Anomaly detection in IoT: Federated Learning approach on the IoT-23 Dataset

MAURICIO LEONEL MERCHAN PRADO, University of Twente, The Netherlands

The use of smart devices with software, sensors and other technologies(IoT) has become more common in the last few years. They aim to be interconnected through a network and exchange data with the purpose of reducing human intervention over specific tasks. This increase in IoT devices has incremented the data transit over a network. These devices' interactions and structure have raised the problem of malware attacks, especially over these devices, which can be manipulated to replicate more attacks. Because of this, there is a concern for identifying potential attacks. AI appears as a solution to this problem, and the Federated Learning approach was used as a technique of Machine Learning and using the IoT23 as Data-set to train and test the model. A Multi-layer Perceptron was used for both central server and federated members and Federated Averaging as an aggregation technique. The results showed that Federated Learning has the potential as a decentralized technique, which means that it benefice each device in its inner ML model. The model scored 100% accuracy on some devices. However, the general learning was affected by this decentralized method having at most 70% of accuracy. The model's performance was affected because of the data, distribution and quantity of each member's dataset.

Additional Key Words and Phrases: Anomaly Detection, Machine Learning (ML), Internet of Things (IoT), IoT-23 (dataset), Centralized Learning, Federated Learning (FL).

1 INTRODUCTION

The essential purpose of the Internet of Things (IoT) is to make communication between different devices easier while increasing system efficiency and improving living quality [10]. Examples are switch bots, intelligent lights and smart personal assistants like Alexa. That is only the beginning; in the last few years, we can make smart any electrical mechanism we possess at home, like a toaster, vacuum cleaner, or refrigerator. As a result, it might be described as a cyber-physical ecosystem of linked sensors and actuators that fall into a continuous cycle of sensing and decision-making according to the data recollected. This data has increased the risk and challenges associated with devices, making them riskier for users and network information. In this sense, the safety, security, and privacy of citizens could be affected [15].

These devices have become more popular over the last few years, and the number of IoT networks has grown parallel. It is harder to control such security devices, not only because of their fragile security but also because of the continued rise of structured attacks. These IoT devices have their disadvantages among DoS attacks because of their structure. Nevertheless, this is not the only attack they can sustain[6]. Since this risk has been growing, The Stratosphere Laboratory, AIC group, FEL, CTU University, Czech Republic has collected and created a network traffic dataset of the IoT. Where they call researchers to use such datasets to develop AI algorithms that help recognize malware over these kinds of devices, this data set contains 20 malware captures and three benign captures executed over IoT devices [13].

Several AI algorithms can be used to manage and treat this DataSet. These methods can go from probabilistic Bayesian Detection to Deep Neural Networks, also well-known as Centralized Machine Learning. The key to these models is that they can answer a specific problem from the given data. Thus, all the data should be gathered and centralized to learn from the group. [3]. On the other hand, there are Decentralized Incremental Learning and Federated Learning. Ekkono Solutions mentions that the advantage of it is that it learns from each device individually without compromising the safety of devices[3].

Furthermore, the difference between federated and decentralized learning is that the later cannot learn across devices. Thus, Federated Learning is the best option among all centralized approaches since it can connect and benefit from all devices involved in a IoT network. Since the investigation is done over several scenarios (23 in total) that conform to the IoT23, FL would be the required model to handle and compute the information for each device across the data set and then create a global model that could learn from each dataset that conforms the IoT23. In addition, several researchers have used this dataset as their training data for their different algorithms but Federated Learning.

Several steps should be followed to implement an FL model: First, a model framework should be chosen; in this case, TensorFlow was chosen. Secondly, the data is processed, cleaned and divided by the devices in the network from the dataset. Next, a general model is developed, which learns from each device or federated member to establish the correspondent training process. This model is a Multi-Layer Perceptron with a multi-class classification model. After that, all learned data would be shared with the central manager, who would then learn and share its results with each device model. Thus, a Client-server model is made to handle the correspondent weights results from each device. The central model uses a Federated Averaging to calculate the new weight values to share back with each Federated member[1].

