ICC'24))

[1]École de Technologie Supérieure, University of Quebec, Montreal, QC H2L 2C4, Canada
[2]Univ. Gustave Eiffel, CNRS, LIGM, 77454 Marne-la-Vallee, France

CORRESPONDING AUTHOR: F. ALALYAN (e-mail: fahdah.alalyan.1@ens.etsmtl.ca)

This work was supported in part by the ANR 5G-INSIGHT Project under Grant ANR-20-CE25-0015, and in part by the Innovation for Defence Excellence and Security (IDEaS) Program of the Department of National Defence Canada.

ABSTRACT The open radio access network (O-RAN) is designed to support the diverse wireless services for beyond 5th-generation (B5G) mobile networks. However, this also expands the potential attack surface, necessitating improved mechanisms for detecting cyberattacks. Advanced artificial intelligence (AI) algorithms, in conjunction with RAN intelligent controllers (RICs), can be utilized to identify threats such as distributed denial-of-service (DDoS) attacks. Nevertheless, AI introduces significant data privacy concern. To address these issues, secure federated learning (FL) can be leveraged to locally train cyberattack detection models and securely transmit the model data for aggregation, thus ensuring protection against eavesdropping. Moreover, peer-to-peer (P2P) FL can be used to avoid the single point of failure inherent in centralized FL. However, securing P2P FL with encryption/decryption or secure average computation (SAC) can result in high communication costs that do not scale well with the number of FL clients. In this paper, we propose a novel P2P FL strategy that ensures secure operation while significantly reducing communication costs. Specifically, we integrate client selection and transfer learning within the RIC-based P2P FL system to detect cyberattacks. Our experiments demonstrate the performance of our method across various scenarios with both balanced and unbalanced dataset distributions. We highlight its superiority in terms of accuracy, robustness, and cost compared to existing benchmarks. Furthermore, we extend our evaluation to a 5G O-RAN testbed, assessing the system's efficiency, accuracy, and adaptability under real-time independent and non-independent and identically distributed (IID/non-IID) traffic conditions. This includes analyzing communication cost, execution time, model loss, and live traffic testing results for practical and real-time deployment.

INDEX TERMS 5G, cybersecurity, cyberattacks, DDoS, federated learning, O-RAN.

I. INTRODUCTION

WIRELESS communication technology has become essential for enabling emerging technologies like vehicle-to-everything (V2X) networks, smart infrastructure, autonomous vehicles, and the Internet of Things (IoT) [1]. Moreover, new applications such as virtual reality (VR) and artificial intelligence (AI) are being spread leading to significant increases in data traffic [2]. As a result, wireless communications have evolved considerably over the past years. With the shift towards fifth-generation (5G) and beyond (B5G) cellular networks, 5G can support a variety of devices, providing them with computational resources and seamless connectivity for intelligent and autonomous operations [3]. Furthermore, 5G enhances data transmission

by offering higher data rates and lower latency, facilitating the growth of data-intensive applications [4].

Nevertheless, the introduction of 5G presents several challenges. Indeed, the complexity of 5G systems increases the threat surface and complicates the definition of system boundaries [5]. Additionally, the rapid deployment of 5G necessitates significant awareness of potential threats. For example, while softwarization, virtualization, and cloudification are essential for network performance, they also create opportunities for security breaches. Similarly, open radio access networks (O-RANs) improve 5G multi-vendor interoperability but pose significant risks due to their open and modular architecture [6]. Therefore, enhanced security measures, such as cyberattack detection, are crucial for O-RAN in 5G. Although there has been substantial research on cyberattack and anomaly detection using machine learning (ML) in RAN, few studies have specifically addressed O-RAN [7].

ML offers robust, innovative, and dynamic solutions for privacy, security, and threat detection in B5G systems. However, a significant challenge is ensuring secure and private knowledge sharing between ML-based detection agents [4]. Alternatively, FL can be leveraged to address data privacy concerns in this context. This distributed ML technique focuses on privacy-preserved collaborative training by sharing model updates rather than raw training data [8]. Consequently, FL is more suitable for maintaining data privacy compared to traditional ML. Despite its advantages, centralized FL faces issues such as the single point of failure and imbalanced data distributions. Introducing the peer-to-peer (P2P) FL concept might help mitigate these issues [7].

In the realm of cyberattack detection, various FL mechanisms have been introduced. P2P FL is particularly preconised for complex O-RAN environments due to the hierarchical structure of RICs and data-driven inputs via open interfaces [9]. A notable advancement in this area is P2P FL with secure average computation (SAC), which employs averaging and n-out-of-n secret partitioning to mitigate risks from semi-honest participants [7]. Although this method shows promising accuracy, it can lead to high communication costs in large-scale systems. Another approach involved using K-means clustering based on client locations within SAC-based P2P FL [7]. This method confined SAC operations to clusters rather than all peers, reducing communication costs. However, clustering did not consider intrinsic characteristics of local datasets or other common criteria like data similarities or peer performance. Alternatively, performance-based neighbor selection (PENS) has been proposed in [10], where clients share their models and training loss to form clusters with similar data distributions. Despite its benefits, sharing models and training loss can potentially expose sensitive information from local datasets.

To circumvent the aforementioned issues, we first proposed in [11] a novel cyberattack detection framework integrating client selection and transfer learning with SAC-based P2P FL in the RIC. In this paper, we extend that foundational work by deploying our system within a 5G O-RAN testbed, thus enabling a comprehensive evaluation of its effectiveness in a realistic network setting. Our results underscore the system's adaptability and robustness in dynamic O-RAN environments, validating its suitability for practical applications in 5G and beyond. Our main contributions can be summarized as follows:

1) We introduce an innovative approach to cyberattack detection using RIC and SAC-based P2P FL, where client peering is subject to a preliminary selection process. This differs from traditional P2P FL, which involves all clients in the training phase.
2) To leverage the advantages of P2P FL with client selection, we suggest implementing transfer learning between selected and non-selected clients. Selected clients participate directly in SAC, while the others utilize the resulting global model through transfer learning. This method reduces communication costs and prioritizes high-performing clients.
3) We conduct a comprehensive performance evaluation of our method, assessing both accuracy and communication efficiency across various dataset distribution scenarios.
4) We validate our proposal using an experimental 5G O-RAN prototype offering a detailed performance assessment in a real deployment scenario. Our analysis covers model's accuracy, communication efficiency, communication cost, execution time, and model robustness under real-time, independent and non-independent and identically distributed (IID/non-IID) traffic conditions.

The remaining of the paper is structured as follows. Section II presents the related works. Section III describes the system model. Section IV details the proposed cyberattack detection mechanism, followed by a presentation of our experimental results in Section V. Finally, Section VI concludes the paper.

II. RELATED WORKS

Besides the attack surface of each 5G subsystem, additional threats for O-RAN expand the attack surface, compared to traditional RAN [5]. For instance, several security and privacy threats arise from the openness of O-RAN and virtualization-related technologies, such as network slicing (NS) [12]. Indeed, NS is a crucial 5G technology that supports heterogeneous services and applications by allocating logical network slices above the physical network [13]. Moreover, the use of AI/ML can further increase the attack surface introducing multiple challenges that threaten the privacy and security of ML models, including poisoning and adversarial attacks on ML models [4]. These issues are also relevant to FL despite its ability to safeguard data privacy [14].

Recent literature has proposed various approaches to tackle the O-RAN security challenges. For instance, the work in [8] aims to build reliable deep reinforcement learning (DRL)-based radio resource management models for mobile virtual

network operators (MVNOs) while preserving data privacy and security. Specifically, it proposed a federated DRL (FDRL) approach for O-RAN slicing, where each MVNO trains a DRL radio resource allocation model and sends the trained model to the RIC for aggregation. A similar problem is solved in [15], aiming to improve NS security. The authors proposed an FL-enabled security orchestrator (FLeSO) to centrally perform security operations in a slicing ecosystem. In contrast, authors in [16] investigated a novel O-RAN cyberattack, called bearer migration poisoning (BMP), to mislead the RIC into triggering a malicious bearer migration procedure.

In the context of FL, authors in [17] proposed an adaptive privacy-preserving FL (Ada-PPFL) scheme based on differential privacy (DP), along with a DP-tolerant cyberattack detection algorithm, to protect the FL server and participants from malicious clients and honest-but-curious servers. Nevertheless, the single point of failure and the imbalance of data distributions in local FL trainers are the main drawbacks of centralized FL. Moreover, split learning (SL) is another technique that splits the model between a server and nodes for training. This method faces several challenges, including slow training, poor scalability, and difficult parallelization [18]. Alternatively, decentralized P2P FL, such as BrainTorrent [19] where each participant may update its local model weights at any given training round, can bypass some of the above challenges. Nevertheless, it might be prone to malicious and semi-honest participants.

To mitigate the effect of semi-honest participants, authors in [20] proposed SAC-based P2P FL that uses an average calculation technique and n-out-of-n secret partitioning method. Also, in [7], the authors proposed SAC-based P2P FL to detect cyberattacks in O-RAN, but it incurred a high communication cost proportional to the number of FL trainers. Consequently, they proposed a variation where K-means clustered trainers based on their locations and SAC operated in each cluster. Thus, the communication cost of SAC is minimized. Some aspects of this approach warrant further consideration. First, clustering was not determined based on characteristics intrinsic to the peer's local dataset or other common criteria, such as peer similarities or performances. Indeed, in real-life applications, it would be more intuitive for clusters to be shaped by shared characteristics or performance among peers. Second, the proposed clustering method poses a potential limitation since clusters remain static even when updates to the peer's local dataset characteristics or performance occur. Overall, the flexibility of clusters and ongoing evaluation are expected as the main drivers in dynamic environments.

