Contents

```
---
title: Intrusion Detection System for IoT devices using Federated Learning
author: Clément Safon, Sarah Ramdani
date: 30-01-2024
place: Télécom SudParis, Palaiseau, France
---

\newpage

# Abstract

> With IoT systems spreading across all the different domains and aspects of our lives for the

\newpage


# 1. Introduction

As the Internet of Things (IoT) continues to proliferate across diverse domains, the security

This paper aims to present Federated Learning, and goes into details on how it works (the theo
The proposed system leverages the decentralized nature of FL to address privacy concerns inher

Our framework enables IoT devices to collaboratively train a global intrusion detection model

The results showcase the potential of Federated Learning as a privacy-preserving solution for

*We would like to point out that as Network Security students, Machine Learning, thus Federate

**As this paper is still a draft, the following statement might evolve over the time:**
This paper is sectionned as follows: first, we expose Federated Larning and explain how it wor
Then, we will explore our two approaches for this experiment, both with with binary and multi-

\newpage

# 2. Federated Learning: An overview

Federated Learning is a machine learning approach that enables model training (and testing) ac

The main steps of the federated learning are the following :

- **Initialisation** : This step can be offline or online. In this part we create the unique s
- **Local Training** : The local devices use their local data to futher train the global model
- **Aggregation** : The servers then collect all the different results and combine them to get
- **Iteration** : The weights of the global model are sent back to each client and we repeat t

> In some cases, we can combine Federated Learning and Transfert Learning to have better resul

So now, what are the different way to merge the client's model in the aggregation step ?

>**(How are the data aggregated? )**

So after the clients train their models, updated model parameters are sent to the server-based

By only exchanging weight parameters instead of raw data, this process is inherently privacy-p

In this study we will almost only use the _FederatedAverage_ method, but in the last part, we

>**Need more sources**

Our Intrusion Detection System is a behavior-based IDS, opposed to a Rule Based Intrusion dete

\newpage

# 3. Dataset presentation

The dataset on which this project is based is the UNSW-NB15 dataset. Before this dataset, two

The dataset creation involves using the IXIA traffic generator configured with three servers,

![Scheme of the generation of network traffic](images/IXIA_scheme.png)

Eleven types of attacks were categorized, as follows: \
- Fuzzers
- Analysis
- DoS
- Exploits
- Generic
- Reconnaissance
- Shellcode
- Backdoors
- Worms

2,540,044 flows of traffic were opened for this dataset.
It is important to notice that in this dataset, the distribution is not equal for every attack
The features were classified into three groups : Basic, Content, and Time. '0' refers to, in t

In this dataset, there were two main problems:
- class imbalance
```

- class overlap.

In multiple experiments which will be presented further in this document, some classes

## 3.1 Preprocessing

The initial stage involves preprocessing the data to tailor the raw dataset to our requirement

Feature No.	Input Feature Name	Description
1	dur	Record total duration
2	proto	Transaction protocol
3	service	Contains the network services
4	state	Contains the state and its dependent protocol
5	spkts	Source to destination packet count
6	dpkts	Destination to source packet count
7	sbytes	Source to destination transaction bytes
8	dbytes	Destination to source transaction bytes
9	rate	Ethernet data rates transmitted and received
10	sttl	Source to destination time to live value
11	dttl	Destination to source time to live value
12	sload	Source bits per second
13	dload	Destination bits per second
14	sloss	Source packets retransmitted or dropped
15	dloss	Destination packets retransmitted or dropped
16	sinpkt	Source interpacket arrival time (mSec)
17	dinpkt	Destination interpacket arrival time (mSec)
18	sjit	Source jitter (mSec)
19	djit	Destination jitter (mSec)
20	swin	Source TCP window advertisement value
21	stcpb	Destination TCP window advertisement value
22	dtcpb	Destination TCP base sequence number
23	dwin	Destination TCP window advertisement value
24	tcprtt	TCP connection setup round-trip time
25	attack_cat	The name of each attack category. In this data set, nine
26	label	0 for normal and 1 for attack records

Some features were redundant in the testing set, such as ... and ... which are actually other.