In this paper, A Literature review of Federated Learning is done, where its advantages are discussed. A step-to-step development of a Federated Learning Method is presented with the dataset used to train this model as well as the steps needed to handle such dataset in terms of cleaning. Moreover, the results of the model are presented in terms of global and individual accuracy, how the FL model behaves with the data, and how this differs from a centralized methodology. Finally, there is a comparison of the learning rate among all federated members within the model.

TScIT 37, July 8, 2022, Enschede, The Netherlands

2 LITERATURE REVIEW

Even though this is a novel technology, cybersecurity is a concern for citizens and the government. In this context, several measures to fight cyber-attacks on IoT devices exist. ENISA is an example of it, where it contains a list of good practices for cybersecurity in the IoT [15]. These measures were just published in 2017, giving guidance about data protection, cryptography, Authentication, privacy and more. Nevertheless, the efficiency of malware attacks has grown so fast that different public and private entities have created different data sets for analyzing and creating AI methods.

Rey V et al. mention several public datasets, such as N-BaloT, Kitsune, TON-loT, IoT23, and more, which have been focused on FL by researchers. In the same paper, they discuss the use of supervised and non-supervised learning methods that have been applied to one of these datasets from 2018, highlighting unsupervised learning and preserving the privacy of the users and devices in this methodology [12]. Moreover, Liu Y et al. research focuses on the efficient communication of using FL and its advantages in a non-supervised method. For their experiments, they used datasets where they split the data into several devices having good performance and accuracy, scoring a 94% of accuracy but having a faster calculation and communication among the clients. [5]. It is important to highlight that these last two papers are from 2022 and 2020, respectively.

The number of research papers that use this FL method is limited by just looking over the internet. It is visible by looking at different papers such as[4, 6, 11] from 2018-2019, where they explain the challenges of IoT networks and even propose to use FL as a new methodology in the seek to protect such devices. They see Federated learning as an important tool within the Artificial intelligent field but also as a method that still needs to be addressed and optimized since it still has to cover several gaps in the machine learning field. Finally, there is the research by Ferrag et al., which also tries to analyze the difference between centralized and federated learning where they are using several datasets, but IoT23 [9].It is comprehensive since its main purpose is to generate a dataset that can work as a guideline of the optimal classes or layers a dataset should have for detecting anomalies for the IoT. In addition, they show no clear conclusion, mentioning that the dataset they used has some limitations. The idea of applying a Federated Learning method over the IoT-23 DataSet is seen as an opportunity to contribute to the investigation of the CTU University [13].

3 IOT23 DATA-SET

As was mentioned before, IoT-23 is the dataset used to train and test this Federated Learning method. This dataset was captured by the Stratosphere Laboratory, which is network traffic executed over IoT devices. This dataset contains in total of 23 scenarios, where 20 of them are malware captures, and three are captured over benign IoT devices' traffic which shows how regular network traffic should look like[13]. There are two versions available. There is the original *.pcap* file, which is the original pcap file from the network traffic. It contains a disadvantage since its extension is associated with Wireshark, and the files can only be opened by this program or similar, making the work more complicated. On the other hand, there is the *conn.log.labeled* files from the lighter version, a document generated from the original version using the Zeek network analyzer. Moreover, the file is labelled in detail and explains its labels. This last and lighter version will be used on this project. The IoT23-Dataset was labelled using the original Network capture and reflected their malware analysis in the last columns that show its possible malware sample. These label names are described in Table1. Where it also contains the names of the attacks and a brief description of what each of them does.

