METALS : seMi-supervised fEderaTed Active Learning for intrusion detection Systems

Ons Aouedi, Gautam Jajoo, Kandaraj Piamrat

METALS : seMi-supervised fEderaTed Active Learning for intrusion detection Systems

Ons Aouedi[‡], Gautam Jajoo[*] Kandaraj Piamrat[†]
[‡] SnT, SIGCOM, University of Luxembourg, Luxembourg
[*] BITS Pilani, India
[†] Nantes University, École Centrale Nantes, IMT Atlantique, CNRS, INRIA, LS2N, UMR 6004, Nantes, France

***Abstract*—Recent studies have explored the potential of Machine Learning (ML) for intrusion detection systems (IDS) in the Internet of Things (IoT) system. However, low latency and privacy requirements are important in emerging application scenarios. Furthermore, due to limited communication resources, sending the raw data to the central server for model training is no longer practical. It is difficult to get labeled data because data labeling is expensive in terms of time. In this paper, we develop a semi-supervised federated active learning for IDS, called (METALS). This model takes advantage of Federated Learning (FL) and Active Learning (AL) to reduce the need for a large number of labeled data by actively choosing the instances that should be labeled and keeping the data where it was generated. Specifically, FL trains the model locally and communicates the model parameters instead of the raw data. At the same time, AL allows the model located on the devices to automatically choose and label part of the traffic without involving manual inspection of each training sample. Our findings demonstrate that METALS not only achieve a high classification performance, comparable to the classical FL model in terms of accuracy but also with a small amount of labeled data.**

***Index Terms*—Federated Learning, Active Learning, Intrusion Detection System, Internet of Things, Cybersecurity.**

I. Introduction

The rapid expansion of wearable devices and sensors in the Internet of Things (IoT) has catalyzed an unprecedented surge in data generation, ushering in an era of ubiquitous computing [1]. This phenomenon, where devices are interconnected in a seamless network, facilitates the continuous exchange of data, thereby enhancing the utility and functionality of IoT ecosystems. However, this technological revolution brings significant security challenges and privacy concerns. The decentralization and ubiquitous nature of data generation and processing in IoT networks increase the vulnerability to cyber-attacks and data breaches. In this context, securing the myriad points of data exchange and processing across the network becomes a paramount concern. Furthermore, the main issues during the development of intelligence-based detection systems are (i) collecting data on a central server and (ii) labeling these data, which is time-consuming [2]. Specifically, analyzing and collecting these data in a centralized framework presents significant challenges, particularly in terms of privacy, latency, and scalability. Centralized data analysis systems, while powerful, often fail to address these critical issues effectively. To solve this problem, Federated Learning (FL) has been used by allowing a group of devices or clients to collaborate on a shared model while keeping their raw data locally stored [3]. Instead of transferring the sensitive data to a central location, FL operates by training machine learning (ML) models on each device and then sharing and aggregating the updated model.

In addition, given the nature of FL and IoT systems where devices are distributed and often beyond human reach, accessing them to label their data can be a difficult and impractical task [2]. Furthermore, the data labeling task does not ensure the selection of high-quality samples, which are more significant to the model and impact its performance. Thus, the use of Active Learning (AL) is a promising approach to deal with the need for a huge amount of labeled data. With AL, the learning algorithm selectively queries labels for informative data points and presents a solution to the labeling challenges in FL. Additionally, it can achieve an exponential acceleration in labeling efficiency [4]. Therefore, the combination of FL and AL can enhance IDS efficiency using both labeled and unlabeled data. Therefore, this paper presents a semi-supervised federated active learning for IDS, called `METALS`, a distributed and intelligent method to help the network security expert to label network data and detect attacks without privacy concerns. The main contributions of this paper are summarized as follows.

- We introduce `METALS`, a semi-supervised federated active learning framework for IDS, enhancing data utilization in a federated context.
- We conduct evaluations of various AL strategies, including *Random Sampling*, *Entropy Sampling*, *Margin Sampling*, and *Least Confidence Sampling*.
- We compare `METALS`' effectiveness against traditional FL, using FedAvg and FedProx, to highlight its performance benefits.