To overcome these limitations, we first proposed in [11], a novel cyberattack detection framework integrating client selection and transfer learning with SAC- based P2P FL in the RIC. In this paper, we extend our prior approach by advancing its applicability and robustness in a real-world environment. Specifically, we deploy the proposed approach, namely SAC-based P2P FL-ASTL for Agent Selection and Transfer Learning, within a 5G O-RAN testbed, enabling comprehensive evaluation in a realistic network setting. In addition, unlike our previous work, we assess in this paper the adaptability and robustness of our approach under dynamic, IID, and non-IID traffic conditions as well as live traffic scenarios.

More specifically, we developed two specialized xApps for our experiments: (1) a "Detection xApp" to support the FL-based attack detection process and (2) a "Monitoring xApp" for collecting and preprocessing data within the testbed environment. The Monitoring xApp continuously gathers live traffic data and applies preprocessing techniques that adapt to non-IID conditions, optimizing the dataset for FL training and inference. This dual-xApp structure enhances real-time data handling, making the system more agile in identifying and responding to new threats. Furthermore, we incorporated several performance metrics, such as training time, model loss, and real-time traffic testing, to validate our approach across different scenarios. These metrics provide a detailed understanding of the system's effectiveness and resilience in real-world conditions, underscoring its suitability for dynamic, high-stakes environments like 5G and beyond networks.

III. SYSTEM MODEL

We examine a 5G O-RAN setup where RAN intelligent controllers (RICs) handle resource management. These controllers, compliant with 3GPP and software-defined RAN (SD-RAN) standards, include both near-real-time (Near-RT) and non-real-time (Non-RT) RICs, which are based on software-defined networks (SDN). They manage radio resources and can utilize AI/ML techniques for these tasks [7], [8], [16].

Considering that RICs may operate within large-scale systems, it is beneficial to implement cooperative mechanisms to leverage shared experiences. FL can be utilized among RICs to achieve this. FL's main goal is to preserve data privacy by allowing collaborative training of ML models on local datasets while sharing only model parameters. Decentralized FL is particularly appealing as it removes the need for an aggregation server, thus reducing the risks associated with a single point of failure, global model corruption, slow convergence, and data misclassification. Also, decentralized FL, such as SAC-based P2P FL, secures the communication of model updates, thus protecting against semi-honest participants [7], [17], [20].

Secure Average Computation (SAC), as shown in Fig. 1, serves a similar purpose in distributed FL as the aggregation server does in centralized FL, i.e., both average participants' model updates to create a global model [20]. However, the way they transmit these updates within the FL framework differs. Indeed, SAC employs two mechanisms, namely lightweight n-out-of-n secret partitioning and secure multi-party average calculation. In the secret partitioning method, each $Agent_j$ ($j \in \{1, \ldots, N\}$) generates N positive random numbers $\{rn_{j1}, \ldots, rn_{jN}\}$, which are then used to compute the

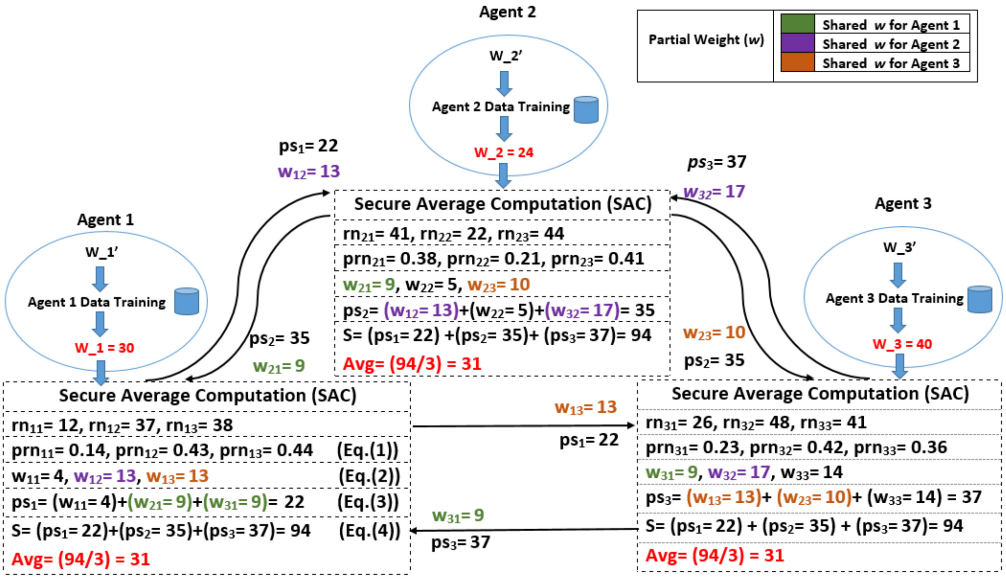

FIGURE 1. Example of SAC with three agents.

percentage distributions, denoted as $prn_{j1}, \ldots, prn_{jN}$, such that:

$$prn_{ji} = \frac{rn_{ji}}{\sum_{k=1}^{N} rn_{jk}}, \quad (i,j) \in \{1,\ldots,N\}^2, \tag{1}$$

where N indicates the number of *Agents* in the P2P FL environment. Then, the percentage distributions are used to generate N partial weights $\{w_{j1}, \ldots, w_{jN}\}$ expressed by

$$w_{ji} = w_j \times prn_i, \quad (i,j) \in \{1,\ldots,N\}^2, \tag{2}$$

where w_j is the model update of $Agent_j$. For instance, $Agent_1$ has $w_1 = 30$ as the model update. This update is partitioned securely into $w_{11} = 4$, $w_{12} = 13$, and $w_{13} = 13$ as presented in Fig. 1. The resulting $\{wj_1, \ldots, w_{j_N}\}$ is used in the multi-party average calculation. In particular, each $Agent_j$ keeps its partial weights w_{jj} and shares the other parts w_{ji} with the respective $Agent_i$, $\forall i \in \{1,\ldots,N\}$ and $i \neq j$. To illustrate this mechanism in Fig. 1, $Agent_1$ keeps $w_{11} = 4$ and shares $w_{12} = 13$ with $Agent_2$ and $w_{13} = 13$ with $Agent_3$. Then, each $Agent_j$ computes its subtotal ps_j as

$$ps_j = \sum_{i=1}^{N} w_{ji}, \quad j \in \{1,\ldots,N\}. \tag{3}$$

Finally, it computes the aggregated SAC weights S and the averaged weight Avg as follows:

$$S = \sum_{j=1}^{N} ps_j \text{ and } Avg = S/N. \tag{4}$$

IV. PROPOSED APPROACH FOR ATTACK DETECTION: SAC-BASED P2P FL-ASTL

A. DESCRIPTION

We introduce a communication-efficient SAC-based P2P FL framework, named SAC-based P2P FL-ASTL, which incorporates agent selection and transfer learning. Initially, clients (or agents) are selected for peering based on their performance in each round. The selected clients then participate in SAC to create a global model. For efficient SAC peering, only agents with high accuracy, determined by their local validation datasets, are chosen in each round. This process ensures the global model is generated while safeguarding both model updates and performance through SAC. To further protect datasets within the FL framework, each agent uses a unique local validation dataset to produce performance metrics, thus differing from conventional FL methods.

In existing research, a balance between communication costs, computation costs, and privacy is often discussed. For example, while encryption enhances privacy, it also raises computation costs due to the encryption/decryption overhead. Our approach seeks to strike a balance between communication costs and privacy. To do so, we introduce a novel method that minimizes communications in large-scale SAC-based P2P FL without compromising the privacy of model updates and local datasets. Our method operates through two main steps: "Initialization" and "Learning Process", which are iteratively executed across all FL rounds.

B. OPERATION

Our system consists of N clients (or agents) denoted by $\mathcal{A}_\mathcal{N} = \{A_1, \ldots, A_N\}$, where each agent A_i has a local training dataset D_i and a local validation dataset V_i from $\mathcal{D} = \{D_1, \ldots, D_N\}$ and $\mathcal{V} = \{V_1, \ldots, V_N\}$, respectively. Our approach follows these steps:

1) *Initialization:* It involves multiple steps as summarized in lines 1 to 6 of Algo. 1. First, in each round, agent A_i begins training from D_i to update its model weights w_i. Then, it uses the validation dataset V_i to generate

Algorithm 1 Proposed SAC-Based P2P FL-ASTL

Input: Number of agents N, training datasets $\mathcal{D}$, validation datasets $\mathcal{V}$, number of FL rounds T.

