A Federated Learning Approach for Improving Security in Network Slicing

Shalitha Wijethilaka*, Madhusanka Liyanage†
*†School of Computer Science, University College Dublin, Ireland
†Centre for Wireless Communications, University of Oulu, Finland
Email: *mahadurage.wijethilaka@ucdconnect.ie, †madhusanka@ucd.ie, †madhusanka.liyanage@oulu.fi

***Abstract*—Network Slicing (NS) is a predominant technology in future telecommunication networks, including Fifth Generation (5G), which supports the realization of heterogeneous applications and services. It allows the allocation of a dedicated logical network slice of the physical network to each application. Security is one of the paramount challenges in an NS ecosystem. Several technologies, including Machine Learning (ML), have been proposed to mitigate security challenges in 5G networks. However, the use of ML for NS security is not properly implemented. Especially, the scarcity of coordination and the difficulties of privacy-protected information sharing between slices cause failures and performance degradation of these ML based NS security solutions. To address this issue, this paper proposes a novel Federated Learning (FL) based coordinated security orchestration architecture named Federated Learning enabled Security Orchestrator (FLeSO) to centrally perform security operations in a slicing ecosystem while preserving the privacy of the data. In addition, the proposed FLeSO architecture enables features such as proactive security deployment and steady security level maintenance independent of the slicing strategy. The proposed architecture is implemented in a real-world slicing testbed, and a comprehensive set of experiments are performed to evaluate the effectiveness of the proposed FLeSO architecture. The test results illustrate the significant advantage of the proposed approach over the legacy system in terms of improving the security of an NS ecosystem.**

***Index Terms*—Network Slicing, Security, Federated Learning, Deep Learning**

I. Introduction

The revolutionary transformation toward the smart world necessitates the connectivity between anything from anywhere, anytime. The advancements in applications such as autonomous vehicles, Augmented Reality (AR), Virtual Reality (VR), and smart cities demand diverse network requirements that can not be facilitated through traditional telecommunication networks. The future telecommunication networks are specifically designed to address these challenges. Network Slicing (NS) is one of the predominant technologies in such networks to facilitate heterogeneous network requirements of diverse applications by dividing the physical network into multiple logical networks, known as network slices [1]. Increasing the scalability and dynamicity, facilitating the diverse Quality of Service (QoS) requirements of different applications, and improving security and privacy, can be identified as some advantages of NS. However, along with these advantages, NS introduces a set of novel security challenges [2].

978-1-6654-3540-6/22/$31.00 © 2022 European Union

Heterogeneous applications that utilize the network have diverse security requirements. Allocating a dedicated network slice supports overcoming this challenge. Also, slice isolation is essential to preserve privacy and reduce the impact of security vulnerabilities. Therefore, managing the security operations within the slice itself, known as in-slice security management, can be proposed as the legacy approach required to facilitate these diverse security requirements. However, this creates a novel set of challenges.

Machine Learning (ML) plays a prodigious role in telecommunication security. Security systems adopt ML models to execute the security operations of the network [3]. However, the network slices are required to be isolated from each other due to privacy concerns. This makes it challenging to collect data and train centralized ML models. Also, the slice owners may not prefer to share the data in their slices as they may contain sensitive information. These challenges in an NS Ecosystem (NSE) degrade the performance of ML models that are used for security purposes. Therefore, a novel centralized approach is required to support the training process of the security-related ML models while preserving the privacy of data and precisely managing security operations in the NSE.