The rest of the paper is organized as follows. Section II provides related work and Section IV presents our proposition `METALS`. The experimental settings and results are presented in Section V. Finally, the conclusion is provided in Section VI.

II. Related Work

Most of the existing FL-based IDS solutions are based on supervised learning algorithms [5] where they have focused only on the labeled data without taking advantage of the

unlabeled data. As a result, under such conditions, the models can overfit on the labeled data. Due to these limitations, researchers have shifted towards semi-supervised FL frameworks to leverage both labeled and unlabeled datasets more effectively. In this context, the authors in [2], [6], proposed for the first time, a semi-supervised FL model for IDS. In particular, the devices learn only the data representations through an unsupervised model (autoencoder). Then, the FL server not only generates a global model, but it exploits a small amount of labeled data to conduct supervised learning. Similarly, the study in [7] proposed a new Federated Self-Supervised Learning framework for IDS, called FSSL. In particular, FSSL also used an autoencoder model for feature extraction tasks using unlabeled data and the transfer of knowledge for model parameters initialization with the encoder layers.

The work in [8] provided a CNN-based semi-supervised method. The authors assumed that each node initially is trained with labeled data so that the traffic information obtained from other unlabeled data is classified using a discriminator considering the previous training. Moreover, the authors in [9] employed a transfer learning-based methodology, where a cloud model is trained with labeled data and this information is used by the end nodes for attack classification on unlabeled data. In particular, the authors develop a better representation of complicated attacks using a graph-based model known as a subgraph aggregated capsule network (SACN). Among all, the most closely related work has been proposed in [10] where the authors proposed a novel FL-empowered semi-supervised active learning (FL-SSAL) framework within a zero-touch network and service management (ZSM) architecture. Using entropy-based active learning, their framework selectively annotates the most informative samples from unlabeled datasets, achieving superior intrusion detection accuracy with reduced annotation and communication overheads compared to traditional techniques.

Although these existing works explore semi-supervised learning and FL for IDS, they primarily focus on using unlabeled data for feature extraction or initial model training. In addition, most of the proposed solutions, including the use of autoencoder for feature extraction and CNN-based methods for semi-supervised learning, are based heavily on specific model architectures. Although the FL-empowered semi-supervised active learning framework proposed by [10] addresses annotation overheads through AL, the authors used the entropy strategy, and the performance analysis was not well explored.

III. Background

For a better understanding of the paper, we first present the two main concepts that have been used in our proposal: federated learning and active learning.

A. Federated Learning (FL)

FL is an iterative process composed of two main steps: local learning and model transmission. Each iteration may enhance the global model [11]. To begin an iteration, the FL server chooses a subset of clients to participate in the learning process, sending them the global model. Once the global model is received, each client performs local training using their local data and then sends back their updated model (i.e., the learned parameters) to the FL server for global aggregation. This process is repeated several rounds until the model's performance meets the desired criteria. Therefore, an FL scenario can be broken down into two primary phases: *local update* and *global aggregation* [12]. This demonstrates that with the help of FL, clients can benefit from other clients' data without sending their sensitive personal data to a central server. In this work, we use Federated Averaging (FedAvg) [3] and FedProx [13] for global aggregation due to their simplicity, effectiveness, and robustness.

B. Active Learning (AL)

AL is an ML technique that focuses on labeling as little amount of data as possible while still achieving performance increases. More specifically, it aims to reduce the need for labeled data by intelligently querying the labels during training. It helps to select the most useful samples from the unlabeled dataset and hand them over to the oracle (e.g., human annotator) for labeling, to reduce the cost of labeling as much as possible while still maintaining performance [14]. AL seeks to optimize the performance of ML models while minimizing the burden and cost associated with labeling numerous instances. The well-known AL strategies are *Random sampling*, *Entropy sampling*, *Margin sampling*, and *Least confidence*. A brief explanation of these methods is provided as follows.

