2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Federated Self-Supervised Learning for Intrusion Detection

Bruno H. Meyer[1], *Aurora T.R. Pozo*[1], *Michele Nogueira*[2], *WagnerM. Nunan Zola*[1]
[1]*Department of Informatics, Federal University of Paraná, Brazil*
[2]*Department of Computer Science, Federal University of Minas Gerais, Brazil*
email: {bruno, aurora, wagner}@inf.ufpr.br, michele@dcc.ufmg.br

***Abstract*—Deep learning and federated learning show significant success in cybersecurity for Intrusion Detection Systems (IDS). This paper presents the Federated Self-Supervised Learning (FSSL) framework proposed for IDSs. FSSL combines Self-Supervised Learning (SSL) with federated learning to obtain a global model. SSL works at the client level, where only unlabeled data is available, and thus it enables the learning from these data. This knowledge enhances the training of the target model. Therefore, FSSL follows a federated learning approach, where private data from multiple clients help to create a global model. Each client learns an unsupervised model, which is then transmitted to a server and combined into a single model. The communication between clients and the server aims to improve model performance and convergence. Conducted experiments compare FSSL with a baseline approach using limited data and a deep learning model. FSSL utilizes an autoencoder to learn a representational model on unlabeled data and transfers knowledge by initializing deep learning model weights with the encoder layers. Results show that FSSL significantly improves the F1-Score of detection systems across three well-known datasets (NSL-KDD, TonIoT, and BotIoT). Moreover, the proposed model demonstrated a noteworthy capability to detect previously unidentified attacks when compared to the baseline.**

***Index Terms*—Federated Learning, Self-Supervised Learning, Intrusion Detection Systems**

I. Introduction

The increasing sophistication and frequency of cyber-attacks have led to the development of a wide range of Intrusion Detection Systems (IDS) to protect computer networks. Within the domain of Cybersecurity, Intrusion Detection Systems (IDS) are designed to tackle the issue of discerning and mitigating cyberattacks. This involves examining various sets of information, such as how software behaves, log files, and data on network traffic. The goal is to spot and address anything unusual or potentially harmful happening in a system. Historically, IDS were first proposed using classical statistical methods to identify anomalies. Although, these methods were limited, due to their unfeasibility to learn with data to identify new types of attacks. In order to surpass these limitations, researchers have explored the use of machine learning (ML) and artificial intelligence techniques. For instance, the utilization of ML approaches have resulted on more accurate detections, lowering false alarms [1]. In industrial control systems specifically, ML algorithms have been utilized to develop anomaly-based detection methods using packet capture files [2].

Artificial neural networks, a fundamental component of machine learning, emulate the interconnected structure of the human brain to process information. Comprising layers of nodes or neurons, these networks specialize in learning patterns from data through a training process. In classification tasks, neural networks analyze input data and assign it to predefined categories based on learned patterns, showcasing their ability to discern complex relationships. In addition, neural networks are also usefull in autoencoders, which is a specific kind of unsupervised learning setup. Autoencoders consist of an encoder and decoder, and they can effectively compress information by transforming input data into an array and subsequently reconstructing the original data from this generated array. This dual functionality of neural networks (enabling classification tasks and facilitating unsupervised learning scenarios like autoencoders) underscores their versatility in the field of machine learning. In the context of Intrusion Detection Systems (IDS), neural networks can be utilized by training these networks with metrics that describe network traffic, associating the traffic with labels indicating whether the traffic is legitimate or indicative of a particular type of attack [3].

Studies focusing on IDS usually consider public datasets, collected in controlled scenarios with labeled data [1], [2], [4]. But this scenario is not what we have on real-world, where network traffic data are mostly unlabeled and distributed among different computers. Two specific approaches have gained attention in recent years to address the data requirements of real-world scenarios. The first is Federated Intrusion Detection System, which involves multiple detection systems being aggregated into a single network to enhance the privacy guarantees during the construction of IDS [1]. Therefore, federated learning avoids situations where models require the centralization of data obtained from different sources, and which can expose private information that can be obtained from network traffic. The second one is the Self-Supervised Learning (SSL) that emerged as an effective method to extract knowledge from unlabeled data to improve robustness of machine learning models trained with few labeled data.