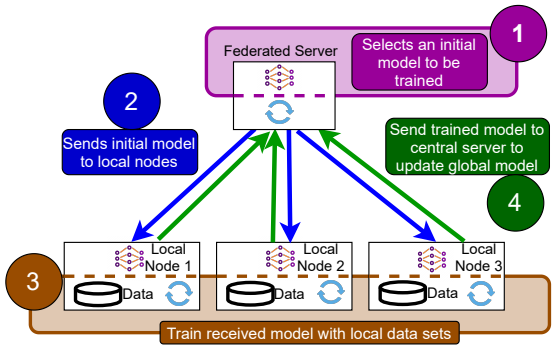

Fig. 1: The Methodology of Federated learning

Federated Learning (FL) is a novel approach in ML that alleviates the challenges in data collection. Learning a global statistical model using the data stored in remote locations can be defined as FL [4], [5]. The primary concept of FL is shown in Fig. 1. A centralized server, known as a federated server which aggregates the received models into one model using a specific method such as Federated Averaging (FedAvg) and FedMA [6] and local data collection nodes are the principal elements in an FL system. In the FL approach, ML models can be trained without exchanging data. Therefore, the privacy

and confidentiality of the data can be preserved.

NS enabled environment can be considered a fitting ecosystem for FL. Each individual network slice can be considered as a local node, and a centralized security orchestration framework can be used as the federated server. Therefore, in this paper, we propose an FL based coordinated security management framework for the NSE. Using FL for security in NS is a novel research area. So, Federated Learning enabled Security Orchestrator (FLeSO) is the first FL based approach proposed to improve security in NS. The modular-level architecture of the proposed FLeSO framework is presented to simplify the implementation. Increased performance of security-related ML models, ability to deploy pro-active security mechanisms and simplified security management are some of the features of our framework. We perform an extensive set of experiments on a real NS testbed which is built using open-source tools, to show the performance of the FLeSO framework. The FLeSO shows an increased accuracy and the ability to catch unseen attacks in the network slices. Also, we show that the distribution of security attacks across the network does not affect the effectiveness of our solution.

The paper consists of six sections. Section II discusses the existing literature in FL, NS, and security. A framework for using FL for security services in NS is proposed in section III. Section IV investigates the results of the approach and comparison of different scenarios, and section V discusses the proposed approach. Finally, the section VI concludes the paper.

II. Related works

FL is a novel approach in ML that Google introduced. In [4], Li et al. provide a comprehensive investigation of FL including unique properties, challenges, and potential future research directions related to FL. Due to the privacy-preserving nature of FL, Nikham et al. identify FL as a relevant technology for training ML models in wireless environments [7]. They present several potential applications, and open research problems, of FL in the context of wireless communications, specifically in Fifth Generation (5G) networks. In [8], Yang et al. provide a comprehensive study on the FL applications for Sixth Generation (6G) networks. The significance of FL for telecommunication systems are emphasized in these researches.

FL utilization for NSEs has been discussed in the research community. In [9], Messaoud et al. propose a novel federated deep Reinforcement Learning (RL) approach to facilitate QoS requirements of an NS enabled Industrial Internet of Things (IIoT) environment via dynamic network management and resource allocation. An FL approach for training ML models in an NSE is proposed in [10]. They build a model to predict service-level Key Performance Indicators (KPIs) of network slices to prove their approach. However, these approaches are not specifically considered to improve the security of an NSE.

FL approaches for improving security can be found in the literature. In [11], Sater et al. present an FL approach using a stacked Long Short-Time Memory (LSTM) for anomaly detection in smart buildings. A hybrid ensemble model for Intrusion Detection System (IDS) in IoT environments is proposed in [12]. They adapt the proposed model to an FL framework, showing that their federated approach achieves performance close to the centralized settings. In [13], Mothukuri et al. propose a Gated Recurrent Units (GRUs) model that uses federated training rounds to detect anomalies for IoT security attacks. They show that their approach outperforms the centralized approach in terms of maintaining the privacy of user data and providing an optimal accuracy in attack detection. Even though these works address the security challenges using federated approaches, none of them specifically discuss how FL can be used to improve security in NS. Therefore, this is the first approach that uses FL to improve security in NS.

III. Proposed FLeSO Framework

We present the proposed FLeSO approach to improve the security in an NSE. The architecture with required elements and connections is shown in Fig. 2. The basic functionalities of the elements in FLeSO are described below.

