FLUIDS: Federated Learning with semi-supervised approach for Intrusion Detection System

Ons Aouedi, Kandaraj Piamrat, Guillaume Muller, Kamal Singh

FLUIDS: Federated Learning with semi-supervised approach for Intrusion Detection System

Ons AOUEDI*, Kandaraj PIAMRAT*, Guillaume MULLER+, and Kamal SINGH+
*LS2N, Univ. de Nantes, BP 92208, 44322 Nantes CEDEX 3, France
{firstname.lastname}@ls2n.fr
+Univ. Jean Monnet, IOGS, CNRS, UMR 5516, LaHC, F - 42023 Saint-Étienne, France
{firstname.lastname}@univ-st-etienne.fr

***Abstract*—In this paper, we present FLUIDS, a Federated Learning with semi-sUpervised approach for Intrusion Detection System. FLUIDS formulates the intrusion detection into a semi-supervised learning where both supervised learning (using labeled data) and unsupervised learning (no label data) are combined in a collaborative way. The combination of federated learning and semi-supervised Learning allows the solution to: better preserve the privacy, improve training and inference efficiency, achieve better accuracy, and be cheaper to deploy.**

***Index Terms*—Federated Learning (FL), Semi-Supervised Learning, Deep Learning (DL), Machine Learning (ML), Auto-Encoder (AE), Intrusion Detection System (IDS).**

I. INTRODUCTION

Doing Machine Learning (ML) in an IoT context can be difficult, as the need for collecting as much data as possible to train the model collides with the unwelcome necessity to transfer sensitive data [1]. In this work, we propose a novel approach, called "FLUIDS", which combines Federated Learning (FL) and Semi-Supervised Learning to solve this dilemma, for Intrusion Detection Systems (IDS). It combines the unsupervised training of an Auto-Encoder (AE) on IoT Gateways (where data have no label) with the supervised training of a classification model on a FL server (where some labels can be determined). This approach does not need to transfer the data, thus better preserves the privacy. Also, it distributes the workload on the devices, thus improving both training and inference efficiency. Finally, by exploiting all the data (labeled & unlabeled), it can achieve better accuracy than models that use only labeled data. It can also prove cheaper, as labels can be costly to obtain while unlabeled data is available in large amounts.

II. METHODOLOGY & ARCHITECTURE

Figure 1 illustrates the architecture of the FLUIDS approach. It consists of IoT devices (bottom), IoT Gateways (middle), and an FL server (top).

The IoT gateways help the IoT devices to connect to the Internet and operate as FL clients and the IoT server acts as the FL server. More specifically, each IoT gateway trains locally an autoencoder model using its unlabeled data in order to learn the representative and low-dimensional features. Then, it sends back the learned model's parameters to the FL server. Unlike the classical FL, in our case, the FL server not only generates a global autoencoder model but it exploits a small amount of

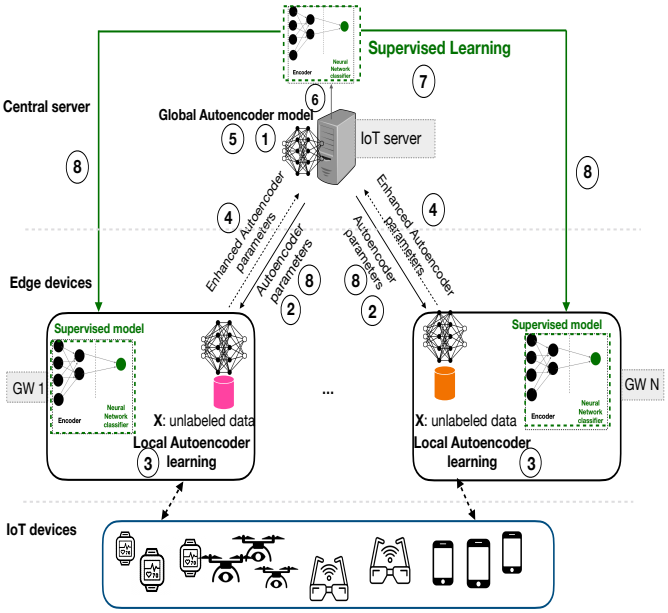

Figure 1. Overview of the proposition

labeled data to conduct supervised learning. The functioning of the system involves multiple rounds and globally follows the steps below:

- **Step 1:** The FL server initializes the hyper-parameters of the Auto-Encoder (*unsupervised learning*).
- **Step 2:** This Auto-Encoder is disseminated to the IoT gateways.
- **Step 3:** Once the training rounds begin, the IoT gateways train the Auto-Encoder model using their *local unlabeled data*.
- **Step 4:** The IoT gateways upload their updated Auto-Encoder parameters to the FL server.
- **Step 5:** The FL server aggregates the Auto-Encoders parameters received from the different IoT gateways'. For the aggregation, the FederatedAveraging (FedAvg) algorithm [2] is used.
- **Step 6:** The FL server extracts the Encoder part of the global Auto-Encoder and concatenates a Fully Connected layer to it (i.e., sigmoid layer), thus obtaining a Neural Network for classification.

