An Efficient Intrusion Detection Solution for Near-Real-Time Open-RAN

Emmanuel N. Amachaghi, Sulyman Abdulkareem, Sotiris Chatzimiltis, Mohammad Shojafar and Chuan H. Foh
*5G/6G Innovation Centre, Institute for Communication Systems, University of Surrey, Guildford, UK

2024 IEEE Symposium on Computers and Communications (ISCC) | 979-8-3503-5423-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISCC61673.2024.10733615

***Abstract*—The rapid adoption of Open Radio Access Network (Open-RAN) architectures has brought unprecedented innovation opportunities in modern telecommunications networks. However, this evolution also introduces novel security challenges, particularly in demanding scenarios where swift decision-making is critical. In this paper, we conduct an in-depth investigation into model poisoning attacks in ensemble learning, highlighting their implications for network security, and provide a detailed demonstration of our proposed Open-RAN Intrusion Detection System (IDS), which is seamlessly incorporated into the security module of the near Real-Time RAN Intelligent Controller (nearRT-RIC). The strategic placement of the IDS within the nearRT-RIC ensures its operation within the demanding 10 ms to 1 second control loop range, enabling nearRT intrusion detection capabilities. Through rigorous evaluation and experimentation, our solution showcases promising results in enhancing network security without compromising performance.**

***Index Terms*—Open Radio Access Network, Machine Learning, Ensemble Learning, Intrusion Detection System.**

I. Introduction

The shift towards Open Radio Access Network (Open-RAN) architecture is transforming modern telecommunications networks by offering disaggregated, and interoperable components, fostering innovation, flexibility, and cost-effectiveness compared to traditional monolithic infrastructures. Open-RAN introduces open interfaces (such as A1, O1, O2, E1) and standards, including open Fronthaul and Midhaul for interconnecting different sections of the RAN, alongside Open-RAN components such as Centralized Unit (CU), Distributed Unit (DU), Radio Unit (RU), Service Management and Orchestration (SMO), Non-Real-Time (Non-RT) RIC, and nearRT RIC. These features empower network operators to freely combine hardware and software components sourced from various vendors. However, this flexibility introduces new challenges, especially regarding security vulnerabilities and threats, particularly in nearRT scenarios. Potential threats such as denial-of-service (DoS) attacks, malware infiltration, supply chain attacks, or unauthorized access may go unnoticed, allowing attackers to exploit vulnerabilities and compromise network resources. The absence of an efficient detection system exacerbates the impact of security incidents, leading to prolonged downtimes and service disruptions. In a cyberattack or intrusion, the lack of real-time detection and response mechanisms hinders the timely containment and mitigation of the threat. As a result, critical network services may be compromised, causing financial losses, reputational damage, and regulatory compliance issues. Further, without proactive security measures, Open-RAN operators face heightened operational risks and liabilities, jeopardizing their ability to deliver reliable and resilient telecommunications services. Functioning in near real-time, the nearRT RIC can dynamically adjust to changing network conditions by managing resource allocation, enhancing real-time decision-making, load distribution, spectrum utilization, and interference mitigation [1]. Interacting directly with base stations and radio units, it swiftly adapts to optimize network performance while ensuring efficient resource utilization and meeting quality of service (QoS) standards for both services and users. Moreover, through standardized open interfaces such as the E2 interface with O-CU-CP, O-CU-UP, and O-DU, the nearRT RIC establishes connections to facilitate seamless communication. It also interfaces with the Non-RT RIC and the Service Management and Orchestration (SMO) framework via the A1 and O1 interfaces, enabling efficient collaboration and coordination across the network ecosystem. The A1 interface is a vital link between the Non-RT RIC and nearRT RIC, facilitating policy-driven directives and accommodating AI/ML workflows for nearRT RIC applications. Hoffmann et al. [2] proposed leveraging xApps (nearRT Apps) to detect signaling storms during device registration procedures, utilizing the Open-RAN Architecture to combat Signaling Storm Attacks (SSA) by capturing and analyzing network messages and statistics in real-time. Furthermore, the collaboration between Non-RT RIC and nearRT RIC enables the optimization and refinement of intelligent Artificial Intelligence / Machine Learning (AI/ML) algorithms, spanning areas such as load balancing, mobility management, multi-connection control, QoS management, and network energy conservation [3]. Thus, there is a critical need for tailored intrusion detection solutions around nearRT Open-RAN environments, leveraging machine learning algorithms, programmability or Software Defined Networking (SDN), anomaly detection techniques, and intelligent traffic analysis to identify and mitigate intrusions while minimizing latency and computational overhead. SDN security enables centralized control for efficient policy enforcement, dynamic adaptation to threats through programmability, and segmen-

