Wireless Network Intrusion Detection and Analysis using Federated Learning

by

Burak Cetin

Submitted in Partial Fulfillment of the Requirements

for the Degree of

Master

of

Computing and Information Systems

YOUNGSTOWN STATE UNIVERSITY

May, 2020

Wireless Network Intrusion Detection and Analysis using Federated Learning

Burak Cetin

I hereby release this thesis to the public. I understand that this thesis will be made available from the OhioLINK ETD Center and the Maag Library Circulation Desk for public access. I also authorize the University or other individuals to make copies of this thesis as needed for scholarly research.

Signature:

Burak Cetin, Student Date

Approvals:

Alina Lazar, Thesis Advisor Date

Dr. Feng Yu, Committee Member Date

Dr. John R. Sullins, Committee Member Date

Dr. Salvatore A. Sanders, Dean of Graduate Studies Date

ABSTRACT

Wi-Fi has become the wireless networking standard that allows short-to medium-range device to connect without wires. For the last 20 years, the Wi-Fi technology has been so pervasive that most devices in use today are mobile and connect to the internet through Wi-Fi. Unlike wired network, a wireless network lacks a clear boundary, which leads to significant Wi-Fi network security concerns, especially because the current security measures are prone to several types of intrusion. To address this problem, machine learning and deep learning methods have been successfully developed to identify network attacks. However, collecting data to develop models is expensive and raises privacy concerns. The goal of this thesis is to evaluate a federated learning approach that would alleviate such privacy concerns. This work on intrusion detection is performed in a simulated environment. During the work, different experiments have concluded to define points that can affect the accuracy of a model to allow edge devices to collaboratively update global anomaly detection models using a privacy-aware approach. Three comparison tests were done in order to find the optimal results; different training rates, different training methods, different parameters. Using different combinations of 5 parameters - training algorithms, number of epochs, devices per round, round numbers and size of the sample set-, these tests with the AWID intrusion detection data set, show that our federated approach is effective in terms of classification accuracy (with an accuracy range of 88-95%), computation cost, as well as communication cost. In our study, the best case had the most rounds, epoch and the devices per round compared to the others.

Acknowledgements

I would first like to thank my thesis advisor Dr. Alina Lazar of the Department of Computer Science and Information Systems at Youngstown State University. The door to Prof. Lazar's office was always open whenever I ran into a trouble spot or had a question about my research or writing. She consistently allowed this thesis to be my own work but steered me in the right direction whenever she thought I needed it.

I would also like to thank the committee members Dr. Feng Yu and Dr. John R. Sullins for their precious time and advice during my thesis process.

I would like to express my sincere gratitude to the Department of Computer Science and Information Systems and the College of Graduate Studies for the financial support they provided during my graduate studies.

Finally, I must express my very profound gratitude to my family, my girlfriend, and to my friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Table of Contents

List of Figures

List of Tables

1 Introduction

During the last 10 years, in the fields of artificial intelligence, machine learning and deep learning, scientists introduced new solutions, algorithms and hybrid methods. Developments in hardware and software brought new systems for the AI world, paving the way to future improvements. In the same time the proliferation of sensors and other wireless devices made possible the collection of large amounts of data. Big Data provides the raw material for the development of any artificial intelligence, machine learning, and deep learning systems [1].

To produce accurate results, prediction systems, speech recognition systems, image recognition systems, chat robots, etc. require large amounts of data for training the models. The typical procedure is to collect data from multiple devices and move it to a central location for training and model development. In this context, low-latency and privacy are two issues to be considered and addressed.

The difference in computational and communication capabilities between devices on the network can affect the efficiency of data collection. The privacy of personal data is one of the biggest issues to be considered when collecting data from many users. For example, sharing the blood test results used to diagnose diseases with other people is not allowed. However, the more data available to doctors, the deep learning methods make it easier to diagnose diseases. With the "Federated Learning" which is newly introduced to the literature, both the privacy of the person is protected, and the big data need is obtained.