- **Step 7:** Then, the FL server fine-tunes this classification model with its local labeled data and hence gets a supervised model.
- **Step 8:** The server sends back the updated the global Auto-Encoder (*unsupervised model*) to the IoT gateways. The server sends as well the concatenated model (*supervised model in green box*) to the IoT gateways, so that they can perform the IDS task, by using the supervised model in inference mode.
- The training rounds are iterated: steps **3, 4, 5, 6, 7, and 8** are repeated for continuous learning and improvement. This process is repeated various rounds until the desired performance is achieved.

It is important to note here, that during step 3, it is not required that all IoT gateways participate in all rounds (e.g., if they are too busy at that time).

III. Experiment and Results

In this section, we first present the dataset used in our experiments. Then, we present the implementation parameters used for intrusion detection. Finally, the results are analyzed and discussed.

A. Dataset description

In order to train our semi-supervised FL model, we use the UNSW-NB15 dataset[1] [3] because it is recent and referenced in many existing papers. The training set contains 175,341 and the testing set contains 82,332 (by default) total observations. The simulation of a partially labeled dataset has been done by randomly selecting of a portion of labeled observations and removing their labels.

B. Results

To evaluate the performance of our semi-supervised FL model, we selected the optimal AE's parameters setting in our simulation.

To validate FLUIDS' efficiency, we compared it to four reference ML models: simple classifiers (Decision Trees – DT – and Support Vector Machine – SVM), ensemble learning (Random Forest – RF), neural network classifier (Multi-Layer Perceptron – MLP). By design, these models use only labeled data. Thus, in our setup, we provided them only with the data located on the FL server. Figure 2 shows the experimentation results. It is worth noting that the semi-supervised federated learning model outperforms all the others. For example, the F1-score is increased by **3.68%**, **5.46%**, **6.21%**, **7.55%** for MLP, RF, SVM, and DT, respectively. This may be attributed to the fact that the use of unlabeled data in the training process boosts the performance of our model.

IV. Conclusion

In this paper, a federated intrusion detection system has been presented. It was trained in a semi-supervised way, using both the labeled and unlabeled data. This model shows better accuracy than traditional supervised models while being cheaper and lighter to train. Also, it is better at protecting sensitive data and hence preserve privacy.

[1]https://research.unsw.edu.au/projects/unsw-nb15-data-set

Table I
Implementation parameters

Dataset	
Nb input variables	197
Nb output variables	1
Training set	175,341
Unlabeled set	122,739
Labeled set	52,602
Unlabeled ratio R_u	2.33
Testing set	82,332 (default)
Deep Learning	
Deep learning tool	PyTorch
DL algorithms	Auto-Encoder (AE) Neural Network (NN)
AE hidden layers	3
NN layers	1
Activation functions	Relu (AE), Sigmoid (NN)
Optimizer	Adam
Learning rate	0.0001
Batch size	64
Loss functions	Mean Squared Error (AE) Binary Cross Entropy (NN)
Federated Learning	
FL server	1
Nb clients (gateways)	100
Clients used in federated updates	10%
AE epochs (client)	5
NN epochs (server)	20
Communication round	6

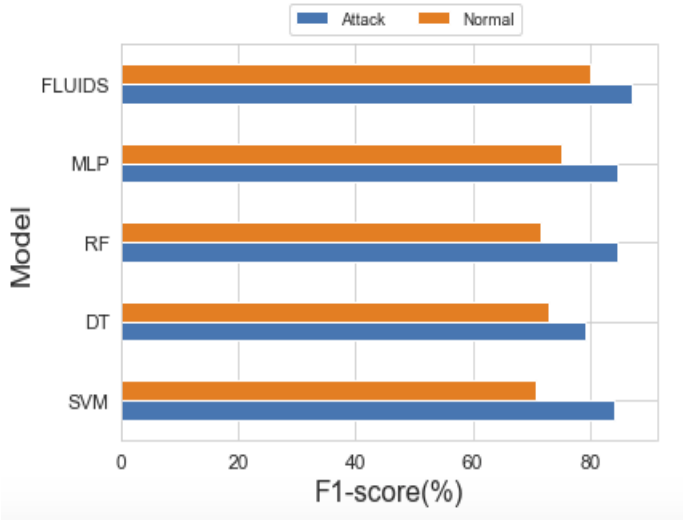

Figure 2. Comparison of FLUIDS performance to identify normal/attack IP flows against supervised models.

References

[1] S. Agrawal, S. Sarkar, O. Aouedi, G. Yenduri, K. Piamrat, S. Bhattacharya, P. K. R. Maddikunta, and T. R. Gadekallu, "Federated learning for intrusion detection system: Concepts, challenges and future directions," *arXiv preprint arXiv:2106.09527*, 2021.

[2] 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*, pp. 1273–1282, PMLR, 2017.

[3] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*, pp. 1–6, IEEE, 2015.