Peer-to-Peer Federated Learning based Anomaly Detection for Open Radio Access Networks

Dinaj Attanayaka*, Pawani Porambage*†, Madhusanka Liyanage*‡, Mika Ylianttila*
* University of Oulu, Finland
†VTT Technical Research Centre, Oulu, Finland
‡School of Computer Science, University College Dublin, Ireland
Email: *[firstname.lastname]@oulu.fi, † pawani.porambage@vtt.fi, ‡madhusanka@ucd.ie

***Abstract*—Open radio access network (O-RAN) has been recognized as a revolutionized architecture to support the multi-class wireless services required in fifth-generation (5G) and beyond 5G networks. The openness and the distributed nature of the O-RAN architecture have created new forms of threat surfaces than the conventional RAN architecture and require complex anomaly detection mechanisms. Moreover, with the introduction of RAN intelligent controllers (RICs), it is possible to utilize advanced Artificial Intelligence (AI)/ Machine Learning (ML) algorithms based on closed control loops to detect anomalies in a data-driven manner. In this paper, we particularly investigate the use of Federated Learning (FL) for anomaly detection in the O-RAN architecture, which can further preserve data privacy. We propose a peer-to-peer (P2P) FL-based anomaly detection mechanism for the O-RAN architecture and provide a comprehensive analysis of four variants of P2P FL techniques. Moreover, we simulate the proposed models using the UNSW-NB15 dataset.**

***Index Terms*—5G, 6G, Network automation, Security, Privacy, O-RAN, RAN Intelligent controllers, Federated learning**

I. Introduction

With the advent of 5G and beyond 5G (B5G) wireless systems, the radio access network (RAN) infrastructure needs to adhere to new technologies and varying customer requirements, which will increase the capital expenditure (CAPEX) and operational expenditure (OPEX) costs [1]. Since the most significant percentage of the total network cost is accounted for by RAN deployments and operations, the motivation to reduce CAPEX and OPEX has been significantly increased [2]. In Cloud-RAN (C-RAN), the digital processing functional part of the typical base station is moved to a regional cloud or edge data center. However, there is a significantly high communication overhead in the fronthaul link between the data center and radio units for maintaining the required low latency. Virtualized RAN (vRAN) is another approach that virtualizes RAN functions by replacing Baseband Units (BBUs) with Commercial Off-the-Shelf (COTS) hardware. Although these two approaches reduce costs and simplify maintenance overhead, the dependency on a single vendor remains a major disadvantage [3]. To overcome the limitations of C-RAN and V-RAN, a standard Open RAN (O-RAN) solution has been proposed by the O-RAN alliance. This new O-RAN architecture supports multi-vendor interoperability and is based on disaggregated, virtualized, and software-based components connected via open, standardized interfaces [2].

With the increased threat surface due to its inherently open and modular architecture, the risks are significantly higher in O-RAN than in conventional RAN [4]. Anomaly detection is a significant security measure for O-RAN in 5G, which is a large-scale heterogeneous system with varying latency and privacy requirements [5]. Although there is some machine learning (ML) research about anomaly detection in RAN, only a few of them are focused on O-RAN [6]. Federated Learning (FL) is a distributed form of ML that preserves data privacy and improves communication efficiency by training models locally and communicating only the parameters for aggregation [7]. In the O-RAN architecture, when performing AI-based operations in a large-scale, complex environment with sensitive data, FL is more suitable than conventional ML models. FL-based anomaly detection mechanisms are proposed for the Internet of Things (IoT) [8], and Zero-Touch Network and Service Management (ZSM) [9] architecture. Peer-to-peer (P2P) FL is also identified as a good match for detecting anomalies in a complex O-RAN environment [7] due to the hierarchical closed-loop architecture of RICs and data-driven inputs via open interfaces [5].

In this paper, we use P2P FL with secure average computation (SAC) [10] to create a distributed anomaly detection mechanism that can be applied to the O-RAN architecture. P2P FL eliminates the single point of failure, and the parameters are not required to be transmitted to a centralized cloud [11]. In addition, we consider two variations of the P2P FL, one being a clustered P2P FL model where each cluster maintains a separate FL model, which is desired when the data is localized, as in RAN. And the other is a hierarchical version of the mentioned clustered P2P FL model, where almost the same model can be derived in each cluster while maintaining a relatively smaller number of communications in training than the normal P2P FL method. In addition to those, a P2P FL method based on multiparty communication via thresholded fully homomorphic encryption (FHE) is suggested to achieve a higher level of security both in communication as well as parameter average calculation [12]. To the best of our knowledge, this is the first research study that proposes and evaluates a P2P FL-based anomaly detection technique to support intelligent automated security management in the O-RAN architecture. The UNSW-NB15 [13] which is a labeled dataset, is used for training the models, and a pre-separated

testing dataset from the same dataset is used for testing.