This work was supported by São Paulo Research Foundation (FAPESP), grants #2018/23098-0 and #2021/04431-2, by the Coordination for the Improvement of Higher Education Personnel (CAPES), Coordination for the Improvement of Higher Education Personnel (CAPES) - Program of Academic Excellence (PROEX), and CNPq grants #141179/2021-0

This study explores the implementation of SSL to address the challenges posed by unlabeled data utilizing Federated Learning (FL) for IDS. FL is an innovative approach that enables the use of machine learning algorithms, like artificial neural networks, to build models trained with distributed and private data. The primary hypothesis investigated in this research is the feasibility of learning SSL models on individual local datasets using Federated Learning (FL) approaches, and subsequently combining them to create a unified representational model for effective IDS implementation. This hypothesis was validated through experimental analysis, wherein clients within a Federated Learning setting learned a representation model using SSL. The resulting model was then transmitted to a server to enhance the training of a supervised learning model.

This paper proceeds as follows. Section II provides an overview of important related works in the field of SSL and federated learning applied to cybersecurity. Section III describes the background knowledge to understand the proposed framework. Section IV outlines the proposed framework for Federated Self-Supervised Learning. Section V details the evaluation methodology. Finally, Sections VI and VII show the results and conclusions.

II. Related works

In the field of IDS, various approaches, including signal processing and supervised/unsupervised ML, have been studied [2]. However, these approaches often assume a centralized setting that may not be suitable for large-scale networks with distributed systems. Additionally, using ML algorithms with cybersecurity data poses challenges like class imbalances, noisy data, and the need for continuous updates to adapt to new attack patterns, which can reduce the efficiency and accuracy of IDS in real-world applications.

Federated Learning, a novel approach in ML, has gained attention due to its ability to address issues of data privacy and centralization while enabling collaboration among multiple parties [5]. This approach has been explored in the context of IDS, where ML models, such as neural networks, are trained in a distributed architecture to create a robust IDS model while preserving data privacy.

Several approaches have been proposed to apply Federated Learning in IDS. However, many of these proposals rely on scenarios where labeled data is available for clients [1], [2], [4]. Another approach involves using publicly available centralized data to train supervised learning algorithms, but this may be unfeasible when labeled data is scarce [2]. Different methods have been proposed, including utilizing autoencoders for anomaly detection [6], searching for thresholds based on reconstruction errors [7], and using Temporal Convolutional Generative Adversarial Networks (TCGAN) for synthetic data generation [8]. However, these approaches have limitations, such as the lack of flexibility in leveraging other strategies besides autoencoders, limited knowledge transfer, and difficulties in complex loss function modifications [9], [10].

While SSL has shown promise in enhancing supervised learning models, its application in cybersecurity, including IDS, is still relatively unexplored [11], [12]. Further research is needed to evaluate the effectiveness of SSL in cybersecurity. The motivation behind introducing the Federated Self-Supervised Learning (FSSL) in this article stems from a notable gap in the current research of Intrusion Detection Systems (IDS). This gap highlights the motivation of new studies to leverage the potential enhancements offered by FSSL. The innovation proposed in this article aims to address this void, specifically targeting the improvement of cyberattack detection. A key feature of this approach involves harnessing unlabeled data, easily accessible through traffic monitoring tools deployed across diverse networks. The utilization of such unlabeled data constitutes a novel avenue for fortifying the capabilities of IDS systems. Also, the adoption of the federated learning paradigm facilitates the application of this method across various scenarios, especially in cases where safeguarding data privacy is paramount. This is achieved by avoiding the transmission of network traffic data beyond local networks.

III. Background

In the field of networking, various machine-learning techniques have been employed to improve network performance and security. This research focuses on IDS, which monitor network traffic to detect and prevent unauthorized access or malicious activities. In these approaches, each network traffic connection is used to create a dataset with several connections using the packet capture file. Then, the dataset is used to build a model that can be, for instance, an artificial neural network, which can then identify patterns of suspicious behavior of new network traffic connections. Each connection is related to a label, which indicates if the connection is normal or malicious. In some applications, a malicious connection can be represented by several types of cyberattacks such as Distributed Denial of Service (DDoS), phishing, and malware infections. Furthermore, each connection is also associated with its network metadata, such as the protocol used and the type of data transmitted, which can be represented by a fixed number of features like strings and numbers.

A. Federated Learning

Federated Learning (FL) has emerged as a promising approach in the field of cybersecurity for situations where publicly labeled data is scarce [1]. It allows for the learning of models on multiple decentralized devices without exchanging sensitive information, thereby preserving privacy. Machine-learning techniques can be extended to federated learning by adapting the learning process of models. Besides, several well-known pre-processing techniques can be applied in data of federated learning, like transforming strings in numeric values and using data normalization.