3.1 Data Cleaning

The IoT23-Dataset contains 23 dataset files which are Zeek Log files (*conn.log.labeled*) using the log format with the IoT network traffic. Each of these log files contains a 23-column with the collected information from the network. Each of these columns represents a connection state during the execution of the captures. There are data like the name of the network protocol, application protocol, IP of the device, and IP where the attack happened, which can be filtered to determine which information is relevant for the model. Since all this data was collected in 24 hours timelapse, there are datasets of devices that contain much more information than others, making the datasets' variance very high among the devices. Moreover, noise and unnecessary data create the need for cleaning and reducing the amount of packages information from the data set. First, this data was managed to be read using python. For cleaning up the data information, a correlation matrix was made to identify how related the columns are. As a result, there were several columns with 0 correlation or close to 0 thus they were deleted. The deleted columns are:

ts, uid, id.orig_h, local_orig, local_resp, missed_bytes, tunnel_parents

Furthermore, columns that contained a single value were identified to be eliminated, it is because a single observation is useless for modelling the AI since they had no variance making the data just more heavy. However, since the information varies from device to device, the same columns were not deleted among devices. It could affect the learning process on that specific device or its interaction with others.

Additionally, rows that contained duplicate Data were identified and deleted from the dataset. Finally, some data needed to be deleted from several of the datasets. It was because the variance among all the devices is high, and the learning rate would be affected. Hence, it was an attempt to maintain the quantity of data of each device in the range of the standard deviation among the devices Thus, the dataset that contained a smaller number of packets than the mean of all devices would remain as they were after cleaning. Otherwise, the number of packets (rows) was reduced by keeping the rows containing much fewer common values among columns and taking random values from the more common ones.

Since Neural Network is the principal model used to develop this Federated Learning method, thus, columns within the datasets, which contain string values, were changed to integer values. Columns such as service, *conn_state, history, label and detailed_label* were encoded. These last two were deleted to create a new column called encoding, which is made of its combination following the parameters in Table 1. While the other would follow the pattern as shown

in Table 2. Finally, the IP address in the network came as strings and trying to have a contribution using this format in the model would be close to nothing. Hence, all IP addresses were converted to their float values.

Table 1. labels of the IoT-23 Data-set

Attack	This indicated that the device is executing an attack to some another host
Benign	the device had no suspicious or malicious behavior
CC	the device is connected to an external server, a CC server. Given its frequent or recurrent connections.
DDoS	It indicates that the device is performing a DDoS attack over other device.
FileDownload	It indicates that the device is downloading a file. It has a connection with an unknown IP destination (CC server)
HearBeat	This indicates the device is used to track another device and returned to the CC server
Mirai	The connection of the infected device contains Mirai botnet characteristics
Okiru	The connection of the infected device contains Okiru botnet characteristics
Part Of Horizontal PortScan	It indicated that the connection of the device is used to gather information by using a Horizontal Port Scan to perform future attacks
Torii	This indicates that the connection of the infected device contains Tori botnet characteristics

Table 2. encoding of label and detailed-label from IoT-23

Label	**Detailed label**	**encoding**
Benign	-	0
Malicious	CC	1
Malicious	CC FileDownload	1
Malicious	CC HeartBeat	2
Malicious	CC HeartBeat Attack	2
Malicious	CC HeartBeat FileDownload	2
Malicious	CC Mirai	3
Malicious	CC Torii	4
Malicious	DDoS	5
Malicious	FileDownload	6
Malicious	Okiru	7
Malicious	Okiru Attack	7
Malicious	PartOfAHorizontalPortScan	8
Malicious	PartOfAHorizontalPortScan Attack	8
Malicious	CC PartOfAHorizontal PortScan	8
Malicious	Attack	8

Table 3. encoding of service column from IoT-23

Service	**Encoding**
-	0
http	1
dhcp	2
dns	3
ssh	4
irc	5
ssl	6

4 BACKGROUND

4.1 Federated Learning

The increase of IoT technoloogies in society, and the variety of its devices in characteristics, data and purpose, have created a more significant concern over the data managed by such machines. Thus, Federated Learning (FL) was raised as an opportunity to compete over any centralized AI method that would require data to be brought to a central point. FL is an approach that contains several advantages. The most important are: Preserving the Privacy of user Data since the data never leaves the device and only the results would be shared with the central server, thus, no raw data is transferred, and the data keeps private; Improving Model Performance, it is because IoT devices could collectively share their trained data by sharing its trained weights with the centralized model, that can distribute the results to all of the connected devices. In this sense, each device would break the constraint of not having sufficient data. Finally, Flexible Scalability is improved by not having to gather all information in a centralized manner, and the amount of data transited is reduced, which also benefits communication costs. These advantages make Federated Learning an optimal model to deploy among devices, it is visible the importance od this model since is already used by some companies such as IBM or Google. The most notable example is the GBoard (Google KeyBoard) on Android implemented by Google, where they focus on word prediction, languages model, voice recognition for texting, or the text entry on touch-screen [7]. The main benefit for both users and developers is that each device would contain a machine learning model and the ability to store such training data. Thus, data or personal information is not compromised. Instead, they share their results (only weights) with the centre point that averages them and send them back to each member, then local devices update their results and continue learning.[8].