1.1 Definition of Federated Learning

To solve the issues mentioned above, Federated Learning [2] has been proposed as a collaborative machine learning solution in which user data never leaves users' devices and the training process is distributed among many users, taking place on the device. Firstly, introduced by scientists at Google, [3], the Federated Learning algorithm is aggregating models trained on the user devices by sending these models from multiple devices to a central location. In this context, data privacy is preserved because only the models are sent instead of the raw data. In classical machine learning, training data is collected in a center location, while in federated learning, personal information remains on user devices.

1.2 Cyber Attacks, Computer Security and Invasion of Privacy

While technological advances make our lives easier, on the other hand, gradually violating our privacy areas. That poses great problems for both individuals and institutions. Securing critical information for organizations or individuals has become a priority. Any security breach or loss of data can have serious consequences, which can lead to personal data corruption, violation of laws and regulations, and loss of money and reputation. Therefore, security is very important, especially on a business scale.

With the increasing amount of information, it has become difficult to secure systems and data. As information about current attacks, vulnerabilities, and methods of defense spread, the attacks become more complex. Because of the ease of access

to information today, it allows individuals to use different types of tools to exploit security vulnerabilities without information. For example, our keyboards learn from our correspondence and suggest the words we will write next [3], and Netflix's banner for the series/movie suggestions according to our marketing technique monitoring preferences. An even more advanced dimension, Target Company, which is one of the important retail companies of the industry operating in the USA, begins to send coupons according to certain stages of their pregnancy by predicting the probability of getting pregnant and when they can give birth by analyzing their buying habits [4]. When one of these coupons comes to the home of a high school student living in the city of Minneapolis, the girl's father calls the store manager and then asks her for information. After this incident, Target officials started to pay attention to their submissions considering that such situations would disturb their customers.

These experiences can give us an idea of how important it is to protect digital personal data and privacy.

1.3 What is an Intrusion Detection System (IDS)?

Increases in the number of spyware, mobile threats and cyberattacks have increased the importance of computer security and the various automated approaches that can be to be used in order to ensure the security of information and computer systems in general [1]. Computer networks are complex structures and are interconnected to other networks with many access points.

The diversification of cyberattacks has shown that complex network systems can no longer be protected only by encryption or firewalls and that makes real-time detection of network traffic and detection of attack attempts inevitable. As the threats

and dangers in the electronic environment mostly endanger the systems from outside, it is very important to detect the threats and attacks entering these environments in advance, and intrusion detection systems are used for this purpose.

Intrusion Detection Systems (IDS) are software and/or hardware components that act as "alarms" in the protection of information systems against attacks over the network [5]. By using IDSs, unauthorized access to the systems and abuse can be determined and attempts by the attackers to infiltrate the systems can be prevented. With the use of IDSs in computer systems, important information such as the type of attacks are made on the system, current deficits, and the attacker's profile can be obtained.

IDS systems have functions [6] such as frequent monitoring of the network, identifying and recording possible threats, stopping attacks and reporting to security administrators. These systems can also be used in some cases to reveal weaknesses in institutions' security policies. IDSs can also detect attackers' network-related information gathering activities and stop attackers at this early stage.

In the internet environment, software or hardware tools such as firewall, antivirus or intrusion detection systems have been designed to prevent internet-based attacks due to the risk of network attacks. The network system uses one or more of these security software to protect important data and the system from attackers or hackers. Relying on a stand-alone firewall system is not enough to prevent attacks on corporate networks or personal networks. For this reason, an intrusion detection system is also used to find the security vulnerabilities.

1.4 Cyberattacks Identification with IDSs

Structures used in intrusion detection systems designed to detect anomaly content are generally divided into three categories: statistical methods, knowledge-based expert systems, and machine learning techniques [7]. Machine learning techniques are widely used in intrusion detection system designs to detect and prevent attacks. Thanks to these systems, the data with the content to be labeled as an attack is normal or not on the network traffic, and the system is tried to be prevented from being damaged by the attacks. Intrusion detection systems that examine network traffic work with two approaches. Misuse detection systems carry out transactions on the misuse signatures/rules registered in the system. Anomaly detection systems, on the other hand, try to detect an unusual situation by examining the compatibility of the system with normal traffic. However, these systems are very vulnerable to new attacks.