- *Random sampling*: Selects data samples in a random way. It is simple and easy to implement.
- *Entropy sampling*: Select the data samples with the highest entropy scores. Therefore, this algorithm focuses on the data samples that are the most uncertain.
- *Margin sampling*: Select the data samples that have the least difference between the probabilities of the two most probable classes, which means that it selects data samples that are on the decision boundary. Thus, it selects the most informative data points for the model.
- *Least confidence*: Select the data samples for which the model is least confident, i.e., with the lowest confidence scores. Thus, it focuses on the data samples that the model is least confident about.

IV. The METALS Proposal

In this section, we present the methodology of METALS (a se**M**i-supervised f**E**dera**T**ed **A**ctive **L**earning for intrusion detection **System**), a novel approach designed to optimize IDS in distributed distributed computing environments, specifically within IoT. As shown in Fig.1 and Algorithm 1, METALS takes advantage of FL and AL to address the challenges of security, data privacy, and efficiency in processing the voluminous unlabeled data generated by IoT devices. In particular, the proposed METALS begins with an initial step

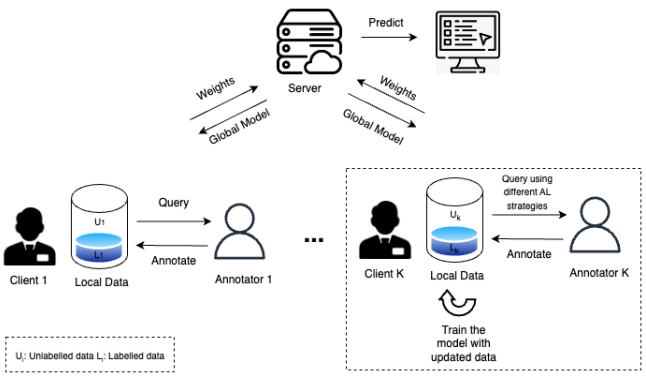

Fig. 1. METALS model for IDS for IoT environments.

of model initialization on the FL server. This initial model, denoted M_0 is deployed in a predefined set K of devices that participate in the network. These devices contribute to the learning process by providing labeled and unlabeled data during the communication round T. In round t, a subset of e devices is selected to actively participate in model training and enhancement. The AL algorithm within the e devices selects a subset S_t^e from $D_{unlabeled}^e$, considered the most informative for learning. This selection is based on an AL strategy designed to identify samples that, once labeled, can significantly enhance the performance of the global model. After identifying and labeling informative samples, each device updates its labeled dataset $D_{labeled}^e$ to include S_t^e and trains a local model M_t^e in the new augmented dataset. Upon completion of the training in all participating devices, the local models M_t^e are aggregated on the FL server to update the global model M_t. This aggregation process is crucial, as it synthesizes the learning achieved across the network into a coherent model that benefits from the diverse data and perspectives of all participating devices. The iterative process of training, AL, and aggregation of models continues for T rounds and continues to find the final model M_T. This model is then evaluated on a separate test dataset to assess its performance and effectiveness as an IDS. Upon satisfactory evaluation, M_T is deployed across IoT devices for inference tasks, thus improving the security posture of the entire distributed IoT environment. Algorithm 1 presents the main learning process of `METALS`.

Algorithm 1: Learning procedure of METALS:

Input: Labeled dataset from IoT devices $D_{labeled}$; Unlabeled dataset from edge devices$D_{unlabeled}$; Edge devices K; Training round T; Participants IoT devices in each round e
Output: Trained IDS model M
Model initialisation M_0 on FL server
Deploy M_0 to K
for each $t = 1, 2, ..., T$ **do**
  for each $K = 1, 2, ..., e$ **do**
    Use M_{t_1} to predict on $D_{unlabeled}^e$
    Apply AL algorithm to select informative S_t^e from $D_{unlabeled}^e$
    Request labels for S_t^e
    Update $D_{labeled}^e$ to include S_t^e
    Train local model M_t^e on updated $D_{labeled}^e$
  End for
  Aggregate local models M_t^e at FL server to update global model M_t
  Send the global model M_t to K
End for
Evaluate the final model M_T on a separate test dataset
Return M_T

V. Experiment and Results

In this section, we first present the dataset used during the experiment. Then, we detail the model architecture used within `METALS` solution. Finally, the results are discussed and analyzed.