4.2 Federated Optimization

Federated Optimization, also known as Federated Averaging, is the optimization step taken to optimize the communication of results among the devices to get a distributed optimization. Even though it is a distributive problem, it contains several particularities that differentiate it from a distributive problem, such as:

- **Non-IID** the data is mainly obtained by the interactions of a particular device; thus, its local data information is not representative of the popular distribution

- **Unbalanced** the amount of data recollected depends on the usage and interactions of the device. It makes the variance of data information among devices get higher.

Since FL is mainly focused on smartphones, there are more points that we have to consider, such as problems like the connectivity of the device and the possible deletion of data [8]. However, these problems would not be the case by applying FL in IoT devices since they need to be online to work. Moreover, in the case of unsupervised learning, the data information of these devices can not be manipulated easily by everyday users.

5 DEVELOPMENT OF THE MODEL

It was already mentioned that FL is the model used in this research project. In this section, we will explain in detail all steps and methods used to develop the model using the IoT23-DataSet. First, how does it work? In general terms, FL is made of a server that works as a central communication that coordinates training activities from all devices. All Federated Members (Clients) are conformed by devices where they would receive the current global model weights from the server, train it together with the local data and generate new updates that will be sent back to the server using an aggregation algorithm. These steps are repetitive where the primary purpose is to obtain a better accuracy after each iteration. The best part is that these devices do not have to send or move any local data, but the weight results instead. TensorFlow was used as a framework to develop the current FL model for practical and better results. TensorFlow is an excellent open-source tool launched by Google that helps developers to create machine learning models where we take advantage of this computational power over a GPU card system, moreover by its flexibility to express different algorithms used for neural network models[14]

5.1 Federated Members

A Federated Members or clients would be each device involved in the system, where there is no data exchange for the model but only its results. Thus, for this research's aim, we have selected each dataset sample of malware as a federated member and the three samples of regular network communication. In other words, our model will contain 23 Federated members whit the sample data already cleaned but also batched, which is the format used on TensorFlow.

5.1.1 Multi-Layer Perceptron (MLP). The multi-layer perceptron is the deep learning model chosen as the central model that each federated member will use as their primary structure. It is because of the format of our data thus, it is optimal to use a multi-class classification in this MLP. To do that, the MLP class contains four layers, one as an input layer with 'relu' activation and one as an output layer with 'softmax' activation, which gives a probability value for each class in the data. Moreover, there are two hidden layers with 'relu' activation as well as "he normal" initialiser. This hidden layer contains 28 and 14 nodes, respectively. Relu function was chosen because an SGD optimisation is used later. Thus, Using Stochastic Gradient Descent allows us to optimise better with Rectified Linear Unit [2]. Additionally, the he_normal initialiser is frequently used with this relu activation which gives better performance since the model based the data on a normal distribution. Additionally, it is use