- **Security Orchestrator Client (SOC)**: Perform security operations in a slice, send security related information to the security orchestrator, and FL model receiving, training, and sending
- **Federated Management Component (FMC)**: ML model distribution, collection and aggregation
- **Slice Security Monitoring Component (SSMC):** Collect outputs from SOCs and pass them to relevant entities in the orchestrator
- **Data Evaluation Component (DEC):** Analyze received information, identify potential attacks, and decide initial action to mitigate identified attacks
- **Solution Life-Cycle Management Component (SLCMC):** Manage the life-cycle of the configured security operations in the network slices
- **System Evaluation Component (SEC):** Evaluate the NSE to perform selected security operations
- **Security Solution Deployment Component (SSDC):** Communicate with Network Slice Manager (NSM) to deploy or configure security operations in the NSE

Fig. 3 shows the functional flow diagram of the FLeSO framework with the involvement of different elements. A SOC is required to be deployed in each network slice to collect data from network functions to prepare data for model training. The FME handles the main model, and it sends the model to each SOC to train using the collected data of a correspondent network slice. After training the received models, SOC sends the model parameters to the FME. The FME aggregates the received models and updates the central model. The simplest method to aggregate model parameters is FedAvg, as mentioned. It takes the average of model weights. Then, the final updated models are deployed in SOCs, and incoming data are evaluated using that model. The SSME collects the model outputs and passes them to the DEE for the initial evaluation. If it identifies an attack, it decides the solution and informs the SSDE to perform it in the slicing environment. Also, the SLCME continuously monitors the attacked slice and alters the mitigation strategy via SSDE according to the strength of the

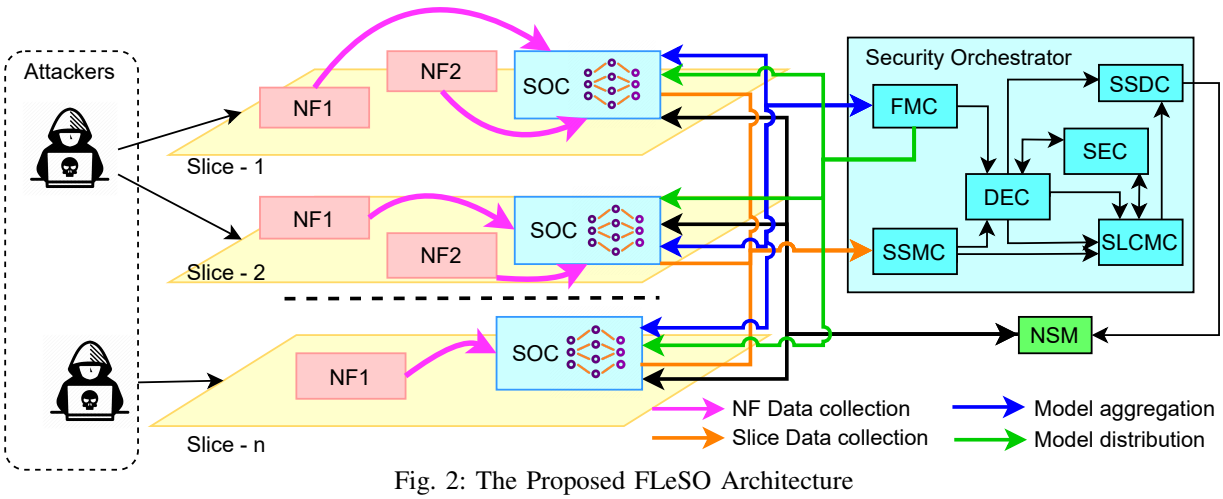

Fig. 2: The Proposed FLeSO Architecture

attack. Finally, deployed security configurations are removed at the end of the attack.