The remainder of the paper is organized as follows: Section II outlines the existing literature on O-RAN security and FL-based anomaly detection. Section III describes the proposed anomaly detection mechanisms, and Section IV explains the simulations and evaluation results in detail. Section V discusses the simulation results, and Section VI concludes the paper with future research directions.

II. Background and Related Work

A. O-RAN security

As shown in Figure 1, the architecture proposed by the O-RAN alliance further disaggregates the radio protocol stack processing at Next Generation NodeB (gNB) into three main units, namely the Control Unit (CU), Distributed Unit (DU), and Radio Unit (RU), by utilizing the 3rd Generation Partnership Project (3GPP) New Radio (NR) 7.2x lower layer split [14].

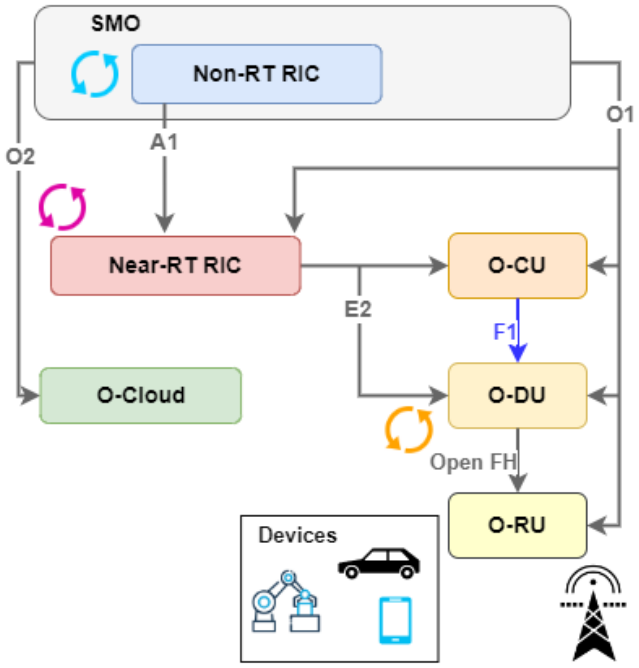


Fig. 1: Logical Architecture of O-RAN [14]

The O-RAN architecture includes two logical controllers called RAN intelligent controllers (RICs). These components, which are based on software-defined networks (SDN), perform particular radio resource management tasks. Near-real-time RIC (Near-RT RIC) operates the control loops with a periodicity between 10 ms and 1 s, and it is the core of the control and optimization of the RAN. The main component of Near-RT RIC is xApps, which are microservices. Non-real-time RIC (Non-RT RIC) is a part of the Service Management and Orchestration (SMO) framework and operates on control loops larger than 1 s. Near-RT RIC utilizes micro-services called rApps, to provide value-added services to support and facilitate the RAN optimizations and operations [14]. Due to the network infrastructure's data streams via open interfaces, RICs have a centralized and abstract perspective on the network. AI and ML techniques are used for the selection and implementation of control rules and actions in O-RAN.

RICs can be deployed in any of the cloud locations, such as the core cloud, regional cloud, or edge cloud. The RIC platform can be used to deploy external RAN control applications created by outside vendors. Compared to earlier proprietary RAN systems, the O-RAN architecture has a substantial benefit because these third-party apps can incorporate many types of cutting-edge RAN control algorithms [15].

Despite the benefits of O-RAN architecture, the new architectural changes significantly alter the RAN attack surface due to additional functions, interfaces, decoupling, virtualization, and the use of open-source codes [16]. The effect of diversity of user equipment (UE), diversity of third-party applications, and the integration of AI and ML has to be analyzed in terms of security threats [3]. The O-RAN specific or general potential vulnerabilities can be exploited through attacks against confidentiality, integrity, and availability. As stated in [16] O-RAN threats can be grouped into seven categories: threats against the O-RAN system, threats against O-cloud, threats against the ML system, threats against 5G radio networks, threats to open source code, and physical threats.

B. FL-based anomaly detection

When performing AI and ML-based operations, data privacy is a major concern. FL is a distributed ML technique introduced by Google to perform privacy-preserving model training [17]. At each trainer location in the federation, a local model is trained using the training dataset, which is retained locally. Only the model parameters are sent to the central aggregator site after the local model training. The aggregator uses the received parameters to create a common global model and returns it to the local trainers. Since actual data is not explicitly accessed or shared, each local trainer can benefit from the datasets of other local trainers without affecting privacy and while reducing the communication cost [18].