tation to reduce attack surfaces, along with support for dynamic reconfigurability to support moving target defense [4], [5], enhanced visibility and monitoring capabilities and robust authentication and authorization mechanisms.

Motivation: This paper addresses the growing security concerns of Ensemble Learning (EL). Thus, we address the following open issues:

- The effect of randomness on aggregation function selection during the FL process needs to be investigated. Also, it is necessary to answer how introducing randomness in the aggregation process can influence the success and efficacy of model poisoning attacks. By exploring this issue, the paper provides insights into the potential advantages and limitations of employing a randomized aggregation function in FL.
- How adversaries can adapt their attack techniques to account for the uncertainty introduced by the server's random aggregation function selection like [6]. The paper presents three distinct adversaries, each with well-defined strategies targeting specific aggregation rules (Krum and Trimmed Mean). Additionally, it introduces a third adversary employing a randomized strategy, alternating between the two aggregation rules.
- How the EL system's robustness is affected by introducing randomness in the aggregation function selection process. Through extensive experiments and evaluations, the paper assesses the performance of each adversary and measures their impact on the overall accuracy of the federated network.

Contributions: The main contributions of this paper include a comprehensive investigation of model poisoning attacks in EL in an Open-RAN environment.

- We design IDS in a NearRT RIC of Open-RAN standardized Oran Alliance architecture with a comprehensive analysis of IDS security. This architecture offers readers valuable insights into the path to a secure software schematic in Open-RAN.
- We utilize EL and several classical classification learning techniques to create enhanced IDS. This represents an efficient, comprehensive, secured, intelligent solution over Open-RAN.
- We used several KPIs to comprehensively assess and compare the performance of the proposed ML techniques. The algorithms were evaluated for their effectiveness using metrics such as accuracy and F1-score.

Roadmap: The paper is organized as follows. Section II overviews existing literature on ML and model poisoning attacks and defense mechanisms. Section III presents the proposed Open-RAN IDS and proposed EL algorithm. We give our experiments and Evaluation in Section IV. Section V interprets the experimental findings and provides insights from our study. Finally, we summarize the achievements in Section VI.

II. Related Work

The existing research on anomaly detection within the Open-RAN architecture has employed various approaches, including ML methods, such as Multilayer Perceptron (MP), Decision Tree (DT), and Support Vector Machine (SVM). To learn various traits, these techniques classify Open-RAN datasets containing diverse Key Performance Indicators (KPIs). The study in [7] demonstrated the efficacy of machine learning techniques in detecting anomalies, showcasing adaptability to different strategies within the Open-RAN architecture. By leveraging ML models and methods, operators and manufacturers can bolster performance and adaptability in the dynamic landscape of 5G networks. [8] proposed a Peer-to-Peer Federated Learning (P2P FL) mechanism for anomaly detection in Open-RAN architecture by training models locally at nearRT RICs and communicating only parameter updates, ensuring data privacy and communication efficiency. It eliminates single points of failure, enhancing network security and demonstrates resilience in hierarchical networks like Open-RAN. This approach is efficient in communication and computation, reducing overhead while maintaining data privacy, and is well-suited for detecting anomalies in complex Open-RAN environments. The discourse on Bearer Migration Poisoning (BMP) within the Open-RAN architecture elucidates the intricacies of this attack vector, its operational mechanics, and its detrimental impact on network performance. Quantitative analyses revealed that BMP attacks could precipitate severe drops in throughput and significant spikes in packet loss rates, thereby compromising user experience [9]. Proposed countermeasures aim to detect and mitigate the risks associated with user service disruption, underscoring the pivotal role of Open-RAN security and the necessity for further research in this domain.