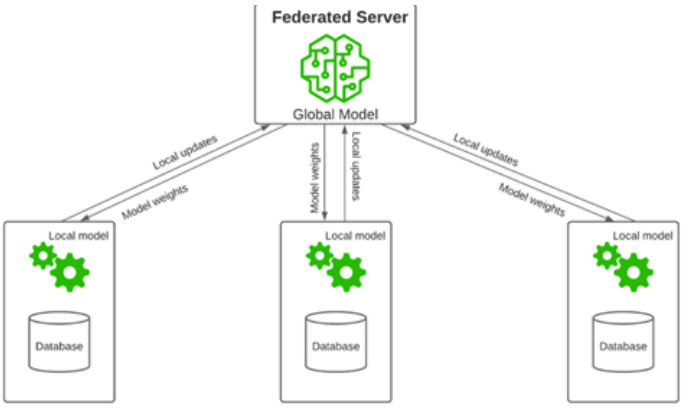

Fig. 1. Federated Learning model

sparse categorical cross-entropy as the loss function to compile the neural network since this function is specific for multi-class classification. Finally, a specific SGD optimisation is used for Federated Members, to be more specific, a Gradient Descent with Nesterov Momentum. It is because of the large amount of data thus the momentum helps to accelerate SGD in the relevant direction, and the Nesterov values would help to rectify and redirect the direction left by the momentum a. To be more specific, we are using the learning rate of (0.01), the momentum of 0.9, decay of 0.01/100 and a clip value of 1, which makes the gradient of each weight will not be more significant than 1.

5.2 Model Aggregation

The main feature of Federated Learning is its Federated Aggregation model, also known as the vanilla algorithm for FL. Since we use a supervised method and already have a predefined dataset for each device, we consider synchronous communication by rounds with all devices communicating. Thus, a round would be after 23 devices trained their models. The central model would only compile and train the scaled weights gathered from each device. For doing it, we use the following formulas:

$$f(w) = \sum_{n=1}^{k} \frac{n_k}{n} F_k(w) \tag{1}$$

where

$$F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{P}k} f_i(w) \tag{2}$$

This mathematical representation is a forward computation. We determine their weight parameters based on each federation member's loss values recorded across all the data points they trained. Then all these parameters are scaled and summed. After gathering all values, those are put in the central MLP to train it. To sum up, Figure1 shows part of its relation where each local model would obtain its local weights, sent to the federated server where the Federated Averaging is done, and its averages are sent back to each model.

5.3 Federated Model

Once all the components were created, the next step was created a basic Server-client iteration that uses the Federated Averaging Algorithm. To do it, The algorithm1 shows a pseudo representation of the steps. The idea was to define a global model that should contain the same structure as all federated members that are developed. The order of all devices are shuffled not to create any pattern that would create a biased result. All devices would have to execute several interactions to ensure the progressive learning rate. Thus, federated members are defined which uses an MLP where it sets the server's weights and trains its data together. Next, the scaled weights of each member are calculated using our Federated Algorithm, which also aims to set all results in the server. All Federated members do this in parallel. Once all weights are set, the server can train its results. Thus, the federated members could obtain the updated results in their next iteration.

Algorithm 1 Federated Model

function FEDERATED SERVER(d) ▷ define the central model that saves all interactions
  $G() \leftarrow GlobalMLP$
  for $iteration = 1, 2, \ldots$ **do**
    x_l
    for $d = 1, 2, \ldots, n_{devices}$ **in parallel do do**
      $member_d \leftarrow set\_global\_weights()$
      $w = \frac{1}{d_k} \sum_{d \in \mathcal{P}k} f_d(w)$
    end for
    $G \leftarrow \sum_{d=1}^{k} \frac{d_k}{d} member_d$
    Train central model
  end for

5.4 Results

To ensure the results, part of the data was split in a relation of 0.3 - 0.7 for train and test. All datasets test parts were merged to test the federated server part. Other metrics were used in the model, which are: *accuracy, precision and recall* as represented in the following equations:

$$Accuracy = \frac{TP_{Attack} + TN_{Normal}}{TP_{Attack} + TN_{Normal} + FP_{Normal} + FN_{Attack}} \quad (3)$$

$$Precision = \frac{TP_{Attack}}{TP_{Attack} + FP_{Attack}} \quad (4)$$

$$Recall = \frac{TP_{Attack}}{TP_{Attack} + FN_{Attack}} \quad (5)$$

Once the model was wholly developed to use the data and handle all devices, its results are represented in the table4. Following this representation, the Federated model obtained 100% as the highest accuracy on some devices and 55% as the worst. Additionally, the global model has got 72% of accuracy. These results are made after ten iterations among all members and the Federated server. The middle column shows the accuracy results after the first iteration with the Federate server, and the right column shows the last iteration.