As mentioned before, the centralized FL method has some considerable drawbacks, such as a single point of failure due to the central aggregator and the imbalance of data distributions in different local trainers [11]. Peer-to-peer (P2P) FL is a novel technique that eliminates the need for a centralized aggregator. There are several proposed techniques in the current state-of-the-art for P2P FL model training. One is Split Learning (SL), which splits the model into two parts, server and node, for training. The training process is relatively slower when multiple nodes are involved. However, SL methods are difficult to parallelize, and they should be running sequentially by design [19]. BrainTorrent is a decentralized P2P FL environment [20]. At any given training round, only one participant updates its local model weights by taking into account its prior weights and model weights received from peer models that are apparently newer. This may introduce a risk that malicious or semi-honest participants are undetectable. In [10], a variant of the P2P FL method is proposed based on secure average computation (SAC), which utilizes an n-out-of-n secret parti-

tioning method and average calculation technique to mitigate the effects of semi-honest participants. However, this model is not optimal when there is a larger number of trainers since the communication cost is significantly higher.

Both Near-RT RIC and Non-RT RIC in the O-RAN architecture can host the AI/ML training by using xApps and rApps respectively. Since RICs are deployed in a hierarchical manner, there is a good motivation to use FL as a ML tool to perform the automated operations. As shown in Figure 2, the Near-RT RIC resides in the regional or central cloud and can act as the central aggregator, whereas Non-RT RICs are located in edge or regional clouds and can serve as distributed local trainers [21]. In P2P FL, only the Near-RT RICs do the training and can communicate via inter-regional or inter-edge cloud connections. The models can be deployed as xApps or within a xApp instance, and deployment can be done image-based or file-based [21].

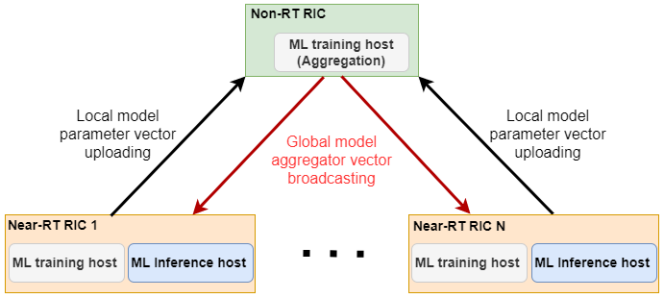


Fig. 2: Federated learning among Non-RT RIC and Near-RT RICs

By default, the communication channel between the UE and the remote radio head unit has not changed significantly from the conventional RAN to O-RAN with respect to the utilized radio frequency (RF). Therefore, O-RAN is vulnerable to some known RF attacks similar to those against conventional RAN. Furthermore, the diversity and pervasiveness of UE types are expanded in O-RAN, increasing the chance that the attack surface will be a target of new threats. There is the risk of Denial of service (DoS) and Distributed DoS (DDoS) attacks on cellular service and control plane, and also DDoS flooding attacks on network/transport layers [3]. Therefore, anomaly detection is an important security measure to be considered in O-RAN. Particularly, a data-driven FL technique suits well when considering a complex system like a RAN in 5G architecture, where there are many ways to cause anomalies.

Going through all the related work discussed above, in the current state-of-the-art, there is no work presented with P2P FL in anomaly detection. In general, FL is also not particularly used for securing O-RAN architecture and its automated security management. Due to the distributed nature and openness of O-RAN architecture, we concluded from the current work that FL-based algorithms are more appropriate for O-RAN architecture. Moreover, we identify that typically FL-based algorithms are applicable for anomaly detection in hierarchical networks whereas P2P FL algorithms are more resilient to a single point of failure. Therefore, we propose our solution of using P2P FL in O-RAN to identify anomalies so that the attacks can be prevented before propagating to the core network. The use of FL also significantly protects data privacy and maximizes communication efficiency.

III. P2P FL based Anomaly Detection Mechanism for O-RAN

In this section, we comprehensively describe how P2P FL can be applied for anomaly detection in the O-RAN architecture and four variants of P2P FL models that can be mapped with the proposed anomaly detection mechanism.