Further into this realm, the susceptibility to Man-in-the-Middle attacks on communication interfaces between network controllers has garnered significant attention. By elucidating the ramifications of such attacks, including data leakage and service denial, researchers advocate for robust security measures, emphasizing adopting a Security-by-Design approach to fortify the Open-RAN architecture against potential vulnerabilities. Addressing the specter of jamming attacks in 5G networks [10] study presents a statistical methodology for detecting downlink jamming utilizing user equipment-generated link quality reports. The study emphasized the suitability of Open-RAN architecture for thwarting jamming attacks owing to its interface openness and analytical capabilities. The efficacy of the proposed detection method underscores its potential implementation within Open-RAN architecture, thereby bolstering network resilience against adversarial activities.

III. Proposed Approaches for Ensemble Learnings

This section presents our proposed architecture and details the proposed EL algorithm over the considered education

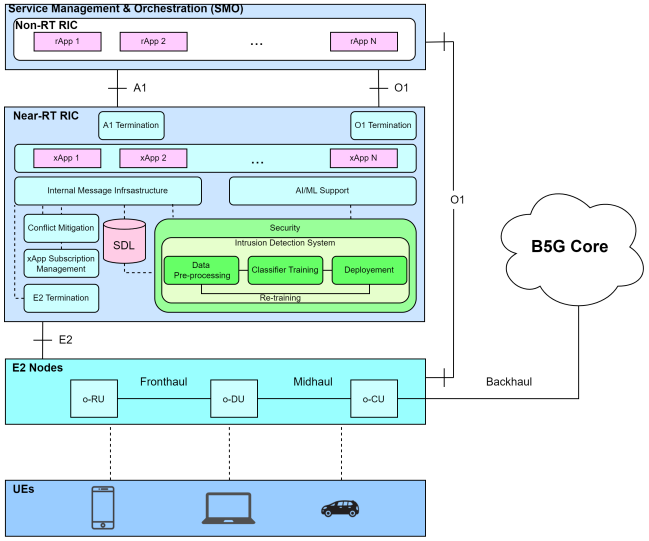

Fig. 1. Proposed Open-RAN IDS Architecture in the nearRT-RIC. SDL: Shared Data Layer; B5G: Beyond 5G.

technique.

A. Proposed Architecture

In Fig. 1, we present a comprehensive illustration of the proposed Open-RAN IDS, seamlessly integrated into the Security module within the near Real-Time RAN Intelligent Controller (nearRT-RIC). Introducing control loops within the Open-RAN environment allows a granular control and management of Open-RAN components. The strategic placement within the nearRT-RIC is intentional, ensuring the IDS operates within the demanding 10ms to 1-second control loop range, allowing nearRT intrusion detection. Data traffic originating from User Equipment (UEs) is captured and stored in the Shared Data Layer (SDL) for subsequent utilization.

Utilizing the APIs provided by the SDL, the IDS efficiently gathers the required data crucial for training purposes. After data gathering, a pre-processing stage ensures the data assumes the appropriate form for further analysis. The AI/ML Support module trains a supervised ensemble learning classifier for detecting network intrusions. An instance of the trained model is then deployed to the system, reinforcing it against potential security threats. Through the nearRT-RIC's assistance, network intrusions from malicious User Equipment (UEs) can be identified, and mitigation strategies can be introduced for quicker response and resolution.

B. FI Dimensionality Reduction Technique

Handling high-dimensional data, commonly known as the curse of dimensionality, is difficult in practice. If the dimensionality of the input dataset increases, any ML classifier will become more complex. As the number of features increases, the chance of overfitting also increases. Hence, reducing the number of features is often required, which can be done with dimensionality reduction. Feature Importance (FI) is a variant of the filter-based dimensionality reduction technique. It uses *Extra Tree Classifier* (ETC) to aggregate feature classification outcomes from multiple de-correlated decision trees derived from a forest. The decision tree construction process employed by the ETC utilizes the entirety of the original features of the dataset. The tree received by each test node comprises a randomly selected subset of features from a more extensive feature set. Each decision tree selects the optimum feature for data partitioning based on the Information Gain and Entropy formulas (1) and (2). Fig. 2 illustrates the dataset features and their importance scores.