Initialization:

1: **for** $i = 1$ to N **do**
2: Get w_i after local training of D_i.
3: Test the updated model w_i on validation dataset V_i.
4: Get $F1_i$ and Acc_i
5: **end for**
6: Update $\mathcal{W} = \{w_1, \ldots, w_N\}$ using *Algorithm 2*

SAC-based P2P FL for subsequent FL rounds:

7: **while** $t \leq T$ **do**
8: **for** $i = 1$ to N **do**
9: Train local dataset D_i using W_{A_k} for ε episodes.
10: Get the updated model w_i'.
11: Test the updated model on validation dataset V_i.
12: Get $F1_i$ and Acc_i.
13: **end for**
14: Update $\mathcal{W} = \{w_1, \ldots, w_N\}$ using *Algorithm 2*
15: **end while**

Algorithm 2 Learning Process

Input: Number of agents N, *performancemetrics*:$[F1_i, Acc_i]_{i=1,\ldots,N}$

Agent selection:

1: **function** SELECT(N, $[F1_i, Acc_i]_{i=1,\ldots,N}$)
2: Get $Avg_{F1} \leftarrow$ **SAC**(N, $[F1_i]_{i=1,\ldots,N}$)
3: Get $Avg_{acc} \leftarrow$ **SAC**(N, $[Acc_i]_{i=1,\ldots,N}$)
4: Initialize an empty set $\mathcal{S}$ for selected agents
5: **for** $i = 1$ to N **do**
6: Assign A_i to set $\mathcal{S}$ based on eq. (5)
7: **end for**
8: **return** $\mathcal{S}$
9: **end function**
10: Get $\mathcal{A}_\mathcal{K} = \{A_1, \ldots, A_K\} \leftarrow \mathcal{S}$ % Set of selected agents
11: Get $\mathcal{A}_\mathcal{R} = \{A_i, \ldots, A_R\} = \mathcal{A}_\mathcal{N} \backslash \mathcal{A}_\mathcal{K}$ % Set of disregarded agents

Global model design:

12: Get $W_{\mathcal{A}_\mathcal{K}} \leftarrow SAC(\mathcal{A}_\mathcal{K}, K)$

Transfer learning:

13: Update $w_i, \forall i \in \mathcal{A}_\mathcal{R}$ using eq. (6)

its local performance metrics, such as the F1-score, denoted $F1_i$, and accuracy Acc_i.

2) *Learning process:* It corresponds to the operations of lines 6 and 14 of Algo. 1. Specifically, it comprises three steps: Agent selection, global model design, and transfer learning, as summarized in Algo. 2. In the first phase (Algo. 2, lines [1-11]), each node engages in SAC to securely exchange validation dataset metrics. Then, the average F1-score and accuracy are shared among all agents. To be selected for peering, each agent A_i verifies that it satisfies the conditions $F1_i \geq Avg_{F1}$ and $Acc_i \geq Avg_{acc}$. Subsequently, we have

$$\mathcal{S} = \{A_i \mid F1_i \geq Avg_{F1} \text{ and } Acc_i \geq Avg_{acc}\}. \quad (5)$$

Let $\mathcal{A}_\mathcal{K} = \{A_1, \ldots, A_K\}$ and $\mathcal{A}_\mathcal{R} = \{A_i, \ldots, A_R\}$ be the sets of selected and disregarded agents, respectively. Then, in the second phase (Algo. 2, line 12), selected agents participate in SAC to generate the global model W_{A_k}, following (1)-(4)[1] In the final phase, transfer learning of the global model W_{A_k} occurs, where the model is transferred to the disregarded agents in $\mathcal{A}_\mathcal{R}$ such that

$$w_i = W_{\mathcal{A}_\mathcal{K}}, \forall A_i \in \mathcal{A}_\mathcal{R}, \quad (6)$$

where w_i refers to the learning model of agent A_i. While "transfer learning" often suggests adapting knowledge from one task or domain to another, here we employ a broader perspective [21]. In our framework, the global model, enriched by insights from selected high-performing agents, is transferred to non-selected agents. This knowledge sharing, even under the same overarching task, effectively transfers learned representations and model parameters, thus enabling all agents to improve their local training performance over subsequent rounds [21].

3) *SAC-based P2P FL-ASTL for subsequent rounds:* In the following FL rounds, the same initialization and learning processes are repeatedly executed until the last FL round is reached. This is emphasized in lines 7-15 of Algo. 1.

Remark 1: To mitigate potential biases arising from local validation sets that may not initially reflect the global data distribution, our framework leverages an iterative and multi-round approach. After each round, selected high-performing agents contribute to the global model, which is then disseminated to all agents for local retraining. In subsequent rounds, agents re-evaluate their updated models, and those demonstrating improved performance can be selected later. Over time, this iterative process integrates insights from a more diverse set of agents, thus preventing any single local model from dominating, and guiding the global model toward a more balanced and representative data distribution.

C. ANALYSIS OF COMMUNICATION COST

In conventional SAC-based P2P FL, each agent must send the computed partitions and subtotals of W model weight values to the other $(N-1)$ agents in each FL round [20]. Consequently, the total communication effort per round is calculated as $2WN(N-1)$, resulting in a total communication cost of

$$c = 2WN(N-1)T. \quad (7)$$

[1]To be noted that this global model, generated at intermediate stages, facilitates incremental improvements during training but is not to be deployed. Only the final global model, obtained after convergence in the last FL round, is used for deployment.

Conversely, our proposed method, which incorporates agent selection, reduces the training effort. In each round t, K_t agents participate in SAC. Therefore, the communication cost for training in round t is $c_{t,1} = 2WK_t(K_t - 1)$, $\forall t = 1, \ldots, T$. Also, the SAC-based exchange of averaged F1 Avg_{F1} and Accuracy Avg_{acc} requires a communication effort of $c_{t,2} = 2QN(N-1)$, where Q represents the averaging cost. Finally, transfer learning necessitates a broadcast cost of $c_{t,3} = W$. Thus, the total communication cost for a single round in our proposed method is $c_{t,1} + c_{t,2} + c_{t,3}$, and the overall cost can be expressed as

$$c' = \sum_{t=1}^{T} \left(c_{t,1} + c_{t,2} + c_{t,3}\right) = \sum_{t=1}^{T} (2WK_t(K_t - 1) + 2QN(N-1) + W). \quad (8)$$

Finally, for the conventional centralized FL [22], the communication cost is given by

$$c'' = W(N+1)T, \quad (9)$$

since it needs N transmissions from the agents to the central aggregator to upload their model parameters and only one broadcast transmission from the aggregator to the agents to update their models.

V. PERFORMANCE EVALUATION

In this section, we assess the performance of our proposed FL method, termed "SAC-based P2P FL-ASTL", for detecting cyberattacks in network traffic, focusing on accuracy and communication efficiency. We compare our approach with the traditional "Centralized FL" [22] and the decentralized "SAC-based P2P FL" [7] in a simulated environment first, where existing datasets were used for assessment.

Subsequently, our evaluation is extended to a real-world deployment within a 5G O-RAN testbed. Within the testbed, we assess these methods across a broad range of metrics, including execution time, communication cost, model accuracy, model loss, and testing of the global model on live traffic. This allows us to evaluate the adaptability and resource efficiency of the approaches under real-time, IID, or non-IID data conditions. The obtained results from both the simulated and real environments provide a comprehensive view of the achievable performances in such conditions.

A. PERFORMANCE ASSESSMENT IN A SIMULATED ENVIRONMENT

1) DATASET SETUP

For detecting cyberattacks, we selected the UNSW-NB15 dataset, a contemporary dataset for network intrusion detection systems (NIDS) [23]. This dataset includes nine attack categories, with each category comprising a set of records. Each record features 49 extracted features.

Preprocessing steps: We have considered a subset of the UNSW-NB15 dataset (2 data files out of the four available) and preprocessed it to be suitable for the purpose of cyber-attack detection in O-RAN. Specifically, we merged the data of the two files and removed six non-relevant features, which are {*srcip*, *dstip*, *attack_cat*, *ct_flw_http_mthd*, *is_ftp_login*, *ct_ftp_cmd*}. The first three features have been eliminated for effective detection since, in real-world scenarios, network flows lack attack categories, and IP addresses may be dynamic or manipulated through IP spoofing. The other three features have been discarded due to their high number of null values. As a result, each record is left with 43 features. Moreover, preprocessing steps such as splitting, feature/categorical encoding, and normalization have been applied. The dataset has been partitioned into training, validation, and testing datasets. Each agent has local training and validation datasets, and a final testing dataset to evaluate the global model.

Dataset distribution: Following preprocessing, the remaining dataset includes 150,000 records, distributed among $N = 100$ agents for training and validation, with an additional separate testing dataset comprising 10,000 records for global model testing. Within the records, 60% represent the attack classes. Each agent receives 1,500 records split into (80%, 20%) between training and validation. Therefore, each agent will have the same dataset size.

The partition of the attack classes at each agent depends on the type of distribution, i.e., IID or non-IID. For an IID distribution, agents have an equal attack class partition of 60%. In contrast, for a non-IID distribution, the partition of the attack class was varied randomly within specific ranges of attack class partitions, in particular, we designed the range [20%, 40%] for the intense non-IID setting and [30%, 60%] for the moderate non-IID setting.

2) FL MODEL ARCHITECTURE AND HYPERPARAMETERS TUNING

We adopt here a deep learning (DL) architecture comprising four layers for our system. Specifically, the input layer is tailored to handle 43 features, followed by two dense hidden layers with 30 and 10 neurons, respectively. The architecture concludes with an output layer consisting of two neurons and is accompanied by a softmax layer for probabilistic classification between "attack" and "no attack". The ReLU activation function is applied to the hidden layers, supplemented by L1 regularization. Throughout our experiments, we set the learning rate to 10^{-4} and the batch size to 100. Moreover, we run FL for $T = 50$ rounds, where, in each round, an FL agent trains locally for $\varepsilon = 10$ episodes.