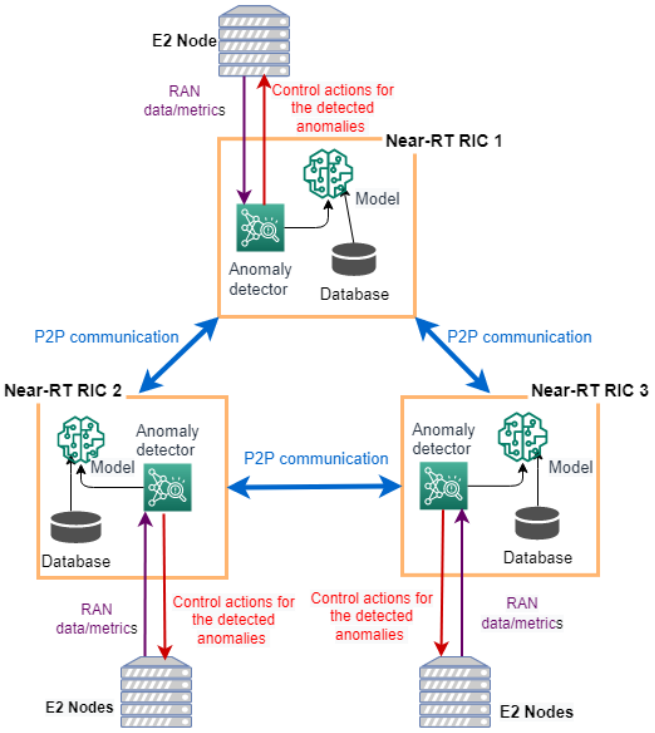


Fig. 3: Proposed anomaly detection mechanism for O-RAN architecture.

The proposed P2P FL anomaly detection model for O-RAN architecture is illustrated in Figure 3. Accordingly, the local trainers of the FL model are hosted at Near-RT RICs, which may reside in the edge clouds, whereas the P2P communication may occur via inter-edge cloud connections. The training model and the detector can be deployed as dedicated xApps, or they can be parts of the same xApp, which can include other functions. These deployments can be image-based or file-based [21]. Moreover, when an anomaly is detected, the detector or the additional xApp hosted at the Near-RT RIC can transmit control actions to the relevant CUs and DUs via the E2 interface. The network flow data stored in the database can be used for model training. After the model is trained, it can be utilized by the anomaly detector in the same Near-RT RIC. Then the security control actions related to the detected anomalies are communicated to the E2 nodes.

Following are the four variants of P2P FL models that can be mapped with the same anomaly detection mechanism:

A. Model 1: Normal P2P FL

The first normal P2P FL model is derived from [10] and designed by locating local models in the Near-RT RICs. As demonstrated in Figure 4, every trainer communicates with every other trainer to calculate average model weights using secure average computation.

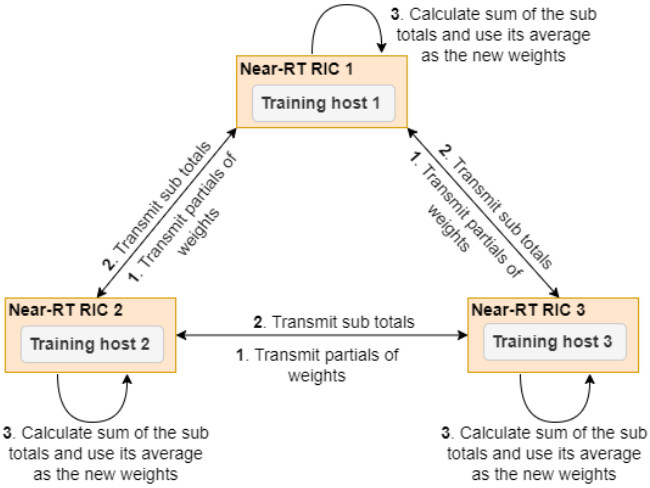


Fig. 4: Secure weight average computation in P2P FL.

When a local trainer trains its model, it randomly partitions the model parameter into partial weights equal to the number of local trainers in the network ($w_i = \sum_{k \in Trainers} w_{ik}$). Then each trainer keeps one partial weight and transmits the rest of the partial weights to other local trainers, one per trainer. Thereafter, the local subtotal is calculated by each trainer with its own and the received partial weights ($t_i^{partial} = \sum_{j \in Trainers} w_{ji}$). One full round is completed after computing the new weight, which is derived as the average of the total previously calculated weights (Equation 1).

$$w_{avg} = \frac{\sum_{i \in Trainers} t_i^{partial}}{\text{total number of trainers}} \tag{1}$$

B. Model 2: Clustered P2P FL

In actual RAN deployments, it is highly likely to encounter unbalanced data and localization (in edge or regional clouds). Hence parameter averaging between all Near-RT RICs in a massive base station deployment will cost higher. In such a scenario, a clustered model is more appropriate, where the clients in the same cluster share the parameters for averaging. Each cluster may have different FL models where the local trainers can be clustered using location-based K-means clustering. The communication in this environment is shown in Figure 5, and within each cluster, the averaging steps will be the same as in Figure 4.