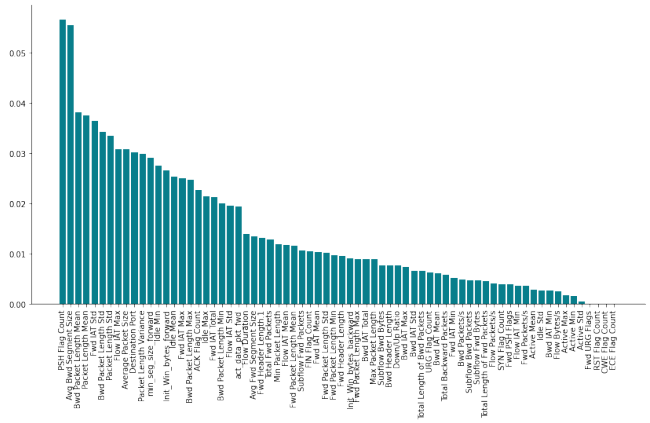

Fig. 2. Important Features of the TU-ML5G-PS-006 dataset.

$$Gain(E, A) = Entropy\,(E) - \sum_{v \in Values(A)} \frac{|Ev|}{E} Entropy\,(Ev) \tag{1}$$

and

$$Entropy\,(E) = \sum_{i=1}^{c} p_i log_2\,(p_i) \tag{2}$$

where E is the entropy value and A is the dataset feature. The quantity c represents the number of category labels, and p_i signifies the proportion of samples within category i.

C. Ensemble Learning

A Stack Ensemble Learner (SEL) classifier can be developed using the same (homogeneous) or dissimilar (heterogeneous) base and meta-learning classifiers. The fundamental distinction between stacking and other ensembles is that the final classification is based on meta-level learning. The basic principle of an ensemble learning classifier is to amalgamate multiple weak learners to construct a single robust learner [11]. The SEL framework utilizes a hierarchical system of classification, where

the initial classification of instances is performed using base learners at the first tier. Once the outputs of the base learners have been predicted, the meta-learner learns them during its training process. Upon rectifying the losses incurred by the base learners, the second-tier classifier (meta-learner) generates the final classification [12]. In SEL, meta-learning classifies using the default dataset features and the predictions of the base learners. One can view the base learner as an *expert*. Then, the meta-learning will have the advantage of accessing *expert opinions* in its learning process, which is the critical factor leading to its more optimal performance. The SEL classifier can be built by implementing the following procedure:

- The whole training and test set of the ITU-ML5G-PS-006 dataset is imported into the experiment environment.
- Using the training set, train k number of base learners WC. Upon completion of the training cycle, k additional features will be incorporated into the training dataset to denote the classification predictions of the k base learners (i.e., expert opinions).
- The process of generating optimal final predictions on test data involves the meta-learner utilizing the Level 1 prediction and learning insight on the most effective approach to combine the predictions of the base classifiers that underlie it.

Algorithm 1 Stack Ensemble Learning

Base-Learning Step
Require: The Whole TU-ML5G-PS-006 Dataset $D = \{\mathbf{x}_i, y_i\}_{i=1}^{N}\ \mathbf{x}_i \in X, y_i \in Y$
Ensure: The labels (WL) predicted by the weak classifiers
The dataset D has been split by default into two distinct sets, namely, a training set S_1 and a test set S_2, respectively.
Step one: Utilize base-learning classifiers
for $k \leftarrow 1\ to\ m$ **do**
Train base learners (expert opinions) WC_k dependent on S_1
Test base learners (expert opinions) WC_k using S_2 and get the predictions WL_k
end for
Meta-Learning Step
Require: The labels (WL) predicted by the ensemble classifiers
Ensure: The final labels (EL) predicted by the weak classifiers
for $i = 1$ **do**
Create an ensemble dataset $D^{'} = F \cup WL$
Utilize a meta-learning classifier
Train meta-learner EC dependent on $D^{'}$
Test meta-learner EC using S_2 and get the predictions EL
Calculate the accuracy and other metrics of the Ensemble Learner
end for

IV. Performance Evaluations

This section presents the evaluation results of testing various cases of the proposed ensemble learning algorithms. The section also describes the setup required to reproduce the presented results. Specifically, we outline the client model utilized in the process and the dataset on which the results were evaluated.