With the evolution and increase in popularity of communication technology, mobile and IoT devices, Wi-Fi technology is widely used because of its advantages in terms of mobility and low price. Compared to wired computer networks, Wi-Fi networks are not only slower but also require additional security layers. The fact that data packets are transmitted through the air, and easily intercepted and tampered with makes Wi-Fi networks vulnerable to various kinds of attacks. Therefore, there is an urgent need for Wi-Fi security defense methods that are fast, cheap and efficient. The server-side intrusion detection is a good approach that can provide security by checking each network transfer to detect any wireless intrusion attacks.

1.5 Federated Learning Implementation of Wireless IDSs

Machine learning techniques are successfully used in wireless intrusion detection systems (WIDs). In the literature, there are many WIDs constituted with different machine learning techniques [8, 9]. Support Vector Machine (SVM), Artificial Neural Networks (ANN), and Deep Neural Networks (DNN) are the most frequently used machine learning techniques. WIDs built based on classical machine learning or deep learning methods provide good performance for detecting such anomalous events. However, collecting wireless network data to be used for server-side machine learning training, is not only expensive but raises user privacy concerns. Federated learning provides a feasible solution to this problem because only the local models are moved to the server instead of the local data.

Network anomaly detection [10, 11] can be defined as the process of identifying unusual activities or attacks taking place in the network. When designing and building intrusion detection tools, especially using machine learning algorithms, the main goal is to correctly predict intrusions. In addition, reducing the number of false positive instances or the attack instances classified as normal is a must. This type of data can be passively collected by computers directly connected to routers or access points. Data from multiple routers would be sent to a central server where all the processing and modeling would occur. Based on the prediction and analysis results, alerts can be sent to the network engineers to take necessary action. All this requires high network bandwidth connections to the centralized servers in order to move the data, threatens the privacy of the users involved and introduces additional latency into the process.

A solution recently developed [12] uses deep learning neural networks to train models locally on computing devices associated with access points in this case. This

approach should be able to identify attacks where they take place quickly, collect data and use the new data to adjust the local models. This allows the system to correctly identify instances of intrusion right where attacks take place.

In this paper, we focus on designing an approach for wireless intrusion detection systems that would alleviate privacy concerns. Federated Learning method is used since it protects the privacy and collects data easily. Using a performance metrics and FedAvg algorithm, we observed the accuracy of classifications made and communication cost during training. For our proposed model, we used AWID intrusion detection dataset and analyzed the results with the performance metrics.

The remaining part of this work is organized as follows. Section 2 focuses on the works done in the literature related to ML, FL, and AI. Section 3 includes methods we have used for our proposed model and gives some brief explanations about them. Section 4 provides knowledge about the data used in the work. Section 5 focuses on the experiments discussed in the methods section and their results and finally, Section 6 summarizes the work done and discusses the future works.

2 Related Works

Internet is an indispensable part of our daily life. Increasing number of web applications and users brought some risks in terms of data security. Intrusion detection systems, one of the most important tools for network security, are successfully used to detect attacks on secure internal networks and unexpected access requests. Applications are expanding and diversifying due to constantly developing technological products and changing living conditions. This change brings new system weak-

nesses and, accordingly, new attack methods. Many IDS models that use machine learning-based analysis methods have been developed since anomaly-based IDSs are more successful in detecting new attack types than signature-based IDSs and machine learning methods are getting more successful results day by day. Today, many researchers are working to achieve a more effective intrusion detection system. For this purpose, there are many intrusion detection systems in the literature that have been realized with different machine learning techniques.

Anomaly detection is an important studied task that has applications to network intrusion and has been studied in the wired and wireless settings. A diverse set of implementation, ranging from statistical approaches [10, 11] to machine learning [8] and deep neural networks is available. The use of deep learning as a state-of-the-art approach for wireless intrusion detection has been investigated in several recent studies [9, 13, 14].

Ramamoorthy et al. have developed two different IDS models using Support Vector Machines and Artificial Neural Networks as the analysis method with the help of DARPA (U.S. Defense Advanced Research Projects Agency) dataset [15]. During their work, they have trained SVM's using DoS attack-like patterns and used benchmark data from DARPA. They have achieved 99% accuracy with their models, showing their classifier efficiency.