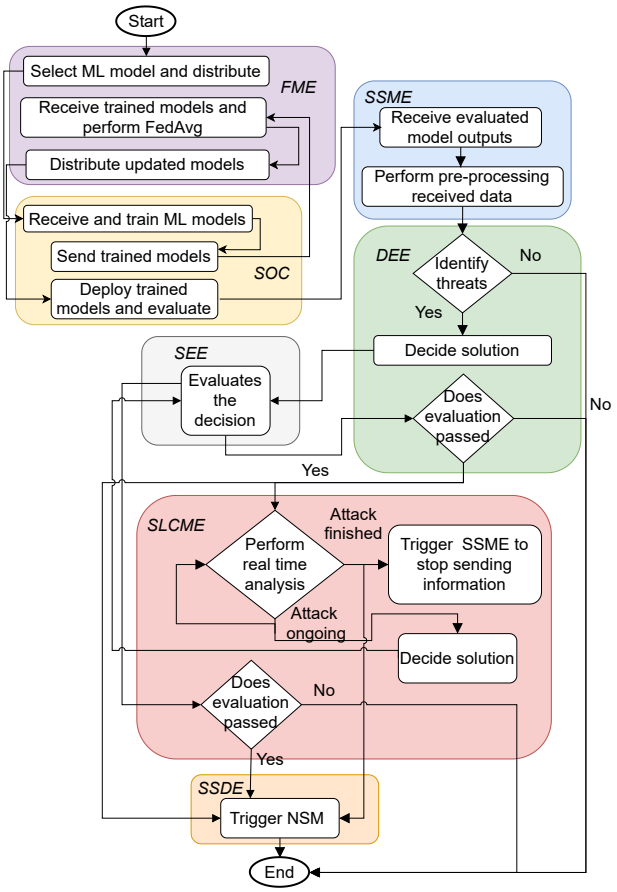

Fig. 3: Interaction of different components to enable the functionality of the proposed FLeSO architecture

A. Enabled features from the FLeSO framework

1) Increased performance of security-related ML models: The FLeSO framework allows to employ distributed information across NSE to train security-related ML models. When the available amount of data increases for the training process, the performance of ML models increases. Therefore, security can be improved in the NSE.

2) Secure information sharing: Due to the security and privacy considerations, information sharing between network slices is unfeasible. However, the FLeSO framework facilitates sharing the crucial features of slice-specific data without sharing sensitive information.

3) Pro-active security deployment: The same security attack can be transpired to different network slices at different times. The FLeSO framework allows extracting attack information from network slices which have been already attacked. Consequently, the updated ML models can be deployed in non-attacked slices in advance.

4) Centralized security management: The FLeSO framework has a centralized architecture in an NSE. It can collect security information from network slices and perform security operations via the NSM. Therefore, any party who needs to perform security operations in the NSE can utilize the FLeSO framework without accessing individual network slices.

IV. Evaluation of the FLeSO Framework

In this section, we present the experiments performed to prove the effectiveness of the FLeSO framework. We implement the FleSO framework on top of a real NS testbed, which is built using open source tools such as Open Source Mano (OSM) and OpenStack as shown in Fig. 4. Python, PyTorch, and Scikit-learn are used to implement the FL framework. The other elements are implemented using Java Springboot. NSL-KDD intrusion detection dataset [14] is used for the experiments as it is one of the widely used data sets among researchers. Data pre-processing techniques such as data cleaning, data transformation, and data reduction are applied to the data set before starting the training process. The data set's composition is balanced when the data set is considered in a high-level manner, i.e. attack and normal. However, the data set contains attack data related to several attack types. It is imbalanced if all the attack types are considered.

We consider the high-level and mid-level classification of attacks in our experiments. Low-level attack classification has