3) SIMULATION RESULTS

In Table 1, we evaluate and compare the communication costs for the proposed "SAC-based P2P FL-ASTL", and the two benchmarks "Centralized FL" and "SAC-based P2P FL". We assume here an IID distribution of datasets and that the FL model of any agent has $W = 1622$ weight values. As shown in Table 1, Centralized FL exhibits the lowest communication cost, approximately 31.25 megabytes (MB),

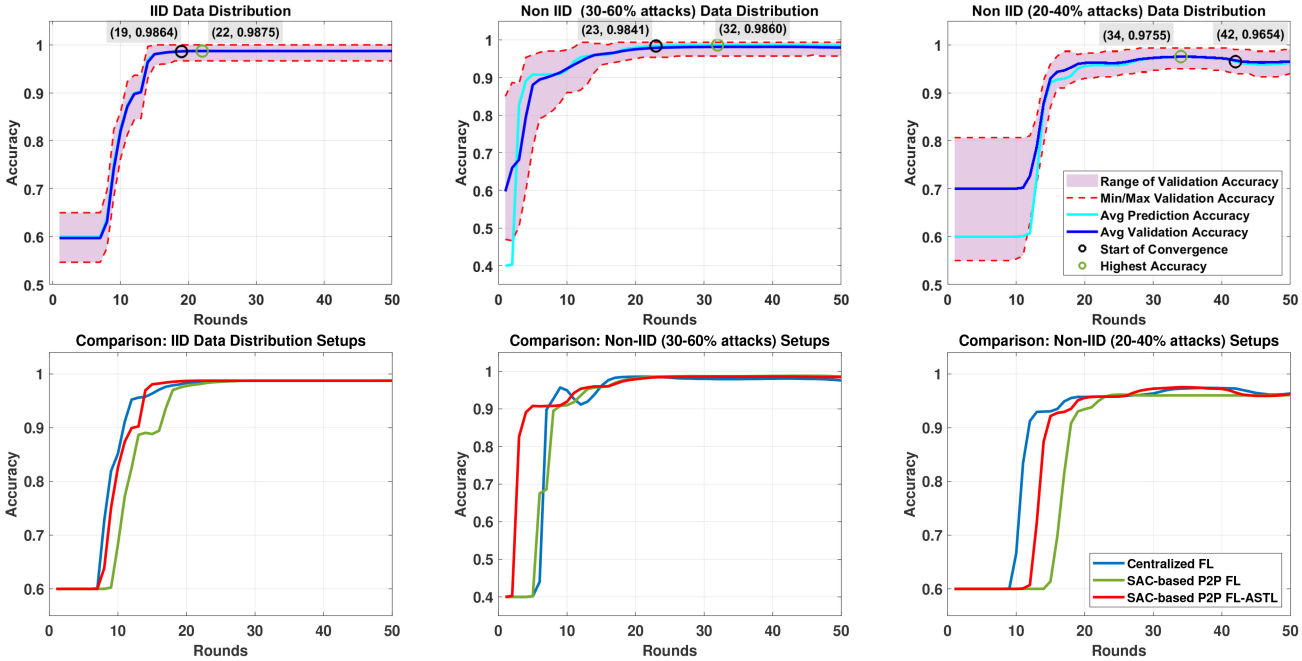

FIGURE 2. Accuracy of: SAC-based P2P FL-ASTL (top row); Several FL methods (bottom row), with different data distributions, using the UNSW-NB15 dataset.

TABLE 1. UNSW-NB15: Communication costs.

FL Method	# model weight values	Avg. commun. cost per round (MB)	Total commun. cost (MB)
SAC-based P2P FL	$c = 32115600$	122.55	6127.5
SAC-based P2P FL-ASTL	$c' = 8068500$	31.92	1595.6
Centralized FL	$c'' = 163822$	0.625	31.25

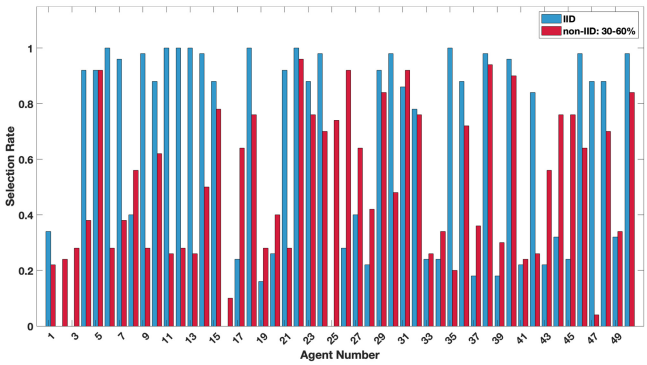

FIGURE 3. Selection rate distribution for 50 agents, sampled from $N = 100$ agents (different IIDness settings), using the UNSW-NB15 dataset.

due to its minimal transmission requirements to and from the aggregation server. In contrast, SAC-based P2P FL incurs the highest communication cost, around 6127 MB, because of the extensive data exchanges among all $N = 100$ agents in the FL system. Our proposed method, however, reduces the communication cost by 74% compared to SAC-based P2P FL, thanks to its incorporation of agent selection and transfer learning. Despite the low communication effort of Centralized FL, it remains vulnerable to security breaches and failures due to its centralized architecture and lack of robust security mechanisms.

Fig. 2 illustrates the global accuracy of our proposed method (top row) compared to benchmarks (bottom row) for both IID and non-IID distributions. In the top row, SAC-based P2P FL-ASTL achieves convergence after 19, 23, and 42 rounds with accuracy levels exceeding 98.6%, 98.4%, and 96.5% for IID, moderate non-IID, and intense non-IID scenarios, respectively. This demonstrates the efficiency of our approach in detecting cyberattacks. However, as non-IIDness increases, convergence takes longer and accuracy slightly decreases (by 0.3% to 2%), showcasing the robustness of our method in non-IID environments. This is also reflected in the increasing variance of accuracy performance at convergence (area between red dashed lines).

When comparing our method to the benchmarks in the bottom row of Fig. 2, we observe that all methods achieve similar accuracy values, indicating comparable robustness to dataset non-IIDness. However, our approach converges more quickly in the IID setting, reaching convergence at round 19, whereas centralized FL and SAC-based P2P FL begin to converge at rounds 23 and 25, respectively.

To analyze the behavior of our proposed method, we plotted the selection rates of 50 agents out of the $N = 100$ available ones performing FL in both IID (blue) and non-IID (red) settings, as shown in Fig. 3. We observed significant variation in selection rates among agents, indicating the

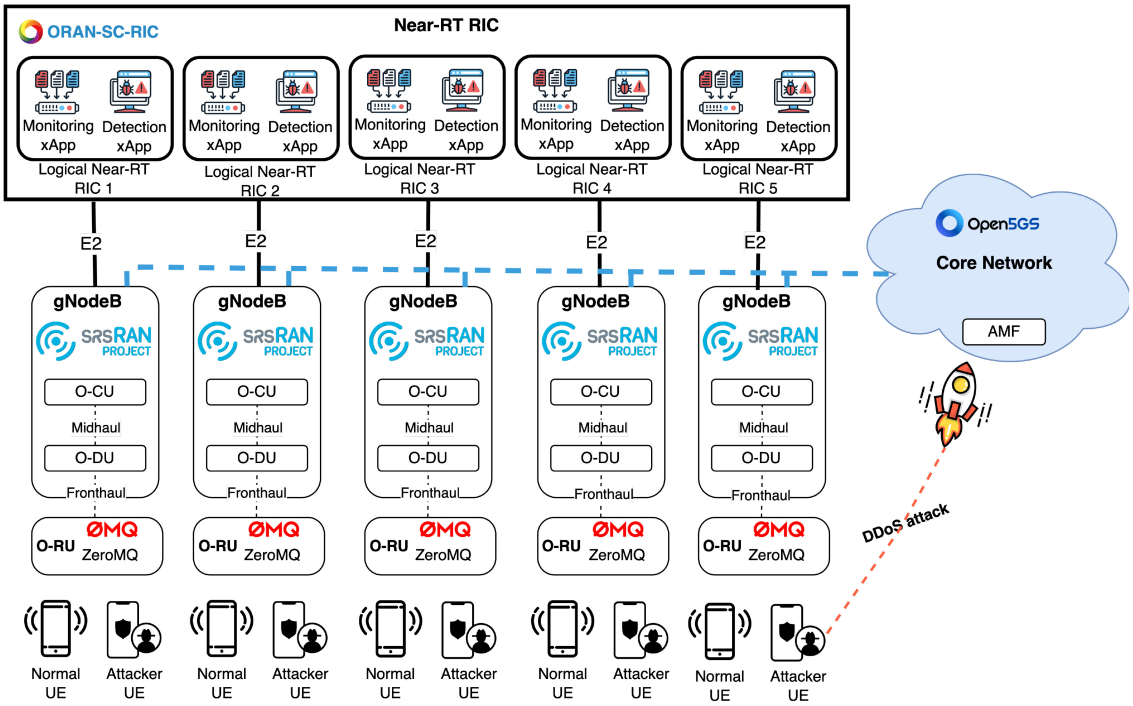

FIGURE 4. Architecture of the 5G O-RAN testbed.

system's adaptability in selecting agents based on their F1 and accuracy performances. A high selection rate (close to 1) suggests that an agent is more likely to be chosen in most rounds. The variation in selection rates is more pronounced in the non-IID setting compared to the IID one. This strategic selection under non-IID conditions demonstrates the model's robustness by maintaining performance and mitigating the effects of unbalanced data distributions through adaptive reliance on different agents.