A. Performance Metrics

Table II illustrates the metrics used to evaluate the effectiveness of an ML model. In scenarios involving imbalanced data, the typical accuracy measurement may not accurately represent the model's performance. This is because a basic classifier can predict all samples to belong to the majority class (Benign Data), thereby achieving a high-accuracy model while neglecting to identify instances that belong to minority classes. Consequently, the overall model may become ineffectual. Precision, recall, and F1-score metrics are employed to better assess the classifier's ability.

TABLE I
ORIGINAL TRAIN AND TEST DATASET FLOW TYPE BREAKDOWN.

A/A	Traffic Flow Type	Training Set	Test Set
0	**Benign**	1337046	410865
1	**Bot**	1228	354
2	**DDoS**	80653	23160
3	**DoS GoldenEye**	6480	1861
4	**DoS Hulk**	109334	41626
5	**DoS Slowhttptest**	3302	994
6	**DoS Slowloris**	3411	1048
7	**FTP-Patator**	3819	1436
8	**Heartbleed**	6	2
9	**Infiltration**	22	6
10	**Portscan**	66659	28728
11	**SSH-Patator**	2086	1067
12	**Web Attack Brute Force**	935	272
13	**Web Attack SQL Injection**	12	4
14	**Web Attack XSS**	410	117

B. Dataset

Data traffic generated from SDNs can closely align with data from Open-RAN as both paradigms share fundamental principles and objectives. In both SDNs and Open-RAN, the concept of decoupling control functions from underlying hardware introduces programmability and flexibility. The E2 nodes, including O-DUs and O-CUs in Open-RAN, can be compared to network devices (switches) in the SDN environment while UEs are the end devices. They act as network components that are controlled by a centralized controller, in the case of Open-RAN, the RAN Intelligent Controller (RIC).

The benchmark dataset adopted for performance comparisons was created by ULAK and provided by the ITU for the "ITU-ML5G-PS-006: Intrusion and Vulnerability Detection in Software-Defined Networks (SDN)" challenge [13]. The initial training set consisted of roughly 1.8 million records and the initial test set had over half a million samples. Each data instance was presented with a collection of 78 features, coupled with a label column describing the data flow category. While cleaning the data, 1,811 corrupted training records and 537 corrupted test records were removed. Additionally, 166,142 duplicate training records were discarded. Following the completion of the data cleaning process, the training set had 1,615,403 records, and the new test set had 511,540 records. All instances, including both training and test sets, where feature

TABLE II
PERFORMANCE METRICS USED FOR EVALUATION.

Metric	Equation
Accuracy	$(TP + TN)/(TP + FP + TN + FN)$
Precision	$(TP/(TP + FP))$
Recall	$(TP/(TP + FN))$
F1-score	$(2 * Precision * Recall)/(Precision + Recall)$

TABLE III
SUMMARY OF THE SINGLE & ENSEMBLE CLASSIFIER

Type	Classifier	Acc.	Pre.	Rec.	F1.
Single Learners	**DT**	99.05	85.18	75.34	77.63
	LR	93.77	41.58	35.14	37.49
	NB	84.96	30.78	20.97	18.16
Ensemble Learner	SEL	**99.07**	**87.79**	**82.05**	**83.34**

values were -1, were substituted with the mean value of the corresponding feature column in the training set. Ultimately, eight features were dropped due to contributing only zero values. As a result, each sample was represented with 70 features. Table I provides insights into how the labels were distributed through the cleaned dataset. Finally, the ETC feature importance algorithm was applied to attain a reduced feature dataset for training.

C. Results

Table III illustrates the preliminary results of the single and ensemble learners. Amongst the single learners, the performance of DT was optimal and superior to the two other classifiers (LR & NB) used in the preliminary experiment. In particular, the precision, recall, and f1-score of the two other classifiers were very low, signifying their inability to classify most traffic flow types correctly. Conversely, the SEL classifier, which is a combination of the three single learners, was adopted from literature [14]–[18], and it achieved the best performance as it leveraged the classification strength of each classifier during its learning phase to give the final classification output.