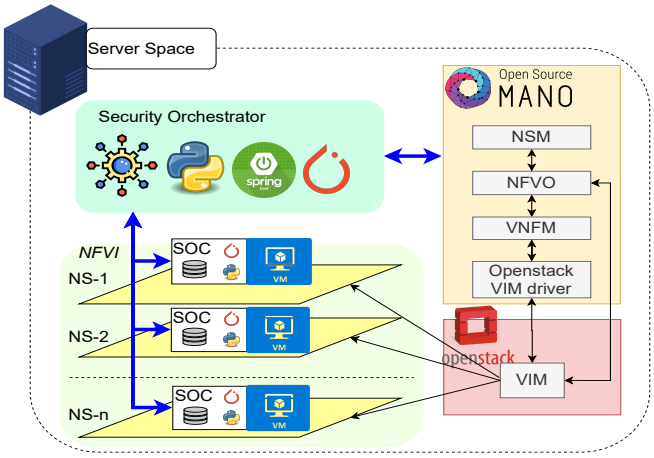

Fig. 4: Testbed implementation of the FLeSO

not been considered due to the deficiency in the number of records per attack type. Denial of Service (DoS), User to Root (U2R), Root to User (R2U), Probe, and Normal are the classes in the mid-level classification of the data set. We use a deep neural network as the ML model to perform the experiments. Table I shows the utilized parameters in the experiments.

Parameter	Value
Batch Size	32
Hidden Layers	2
Nodes in each hidden layer	200
Test Size	10%
Learning Rate	0.01
Number of Epochs	20
Optimizer	SGD
Criterion	Cross Entropy Loss

TABLE I: Experiment Parameters

In our FL implementation, we assume that the SOC in a particular network slice is correspondent to a local node, and the FME is correspondent to the FL server. Also, we distribute data in the NSL-KDD data set across the SOCs (local nodes) in different fashions. Hence, in the network, the data received for the SOC by the network slice is considered to be the allocated data from the data set for that particular SOC. NSE with one network slice is not considered in the experiments as NS is not required in this scenario.

A. *Performance comparison between legacy and FLeSO*

In this experiment, our objective is to investigate the impact of the training data distribution across the NSE on the accuracy of the FLeSO framework. Here, the training data set is Independently and Identically distributed (IID) across the network slices. We compare the FLeSO based approach and the legacy approach in this experiment. With the FLeSO framework, a centralized model is trained using all the data in the NSE. In the legacy approach, as the data cannot be shared between network slices, models are trained only using the slice specific data. When the number of network slices increases, the traffic per slice decreases since the number of users in the network is the same. The experiment is performed with two types of data classification, i.e. high-level and mid-level. In both scenarios, we measure the accuracy of the models that were trained using the normal data set and modified data set by applying the Synthetic Minority Oversampling Technique (SMOTE) [15]. SMOTE is one of the widely used techniques to remove the imbalanced nature on data. It helps to reduce the biasness of classifications of ML models towards the majority class. For each number of network slices, the experiment is performed numerous times and taken the average to increase the accuracy of the results.

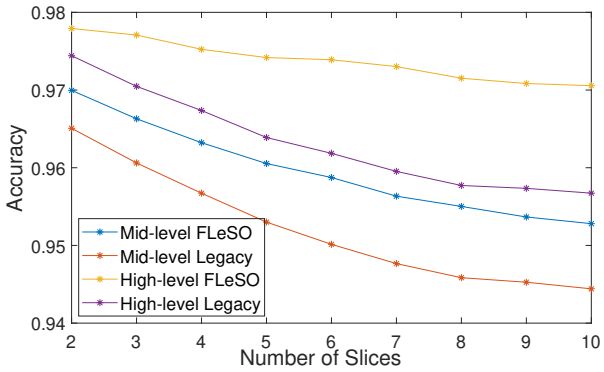

Fig. 5: Accuracy comparison without SMOTE transformation

Fig. 5 shows the results for the model, which is trained using the standard data set. In terms of accuracy, in both attack classification scenarios, the FLeSO framework shows a higher accuracy when the number of slices in the network increases. High-level attack classification always shows a higher accuracy due to the lower number of classifiers in the data set. Moreover, when the number of network slices increases, the accuracy decreases in each scenario. In the legacy approach, the accuracy is reduced due to the deficit of the training data in the network slices. The deviation from the optimal weight values of the ML models due to the FedAvg aggregation method is the cause of the accuracy reduction of the FLeSO framework.