A parametric model contain several values called parameters (or weights), which are used to perform predictions. The most common algorithm used in applications of FL is the

Federated Averaging (FedAvg). FedAvg consists in initializing the weights of a parametric model (usually an artificial neural network) in a central node referred to as server, which is then distributed to multiple devices in the network called clients. For neural networks, the parameters represent the set of weights that are learned in the training process. Then, several iterations named rounds are executed. A round in a FL involves that each client updates the weights of the model using its local training data and returning an updated set of weights to the server. The server then receives these updated weights and averages them to generate a new set of global model weights, which are subsequently distributed back out to the clients for the next round.

A common problem faced in applications of FL for cybersecurity data is the class imbalance issue, where normal instances outnumber[1] malicious instances [13]. The imbalance can be of different kinds. Also, a problem emerges when data is not independent and identically distributed (IID), in which each client might have a different distribution of data with different local dataset sizes. IID stands for "independent and identically distributed," which refers to a set of random variables that are statistically independent of each other and follow the same probability distribution. Usually, a solution to these issues involves techniques such as class weighting, oversampling or undersampling and incorporating regularization in applications where neural networks are used to obtain the model of classification. However, care should be taken when applying these techniques to maintain the privacy of client data, which is a concern when using Federated Learning.

B. Self-Supervised Learning

Self-Supervised Learning is a ML approach that enables models to learn from unannotated data, reducing the reliance on labeled training data [11]. SSL is particularly valuable in scenarios where labeled data is limited or costly to obtain, such as in cybersecurity applications. By leveraging unlabeled data, SSL can enhance model performance and aid in identifying and mitigating security threats.

SSL involves two steps: pre-training on unlabeled data to capture the underlying data structure and fine-tuning on labeled data for downstream tasks like classification or anomaly detection. During pre-training, models can learn to predict data characteristics or generate synthetic examples, such as rotating unlabeled images and using the rotation angle as labels [11]. This pre-training step is known as the pretext task.

In the second step, called knowledge transfer, the pre-trained model is utilized. Knowledge transfer involves feature extraction or fine-tuning. Feature extraction transforms the input data into abstract representations that serve as input for downstream models trained with labeled data. Fine-tuning modifies the pre-trained model by training it on labeled data for more accurate classification or anomaly detection. For instance, two neural networks can be employed (one for the pretext task and one for the target task) with shared architecture [9]. The parameters learned by the pretext task model can partially initialize the target task model. Although SSL offers advantages, challenges like pretext task definition and effective knowledge transfer need to be addressed.

C. AutoEncoders

An autoencoder is a neural network architecture that aims to reconstruct its input as accurately as possible [6]. It consists of two components: the encoder, which compresses the input data into a lower-dimensional representation called the "latent space", and the decoder, which attempts to reconstruct the original input from the latent representation. What makes autoencoders particularly useful is their ability to learn from unlabeled data, enabling unsupervised learning.

Autoencoders can be seen as compression models, with the choice of loss function playing a crucial role in measuring the similarity between the input and the reconstructed output. While achieving a perfect reconstruction is typically unattainable, autoencoders have proven successful in various applications. Two common uses of autoencoders include generating meaningful representations by training on unlabeled data and leveraging the reconstruction error to detect anomalies. By training an autoencoder on unlabeled data, the encoder can extract valuable features as inputs encodings using high-dimensional arrays for visualization or clustering. Alternatively, the reconstruction error can be employed to identify instances that significantly deviate from the majority of the training data, suggesting the presence of anomalies or outliers. In this article, autoencoders will be explored by their capacity to create representation models using the encoder part of the architecture, which can be trained using unlabeled data.

D. Intrusion Detection Systems

IDS is a critical component of network security, providing mechanisms for detecting and reacting to suspicious activity patterns outside of normal behavior [2]. One approach to improving IDS involves applying ML algorithms, such as anomaly-based and deep learning-based methods. In IDS, the network traffic can be captured in a packet capture file and used to develop machine-learning algorithms for anomaly detection. These algorithms use mathematical techniques to learn, profile, classify and predict unusual results in the network traffic, which are indicative of potential attacks on a system.