B. PERFORMANCE ASSESSMENT IN A 5G O-RAN TESTBED

1) TESTBED SETUP

To evaluate the effectiveness of the FL algorithms, we designed an O-RAN based 5G testbed, as illustrated in Fig. 4. This testbed encompasses the 5G core network (CN), logical near real-time RICs hosted in a physical real-time RIC [24], gNodeBs (gNBs), and multiple user equipment (UEs) with either a normal/legitimate profile or an attacker profile. The CN leverages Open5GS, an open-source containerized/dockerized 5G core framework that offers essential network functions and provides seamless connectivity between the RAN and external networks [25]. The containerized structure enables efficient integration with other network components, fostering a robust platform for evaluating our security techniques within the testbed. For network intelligence, the near real-time RIC (Near-RT RIC) is deployed using the O-RAN Software Community's ORAN SC framework [26]. Our setup includes five logical RICs, each configured as an FL agent hosting its respective xApp. These RICs operate on a shared physical infrastructure, allowing inter-agent communication for collaborative online learning. Note that in "Centralized FL", RIC 1 serves also as the aggregation server, thus managing and coordinating the learning process across all agents. This testbed design allows us to evaluate both distributed and centralized FL paradigms within the same environment. The RAN setup utilizes the open-source srsRAN project to deploy modular 5G gNBs, each partitioned into a central unit (CU) and a distributed unit (DU) in line with the O-RAN modular architecture [27]. The DU manages virtual UEs through ZeroMQ (zmq) communication protocol [28]. This protocol enables efficient data exchange, supporting the srsUE framework in emulating realistic user behaviors. Each gNB instance connects to two virtual UEs, allowing a total of 10 emulated users across the testbed. To evaluate our FL-based attack detection framework under various conditions, we simulated user activity by generating HTTP traffic to mimic benign behavior, and DDoS attacks traffic targeting the CN (specifically, the AMF network function). The Mausezahn tool was used to emulate attacks, including TCP, UDP and mixed traffic types [29]. Our experimental setup provides a rigorous and realistic 5G environment, enabling us to perform in-depth evaluations of the FL algorithms in response to diverse traffic patterns.

2) DATASET SETUP

To effectively detect cyberattacks in the 5G O-RAN environment, we generate datasets that support both online and offline training modes. This approach enables real-time detection and the analysis of historical data. We generate Distributed Denial of Service (DDoS) attack traffic alongside

TABLE 2. Testbed network configuration.

gNodeB	UE	UE IP Address	Attack Status
gNodeB 1	UE1	10.45.1.2	Attacker
	UE2	10.45.1.4	Benign
gNodeB 2	UE3	10.45.1.6	Attacker
	UE4	10.45.1.7	Benign
gNodeB 3	UE5	10.45.1.8	Attacker
	UE6	10.45.1.9	Benign
gNodeB 4	UE7	10.45.1.10	Attacker
	UE8	10.45.1.11	Benign
gNodeB 5	UE9	10.45.1.12	Attacker
	UE10	10.45.1.13	Benign

benign traffic, equipping our model to differentiate between attack and non-attack scenarios. The datasets are collected and monitored through a dedicated xApp, allowing us to observe and capture network behaviors for further analysis. Below, we outline each step of this setup:

Data generation: DDoS attacks are designed to overwhelm network services by flooding targeted servers with high volumes of traffic from various distributed sources, such as compromised Internet-connected devices, e.g., bots, or attacker's computers. DDoS attacks can be of several types, including volume-based attacks, protocol-layer attacks, application-layer attacks, and zero-day attacks [30].

Our study focuses on Transmission Control Protocol Synchronize (TCP SYN) and User Datagram Protocol (UDP) floods as representative examples of protocol-layer and volume-based attacks. A TCP SYN flood exploits the TCP handshake process by sending numerous SYN requests without completing the handshake, leaving server resources in half-open connections and leading to denial of service. Meanwhile, UDP floods overwhelm the target by sending many UDP packets to random ports. This forces the server to continuously search for non-existent applications and respond with Internet Control Message Protocol (ICMP) error packets, ultimately exhausting the bandwidth and server resources [30]. In our testbed configuration, we generate attack and benign traffic from two UEs associated with each gNodeB, as defined in Table 2. This setup enables a realistic testing environment by distributing both types of traffic across multiple sources. For each gNodeB, we designate specific UEs to generate DDoS attack traffic. The "Attacker" associated with each gNodeB generates TCP SYN and UDP flood, while the "Benign" UE generates Hypertext Transfer Protocol (HTTP) traffic, simulating normal network behavior. Consequently, we can observe and analyze the network's response to cyberattacks alongside regular traffic.

Monitoring xApp: In our testbed, the monitoring xApp, deployed within the Near-RT RIC, is essential for continuous traffic data collection and monitoring, enabling a flexible FL lifecycle in line with O-RAN standards. This xApp operates offline and online to support model training and adaptation based on historical and real-time data. In offline mode, the collected dataset is stored as a historical record, serving as the basis for training the FL model. As a result, an initial model is deployed for inference, relying on accumulated network behavior data to detect cyber threats effectively, as outlined in O-RAN guidelines [31].

In online mode, the monitoring xApp periodically collects live traffic data in predefined intervals, e.g., 10 seconds across four sampling windows for 40 seconds. This live dataset undergoes immediate preprocessing and integration into the FL training cycle, enabling continuous model adaptation. We ensure the model dynamically adjusts to emerging network patterns and threat behaviors by updating the model with recent traffic data, which aligns with key recommendations for adaptive model training in the O-RAN's standards [31]. This dual-mode data handling provides a robust foundation for maintaining precise and responsive cyberattack detection in the evolving 5G O-RAN environment.

In both offline and online modes, the monitoring xApp follows predefined data collection intervals that are identical for all agents, thus ensuring that each agent naturally obtains a similar dataset size. Moreover, aligning with O-RAN standards [31], the offline mode corresponds to AI/ML inference over cumulative historical datasets. In contrast, the online mode corresponds to AI/ML training that periodically incorporates newly captured live traffic data. In both modes, the datasets continually evolve, either by accumulating additional historical data or by integrating fresh traffic samples, thereby preventing them from being static. This continuous adaptation supports frequent re-evaluation, ensuring our approach remains practical and responsive in real-time network environments.

Preprocessing steps: To simplify preprocessing, we incorporated the LUCID tool right after data collection, thus minimizing delay and agreeing with O-RAN's requirement for scalable and low-resource processing. The LUCID tool enables fast and lightweight preprocessing to handle the large volume of data generated by the monitoring xApp within the Near-RT RIC. Specifically, with LUCID, we efficiently extract the necessary features and apply essential transformations to data while omitting features like application-layer attributes, link layer encapsulation types, TCP/UDP ports, and IP addresses, that could hinder model generalization [32].

In offline mode, the labeling process directly references attacker IP addresses as predefined in our configuration, which is feasible since these addresses are known beforehand. However, in online mode, to handle real-time data adaptively, we utilize the offline trained model to predict attackers' IP addresses for dynamic labeling in real-time environments while enhancing the adaptive detection capabilities. The integration and customization of LUCID with our FL framework ensure a robust and efficient preprocessing pipeline.

Dataset distribution: Each FL agent corresponds to an xApp in a Near-RT RIC, providing 5 agents deployed across 5 RICs, with each xApp handling its unique dataset. These

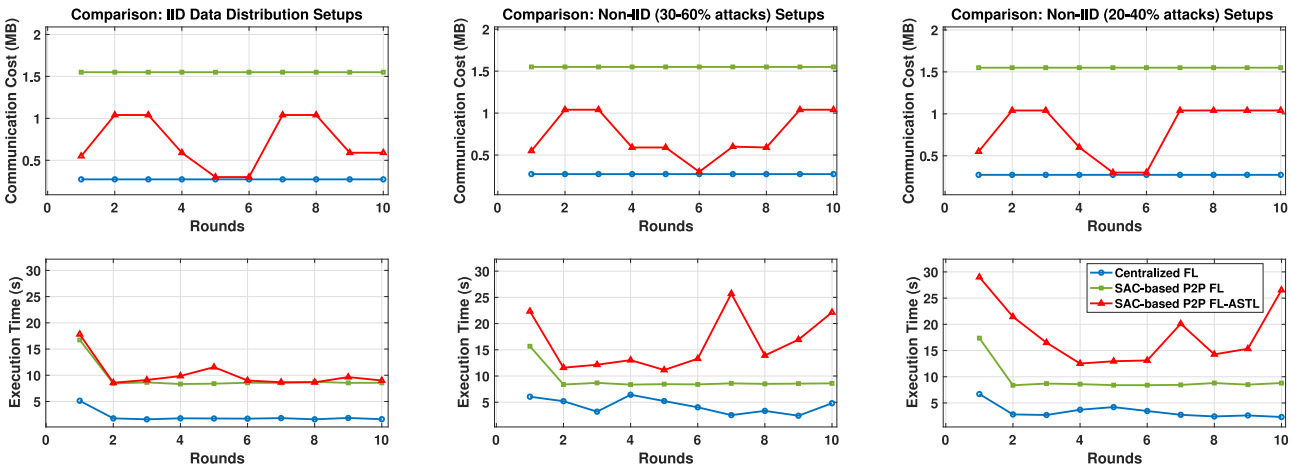

FIGURE 5. Communication cost (top row) and execution time (bottom row) across several FL methods under various data distributions in the testbed-collected dataset.