C. Model 3: Hierarchical P2P FL

In Model 3, we present a hierarchical clustered model where a master trainer is selected for each cluster. The communication in this environment is shown in Figure 6 and the averaging steps within the cluster are similar to Figure 4.

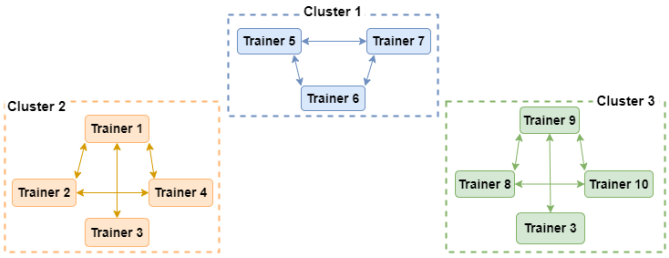


Fig. 5: Parameter sharing of Clustered P2P FL.

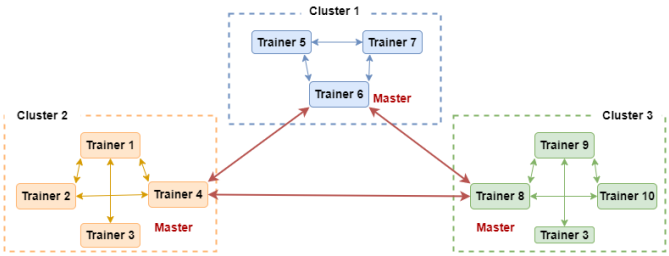


Fig. 6: Parameter sharing of Hierarchical P2P FL.

The selection of the master trainer is performed based on resource availability. After a fixed number of global rounds, parameter values are shared and averaged between masters of each cluster. Here the ratios correspond to the number of clients in each cluster, as shown in Equation 2. Then, the master client will share those average values with other clients in its own cluster. This may increase the probability of having the same model in each cluster while maintaining a relatively smaller number of communications compared to Model 1, the normal P2P FL model.

$$w_{avg}^{master} = \sum_{k \in Clusters} \frac{\text{number of trainers in k}}{\text{total number of trainers}} w^{k^{th} master} \tag{2}$$

D. Model 4: Homomorphic P2P FL

In Model 4, we consider a parameter averaging method with secure communication and the threshold Fully Hormomophic Encryption (FHE) method as explained in [12]. A key generation protocol can be used to generate a common public key (P_k) and secret shares of the private key (S_k), and each trainer receives P_k and a share of S_k. Thereby, each trainer can encrypt; nevertheless nobody can decrypt without the consent of other trainers. Hence, for parameter averaging in P2P FL, each trainer can encrypt parameter values and share them with each other. Then each trainer performs homomorphic addition to compute the encrypted total values, followed by partial decryption using an individual secret key share. After that, these partial decryption values are shared with each other, and the final decryption of the total is performed at each trainer, and the average can be calculated. As illustrated in Figure 7, this provides secure communication of parameter values and protection against semi-honest trainers in the network.

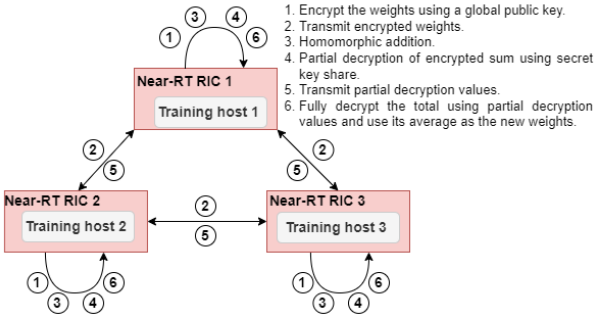


Fig. 7: Weight average computation using FHE.

IV. Simulation and Results

In this section, we present the simulated results in a concise manner. The three models based on SAC (Model 1, Model 2, and Model 3) are extensively compared, and the performance of Model 4 is observed in terms of accuracy. For comparison purposes, a Centralized FL model and a General ML model (where each trainer's model is trained separately) are simulated. The UNSW-NB15 dataset is selected for model training and testing. This data set consists of nine types of attacks, such as DoS, backdoors, and worms, along with normal network traffic [13]. Out of all the features, 42 are selected for the model training.