Several datasets were created to evaluate IDS solutions [12]. These datasets usually contain several network flows (or connections), where each flow represents a set of network packets sent from an origin to a destination using a fixed number of metrics like timestamp, connection duration time, and transmitted bytes, among others. Also, each flow is related to a label that indicates if the flow is related to normal traffic or a type of cyberattack. Therefore, by analyzing these flows and identifying anomalous behavior patterns, IDS can be evaluated by detecting potential threats and compared with the labels. It is important to avoid false negatives in IDS to minimize the risk of missing cyberattacks, while also minimizing false

[1]There are exceptions like DDoS attacks.

positives to avoid unnecessary alerts leading to disruption of normal system operations. For this reason, it is important to investigate novel approaches that achieve higher accuracy in intrusion detection while keeping false positives and false negatives to a minimum [1].

IV. A framework for Federated Self-Supervised Learning for IDS

The motivation behind this research is to address the challenges faced by Intrusion Detection Systems (IDS) that employ Federated Learning (FL) in detecting cyber-attacks. A novel framework for Federated Self-Supervised Learning is proposed in this article. The framework aims to combine the benefits of federated learning and SSL techniques. The framework considers the use of unlabeled data, a characteristic that is not considered in all Federated Learning techniques.

The framework leverages SSL allowing the model to learn from large quantities of unannotated data by generating annotations and labels internally in each federated client. In this way, privacy and critical data can be protected without leaving the clients while improving intrusion detection models. In Figure 1, a scenario is presented where a target task is being solved using a supervised learning model trained with limited data. Additionally, the server can take advantage of clients, where a large amount of unlabeled data is available. In this problem, the supervised learning model should be able to deal with new data that can have potentially different data distribution compared to the training data, see Figure 1.

The proposed framework assumes a pool of pretext tasks to be executed in each client, which corroborates with strategies based on personalized learning. Each client creates their task-specific patterns autonomously depending on patterns observed in their data. However, an essential key is required in this framework: all clients should share a common representation model. This characteristic is necessary because these representation models are aggregated during federated training to compose a single global representation model. For instance, neural networks can solve each client pretext task, which all neural networks shares at least the initial layers that serve as a common representation model, but each model will be trained with different clients data. Then, several rounds can be executed until all clients reach desirable convergence criteria, providing an effective model representation using unlabeled data. A simple and effective method for creating representation models from unlabeled data involves training AutoEncoders on this data. AutoEncoders are trained to encode the input data into a latent vector, which is then utilized to reconstruct the original input. By assessing the reconstruction error, we can determine the model's ability to learn an effective representation through its encoder. This learned representation supports the transfer of learning approaches.

After training a model representation using one or more pretext tasks among clients in federated learning, knowledge transfer is performed. The goal is to utilize the representation method to enhance training efficiency with limited labeled data. This can be achieved using various methods discussed in the literature on self-supervised learning (SSL) [11] and transfer learning [10]. For example, the representation method can be applied in a straightforward process where all input data is transformed before being used in training the supervised learning model. Although this technique is simple, it allows for the combination of different approaches. For instance, representation models can be learned using neural networks, and then these models can be used to transform data for other machine learning algorithms besides neural networks.

Another possibility for implementing knowledge transfer is sharing the first layers of neural network architectures of the pretext task models and the model that solves the target task and using the representation model learned through SSL to initialize the parameters (weights) of the supervised learning model. This approach can boost the convergence of the supervised model and improve its accuracy, besides enabling the training process to achieve smaller minimum local errors through common optimization methods used to train neural networks like Adam [14] and SGD [15].

Achieving good minimum local errors is an essential factor when training neural networks, particularly for complex tasks that require a significant amount of labeled data. However, it is known that it is impossible to ensure convergence when the desired neural network output is similar or equal to non-convex functions. This is the case of several problems like IDS, where is possible to exist non-linear data where different types of network traffic share the same representation. Therefore, the proposed framework can help mitigating the convergence challenge faced in the scenarios mentioned before. The mitigation can be achieved through knowledge transfer, which allows the supervised learning model to leverage the advantage of pre-trained representation learning. This can help reducing the occurrence of multiples local minima during training, providing a head start in the training process.

V. Experimental setup

The proposed FSSL framework was compared to a baseline framework approach using three datasets of network traffic data commonly used for evaluating IDS. Both of these frameworks are described in Figure 1. The baseline approach consists in a centralized (non federated) training, where private and unlabeled data distributed among clients is not used. This section describes the datasets used in the experiments, as well as the details of the implementation of the baseline and FSSL approaches. Also, it details the evaluation methods. The datasets used in the experiments[2] are: TonIoT (461.043 instances and 10 classes), BotIoT (2.942.153 instances and 5 classes), and NSL-KDD (131.594 instances and 30 classes). These datasets were processed to create settings equivalent to the presented in Figure 1.