Furthermore, Table IV gave a broader overview of the metrics we used to evaluate all the classifiers. In this table, we present the performance of the classifiers on each of the traffic flow types. This provides a more comprehensive insight into the earlier given Table III. It can be deduced from the table that for most of the flow types, the SEL evaluation metric values were optimal primarily compared to the other classifiers. In addition, for some flow types, the single classifiers recorded no metric values, signifying no detection of the flow type during the classification task. This was, however, different for the SEL as it recorded classification metrics for all the flow types, which signifies its superiority over the other classifiers.

Table V presents the performance summary of the SEL classifier after we applied the FI feature reduction technique to the dataset. The result indicated that using all the dataset features produced the best classification performance for all the metrics used except for accuracy. The other metrics essentially looked into how well the classifier can classify the different flow types correctly, hence the reason why the performance analysis can not be based on accuracy alone as it can be seen that 19, 15, and 10 features have a higher accuracy compared to using all the features but had a reduced performance for the other metrics. The lowest performance was recorded when the 5 most important features of the dataset were used to evaluate the SEL. However, regarding training (TrT.) and test (TeT.) time, the 5 most important features recorded the fastest times, signifying quicker classification by the SEL compared to when other features were evaluated. The 10 most important features of the dataset are the most optimal compared to other features; in particular, when all dataset features are used, the reduction in performance is not too significant. In addition, it made up for the reduced performance with its training and test times, which is faster than when a higher number of features are used to evaluate the SEL.

Fig. 3 and Fig. 4 illustrate the explainable artificial intelligence (XAI) of this contribution. In Fig. 3, a branch of a DT classifier is presented after it was applied to the dataset, and it can be seen that at every node, it made use of the features of the dataset with the least entropy value to classify the flow types the best possible way. Similarly, in Fig. 4, we illustrated the working mechanism of the DT meta-learner extract of our SEL with the same branch flow as of the previous. However, in this, we can see that the flow type classification is different. This is because of the addition of the prediction outputs of the base learners to the learning phase of the meta-learner. In particular, it can be seen that the feature of the root node in the first tree is the feature *Bwd Packet Length Mean*, which was replaced as *NB Label*, which is the prediction label of the NB base-learner. The entropy values for both trees were calculated, which summed up to 5.076 and 4.464 for the first and second trees, respectively, which signifies that the DT extracted from the SEL has less impurity compared to the previous and can classify the flow types more correctly, hence the better performance in earlier presented in Tables III, IV, and V.

V. DISCUSSION AND LIMITATIONS

This paper overviews the ML algorithms utilized in Open-RAN IDS, encompassing supervised learning techniques. We evaluated the performance of some classifiers on the ITU-ML5G-PS-006 dataset. We then analyzed the results by comparing the general performance of the classifiers and then analyzing their performance on individual dataset flow types more in-depth. The internal working mechanism of the SEL classifier used in the paper was illustrated to explain how it achieved superior performance over other classifiers utilized in this contribution. However, our contribution is limited to

TABLE IV
PRECISION, RECALL AND F1 SCORE FOR THE ITU-ML5G-PS-006 DATASET MULTICLASS NETWORK CATEGORIES.