datasets are based on the data collected from associated UEs, as explained in Table 2 above. In the IID configuration, we apply LUCID's standard data distribution, which equally balances the dataset between attack and benign traffic at 50% each. This balanced approach establishes our baseline and aligns with LUCID's setup for consistent data handling. Given the need to replicate real-world traffic conditions, we introduce two non-IID scenarios by customizing LUCID to achieve varying attack distributions. Specifically, we test with attack distributions in the ranges of [30%-60%] and of [20%-40%], respectively, ensuring realistic data heterogeneity between agents. The attack distributions are designed as follows:

- *[30%-60%] distribution:* [30%, 40%, 50%, 55%, 60%] across Agents 1 through 5, respectively.
- *[20%-40%] distribution:* [20%, 25%, 30%, 35%, 40%] across Agents 1 through 5, respectively.

3) FL MODEL ARCHITECTURE AND HYPERPARAMETERS SELECTION

Here, we consider a Convolutional Neural Network (CNN) architecture with four primary layers for our system. The input layer accepts data with dimensions $10 \times 11 \times 1$, representing the features in our dataset. A convolutional layer with 64 neurons follows, each with a kernel size of 3×11 and "valid" padding, thus reducing the spatial dimension and producing an output shape of 1×64. This configuration effectively captures spatial features across the input data. The activation function used for this layer is ReLU, introducing non-linearity and enabling the network to learn complex patterns. Subsequently, we introduce a global max pooling layer that reduces each feature map to its maximum value, producing an output shape of 64. This step simplifies computations and minimizes overfitting risks. A flattening layer follows, converting the pooled feature maps into a one-dimensional vector, thus preparing the data for the final classification stage. The model concludes with a dense output layer of a single neuron, paired with a sigmoid activation function for binary classification between "attack" and "no attack".

Throughout our experiments, we set the learning rate to 10^{-3} and the batch size to 1024 to balance convergence speed and stability. The FL is conducted over $T = 10$ rounds, with each FL agent training locally for $\varepsilon = 10$ episodes per round.

TABLE 3. Testbed dataset: Communication costs.

FL Method	Avg. commun. cost per round (MB)	Total commun. cost (MB)	Data Distribution
SAC-based P2P FL	1.55	15.5	All distributions
SAC-based P2P FL-ASTL	0.70 0.738 0.799	7.08 7.38 7.99	IID non-IID [30%-60%] non-IID [20%-40%]
Centralized FL	0.272	2.72	All distributions

4) EXPERIMENTAL RESULTS

Table 3 presents the communication costs (in megabytes) of the proposed "SAC-based P2P FL-ASTL", and the two benchmarks "Centralized FL" and "SAC-based P2P FL", when using data from our testbed. Similarly to the results in Table 1, "Centralized FL" achieves the lowest communication cost (2.72 MB) given its broadcast nature of the global model while "SAC-based P2P FL" realizes the highest cost of 15.5 MB due to the extensive data exchange between the $N = 5$ agents, given any data distribution. In contrast, the proposed "SAC-based P2P FL-ASTL" incurs a communication cost between 7.08 MB (IID case) and 7.99 MB (worst non-IID case), which is up to 55% less than SAC-based P2P FL.

Fig. 5 presents a more detailed landscape of the communication cost (top row) across the FL rounds, with an

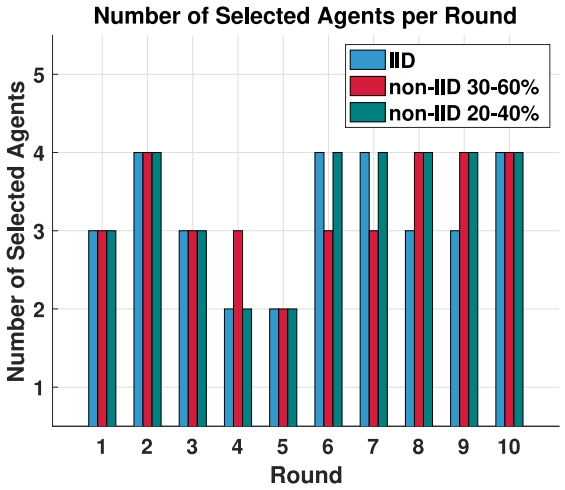

FIGURE 6. Number of selected agents per round in different IIDness settings, using the testbed-collected dataset.

assessment of the execution times (bottom row). As can be seen, the communication cost of the proposed method fluctuates between the minimal cost (by "Centralized FL") and is consistently below the worst cost (by "SAC-based P2P FL"). This fluctuation is due to the dynamic agent selection mechanism, as shown in Fig. 6. Indeed, this mechanism adjusts the number of participating agents based on the agents' performances in each round, thus resulting in decreased communication costs during rounds where fewer agents are selected (e.g., rounds 4 to 6). However, the increase in communication cost in rounds 7 and 8 reflects a reintroduction of more agents to maintain model performance as the data distribution stabilizes, especially in non-IID scenarios. However, we notice that the execution time of "SAC-based P2P FL-ASTL" is often higher than that of the other methods. This is expected since the agent selection mechanism is a multi-step process that involves (1) executing SAC for performance evaluation, (2) agent selection, (3) SAC for model aggregation among selected agents, and (4) transfer learning for model refining. In contrast, the "SAC-based P2P FL" method only performs SAC for model aggregation without additional steps, making it faster. Nevertheless, the extra steps in "SAC-based P2P FL-ASTL" are essential for achieving robust performance and adaptability in varied data distributions, as they allow for a more selective and efficient approach to P2P FL.

Fig. 7 illustrates the accuracy performance of the aforementioned FL methods in different data distribution settings within the 5G O-RAN testbed. The top row displays the accuracy performances (average validation, average prediction, Min/Max validation, and range of validation) of the "SAC-based P2P FL-ASTL" only, while the bottom row compares its average results with those of "Centralized FL" and "SAC-based P2P FL" benchmarks, as functions of the number of FL rounds. In the IID scenario (top row, left), "SAC-based P2P FL-ASTL" converges rapidly within the first few rounds, achieving accuracy levels exceeding 99%. The narrow range of accuracy variance throughout the training indicates stability when handling homogeneous data distributions. In the non-IID 30%-60% attack distribution setting (top row, middle), the model converges by round 7, achieving an accuracy of 99.81% with limited fluctuations. This outcome demonstrates the model's robustness against moderate non-IIDness, where attack traffic affects data distributions. Under the more intense non-IID 20%-40% attack distribution (top row, right), the "SAC-based P2P FL-ASTL" converges slightly earlier, around rounds 5 and 6, and stabilizes at a similarly high accuracy level of 99.8%. These results demonstrate the model's robustness and adaptability to any level of attack intensity and data distribution. In the bottom row of Fig. 7, "SAC-based P2P FL-ASTL" consistently achieves better accuracy than the benchmarks. It converges to its best performance in about 6 or 7 rounds, against a faster convergence of "Centralized FL" (in 4 to 6 rounds) and "SAC-based P2P FL" (in 3 to 6 rounds).

In Fig. 8, we depict the different methods' training/validation losses as functions of the FL rounds. First, for any method, we notice that the training and validation losses are almost identical, demonstrating their stable operation. Then, "SAC-based P2P FL-ASTL" stabilizes its loss in 5 rounds only (achieving a loss of 10^{-2}), compared to 10 FL rounds for "SAC-based P2P FL" and a higher number of rounds for "Centralized FL", to reach the same loss value. The loss analysis confirms that high accuracy can be reached with minimal loss, supporting the decision to limit the number of rounds for optimal resource use. Hence, "SAC-based P2P FL-ASTL" is suitable for efficient and reliable real-time deployments.

Fig. 9 illustrates the agent selection rate distribution when using "SAC-based P2P FL-ASTL" across $N = 5$ agents. Similarly to the results of Fig. 3, variations in selection rates reflect the system's capability to adapt based on the agents' F1 and accuracy performances. This fluctuation is accentuated in the non-IID settings showcasing flexibility in mitigating the unbalanced data effect. Interestingly, we notice that Agent 3 is always selected in any IIDness setting, suggesting its role as a central agent (with better data distributions) to improve the overall FL performances.

Finally, Tables 4, 5 and 6 present the live traffic testing results for the IID, non-IID (30%-60%), and non-IID (20%-40%) settings, respectively, in terms of DDoS detection rate, benign traffic detection rate, and identification or not of the attacker(s), and given different types of traffic (benign HTTP and/or TCP attack and/or UDP attack). When only one type of traffic is present (HTTP, TCP, or UDP), all methods perform perfectly in any data IIDness setting, correctly classifying 100% of the traffic and identifying the attacker.