The results when we apply SMOTE transformation on the data set are shown in the Fig. 6. The results are almost identical to the previous experiment. However, the FLeSO framework shows an almost constant accuracy value when increasing the number of network slices in the system in this scenario. Due to the SMOTE transformation, the ML-model weight values are much closer to the optimal values, and therefore, the impact of the FedAvg aggregation mechanism is minimum. It is the reason for the almost constant accuracy value in the FLeSO framework. Furthermore, the accuracy values in each scenario show relatively lower values than the results received through training with standard data. Removing the biasness of model classification towards the majority class due to the SMOTE transformation could be the reason for this accuracy reduction.

B. *Proactive security implementation with FLeSO framework*

This experiment aims to investigate how proactive security mechanisms can be implemented using the FLeSO framework, hence making the network slices capable of identifying unseen

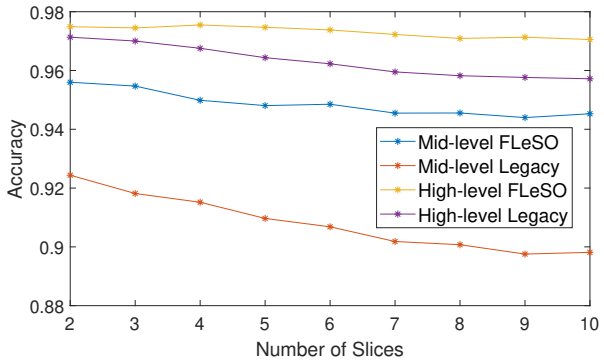

Fig. 6: Accuracy comparison with SMOTE transformation

attacks. We use an NSE with five network slices for this experiment since the data set has five attack types under the mid-level attack classification. Non-IID data is used to perform this experiment. Data analogous to a specific attack type is sent to each network slice with a portion of normal data. Attack data distribution across the NSE is shown in table II. However, the test set consists of all attack types.

	Normal	DoS	U2R	R2U	Probe
S1	✓	✗	✗	✗	✗
S2	✓	✓	✗	✗	✗
S3	✓	✗	✓	✗	✗
S4	✓	✗	✗	✓	✗
S5	✓	✗	✗	✗	✓

TABLE II: Data distribution across slices in experiment B

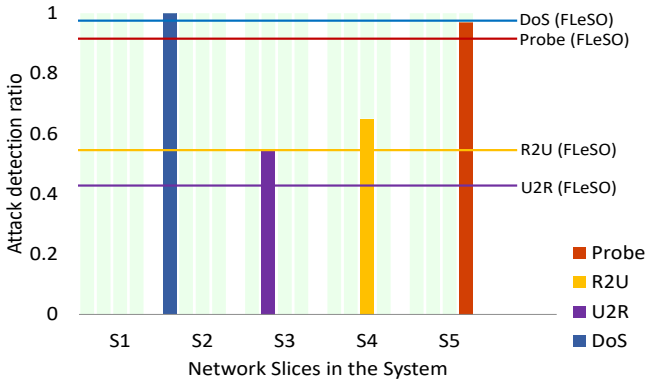

Fig. 7: Attack detection in experiment B

Fig. 7, shows the received results in this experiment. In the legacy approach, the models in individual slices can not detect any attack type other than the attack in the training data. However, with the FLeSO framework all the attacks can be identified. This supports the implementation of proactive security mechanisms in the NSE. However, the limitation in the FLeSO framework in this experiment is the requirement of a higher number of federated rounds to achieve this performance. In this experiment, we performed 500 federated rounds to receive this result. Significant deviations of the weight values from the optimal values in the hidden layers due to the FedAvg algorithm and the specific individual attack distribution across the network are reasons for this observation. Moreover, the U2R and R2U attack detection is relatively low due to the fewer amount of training data in the data set related to those attacks.

C. Performance evaluation when a slice is under multiple attacks

This experiment aims to analyze the performance of the FLeSO framework when we alter the number of attack types in a particular network slice. This experiment is an extension of the previous experiment. The data set is not limited to one specific attack type. Instead, an amalgam of attack types is sent to each network slice. Attack data distribution across the network can be seen in the table III. The training data is non-IID, and the training data distribution is different from the previous experiment.