For each dataset, a stratified random subsample equivalent to 1% of the entire set was chosen as labeled data, resulting in a reduced training data size. The remaining 99% of each dataset was used to train the autoencoder models. The labeled data

[2]Datasets public available: TonIoT, BotIoT, and NSL-KDD

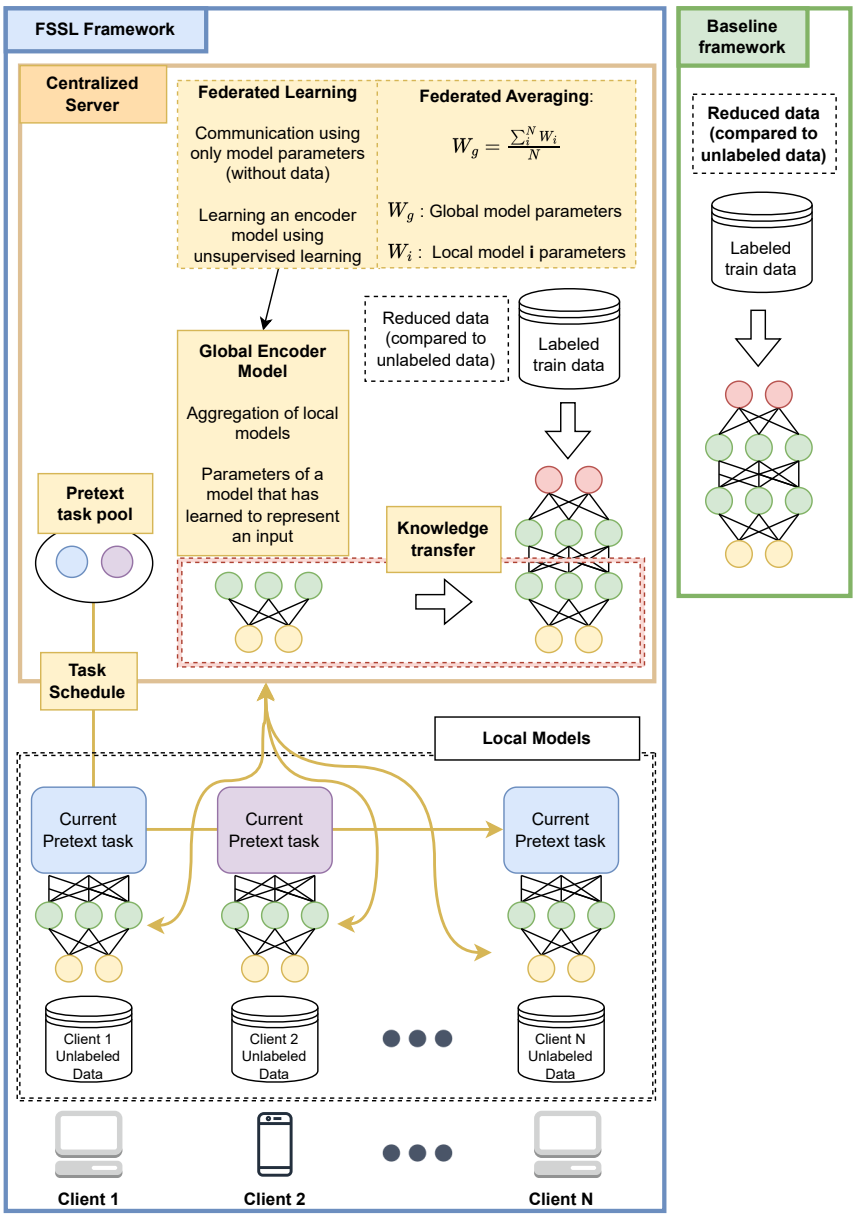

Fig. 1. The FSSL and baseline frameworks.

was carefully selected to represent both normal and anomalous cyberattack samples. To create different scenarios, the remaining data was divided into 10 groups representing different federated clients. The clients' data can be categorized as either IID or Non-IID based on their level of data homogeneity or heterogeneity. This was achieved using a technique from a previous work [16] that employs the Dirichlet distribution to control the degree of data homogeneity/heterogeneity across clients. We used two parameter values, 0.1 for high heterogeneity and 100.0 for low heterogeneity, to create the two considered scenarios.