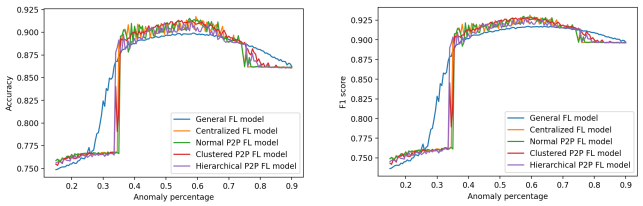
For the simulations, a common Multilayer Perceptron (MLP) model, which is a fully connected feed-forward Artificial neural network (ANN) was used. It had four layers, including two hidden layers. TensorFlow libraries were used for the simulations. An independent and identically distributed (IID) data distribution was considered, where each trainer has a training dataset with the same number of anomalies.

First, we simulated the behavior of the P2P SAC-based models when the training anomaly percentage is varied while keeping all the other parameters fixed, as shown in Table I. Figure 8 illustrates how the overall accuracy and F1-score of each method behave when the training anomaly percentage is varied.

TABLE I: Simulation parameters for varying training anomaly percentage.

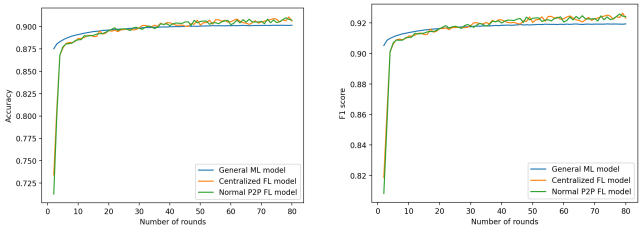
Parameter	Value
Number of trainers	100
Number of clusters	5
Number of epochs	10
Number of rounds	50
Batch size	100
Training sample size	100000
Testing sample size	10000
Testing anomaly percentage	60%

Then the number of training rounds was varied while other parameters were kept the same, as depicted in Table II. Figure 9 shows how accuracy and the F1-score behave when the number of training rounds is varied in the Normal P2P model compared to the reference Centralized FL model and General ML model. Moreover, the accuracy and F1-score of the three SAC-based P2P FL models are illustrated in Figure 10. The



(a) Accuracy vs Anomaly percentage. (b) F1-score vs Anomaly percentage.

Fig. 8: Accuracy with varying anomaly percentage.



(a) Accuracy vs Number of rounds. (b) F1-score vs Number of rounds.

Fig. 9: Performance of the Normal P2P model vs Number of rounds.

accuracy of the Clustered and Hierarchical P2P FL models is compared in Figure 11. Moreover, as illustrated in Figure 12, the transmission (Tx) communication costs of considered FL methods in model training are compared.

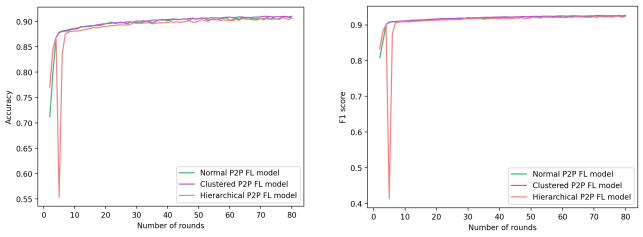
TABLE II: Fixed parameters for the simulations with varying training rounds.

Parameter	Value
Number of trainers	100
Number of clusters	5
Number of epochs	10
Batch size	100
Training sample size	150000
Training anomaly percentage	60%
Testing sample size	10000
Testing anomaly percentage	60%
Cluster anomaly proportions	[0.6, 0.5, 0.4, 0.7, 0.6]

Finally, Homomorphic P2P FL method performance is compared with Centralized FL and Normal P2P FL method as shown in Figure 13.

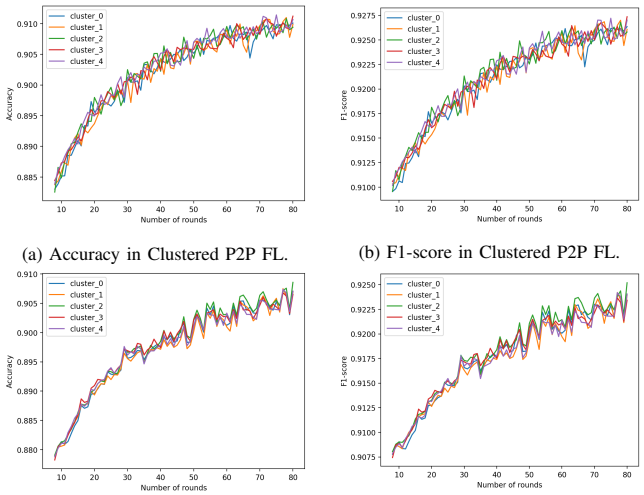
V. Discussion

When the number of anomalies is varied, the maximum accuracy and F1 score values are achieved when the training anomaly percentage is around 60% in all models. As depicted



(a) Accuracy vs Number of rounds. (b) F1-score vs Number of rounds.