A. Dataset description

To evaluate the performance of `METALS`, we have chosen one of the most recent datasets, which is called *Edge-IIoTset* published in 2022 [15]. This dataset consists of 1526482 observations, 61 features, and 1 label, the label contains 15 possible values, benign, and seven different types of attacks[1]. Before training the models, the data are normalized using the Min-Max scaling technique, so all their values are in the range of [0, 1] to optimize the performance of the training process.

B. Experimental Setup

We conducted our experiments with `Python3` as a programming language, `Scikit-learn` for the conventional models, and `PyTorch` for `METALS`. Also, all experiments were run using four core Intel Core i7-6700 CPU@3.40GHz processor, and 32.00 GB of RAM. The code is available online.[2]

1) Data preprocessing: The preprocessing includes steps such as column selection to remove non-informative or unnecessary flow features, handling missing values by removing rows, duplicate removal, one-hot encoding for categorical data transformation, mapping target labels for the 'Attack_type' variable, feature removal for noise reduction, and dataset splitting into training, validation, test, and labeled sets. Similarly to the dataset paper [15], we use the Synthetic Minority Oversampling Technique (SMOTE) [16] to synthesize new examples to oversample minority classes (precisely MITM, Ransomware, and Fingerprinting). This targeted oversampling aims to improve the overall effectiveness of the model. Furthermore, to simulate a partially labeled dataset, we select a portion of the samples randomly and use them as unlabeled data.

[1]https://www.kaggle.com/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot

[2]https://github.com/gautamjajoo/FAL

2) Model: Followed by the model proposed in [15], our model architecture consists of an input layer, two hidden layers, and an output layer. Each layer (except the output layer) is followed by a Rectified Linear Unit (ReLU) activation function, which introduces non-linearity into the model. The output layer does not have an activation function since we use it for multiclass classification. Table I shows the different parameters of our model. Table II, and Table III summarize the METALS and the classical FL settings in our simulations, respectively.

TABLE I
MODEL HYPERAPARAMETERS

Hidden Layers	2
Hidden Nodes	90
Regularization	L2
Activation function	ReLU
Classification function	softmax
Optimizer	Adam
Learning Rate	0.001
Client epochs	10
Server Epochs	10
Batch Size	100

TABLE II
METALS IMPLEMENTATION PARAMETERS

No. of clients	100
Fraction of clients for each round	0.2
Initial labeled data	20%
% of data labeled in each round per client	20%
Communication round	50
Federated Aggregator	FedAvg/FedProx

TABLE III
CLASSICAL FL IMPLEMENTATION PARAMETERS

No. of clients	10
Fraction of clients for each round	1
Possible values of μ	0.3, 0.5, 0.8, 1
Communication round	10
Federated Aggregator	FedAvg/FedProx

C. Results

In this section, we provide a comprehensive benchmark of METALS against classical FL while varying aggregation algorithms (e.g., FedAvg and FedProx) under different label percentages and AL strategies. We also provide an analysis of the impact of communication rounds.

1) Performance of classical FL: In this subsection, we provide the performance evaluation of classical FL with two aggregators (FedAvg and Fedprox). With FedProx, we also varied the value of μ, which affects the performance of FedProx, and Fig. 2 presents the obtained accuracy. In particular, all algorithms improved their performance from the 1^{st} round to the 10^{th} round and converged to almost the same global accuracy. This suggests that the FL process was effective in improving the model's performance. Furthermore, it can be observed that FedProx(0.5), meaning $\mu = 0.5$, gives the highest accuracy in all rounds. The improvement between rounds is smaller than that of other FedProx variants and FedAvg. This suggests that FedProx(0.5) might converge faster to a locally optimal solution, while other FedProx variants need more rounds to improve. Also, it can be seen that FedProx(1) might converge faster initially but potentially sacrifice final performance due to its larger penalty. In general, a higher value of μ can accelerate learning but can also increase instability and model overfitting. Moreover, we can notice that FedProx generally achieves better accuracy and F1-score, indicating that it might better balance precision and recall compared to FedAvg. This could be due to the regularization effect of the FedProx penalty, which can help prevent overfitting and improve generalization. Nevertheless, FedAvg's performance is more stable across rounds, while FedProx variants show more variation. This could be due to the FedProx penalty that introduces some noise into updates, which could lead to occasional performance fluctuations.