However, differences emerge in mixed traffic scenarios. Under the IID setting, as shown in Table 4, while all methods achieve high detection performances for DDoS attacks, "SAC-based P2P FL" occasionally misclassifies a benign user as an attacker (false positive). In contrast, "Centralized FL" and "SAC-based P2P FL-ASTL" methods

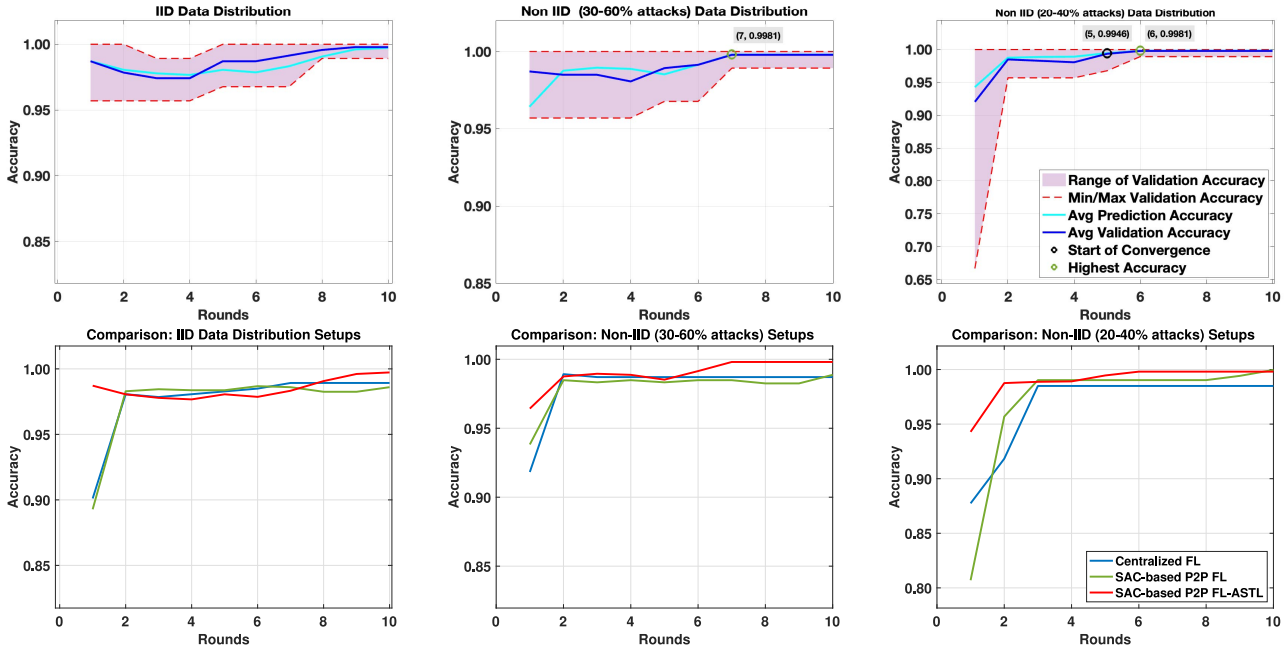

FIGURE 7. Accuracy of: SAC-based P2P FL-ASTL (top row); Several FL methods (bottom row), with different data distributions, using the testbed-collected dataset.

TABLE 4. Live traffic testing results for global models of FL methods (IID scenario).

Scenario	FL Method	Detected DDoS	Detected Benign	Detected Attacker IP Address
Benign Only	Centralized FL, SAC-based P2P FL, SAC-based P2P FL-ASTL	0%	100%	✗
TCP Only	Centralized FL, SAC-based P2P FL, SAC-based P2P FL-ASTL	100%	0%	10.45.1.6
UDP Only	Centralized FL, SAC-based P2P FL, SAC-based P2P FL-ASTL	100%	0%	10.45.1.6
TCP & UDP	Centralized FL, SAC-based P2P FL, SAC-based P2P FL-ASTL	100%	0%	10.45.1.6
Benign & TCP	Centralized FL	86.49%	13.51%	10.45.1.6
	SAC-based P2P FL	86.53%	13.47%	10.45.1.6, 10.45.1.7*
	SAC-based P2P FL-ASTL	87.37%	12.63%	10.45.1.6
Benign & UDP	Centralized FL	86.78%	13.22%	10.45.1.6
	SAC-based P2P FL	86.82%	13.18%	10.45.1.6, 10.45.1.7*
	SAC-based P2P FL-ASTL	86.78%	13.22%	10.45.1.6
Benign & TCP & UDP	Centralized FL	95.02%	3.79%	10.45.1.6
	SAC-based P2P FL	94.30%	5.70%	10.45.1.6, 10.45.1.7*
	SAC-based P2P FL-ASTL	93.98%	6.20%	10.45.1.6

* Misclassified as an attacker (false positive) or as benign user (false negative).

correctly classify all users, demonstrating a better detection performance. In non-IID scenarios with moderate attack distributions (30%-60%), "Centralized FL" and "SAC-based P2P FL" begin to exhibit misclassifications, leading to increased false positives, as shown in Table 5. This issue becomes more pronounced under intense non-IID conditions (20%-40%), as presented in Table 6. Indeed, when the TCP SYN flood attack is mixed with the benign traffic, the benchmarks fail to detect attackers, and they incorrectly classify all users as benign, thus leading to increased false negatives.

TABLE 5. Live traffic testing results for global models of FL methods (Non-IID (30-60% attacks)).

Scenario	FL Method	Detected DDoS	Detected Benign	Detected Attacker IP Address
Benign Only	Centralized FL, SAC-based P2P FL, SAC-based P2P FL-ASTL	0%	100%	✗
TCP Only	Centralized FL, SAC-based P2P FL, SAC-based P2P FL-ASTL	100%	0%	10.45.1.6
UDP Only	Centralized FL, SAC-based P2P FL, SAC-based P2P FL-ASTL	100%	0%	10.45.1.6
TCP & UDP	Centralized FL, SAC-based P2P FL, SAC-based P2P FL-ASTL	100%	0%	10.45.1.6
Benign & TCP	Centralized FL	86.82%	13.18%	10.45.1.6, 10.45.1.7*
	SAC-based P2P FL	86.82%	13.18%	10.45.1.6, 10.45.1.7*
	SAC-based P2P FL-ASTL	86.49%	13.51%	10.45.1.6
Benign & UDP	Centralized FL	86.82%	13.18%	10.45.1.6, 10.45.1.7*
	SAC-based P2P FL	86.82%	13.18%	10.45.1.6, 10.45.1.7*
	SAC-based P2P FL-ASTL	86.78%	13.22%	10.45.1.6
Benign & TCP & UDP	Centralized FL	95.00%	5.00%	10.45.1.6, 10.45.1.7*
	SAC-based P2P FL	96.14%	3.86%	10.45.1.6, 10.45.1.7*
	SAC-based P2P FL-ASTL	94.70%	5.30%	10.45.1.6

* Misclassified as an attacker (false positive) or as benign user (false negative).

TABLE 6. Live traffic testing results for global models of FL methods (Non-IID (20-40% attacks)).

Scenario	FL Method	Detected DDoS	Detected Benign	Detected Attacker IP Address
Benign Only	Centralized FL, SAC-based P2P FL, SAC-based P2P FL-ASTL	0%	100%	✗
TCP Only	Centralized FL, SAC-based P2P FL, SAC-based P2P FL-ASTL	100%	0%	10.45.1.6
UDP Only	Centralized FL, SAC-based P2P FL, SAC-based P2P FL-ASTL	100%	0%	10.45.1.6
TCP & UDP	Centralized FL, SAC-based P2P FL, SAC-based P2P FL-ASTL	100%	0%	10.45.1.6
Benign & TCP	Centralized FL	0%	100%	✗*
	SAC-based P2P FL	0%	100%	✗*
	SAC-based P2P FL-ASTL	86.49%	13.51%	10.45.1.6
Benign & UDP	Centralized FL	86.49%	13.51%	10.45.1.6
	SAC-based P2P FL	86.78%	13.22%	10.45.1.6
	SAC-based P2P FL-ASTL	86.78%	13.22%	10.45.1.6
Benign & TCP & UDP	Centralized FL	36.92%	63.08%	✗*
	SAC-based P2P FL	34.22%	65.78%	✗*
	SAC-based P2P FL-ASTL	94.95%	5.05%	10.45.1.6

* Misclassified as an attacker (false positive) or as benign user (false negative).

In contrast, "SAC-based P2P FL-ASTL" consistently shows robust detection across all scenarios, where it accurately identifies attacker IP addresses, even in challenging non-IID settings. Adaptive agent selection and SAC successfully manage diverse data sources and reduce misclassifications. These findings highlight our proposed method's strength in

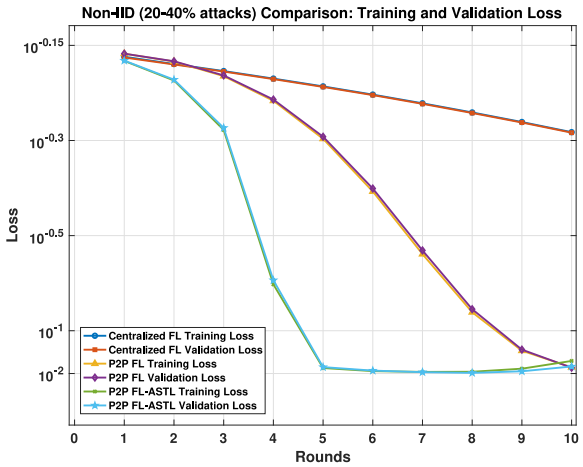

FIGURE 8. Training/Validation loss as a function of FL rounds (Non-IID (20-40% attacks), several FL methods).