After this step, we reduce the normal traffic by randomly taking a part of the traffic with a

After having randomly split the entier dataset in a training set (80%) and a testing set (20%)

```
![Preprocessed UNSW-NB15 Data Distribution](images/number_traffic_instances_per_attack_cat.png

Here we have taken 5% of the normal trafic entries.

We can also mention that some other preprocessing will be done juste before the simulation pro

>**Copier la figure des centroids ?**

## 3.2 Visualization

![UNSW-NB15 Data Distribution](images/number_traffic_instances_per_category.png)


In order to understand the results better and to get a clear view on what information they do

175,341 records were selected for the training set, 82,332 records for the testing set.
All of the features may not be relevant. The nominal features are converted to numeric feature
After the distance between centroids has been calculated, the results are plotted.
On the plot, a colored scaled represents as follows: the darker shades mean that the centroids

insert the images of the visualisation artiicle (or just cite the article)
->
Principal Component Analysis (PCA) is
Overlap problem: many attacks have a similar behavior comparing to

## 3.3 Evaluations

In order to evaluate the different models, a large panel of metrics came in handy. Among them,

- Precision: this metric is used to evaluate the performance of the FL model. It assesses the

$$
Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}
$$

- Recall: also referred to as the true positive rate, quantifies the ability of a classificati

$$
Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}
$$

- F1 Score: the F1 Score will then compute the harmonic mean between the precision and recall.

$$
F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall}
```

```
$$

- Accuracy:
$$
Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Examples}
$$


Centralized EValuations.
Federated Evaluation


old part :
Different methods exist to evaluate models: FedFomo [[4]](#4) and L2C (Learning to Collaborate
**Important remark: these evaluation methods were only studied and researched at the beginning

For client selection in federated learning :
Interesting topic to learn more about how Trusted Execution Environment (TEE) may make FL more
Y. Chen et al., "A Training-Integrity Privacy-Preserving Federated
Learning Scheme with Trusted Execution Environment,"
Information Sciences, vol. 522, 2020, pp. 69-79.

The device's resource, time consumption, as well as communcation cost should be looked upon in

\newpage

# 4. Implemented IDS

[IBM-FL](https://github.com/IBM/federated-learning-lib/tree/main) was the first notebook used
Flwer is the second module which was used in this experiment.

The computer which was used to train all the data is

Number of epochs + rounds for each training.

## 4.1. The Simulation Environment

In this part we are just going to describe the simulation process that we used in our set up.


## 4.2. First experiment: binary classification

In the first experiment, we went for the most simple IDS behavior, the one that can classify a

This first experiment consisted in two main things:
```

First, after having tested many configurations, we had some good results in terms of F1-score

To prove this, we have to split the dataset into the N clients, but this time, not randomly si

So now, we are able to split all the data between N client and destroy all the data categorise

![Binary Classification Metrics Evolution](images/metrics_centralized.png)

## 4.3. Second experiment: multi-class classification

In this section, we explore our second experiment, this time with multi-class classification F

## 4.4. Data repartition

Will be presented in this section the different data repartitions of the three experiments we
The first one, there is only one client which is trained on the whole dataset.
![Data repartition of the client 0](images/Multi_class/data_repartition_client0_all.png)

The second one, still with only one client the first which has no network traffic from the "G
![Data repartition, client 0 with no generic](images/Multi_class/data_repartition_client0_no_g

The third one, which was the last experiment we conducted, where each client had an even repar

![Data repartition for three clients](images/Multi_class/data_repartition_three_clients.png)

The third experiment was done in order to test if client 0 would benefit from the other client
Pourquoi ne pas avoir fait CL, FL avec tout, FL avec un client sans une classe?

In this experiment, the feature "rate" was not taken in the features we wanted to keep.
Even though in most litterature, only 24 features out of the 41 were kept, amongst which the f

What we didn't take into account is that the FE is different for every client. They are going

## 4.5 Results

The evaluation is then done after the aggregation part.

After all the experiments, we can now conclude that the clients get better and they learn from

![Metrics evolution for client 0 which trains its model with the entire dataset (CL)](images/M

![Metrics evolution for client 0 which trains its model with the entire dataset without the ge

![Metrics evolution for the global model with a three client infrastructure](images/Multi_clas

# 5. Attacks
## 5.1 Attack models

After the end of the implementation, the following part consisted in testing if the models wer
In the first two attack attack scenario, the attacker has no knowledge of the actual infrastru

### 5.1.1 Fuzzing attack

For this attack scenario, the goal was to test the system's robustness and security by simulat

The code we used to perform the fuzzing attack is the following:

```python
for i in range(len(parameters)):
    random_values = np.random.uniform(low=-5, high=5, size=parameters[i].shape)
    parameters[i] = random_values
return parameters, len(self.x_train), {}
```