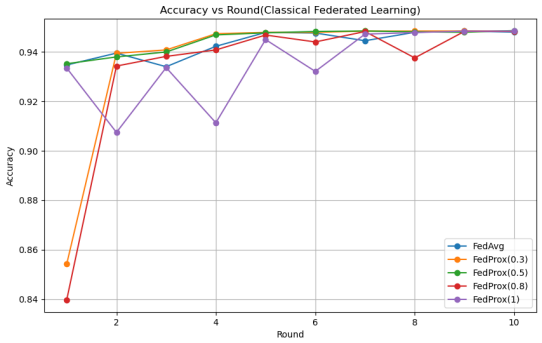

Fig. 2. Accuracy with classical FL framework

2) Performance with different number of labeled samples: To evaluate the effectiveness of the proposed METALS, we conducted several experiments with varying amounts of labeled data. In these first experiments, we assume that 20% of the client dataset is initially labeled and 20% of the unlabeled data of the clients is labeled and then added to the labeled dataset at the sampling step of each round. Fig. 3, Fig. 4, and Fig. 5 illustrate the impact of different AL sampling strategies on global accuracy in an FL context using Fedprox. From these figures, we can interpret that all AL sampling strategies show an increase in global accuracy as more labeled data are added. This trend is expected since more labeled data typically provide the model with more information to learn from, which improves its ability to generalize to unseen normal/attack traffic. However, we also noticed that adding more labeled samples does not always increase global accuracy. Specifically, there is a noticeable decrease in performance for all strategies with a huge amount of data. This decrease could be indicative of overfitting, where the model starts to memorize the training data rather than learning to generalize from it. Overfitting typically occurs when a model is trained on too much data that includes noise or noninformative samples, leading to decreased performance during the inference.

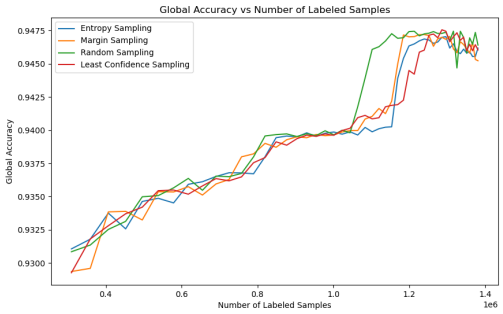

Fig. 3. Global modal accuracy for various AL methods with numbers of labeled samples - Fedprox(μ = 0.3)

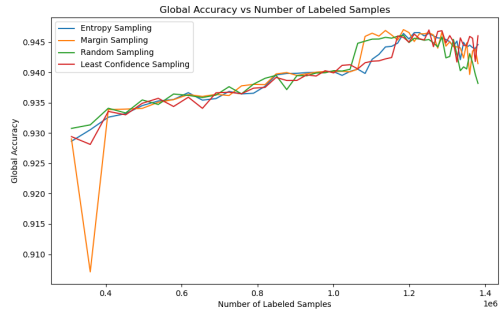

Fig. 4. Global modal accuracy for various AL methods with different numbers of labeled samples using Fedprox(μ = 0.5)

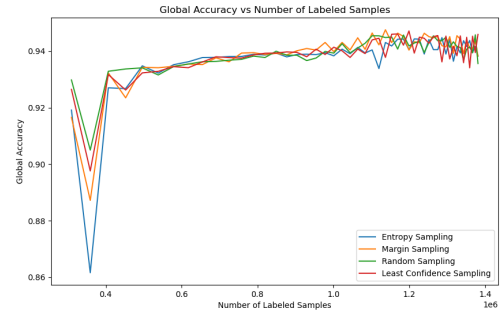

Fig. 5. Global modal accuracy for various AL methods with different numbers of labeled samples using Fedprox(μ = 0.8)