Three methods were used to create a detection model for cybersecurity using three datasets, which in each method can be aided by non-IID or IID non-labeled data distributed among ten clients. The first method assumes a simple and common approach where only labeled data is used to train a supervised learning neural network model, which will be referred to as a baseline method. In contrast, the FSSL framework was implemented to take advantage of the nonlabeled data in the same scenario using the non-IID or IID data. An autoencoder was used to implement the pretext task of the FSSL framework for learning representations from non-labeled data. The autoencoder shares the first layer of the neural network that was used in the target model (the neural network used to detect attacks). The autoencoders were trained using 10 rounds with the FedAvg algorithm, and then the averaged autoencoder was used to initialize the first layer of the target model, which is then trained in the same way as in the baseline method.

A. Architectures and validation methods

The implementation of the autoencoder and classification deep learning models was carried out using the Keras and Tensorflow libraries (version 2.6) in Python3. Note that this research does not focus on exploring neural network architectures, leaving room for other works to extend the presented experiments with different configurations. The autoencoder was inspired by the architecture used by [17] and consisted of a multilayer perceptron with one hidden layer containing 1000 units, utilizing linear activation and a dropout ratio of 0.2. The final layer of the autoencoder employed the relu activation function and the categorical cross-entropy loss function. For the classification model, the initial layers were shared with the autoencoder architecture, while the final layers used softmax activation function and the categorical cross-entropy loss function. The RMSprop algorithm was selected as the optimizer for training the neural networks, with the default parameters of the Tensorflow library being utilized. The autoencoders were trained using 10 rounds of federated learning communication. The classification neural networks were trained using 300 epochs. The evaluation of the three models was made using a 10-fold validation (cross validation) [5] and the average F1-score of each trained model was measured. The results were compared between scenarios to different levels of homogeneity/heterogeneity by the t-student statistical test to identify significant differences in the performance considering the F1-score. To guarantee the reproducibility of our results, we will provide access to the source code, result data, and specific analyses, such as the t-student test, through the following GitHub repository: https://github.com/BrunoMeyer/FSSL-SCCI2023.

VI. Results

In Table I, it is possible to observe the macro average F1-score for each method and scenario, as well as the specific F1-score for the detection of each network traffic type. In the table, FSSL* represents the FSSL model trained with Non-IID data). The results of the experiments conducted in this study demonstrate the effectiveness of using FSSL over a baseline approach, especially in scenarios where IID data is used. An example of this can be seen in the BotIoT dataset, where the FSSL achieved a macro average F1-Score of 0.508, while the baseline approach only achieved 0.305. All macro-average F1-Scores showed significant statistical differences when evaluated using the t-student test and a 10-fold validation strategy for Ton-IoT and Bot-IoT (p-values smaller than 0.025). While FSSL achieved a slightly better average F1-Score in the NSL-KDD dataset, there was no statistically significant difference (p-value ¡ 0.05). This might be due to the dataset's complexity with multiple classes, necessitating other techniques to address challenges not explored in this study. An interesting result from the experiments is that in the scenario where non-IID data is used, the FSSL framework achieved a slightly lower average F1-score compared to the scenario where IID data is used to train the pretext task model. This characteristic is a promising discovery for cybersecurity data, where datasets are often non-IID due to the diversity and variability of attacks, requiring strategies to normalize or compensate for the fact that some federated clients have more data than others.

TABLE I
AVERAGE F1-SCORE AND TRAINING/TESTING INSTANCES ARE SHOWN, INCLUDING FSSL* (TRAINED WITH NON-IID DATA). MACRO-AVERAGE'S STANDARD DEVIATION IS PRESENTED USING 10-FOLD VALIDATION.