	Normal	DoS	Probe	R2U	U2R
S1	✓	✗	✗	✗	✗
S2	✓	✓	✗	✗	✗
S3	✓	✓	✓	✗	✗
S4	✓	✓	✓	✓	✗
S5	✓	✓	✓	✓	✓

TABLE III: Data distribution across slices in experiment C

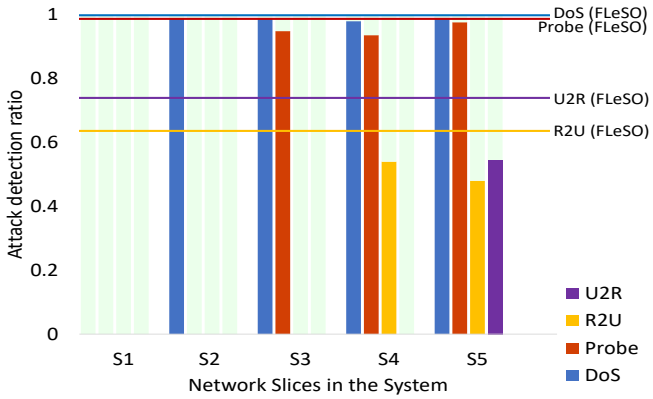

Fig. 8: Attack detection in experiment C

As shown in Fig. 8, even in the legacy approach, when the number of attacks in the training set is increasing in a particular slice, the detectable attack types of the ML model are increasing. However, the FLeSO does not depend on the number of attack types in the network slice. Also, it outperforms all the scenarios than the legacy security models, even when the slice is provided with the highest attack diversity.

D. Convergence analysis for optimal performance of FLeSO

This experiment aims to analyse the performance of the FLeSO framework with different training data distributions and with different number of federated rounds. We consider IID data, where all the attack data can be found in each network slice and non-IID data, where specific attack data can be found in a particular network slice in this experiment.

Fig. 9 depicts the obtained results in this experiment. The FLeSO framework converges to the highest accuracy rapidly with IID data. Then, it converges with cumulative attack distribution and specific attack per slice distribution, respectively. However, when the number of federated rounds are increasing,

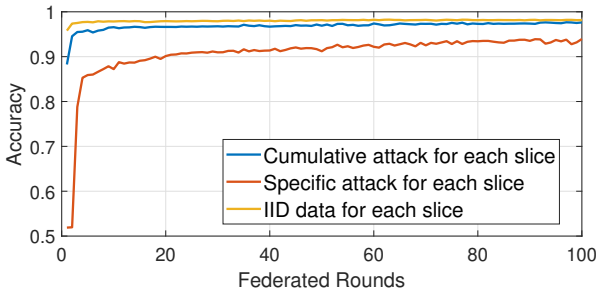

Fig. 9: Accuracy comparison with federated rounds

the FLeSO framework converges to the same level of accuracy with all the data distributions.

V. Discussion

Table IV shows the feature-wise comparison of the proposed framework with existing related works. The key-related works were selected considering the relevance to our approach, implementation details, and experimental results. The feature list is extracted from the proven results of the experiment. From the comparison of the table IV, we can deliberate that the FLeSO framework outperforms considering an NSE.

Features	[9]	[11]	[12]	[13]	FLeSO
Security improvement	✗	✓	✓	✓	✓
Increased performance of security related ML models	✗	✓	✓	✓	✓
Secure information sharing	✓	✓	✓	✓	✓
Pro-active security deployment	✗	✗	✗	✗	✓
NS-specific implementation	✓	✗	✗	✗	✓

TABLE IV: Feature comparison with key related works

Apart from the discussed features, some limitations of the FLeSO framework can be discovered through the performed experiments. Data distribution across the network has a significant impact on the performance of the ML models in our approach. Even though a higher number of federated rounds allows for rectifying the performance reduction, it significantly increases the training time. Another limitation is the accuracy reduction when the number of network slices increases. Moreover, inherent security vulnerabilities in FL approaches, such as communication bottlenecks, poisoning, and backdoor attacks, can be found in this approach.