3) Communication Round Effects on METALS: Table IV, Table V, Table VI, and Table VII provide the METALS performance details using both FedAvg and FedProx algorithm with various client sampling strategies. Performance metrics also include the best client accuracy (B), the worst client accuracy (W), and the global model accuracy (G) at the 1^{st} and 50^{th} training rounds. Across all sampling strategies, we can notice an improvement in both the best and worst client accuracies, as well as the global model accuracy from the 1^{st} and 50^{th} rounds. This trend underscores the effectiveness of FL in improving model performance in successive rounds of training with AL strategies. Additionally, performance varies between different sampling strategies, indicating the importance of selecting an appropriate strategy for data labeling in FL environments. However, the choice of μ can affect the performance. Generally, a lower μ (0.3, 0.5) often yields a higher global model accuracy by the 50^{th} round across all sampling strategies, suggesting that a smaller regularization term favors more significant contributions from local updates to the global model in this context. In addition, these tables provide a comprehensive view of the model's performance from individual clients to the global model. In addition, the improvement in both the best and worst client accuracies along with the global model accuracy highlights the METALS model's ability to balance local and global objectives. Maintaining this balance can address the security requirements and difficulties of individual nodes.

TABLE IV
METALS WITH ENTROPY SAMPLING

Aggregator	**1st round**			**50th round**		
	B	W	G	B	W	G
FedAvg	0.9303	0.9246	0.93	0.9413	0.9300	0.9422
Fedprox(0.3)	0.9305	0.9181	0.9310	0.9451	0.9366	0.9461
Fedprox(0.5)	0.9307	0.9205	0.9286	0.9416	0.9284	0.9445
Fedprox(0.8)	0.9314	0.9268	0.9191	0.9420	0.9287	0.9382

Note: (B): Best client accuracy; (W): Worst client accuracy; (G): Global model accuracy

TABLE V
METALS WITH MARGIN SAMPLING

Aggregator	**1st round**			**50th round**		
	B	W	G	B	W	G
FedAvg	0.9306	0.9213	0.9292	0.9471	0.9442	0.9480
Fedprox(0.3)	0.9308	0.9247	0.9293	0.9445	0.9297	0.9452
Fedprox(0.5)	0.931	0.9192	0.9290	0.9424	0.9305	0.9414
Fedprox(0.8)	0.9305	0.9228	0.9165	0.9447	0.9299	0.9382

(B): Best client accuracy; (W): Worst client accuracy; (G): Global model accuracy

TABLE VI
METALS WITH RANDOM SAMPLING

Aggregator	**1^{st} round**			**50^{th} round**		
	B	W	G	B	W	G
FedAvg	0.9306	0.9280	0.9313	0.9472	0.9457	0.9482
Fedprox(0.3)	0.9309	0.9265	0.9308	0.9436	0.9248	0.9464
Fedprox(0.5)	0.9302	0.9238	0.9307	0.9429	0.9324	0.9381
Fedprox(0.8)	0.9311	0.9253	0.9297	0.9433	0.9141	0.9356

(B): Best client accuracy; (W): Worst client accuracy; (G): Global model accuracy

D. Discussion

METALS represents a significant advancement in the development of IDS in IoT environments. By integrating FL with AL, METALS addresses the dual challenges of data privacy

TABLE VII
METALS WITH LEAST CONFIDENCE SAMPLING

Aggregator	1^{st} round			50^{th} round		
	B	W	G	B	W	G
FedAvg	0.9310	0.9232	0.9313	0.9471	0.9421	0.9475
Fedprox(0.3)	0.9310	0.9207	0.9292	0.9441	0.9319	0.9460
Fedprox(0.5)	0.9312	0.9230	0.9294	0.9450	0.9298	0.9460
Fedprox(0.8)	0.9311	0.9252	0.9264	0.9449	0.9244	0.9459