Dataset	Instance type	Train	Test	Baseline	FSSL	FSSL*
TonIoT	*Macro Average*	502	4519	0.853 ± 0.013	**0.873** ± 0.013	0.871 ± 0.017
	backdoor	21	196	0.914	0.941	0.954
	ddos	21	195	0.721	0.737	0.743
	dos	21	196	0.890	0.887	0.876
	injection	21	196	0.461	0.588	0.590
	mitm	1	10	0.000	0.013	0.000
	normal	326	2940	0.917	0.924	0.925
	password	21	196	0.592	0.738	0.731
	ransomware	21	196	0.736	0.706	0.683
	scanning	21	196	0.826	0.861	0.846
	xss	21	196	0.795	0.791	0.795
BotIoT	*Macro Average*	997	8980	0.305 ± 0.036	**0.508** ± 0.069	0.384 ± 0.047
	DDoS	524	4716	0.211	0.567	0.393
	DoS	448	4040	0.435	0.435	0.368
	Normal	1	1	0.000	0.000	0.000
	Reconnaissance	24	222	0.000	0.536	0.483
NSL KDD	*Macro Average*	573	5161	0.883 ± 0.011	0.889 ± 0.012	**0.892** ± 0.014
	apache2	18	165	0.934	0.928	0.917
	back	9	81	0.151	0.293	0.362
	mailbomb	1	4	0.693	0.763	0.792
	mscan	30	276	0.860	0.872	0.870
	multihop	3	29	0.000	0.000	0.000
	named	3	34	0.040	0.000	0.040
	neptune	1	1	0.989	0.989	0.990
	nmap	7	65	0.951	0.965	0.962
	normal	24	224	0.915	0.915	0.918
	pod	1	3	0.338	0.292	0.295
	buffer_overflow	1	3	0.000	0.000	0.000
	portsweep	120	1080	0.689	0.683	0.677
	processtable	1	17	0.924	0.911	0.909
	ps	248	2239	0.031	0.043	0.046
	rootkit	1	9	0.107	0.117	0.107
	saint	4	37	0.470	0.477	0.461
	satan	17	153	0.782	0.786	0.808
	sendmail	1	2	0.000	0.000	0.000
	smurf	1	2	0.873	0.868	0.867
	snmpgetattack	7	71	0.551	0.605	0.614
	snmpguess	18	167	0.633	0.669	0.681
	teardrop	1	2	0.042	0.050	0.043
	warezclient	16	152	0.000	0.000	0.000
	warezmaster	4	39	0.867	0.860	0.860
	xlock	8	73	0.000	0.000	0.000
	xsnoop	1	3	0.000	0.000	0.000
	xterm	1	1	0.000	0.000	0.000
	guess_passwd	23	212	0.896	0.905	0.904
	httptunnel	1	1	0.730	0.743	0.729
	ipsweep	1	1	0.801	0.767	0.768
	land	1	2	0.000	0.000	0.000

When examining the F1-score for each individual class, it becomes evident that certain classes, such as "Normal" in the BotIoT dataset, present significant challenges in terms of accurate identification. In this particular case, the training data for the detection model consisted of only one sample of the "Normal" class, resulting in an extremely imbalanced scenario that needs the implementation of additional techniques to enhance the F1-score. Both FSSL and FSSL with non-IID data demonstrated notable improvements in detecting certain classes, such as "Reconnaissance" in the BotIoT dataset. The F1-score for this class increased from 0.0 in the baseline approach to 0.536 when utilizing FSSL. This finding provides

evidence that the proposed framework enhances the detection of cyberattacks that conventional supervised learning models struggle to identify when trained with limited training data.

One important aspect to analyze in the experimental results is the relationship between the average number of instances used to train the supervised model and the achieved F1-score. Similar to real-world scenarios, the network traffic in the datasets used in the experiments exhibits a high degree of data imbalance. For example, in the BotIoT dataset, the class "Normal" has an average of only one instance for training and another for testing. As a result, the classification inefficiency is observed across all compared methods for this class, leading to an F1-score of 0.

A possible evidence for why FSSL outperforms the baseline approach in terms of efficiency becomes apparent when examining the number of training instances for certain classes and the resulting F1-Score. For example, consider the "Reconnaissance" class in the BotIoT dataset, which only has 24 training instances out of a total of 997 instances, and was tested using 222 instances. Although this is not as extreme as the previously mentioned "Normal" class, it still represents an imbalanced scenario. FSSL achieved an average F1-Score of 0.536 for IID data and 0.483 for non-IID data, whereas the baseline approach achieved an F1-Score of 0. In other words, FSSL generally enhances the detection of certain classes even when a small number of training instances are provided to the classification model. However, it is still necessary to have at least a minimal amount of training data. This characteristic can also be observed for other classes in the experimental results presented in Table I. For instance, in the TonIoT dataset, the "Men-In-The-Middle" class ("mimt") showed a slight improvement in F1-Score from 0.0 to 0.013, and the "password" class shows an improvement from 0.592 to 0.738.