Depren et al. have developed a model trained with SOM (Self Organizing Map), a method of unsupervised machine learning, using the KDDCup99 data set [16]. They have utilized a self-organizing map (SOM) structure to construct a model to detect anomalies and describe normal traffic. Authors suggest that normal traffic representing normal behavior is clustered around cluster centers, while irregular traffic is

clustered outside of the normal traffic clustering, having high quantization error rates when classifying behavior types. They have trained their proposed SOM with normal traffic data and determine the type of traffic by clustering location and quantization errors. The authors claim an accuracy of 99.1%.

For example, Wang et al. [9] analyzes network attacks in the Wi-Fi setting by comparing the results of two Deep Neural Network (DNN) architectures and one Stacked Autoencoder (SAE) in terms of network attacks classification. The approach presented in this paper improves upon the method described by Thing [14]. According to the paper, they used the Aegean Wi-Fi Intrusion Dataset (AWID) reduced dataset and classified the network records into four categories: normal, injection attacks, impersonation attacks and flooding attacks. They report classification accuracy above 98.3% for three of the classes and 73% only for the flooding attack class. Also, [13] proposed a different architecture based on the popular ladder network implementation and achieved even better results with an overall accuracy of 98.54%.

Wang et al. Used the 5-class KDDCup dataset to compare the model they proposed in their study that is the fuzzy clustering artificial neural networks model (FC-ANN) by decision trees, Naive Bayes (NB), and ANN [17]. Their proposed model utilizes fuzzy clustering, ANN models and a fuzzy aggregation module as a meta-learner. Their experimental results show that their proposed approach outperforms some of the well-known methods in terms of precision and detection. They claim that their proposed methods provide improvements on detecting R2L and U2L classes.

It is seen that multi-user mobile services are used in the application areas of Federated Learning methods. The work done by Yang et. al. focuses on the application aspect of Federated Learning on their Gboard application to improve some

application functions without breaching user privacy [3]. In the work done, they have improvement their click-through-rate by up to +33.95%. On another example, Firefox used Federated Learning to develop the URL bar. In a study of about 360,000 users, the Firefox URL bar shows suggestions when users type in a search query [18]. Some of these suggestions are provided directly by a search engine. Others are produced by Firefox based on the user's history, bookmarks, or open tabs.

The concept of federated learning is an emerging paradigm, initially proposed by Google researchers [12]. This first paper showed the applicability of deep convolutional neural networks in the federated setting for image classification tasks and next word prediction tasks [12, 19]. While we aim to use a similar general architecture, we apply federated learning to the intrusion detection task. However, this problem is harder, compared to image classification tasks because of the data imbalance issue. The AWID dataset has a 10:1 ratio between the normal and abnormal instances.

By the work done by Nguyen et al., an autonomous self-learning distributed system named DIOT is proposed to detect compromised IoT devices [20]. Their system utilizes federated learning to aggregate behavior profiles to detect anomalies caused by malicious parties. They have tested their system on 30 IoT devices and claim a detection rate of 95.6% and a detection speed of 257 ms when testing against Mirai malware.

Previous studies [21, 22, 23] describe experiments for the intrusion detection task in a federated environment using the KDD 1999 cup data and the AWID dataset respectively. In the study developed by Preuveneers [23], federated learning is combined with block-chain technology to prevent malicious cyber-attacks. The experiments described here were applied to a realistic intrusion detection use case (AWID

dataset) and using SAE models for anomaly detection. They show that the addition to block chain has small effects on the performance of the federated learning.

Tuor et. al. proposes a method for selecting relevant data distributively, which is a benchmark model trained on a task specific small-benchmark dataset [24]. The goal is to evaluate the relevance of data samples and selecting a highly relevant data. Their proposed method utilizes federated learning on clients' selected subsets of their data. During the work, they have evaluated their proposed approaches effectiveness using different real-world datasets with a large number of clients and claim a 25% improvement in model accuracy.

As seen in all these studies, there are a lot of researches related to intrusion detection using machine learning, deep learning, and artificial intelligence methods. Most of the time, the data they used in their works was obtained by using certain data sets that were previously collected. In other words, data from daily life was not used. This is due to the fact that instant data collection is expensive and difficult, as well as the problem of personal data privacy that may result. At this point, the federated learning method, a new machine learning method, is used to protect data privacy to make anomaly detection on wireless networks.