Clf.	Metric (%)	Benign	Bot	DDoS	DoS Golden Eye	DoS Hulk	DoS Slow http test	DoS Slow loris	FTP Pata-tor	Heartbleed	Infiltration	Port scan	SSH Pata-tor	W.A Brute Force	W.A SQL Injec-tion	W.A XSS
DT	Prec.	**98.97**	68.02	**99.96**	**99.14**	99.80	98.29	98.48	**99.93**	0.00	**100.00**	99.40	99.72	73.80	**100.00**	42.24
	Rec.	99.90	70.90	99.95	99.41	90.68	**98.09**	99.14	99.79	0.00	**33.33**	99.09	99.34	73.53	**25.00**	41.88
	F1	99.43	69.43	**99.96**	**99.28**	95.02	98.19	98.81	99.86	0.00	**50.00**	**99.24**	99.53	73.66	**40.00**	42.06
LR	Prec.	94.99	0.00	95.77	93.83	93.10	77.93	88.91	0.00	0.00	0.00	79.23	0.00	0.00	0.00	0.00
	Rec.	97.86	0.00	83.61	63.73	65.83	68.91	48.19	0.00	0.00	0.00	99.02	0.00	0.00	0.00	0.00
	F1	96.40	0.00	89.28	75.90	77.12	73.14	62.50	0.00	0.00	0.00	88.03	0.00	0.00	0.00	0.00
NB	Prec.	87.64	0.00	74.30	87.29	71.02	31.61	07.61	0.00	0.00	02.20	100.00	0.00	0.00	0.00	0.00
	Rec.	96.68	0.00	47.55	08.49	61.19	19.01	48.19	0.00	0.00	**33.33**	00.14	0.00	0.00	0.00	0.00
	F1	91.93	0.00	57.99	15.48	65.74	23.74	13.15	0.00	0.00	04.12	00.28	0.00	0.00	0.00	0.00
SEL	Prec.	**98.97**	**79.62**	**99.96**	98.93	**99.81**	**98.49**	**98.59**	**99.93**	**100.00**	25.00	**99.41**	**100.00**	**73.82**	**100.00**	**44.35**
	Rec.	**99.91**	**83.90**	**99.96**	**99.52**	**90.69**	**98.49**	**99.43**	**99.93**	**100.00**	16.67	99.07	**99.91**	**73.63**	**25.00**	**43.59**
	F1	**99.45**	**81.71**	**99.96**	99.22	**95.03**	**98.49**	**99.00**	**99.93**	**100.00**	20.00	**99.24**	**99.95**	**74.22**	**40.00**	**43.97**

TABLE V
SEL RESULT SUMMARY OF REDUCED FEATURE DIMENSIONS USING FI

Number of Features	Acc.	Pre.	Rec.	F1.	TrT.	TeT.
All	99.07	87.79	82.05	83.34	9.49mins	2.37s
19	99.17	83.13	82.54	82.81	8.29mins	1.86s
15	99.71	79.05	78.35	78.64	4.55mins	1.85s
10	99.71	80.03	78.86	79.14	4.27mins	1.44s
5	96.99	84.57	68.66	73.68	4.07mins	1.38s

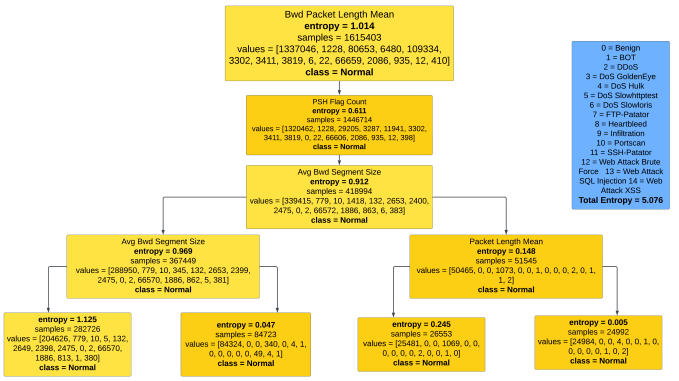

Fig. 3. Tree created using single DT.

the classifier, feature reduction technique, and dataset we employed, which will be considered in the future.

VI. CONCLUSIONS AND FUTURE WORK

This paper evaluated the applicability of the SEL classifier on the TU-ML5G-PS-006 dataset. Experiment results demonstrate that by using EL, essential features of the dataset can be used, leading to the removal of redundant and less important features. Thus reducing computational resource usage and, most importantly, the dataset feature dimensions. We reduced the dataset's feature dimension to 5 before a more reduced performance was recorded for the SEL classifier. However, SEL

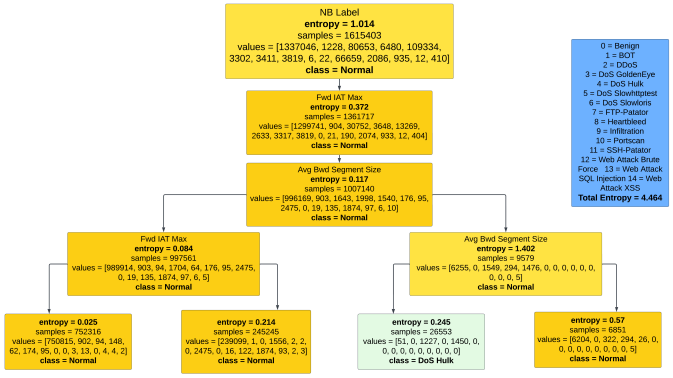

Fig. 4. Tree created using DT extract from the SEL.

maintained the most optimal classification performance across all the evaluation metrics for the different flow types, with the 10 most important features of the dataset deduced from using the EL technique. Despite the optimal performance of the SEL on the EL-reduced feature dimensions, further evaluation of the SEL classifier with other datasets is needed to validate its applicability. Other feature dimensionality reduction techniques can also be adapted to further evaluate the performance of the SEL.