(B): Best client accuracy; (W): Worst client accuracy; (G): Global model accuracy

and the efficient IDS, while simultaneously enhancing the learning process through the targeted labeling and incorporation of informative data samples. This methodology not only improves the efficiency and performance of intrusion detection models but also contributes to the goal of securing distributed networks against evolving cybersecurity threats. As shown in the experiment results, the trend of first increasing the accuracy of METALS with more labeled data and then decreasing thereafter highlights the complex function of AL in increasing the effectiveness of the models. AL, by design, selectively targets the most informative samples for labeling, thereby maximizing the model's learning efficiency from a relatively smaller dataset. This strategy not only optimizes resource utilization by focusing on data that contribute the most significantly to learning, but also plays a pivotal role in mitigating the risk of overfitting. Furthermore, the differentiation in performance across various AL strategies and the impact of the FedProx algorithm's μ value elucidate the importance of careful strategy selection and parameter tuning. For example, too much regularization (a high μ value) may hinder the METALS ability to integrate valuable insights from local data, while too little can lead to METALS to overfit to specific client datasets.

VI. CONCLUSION

This article introduced METALS, a new semi-supervised federated active learning framework designed to improve intrusion detection systems (IDS) within the IoT environment. By combining the strengths of AL and FL, METALS efficiently leverages unlabeled data from IoT devices, addressing a significant challenge in cybersecurity with minimal privacy concerns. Through comprehensive evaluations of various AL methods, including FL aggregator and AL strategies, this study has demonstrated the efficacy of each strategy in the context of federated settings. The preliminary results of our work pave the way for future research and implementations, driving towards semi-supervised-FL efficient IDS solutions.

ACKNOWLEDGEMENT

This work was partially supported by the European Union Horizon-CL4-2021 Research and Innovation Program under Grant Agreement 101070181 (TALON) and partially supported by the ANR CHIST-ERA project Di4SPDS-Distributed Intelligence for Enhancing Security and Privacy of Decentralised and Distributed Systems. The paper reflects the authors' views, and the Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] A. Alam, S. Qazi, N. Iqbal, and K. Raza, "Fog, edge and pervasive computing in intelligent internet of things driven applications in healthcare: Challenges, limitations and future use," *Fog, edge, and pervasive computing in intelligent IoT driven applications*, pp. 1–26, 2020.

[2] 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.

[3] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[4] M.-F. Balcan, A. Beygelzimer, and J. Langford, "Agnostic active learning," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 65–72.

[5] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, M. Alazab, S. Bhattacharya, P. K. R. Maddikunta, and T. R. Gadekallu, "Federated learning for intrusion detection system: Concepts, challenges and future directions," *Computer Communications*, 2022.

[6] O. Aouedi, K. Piamrat, G. Muller, and K. Singh, "Intrusion detection for softwarized networks with semi-supervised federated learning," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 5244–5249.

[7] B. H. Meyer, A. T. Pozo, M. Nogueira, and W. M. N. Zola, "Federated self-supervised learning for intrusion detection," in *2023 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2023, pp. 822–828.

[8] R. Zhao, Y. Wang, Z. Xue, T. Ohtsuki, B. Adebisi, and G. Gui, "Semi-supervised federated learning based intrusion detection method for internet of things," *IEEE Internet of Things Journal*, 2022.

[9] X. Pei, X. Deng, S. Tian, L. Zhang, and K. Xue, "A knowledge transfer-based semi-supervised federated learning for iot malware detection," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[10] F. Naeem, M. Ali, and G. Kaddoum, "Federated-learning-empowered semi-supervised active learning framework for intrusion detection in zsm," *IEEE Communications Magazine*, vol. 61, no. 2, pp. 88–94, 2023.

[11] O. Aouedi, A. Sacco, K. Piamrat, and G. Marchetto, "Handling privacy-sensitive medical data with federated learning: challenges and future directions," *IEEE Journal of Biomedical and Health Informatics*, vol. 27, no. 2, pp. 790–803, 2022.

[12] D. C. Nguyen, M. Ding, Q.-V. Pham, P. N. Pathirana, L. B. Le, A. Seneviratne, J. Li, D. Niyato, and H. V. Poor, "Federated learning meets blockchain in edge computing: Opportunities and challenges," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12 806–12 825, 2021.

[13] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," 2020.

[14] B. Settles, "Active learning literature survey," 2009.

[15] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-iiotset: A new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40 281–40 306, 2022.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, p. 321–357, Jun. 2002. [Online]. Available: http://dx.doi.org/10.1613/jair.953