Table 4. Accuracy results at 1st and last interaction

F. Members	1st iteration accuracy	last iteration accuracy
device01	0.55	0.55
device03	0.939	0.939
device07	0.999	0.997
device08	0.899	1.0
device09	0.993	0.999
device17	0.501	0.55
device20	0.987	0.67
device21	0.997	0.999
device33	0.726	0.737
device34	0.917	0.920
device35	1.00	1.00
device39	1.00	1.00
device42	0.988	0.988
device43	0.562	0.562
device44	0.14	0.543
device48	0.997	0.997
device49	1.00	1.00
device52	1.00	1.00
device60	0.994	1.00
HoneyPot4	1.00	0.600
HoneyPot5	1.00	0.623
HoneyPot7	1.00	0.76
F_server	0.669	0.70

It is important to notice some changes in accuracy on some devices. Referring to the 4, Device08 goes from 89% accuracy to 100%, which clearly shows an improvement. On the other hand, the device HoneyPot4 shows a decrement in its accuracy. This device had 100% of accuracy in the first iteration but 60% at last. It is essential to highlight that none of the HoneyPot devices contained malicious data. Furthermore, there is an improvement through each iteration in the federated server. The accuracy of the model improves from 66.9% to 72% of accuracy.

Furthermore, the model obtained 0.92 as recall and 0.85 precision at the first iteration. There was an improvement in the recall and precision after ten iterations since they got 0.98 and 0.883, respectively. Thus, the model is more able to distinguish the true positives from the false negatives given its recall. However, its precision says the model conflicts in differentiating true and false positives.

It is important also to highlight that this experiment was conducted on a virtual machine from a server with 8 core virtual CPU, it has an architecture ARMc8.2 with 32GB RAM and NVIDIA GPU with 64 Tensor Cores, and the algorithms used in this research project were implemented with the use of python3.9 using libraries like "Pandas" and "Keras" by TensorFlow.

5.5 Interpretation

Surprisingly, the model's accuracy is not the best. However, the precision and recall show that the model is quite biased for the categories among all data. It is because the model achieved a good

recall, which means it identifies true positives from false negatives. However, it struggles with precision; thus, there is no patron to identify false positives. It is visible on the HoneyPot devices, where their accuracy decreases with the interaction with the other devices that contain malicious data. Hence, the large amount of data in the category of malware makes the data unbalanced even as separate devices. Theoretically, these devices should maintain their 100% of accuracy, but it does not happen because of the large amount of malware data and the Federated Averaging algorithm. It is essential to remember that this algorithm makes use of the weights. Thus, the results of the devices with more data will predominate on this average. In summary, the model either can not distinguish benign or malicious traffic or is biased because of the amount of malicious traffic data.

6 CONCLUSION

Using Federated Learning is a promising feature in the IoT field. However, the use of FL in this data set was limited by its structure and the high correlation among columns and the label column. Moreover, the dataset is unbalanced between malware and benign, making the model biased because of the significant amount of malware data. It is essential to highlight that the performance of the FL was beneficial among all independent members (devices) of this method, and its accuracy increased after each iteration, even if the global accuracy was not as expected.

7 LIMITATIONS AND FUTURE RESEARCH

The most evident limitation was in the DataSet by cleaning the noise values and encoding to accomplish two things: to avoid a dataset with several categories and to validate it for use with the Neural Network in the FL method thus, basic cleaning of data was made. Moreover, the dataset is unbalanced, making the learning process biased and harder to predict. Federated Learning has vast potential as an AI model that should be improved. There are novel and more complex algorithms that can be implemented. These algorithms have not been implemented and only exist as mathematical proof. Thus, a vast field of FL can be applied to IoT. This method can be more beneficial using unsupervised learning since the federated members need an update, not only from the federated server but also by getting new data.

ACKNOWLEDGMENTS

Words cannot express my gratitude to my supervisor and chair for their help and valuable feedback. I could not have done this without the support of my friends that followed me through this journey.