ACKNOWLEDGEMENT

This work was partially supported by the U.K. Department for Science, Innovation, and Technology under Project 5G MoDE (Mobile oRAN for highly Dense Environments).

REFERENCES

[1] G. M. Almeida *et al.*, "RIC-O: Efficient placement of a disaggregated and distributed RAN Intelligent Controller with dynamic clustering of radio nodes," *IEEE Journal on Selected Areas in Communications*, pp. 1–30, 2023.

[2] M. Hoffmann *et al.*, "Signaling Storm Detection in IIoT Network based on the Open RAN Architecture." IEEE INFOCOM Workshop, 2023, pp. 1–2.

[3] D. Sabella *et al.*, "Energy efficiency benefits of ran-as-a-service concept for a cloud-based 5g mobile network infrastructure," *IEEE Access*, vol. 2, pp. 1586–1597, 2014.
[4] M. K. Motalleb *et al.*, "Moving target defense based secured network slicing system in the o-ran architecture," in *IEEE GLOBECOM*. IEEE, 2023, pp. 6358–6363.
[5] C. Benzaïd and T. Taleb, "Ai for beyond 5g networks: a cyber-security defense or offense enabler?" *IEEE network*, vol. 34, no. 6, pp. 140–147, 2020.
[6] S. Chatzimiltis *et al.*, "A collaborative software defined network-based smart grid intrusion detection system," *IEEE open journal of the Communications Society*, 2024.
[7] P. V. Alves *et al.*, "Machine learning applied to anomaly detection on 5g o-ran architecture," *Procedia Computer Science*, vol. 222, pp. 81–93, 2023.
[8] D. Attanayaka *et al.*, "Peer-to-peer federated learning based anomaly detection for open radio access networks," in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 5464–5470.
[9] S. Soltani *et al.*, "Poisoning Bearer Context Migration in O-RAN 5G Network," *IEEE Wireless Communications Letters*, vol. 12, no. 3, pp. 401–405, mar 2023.
[10] P. Kryszkiewicz *et al.*, "Open RAN for detection of a jamming attack in a 5G network," in *IEEE Vehicular Technology Conference*, vol. 2023-June. Institute of Electrical and Electronics Engineers Inc., 2023.
[11] S. Bagui *et al.*, "Resampling imbalanced data for network intrusion detection datasets," *Journal of Big Data*, vol. 8, no. 1, pp. 1–41, 2021.
[12] R. Qaddoura *et al.*, "A multi-stage classification approach for iot intrusion detection based on clustering with oversampling," *Applied Sciences*, vol. 11, no. 7, p. 3022, 2021.
[13] Machine Learning for SDN security: improving intrusion and vulnerability detection. [Online]. Available: https://challenge.aiforgood.itu.int/match/matchitem/81
[14] H. A. Ahmed *et al.*, "Network intrusion detection using oversampling technique and machine learning algorithms," *PeerJ Computer Science*, vol. 8, p. e820, 2022.
[15] A. Abbas *et al.*, "A new ensemble-based intrusion detection system for internet of things," *Arabian Journal for Science and Engineering*, pp. 1–15, 2021.
[16] V. Pai, N. Adesh *et al.*, "Comparative analysis of machine learning algorithms for intrusion detection," in *IOP Conference Series: Materials Science and Engineering*, vol. 1013, no. 1. IOP Publishing, 2021, p. 012038.
[17] K. Upadhyay, "Network intrusion detection system based on machine learning," *Annals of RSCB*, vol. 25, no. 4, pp. 12 445–12 451, 2021.
[18] S. A. Abdulkareem *et al.*, "Smote-stack for network intrusion detection in an iot environment," in *2022 IEEE ISCC*, 2022, pp. 1–6.