Fig. 10: Performance of the SAC-based P2P FL models vs Number of rounds.



(a) Accuracy in Clustered P2P FL.

(b) F1-score in Clustered P2P FL.

(c) Accuracy in Hierarchical P2P FL.

(d) F1-score in Hierarchical P2P FL.

Fig. 11: Performance of Clustered and Hierarchical P2P Fl models with varying Number of rounds.

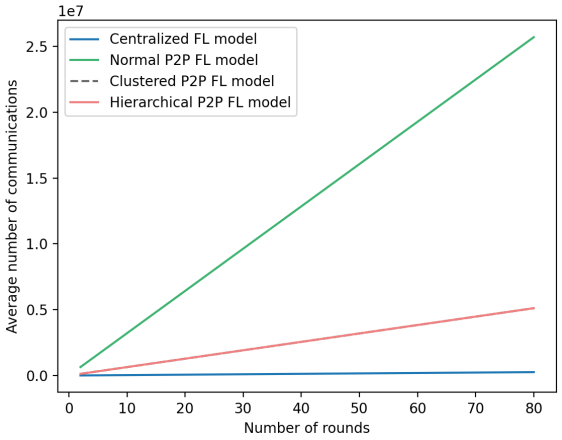
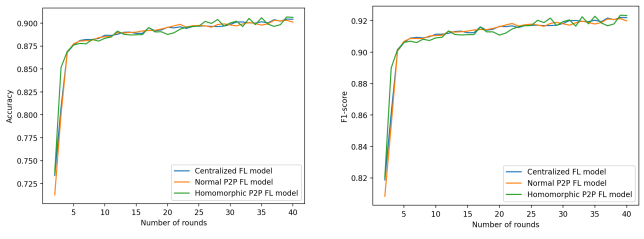


Fig. 12: Comparison of Tx communication costs vs Number of rounds.



(a) Accuracy vs Number of rounds.

(b) F1 score vs Number of rounds.

Fig. 13: Homomorphic P2P FL model performance with varying Number of rounds.

in Figure 8, when the anomaly percentage is considerably lower, training is not accurate because there are not enough anomalies. When the anomaly percentage is quite high, the normal data captured in the training set is not enough. Hence, there can be over-fitting, which leads to somewhat decreased performance.

For the varying training rounds, the Centralized FL model and the Normal P2P FL model have similar accuracy and F1 score values, which is expected. As illustrated in Figure 9, the accuracy of these two FL models is higher than that of general ML model training after about 45 training rounds. After 80 rounds, the accuracy of the Normal P2P FL model is 90.8% which is about 0.5% greater than the General ML model.

When the three types of P2P models are compared, the average accuracy and F1 curves of the Clustered P2P FL model are similar to the Normal P2P model. However, the Hierarchical P2P FL model's performance is relatively worse. Nevertheless, due to parameter averaging between the clusters, the clusters have quite similar accuracy and F1-score values in the hierarchical FL model scenario compared to the clustered FL model, which can be observed in Figure 11.

As shown in Figure 12, the number of transmissions for training gradually increases with the number of rounds in all FL methods. However, the total number of Tx communications in a particular round is significantly high in the Normal P2P FL model, which is (number of clients - 1) times more than the centralized FL model. This is due to the SAC method used for averaging, which provides protection against semi-honest clients. However, Clustered and Hierarchical P2P FL models have about five times smaller total communication cost than the Normal P2P FL model due to clustering.

As shown in Figure 13, the Homomorphic P2P FL model performs the same as the Centralized FL and Normal P2P FL models. However, some degradation of performance can be observed due to higher precision errors in the averaging process due to encryption and decryption. Moreover, the computation cost is significantly higher in Homomorphic P2P FL. However, this performance penalty can be neglected because of the additional security it offers in terms of secure communication and security against semi-honest clients.

When comparing SAC and HE-based average computation, a semi-honest client receives a partial weight value of an honest client in the SAC-based P2P FL methods. If the actual weight value is negative, then the partial weight values must be negative, and vice versa. Therefore, the semi-honest client can reduce the search range by half, unlike the Homomorphic P2P FL method.