VI. Conclusion & Future Works

In this paper, we propose the FLeSO framework that can be used to improve security in an NS environment. FLeSO is a novel FL based coordinated security orchestration architecture which can outcome the drawbacks of legacy ML-based approaches in slice security management. The experiments performed on top of a real NS testbed prove that the FLeSO achieves very high accuracy even under limited data availability in network slices while preserving slice isolation. Moreover, the FLeSO framework allows deploying security mechanisms pro-actively for unseen security attacks in network slices. The results exhibit that the FLeSO can obtain a significant accuracy for any data distribution across the network by increasing the federated training rounds. Furthermore, the discussed feature-wise comparison of the FLeSO framework with existing related works manifests the significance of our framework in an NSE. Finally, the discussed limitations of the framework open some future research directions.

The future work focuses on testing the entire framework with real-time network traffic. Investigating optimal model aggregation mechanisms for NS would be one of the interesting research directions of this work. Moreover, enhancing the framework by reducing the federated rounds required for convergence is another potential future work.

VII. Acknowledgement

This research is funded by the Academy of Finland under 6Genesis Flagship (grant 318927), the Science Foundation Ireland under Connect Center (13 RC/2077_P2), and the European Commission in SPATIAL (Grant no: 101021808) projects.

References

[1] T. Umagiliya, S. Wijethilaka, C. De Alwis, P. Porambage, and M. Liyanage, "Network slicing strategies for smart industry applications," in *2021 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2021, pp. 30–35.

[2] V. A. Cunha, E. da Silva, M. B. de Carvalho, D. Corujo, J. P. Barraca, D. Gomes, L. Z. Granville, and R. L. Aguiar, "Network slicing security: Challenges and directions," *Internet Technology Letters*, vol. 2, no. 5, p. e125, 2019.

[3] G. Bugár, M. Vološin, T. Maksymyuk, J. Zausinová, V. Gazda, D. Horváth, and J. Gazda, "Techno-economic framework for dynamic operator selection in a multi-tier heterogeneous network," *Ad Hoc Networks*, vol. 97, p. 102007, 2020.

[4] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[5] N. Wang, W. Yang, Z. Guan, X. Du, and M. Guizani, "Bpfl: A blockchain based privacy-preserving federated learning scheme," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021, pp. 1–6.

[6] S. Shukla and N. Srivastava, "Federated matched averaging with information-gain based parameter sampling," in *The First International Conference on AI-ML-Systems*, 2021, pp. 1–7.

[7] 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.

[8] Z. Yang, M. Chen, K.-K. Wong, H. V. Poor, and S. Cui, "Federated learning for 6g: Applications, challenges, and opportunities," *Engineering*, 2021.

[9] S. Messaoud, A. Bradai, O. B. Ahmed, P. T. A. Quang, M. Atri, and M. S. Hossain, "Deep federated q-learning-based network slicing for industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5572–5582, 2020.

[10] B. Brik and A. Ksentini, "On predicting service-oriented network slices performances in 5g: A federated learning approach," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*. IEEE, 2020, pp. 164–171.

[11] R. A. Sater and A. B. Hamza, "A federated learning approach to anomaly detection in smart buildings," *ACM Transactions on Internet of Things*, vol. 2, no. 4, pp. 1–23, 2021.

[12] S. Chatterjee and M. K. Hanawal, "Federated learning for intrusion detection in iot security: a hybrid ensemble approach," *arXiv preprint arXiv:2106.15349*, 2021.

[13] V. Mothukuri, P. Khare, R. M. Parizi, S. Pouriyeh, A. Dehghantanha, and G. Srivastava, "Federated learning-based anomaly detection for iot security attacks," *IEEE Internet of Things Journal*, 2021.

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*. Ieee, 2009, pp. 1–6.

[15] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.