It replaces the original values in each element of the parameters list with randomly generated

![Fuzzer poisoning attack](images/fuzzer_poisoning_attack.png){ width = 70% }

![Metrics evolution - fuzzer poisoning attack](images/metrics_fuzzer_poisoning.png)

### 5.1.2 Simple poisoning attack

![Simple poisoning attack](images/simple_poisoning_attack.png){ width = 70% }

![Metrics - simple poisoning attack](images/metrics_simple_poisoning.png)

### 5.1.3 Targeted poisoning attack

This third attack consisted in a targeted attack, in which the attacker has the knowledge how
In the attack we conveyed, there are three steps:

1. The attacker first trains the model in order to estimates the weights of the other clients.
2. Then, the attacker will

![Targeted Poisoning attack](images/targeted_poisoning_attack.png)

![Metrics evolution - targeted Poisoning attack](images/metrics_targeted_poisoning.png)

This is, by far, the most effective attack scenario. According to the Confusion Matrix (Figure

# 6. Counter-measures

(To be discussed.)

# 7. Issues with Federated Learning and with Machine Learning in general

In this section, many issues and challenges which occur in Machine Learning as well as in Fede

In any type of Machine Learning in general, the cleanliness of the data as well as the accurat

IDSs, which many now rely on machine learning, play a critical role in cybersecurity, but misc

> What is the rate of false positives which make IDS unusable for real life conditions?

The diversity in the network traffic also appears to be an issue. Let's imagine that we train
Diversity in the network traffic.

In ML, it is needed to have a suffiscient amount of data for each and every class in order to

\newpage

# 8. Conclusion

In this research project, we explored the application of FL in the context of Intrusion Detect

The paper provided an overview of Federated Learning, explaining its principles and detailing

Throughout our experiments, including both binary and multi-class classifications, using the U

The research highlighted the challenges and issues in the application of FL to IDS, such as th

The paper concluded by addressing potential counter-measures to the identified issues and emph

Our experiments, ranging from binary to multi-class classifications using the UNSW-NB15 datase

In acknowledging the challenges encountered, including data cleanliness, accurate labeling, an

```
[commment] (#
- Expériences, bilan, enseignements à tirer comment bien effecuter l'apprentissage
- Présenter les résultats avec hopefully les résultats cohérents qui co
- état de l'art/background/restituer les recherches
- Attention
- Dire les obstacles qu'on a rencontré
- Fuzzing
- Fonction)

\newpage
```

```
# References
<a id="1">[1]</a>
Moustafa, Nour & Slay, Jill. ([2015](https://www.researchgate.net/publication/287330529_UNSW-N
*UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 networ
10.1109/MilCIS.2015.7348942.

<a id="2">[2]</a>
Zoghi, Zeinab & Serpen, Gursel. ([2021](https://arxiv.org/abs/2101.05067)).
*UNSW-NB15 Computer Security Dataset: Analysis through
Visualization.*
Electrical Engineering & Computer Science, University of Toledo, Ohio, USA.

<a id="3">[3]</a>
([2021](https://www.icir.org/robin/papers/oakland10-ml.pdf))
*Outside the Closed World: On Using Machine Learning For Network Intrusion Detection*

<a id="4">[4]</a>
Zhang, Michael, et al. ([2020](https://arxiv.org/abs/2012.08565)).
"Personalized federated learning with first order model optimization." arXiv preprint arXiv:20

<a id="5">[5]</a>
Shuangton Li, et al. ([2022](https://openaccess.thecvf.com/content/CVPR2022/papers/Li_Learning
"Learning to collaborate in decentralized learning."

<a id="6">[6]</a>
Arp, Daniel et al. ([2020](https://www.usenix.org/system/files/sec22summer_arp.pdf)).
"Do's and Don'ts of Machine Learning in Computer Security"

<a id="7">[7]</a>
Aguessy, François-Xavier. ([2023](https://moodle.imtbs-tsp.eu/mod/resource/view.php?id=22573))
"Intrusion detection and alert correlation (as part of a Télécom SudParis class)"
```

<a id="8">[8]</a>
Fraboni, Yann, et al. ([2023](https://www.jmlr.org/papers/volume24/22-0689/22-0689.pdf))).
"A General Theory for Federated Optimization with Asynchronous and Heterogeneous Clients Updat