VI. Future Directions and Conclusion

FL is a distributed ML technique that improves privacy as well as communication efficiency. P2P FL is a novel variation of FL and is more suitable for complex systems like O-RAN in 5G. In this paper, we present a Normal P2P FL model, a Clustered P2P model, and a Hierarchical clustered P2P FL model with SAC to detect anomalies in the O-RAN architecture. Furthermore, a more secure approach to

P2P FL training called the Homomorphic P2P FL model is proposed, where FHE with secret key sharing is used for average computation. The UNSW-NB15 networking dataset was used for the simulation of the mentioned models. It is visible from the results that the accuracy and F1-score values are higher in FL methods compared to the general ML method, and P2P FL achieved similar performance as the centralized FL method. However, there is a penalty of high communication costs when using P2P FL instead of centralized FL.

In future work, we expect to increase the accuracy and F1 score of the proposed models and use more advanced federated reinforcement learning to support the online training of the models. Moreover, we plan to simulate different sub-optimal scenarios in training, such as offline trainers and semi-honest trainers, and finally implement the models in the O-RAN architecture.

ACKNOWLEDGMENT

This work is partly supported by VTT Technical Research Centre of Finland and by Business Finland in SUNSET-6G, European Union in SPATIAL (Grant No: 101021808), Academy of Finland in 6Genesis (grant no. 318927) and Science Foundation Ireland under CONNECT phase 2 (Grant no. 13/RC/2077_P2) projects.

REFERENCES

[1] C. de Alwis, Q.-V. Pham, and M. Liyanage, *6G Frontiers: Towards Future Wireless Systems*. John Wiley & Sons, 2022.

[2] B. Brik, K. Boutiba, and A. Ksentini, "Deep Learning for B5G Open Radio Access Network: Evolution, Survey, Case Studies, and Challenges," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 228–250, 2022.

[3] D. Mimran, R. Bitton, Y. Kfir, E. Klevansky, O. Brodt, H. Lehmann, Y. Elovici, and A. Shabtai, "Security of Open Radio Access Networks," *Computers & Security*, vol. 122, p. 102890, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016740482200284X

[4] M. Liyanage, A. Braeken, S. Shahabuddin, and P. Ranaweera, "Open RAN Security: Challenges and Opportunities," *arXiv preprint arXiv:2212.01510*, 2022.

[5] P. H. Masur, J. H. Reed, and N. K. Tripathi, "Artificial Intelligence in Open-Radio Access Network," *IEEE Aerospace and Electronic Systems Magazine*, vol. 37, no. 9, pp. 6–15, 2022.

[6] Y. Yuan, J. Yang, R. Duan, I. Chih-Lin, and J. Huang, "Anomaly Detection and Root Cause Analysis Enabled by Artificial Intelligence," in *2020 IEEE Globecom Workshops (GC Wkshps)*, 2020, pp. 1–6.

[7] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6G communications: Challenges, methods, and future directions," *China Communications*, vol. 17, no. 9, pp. 105–118, sep 2020. [Online]. Available: https://doi.org/10.23919%2Fjcc.2020.09.009

[8] Y. Liu, N. Kumar, Z. Xiong, W. Y. B. Lim, J. Kang, and D. Niyato, "Communication-Efficient Federated Learning for Anomaly Detection in Industrial Internet of Things," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[9] S. Jayasinghe, Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila, "Federated Learning based Anomaly Detection as an Enabler for Securing Network and Service Management Automation in Beyond 5G Networks," in *Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, 2022, pp. 345–350.

[10] T. Wink and Z. Nochta, "An Approach for Peer-to-Peer Federated Learning," in *51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2021, pp. 150–157.

[11] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[12] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, 2012, pp. 1219–1234.

[13] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[14] O-RAN ALLIANCE, "O-RAN Architecture Description," O-RAN.WG1.O-RAN-Architecture-Description-v06.00, 2022.

[15] A. Garcia-Saavedra and X. Costa-Pérez, "O-RAN: Disrupting the Virtualized RAN Ecosystem," *IEEE Communications Standards Magazine*, vol. 5, no. 4, pp. 96–103, 2021.

[16] O-RAN ALLIANCE, "O-RAN Security Threat Modeling and Remediation Analysis," O-RAN.SFG.Threat-Model-v02.01, 2022.

[17] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Singh and J. Zhu, Eds., vol. 54. PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[18] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: Motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[19] V. Turina, Z. Zhang, F. Esposito, and I. Matta, "Combining split and federated architectures for efficiency and privacy in deep learning," in *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, 2020, pp. 562–563.

[20] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "Braintorrent: A peer-to-peer environment for decentralized federated learning," *arXiv preprint arXiv:1905.06731*, 2019.

[21] O-RAN ALLIANCE, "AI/ML workflow description and requirements," O-RAN.WG2.AIML-v01.03, 2021.