I want to express my deepest gratitude to SENESYT (Secretaria Nacional de Educación Superior, Ciencia, Tecnología e Innovación del Ecuador) because this endeavour would not have been possible without their support. As a GAR student, I am grateful for the financial support to fulfil the bachelor's program in Technical Computer Science.

Lastly, I would be remiss in not mentioning my family, especially my parents and sister, since they always believe in me and my potential. Their motivation and support helped me during this process.

REFERENCES

[1] 2022. ML Brain Enterprise AI Design a federated learning system in seven steps. *Integrate.ai* (2022). https://integrate.ai/blog/design-a-federated-learning-system-in-seven-steps-pftl/

[2] S. Abirami and P. Chitra. 2020. Energy-efficient edge based real-time healthcare support system. 339–368. https://doi.org/10.1016/bs.adcom.2019.09.007

[3] Ekkono Solutions AB. 2020. Federated Learning. https://www.ekkono.ai/wp-content/uploads/2020/12/SWP_Federated_Learning_Ekkono_Solutions_May_2020.pdf

[4] Kenneth Kimani, Vitalice Oduol, and Kibet Langat. 2019. Cyber security challenges for IoT-based smart grid networks. *International Journal of Critical Infrastructure Protection* 25 (6 2019), 36–49. https://doi.org/10.1016/j.ijcip.2019.01.001

[5] Yi Liu, Neeraj Kumar, Zehui Xiong, Wei Yang Bryan Lim, Jiawen Kang, and Dusit Niyato. 2020. Communication-Efficient Federated Learning for Anomaly Detection in Industrial Internet of Things. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*. IEEE, 1–6. https://doi.org/10.1109/GLOBECOM42002.2020.9348249

[6] Yang Lu and Li Da Xu. 2019. Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics. *IEEE Internet of Things Journal* 6, 2 (4 2019), 2103–2115. https://doi.org/10.1109/JIOT.2018.2869847

[7] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2016. Communication-Efficient Learning of Deep Networks from Decentralized Data. (2 2016).

[8] McMahan Brendan and Ramage Daniel. 2017. Federated Learning: Collaborative Machine Learning without Centralized Training Data.

[9] Mohamed Amine Ferrag, Othmane Friha, Djallel Hamouda, Leandros Maglaras, and Helge Janicke. 2022. Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. (1 2022). https://doi.org/10.36227/techrxiv.18857336.v1

[10] Sandro Nižetić, Petar Šolić, Diego López-de-Ipiña González-de Artaza, and Luigi Patrono. 2020. Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future. *Journal of Cleaner Production* 274 (11 2020), 122877. https://doi.org/10.1016/j.jclepro.2020.122877

[11] Jianli Pan and Zhicheng Yang. 2018. Cybersecurity Challenges and Opportunities in the New "Edge Computing + IoT" World. In *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*. ACM, New York, NY, USA, 29–32. https://doi.org/10.1145/3180465.3180470

[12] Valerian Rey, Pedro Miguel Sánchez Sánchez, Alberto Huertas Celdrán, and Gérôme Bovet. 2022. Federated learning for malware detection in IoT devices. *Computer Networks* 204 (2 2022), 108693. https://doi.org/10.1016/j.comnet.2021.108693

[13] Sebastian Garcia, Agustin Parmisano, and Maria Jose Erquiaga. 2020. IoT-23: A labeled dataset with malicious and benign IoT network traffic . *Zenodo* (2020). https://doi.org/10.5281/zenodo.4743746

[14] TensorFlow Developers. 2022. TensorFlow. https://doi.org/10.5281/zenodo.6574269

[15] The European Union Agency for Cybersecurity. 2017. Internet of Things (IoT). https://www.enisa.europa.eu/topics/iot-and-smart-infrastructures/iot

[16] Tuo Zhang, Lei Gao, Chaoyang He, Mi Zhang, Bhaskar Krishnamachari, and A. Salman Avestimehr. 2022. Federated Learning for the Internet of Things: Applications, Challenges, and Opportunities. *IEEE Internet of Things Magazine* 5, 1 (3 2022), 24–29. https://doi.org/10.1109/IOTM.004.2100182