VII. Conclusion

This paper presented and evaluated a Federated Self-Supervised Learning (FSSL) framework for Intrusion Detection Systems. The framework demonstrated promising results in accuracy and F1-score, outperforming baseline approaches and highlighting the potential of using unlabeled data to improve supervised models in cybersecurity intrusion detection using federated learning settings. The FSSL framework has the potential to enhance the effectiveness and convergence speed of supervised models while addressing privacy and security concerns associated with centralized data collection in IDS implementation. Overall, the results of the experiments showed that the proposed Federated Self-Supervised Learning framework outperformed the baseline approach improving the F1-score ratio in up to 20%. The autoencoders used in the pretext task were used mainly due to their simplicity to be implemented and to avoid possibly unnecessary complexity in the evaluation of the proposed FSSL framework. Although it is expected that specific cybersecurity-related pretext tasks can be explored to improve the advantage of knowledge transfer using unlabeled data, we will address further analysis in future work to confirm this hypothesis. Furthermore, a simple neural network architecture was also adopted to avoid unnecessary complexity to interpret the results. Future works can explore the usage or the search for more complex architectures and identify if the results found in this research continue to show the same behavior in the analyzed scenarios or other possible evaluation methods. Additionally, this study focused on scenarios with limited labeled training data compared to unlabeled data, and further investigation is required to assess the advantages and disadvantages of the proposed framework in other scenarios.

References

[1] S. Arisdakessian, O. A. Wahab, A. Mourad, H. Otrok, and M. Guizani, "A survey on iot intrusion detection: Federated learning, game theory, social psychology and explainable ai as future directions," *IEEE Internet of Things Journal*, 2022.

[2] E. M. Campos, P. F. Saura, A. González-Vidal, J. L. Hernández-Ramos, J. B. Bernabé, G. Baldini, and A. Skarmeta, "Evaluating federated learning for intrusion detection in internet of things: Review and challenges," *Computer Networks*, vol. 203, p. 108661, 2022.

[3] N. Moustafa, "A new distributed architecture for evaluating ai-based security systems at the edge: network ton_iot datasets. sustain. cities soc. 72, 102994 (2021)," 2021.

[4] M. Alazab, S. P. RM, M. Parimala, P. K. R. Maddikunta, T. R. Gadekallu, and Q.-V. Pham, "Federated learning for cybersecurity: concepts, challenges, and future directions," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 5, pp. 3501–3509, 2021.

[5] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, "A review of applications in federated learning," *Computers & Industrial Engineering*, vol. 149, p. 106854, 2020.

[6] J. Schneible and A. Lu, "Anomaly detection on the edge," in *MILCOM 2017-2017 IEEE military communications conference (MILCOM)*. IEEE, 2017, pp. 678–682.

[7] T. Zhang, C. He, T. Ma, L. Gao, M. Ma, and S. Avestimehr, "Federated learning for internet of things," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, 2021, pp. 413–419.

[8] M. Abdel-Basset, N. Moustafa, and H. Hawash, "Privacy-preserved cyberattack detection in industrial edge of things (ieot): A blockchain-orchestrated federated learning approach," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7920–7934, 2022.

[9] O. Aouedi, K. Piamrat, G. Muller, and K. Singh, "Federated semisupervised learning for attack detection in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 286–295, 2022.

[10] T. V. Khoa, D. T. Hoang, N. L. Trung, C. T. Nguyen, T. T. T. Quynh, D. N. Nguyen, N. V. Ha, and E. Dutkiewicz, "Deep transfer learning: A novel collaborative learning model for cyberattack detection systems in iot networks," *IEEE Internet of Things Journal*, 2022.

[11] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2020.

[12] P. K. Mvula, P. Branco, G.-V. Jourdan, and H. L. Viktor, "A systematic literature review of cyber-security data repositories and performance assessment metrics for semi-supervised learning," *Discover Data*, vol. 1, no. 1, p. 4, 2023.

[13] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, and K. Rieck, "Dos and don'ts of machine learning in computer security," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 3971–3988.

[14] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," *arXiv preprint arXiv:1904.09237*, 2019.

[15] J. Konečnỳ, B. McMahan, and D. Ramage, "Federated optimization: Distributed optimization beyond the datacenter," *arXiv preprint arXiv:1511.03575*, 2015.

[16] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 713–10 722.

[17] J. Wang, J. Hu, J. Mills, G. Min, M. Xia, and N. Georgalas, "Federated ensemble model-based reinforcement learning in edge computing," *IEEE Transactions on Parallel and Distributed Systems*, 2023.