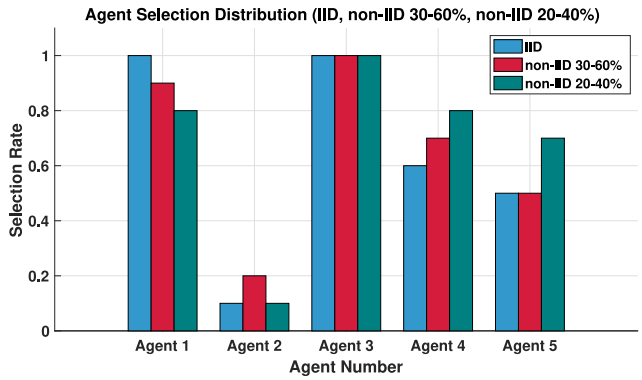

FIGURE 9. Selection rate distribution across $N = 5$ agents in different IIDness settings.

achieving reliable detection in decentralized settings such as 5G O-RAN.

VI. CONCLUSION

In this work, we proposed a novel SAC-based P2P FL-ASTL method adapted for use within the RICs of an O-RAN based 5G network to efficiently detect cyberattacks. Unlike the conventional SAC-based P2P FL, we aim to reduce the communication cost through the integration of two mechanisms, namely agent selection and transfer learning. Agent selection has been developed in a secure manner where only agents presenting high performances, in terms of F1-score and accuracy, are allowed into P2P FL, while transfer learning ensures that all involved FL agents benefit from the SAC-based P2P FL. Our method enhances security in parameter sharing and reduces SAC's computational burden, paving the way for a more secure and streamlined O-RAN in 5G and beyond. Through both simulations and real experiments using our developed 5G O-RAN testbed, we showed that SAC-based P2P FL-ASTL successfully cuts the communication cost by up to 74% compared to the conventional SAC-based P2P FL, while achieving equivalent or higher accuracy than the benchmarks. Our approach has also been proven robust against moderate and intense dataset non-IIDness with a negligible degradation in accuracy (below 2%). In addition, the proposed deployment and led experiments demonstrated the system's efficiency and adaptability in handling real-time non-IID traffic. Specifically, we observed high detection accuracy, efficient model convergence, stable loss around 10^{-2}, and effective handling of live traffic. Indeed, SAC-based P2P FL-ASTL consistently maintained detection rates around 99% for DDoS attacks in mixed traffic scenarios, even under challenging non-IID distributions. The combination of the simulation and experimental results confirmed the suitability of SAC-based P2P FL-ASTL for real-world applications consistently maintaining high detection accuracy and operational efficiency in O-RAN networks.

REFERENCES

[1] K. Ramezanpour and J. Jagannath, "Intelligent zero trust architecture for 5G/6G networks: Principles, challenges, and the role of machine learning in the context of O-RAN," *Comput. Netw.*, vol. 217, Nov. 2022, Art. no. 109358.

[2] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6G wireless communication systems: Applications, requirements, technologies, challenges, and research directions," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 957–975, 2020.

[3] U. Ghafoor, M. Ali, H. Z. Khan, A. M. Siddiqui, and M. Naeem, "NOMA and future 5G & B5G wireless networks: A paradigm," *J. Netw. Comput. Appl.*, vol. 204, Aug. 2022, Art. no. 103413.

[4] A. Afaq, N. Haider, M. Z. Baig, K. S. Khan, M. Imran, and I. Razzak, "Machine learning for 5G security: Architecture, recent advances, and challenges," *Ad Hoc Netw.*, vol. 123, Dec. 2021, Art. no. 102667.

[5] V. Sritapan, D. Massey, and B. Talbot, "5G security evaluation process investigation," Washington, DC, USA, White Paper, May 2022. [Online]. Available: https://www.cisa.gov/sites/default/files/publications/5G_Security_Evaluation_Process_Investigation_508c.pdf

[6] M. Liyanage, A. Braeken, S. Shahabuddin, and P. Ranaweera, "Open RAN security: Challenges and opportunities," *J. Netw. Comput. Appl.*, vol. 214, May 2023, Art. no. 103621.

[7] D. Attanayaka, P. Porambage, M. Liyanage, and M. Ylianttila, "Peer-to-peer federated learning based anomaly detection for open radio access networks," in *Proc. IEEE Int. Conf. Commun.*, 2023, pp. 5464–5470.

[8] A. Abouaomar, A. Taik, A. Filali, and S. Cherkaoui, "Federated deep reinforcement learning for open RAN slicing in 6G networks," *IEEE Commun. Mag.*, vol. 61, no. 2, pp. 126–132, Feb. 2023.

[9] *O-RAN-WG1-O-RAN Architecture Description v01.00.00*, O-RAN Alliance, New York, NY, USA, 2020.

[10] N. Onoszko, G. Karlsson, O. Mogren, and E. L. Zec, "Decentralized federated learning of deep neural networks on non-IID data," 2021, *arXiv:2107.08517*.

[11] F. Alalyan, B. Bousalem, W. Jaafar, and R. Langar, "Secure peer-to-peer federated learning for efficient cyberattacks detection in 5G and beyond networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2024, pp. 1752–1757.

[12] M. Polese, L. Bonati, S. D'oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1376–1411, 2nd Quart., 2023.

[13] C. De Alwis, P. Porambage, K. Dev, T. R. Gadekallu, and M. Liyanage, "A survey on network slicing security: Attacks, challenges, solutions and research directions," *IEEE Commun. Surveys Tuts.*, vol. 26, no. 1, pp. 534–570, 1st Quart., 2024.

[14] B. Mao, J. Liu, Y. Wu, and N. Kato, "Security and privacy on 6G network edge: A survey," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1095–1127, 2nd Quart., 2023.

[15] S. Wijethilaka and M. Liyanage, "A federated learning approach for improving security in network slicing," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, 2022, pp. 915–920.
[16] S. Soltani, M. Shojafar, A. Brighente, M. Conti, and R. Tafazolli, "Poisoning bearer context migration in O-RAN 5G network," *IEEE Wireless Commun. Lett.*, vol. 12, no. 3, pp. 401–405, Mar. 2023.
[17] J. Le, D. Zhang, X. Lei, L. Jiao, K. Zeng, and X. Liao, "Privacy-preserving federated learning with malicious clients and honest-but-curious servers," *IEEE Trans. Info. Forensics Security*, vol. 18, pp. 4329–4344, 2023.
[18] V. Turina, Z. Zhang, F. Esposito, and I. Matta, "Combining split and federated architectures for efficiency and privacy in deep learning," in *Proc. 16th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2020, pp. 562–563.
[19] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "BrainTorrent: A peer-to-peer environment for decentralized federated learning," 2019, *arXiv:1905.06731*.
[20] T. Wink and Z. Nochta, "An approach for peer-to-peer federated learning," in *Proc. Ann. IEEE/IFIP Int. Conf. Depend. Syst. Netw. Workshop (DSN-W)*, Aug. 2021, pp. 150–157.
[21] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, Jan. 2019.
[22] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of FedAvg on non-IID data," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–26.
[23] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Info. Syst. Conf. (MilCIS)*, 2015, pp. 1–6.
[24] *O-RAN Working Group 1 (Use Cases and Overall Architecture), O-RAN Architecture Description*, O-RAN Alliance, New York, NY, USA, 2023.
[25] "Open5GS." Oct. 15, 2024. [Online]. Available: https://open5gs.org/
[26] (O-RAN Alliance, New York, NY, USA). *O-RAN SC Projects*. Oct. 15, 2024. [Online]. Available: https://docs.o-ran-sc.org/en/latest/projects.html#near-realtime-ran-intelligent-controller-ric
[27] "SrsRAN project." SRS. Oct. 15, 2024. [Online]. Available: https://www.srsran.com/5g
[28] "ZeroMQ," Oct. 15, 2024. [Online]. Available: https://zeromq.org/
[29] "Mausezahn." Netsniff-ng. Oct. 15, 2024. [Online]. Available: http://netsniff-ng.org/
[30] A. A. Nafea, M. M. Hamdi, B. saad Abdulhakeem, A. T. Shakir, M. S. I. Alsumaidaie, and A. M. Shaban, "Detection systems for Distributed Denial-of-Service (DDoS) attack based on time series: A review," in *Proc. 21st Int. Multi-Conf. Syst., Signals Devices (SSD)*, 2024, pp. 43–48.
[31] *O-RAN AI/ML Workflow Description and Requirements 1.03, ML Workflow Description Requirements*, vol. 1, O-RAN Alliance, New York, NY, USA, 2021.
[32] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martinez-del Rincon, and D. Siracusa, "LUCID: A practical, lightweight deep learning solution for DDoS attack detection," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 2, pp. 876–889, Jun. 2020.