On the next part of this paper, methods on federated learning are explained and a methodology is proposed.

3 Methods

In this thesis, we propose and evaluate a federated learning [12] method for building WIDs models. As you can see in the Figure 1, this approach allows edge

devices to use locally collected data to train their local models first. Next, a global model is constructed by averaging the local models. In this way, the edge devices do not have to share their raw training data that may contain sensitive information. Mobile or edge devices train a local model, and send only model parameters to the server, instead of the raw training data. We apply this federated learning approach

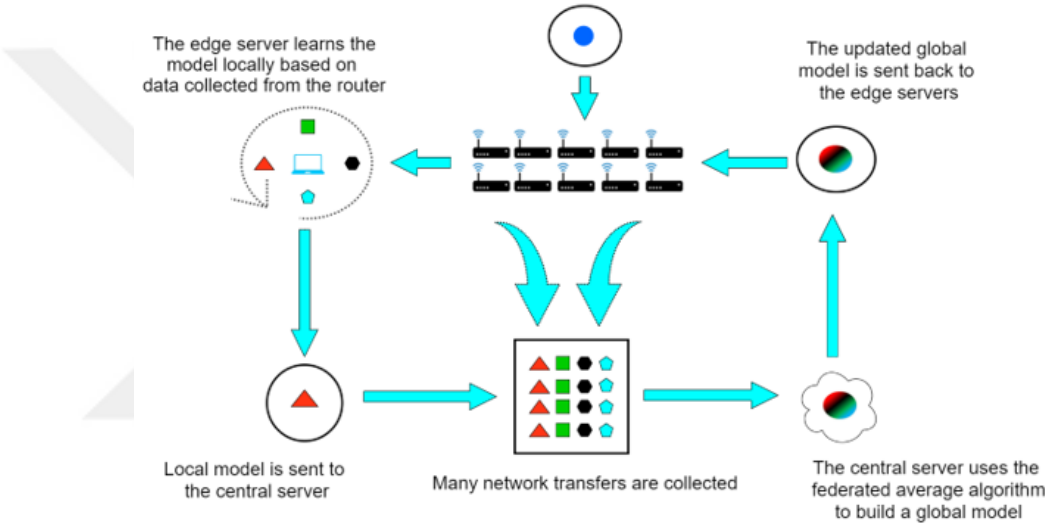

Figure 1: Federated learning process diagram

to classify outgoing network transfers, namely predicting whether a transfer is normal or attack. The approach we used for this research uses an unsupervised method called Stacked Autoencoders for the anomaly detection. Methodological challenges we had to address in order to apply federated learning to perform intrusion detection include: feature selection, deep learning model choice and tuning the federated learning parameters. We also evaluated our methodology using the AWID wireless intrusion dataset.

3.1 Stacked Autoencoders (SAE)

An autoencoder is a neural network that copies values from the input layer to the output layer. In other words, we recreate the data we provide as an input to the neural network in the output layer [25]. It is an unsupervised learning model in which labels are not clearly identified when training the data set (unlabeled). An autoencoder is also known as a self-supervised learning model because it produces its own labels while training the data. This neural network consists of two separate phases; encoder and decoder. As we can see in the Figure 2, the encoder creates the h code with the function f from the input $x, h = f(x)$, the decoder converts this h code with the function g into the r output, since $r = g(h)$, where the input (x) and the output (r) will be equal to each other so this process is called reconstruction. One point to note here is that if model $g(f(x)) = x$ is successfully learned in everywhere, this will not actually be of much use to us; because it copied the input exactly as the output and didn't create a useful code sample to represent the input.

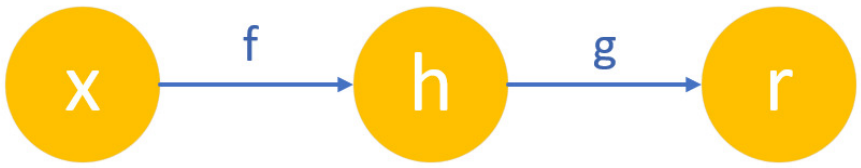

Figure 2: Simple Visualization of Function of Autoencoders

The hidden layer that represents the entry is called code. The simplest description of autoencoders is a feed-forward, non-repetitive neural network with an input layer, an output layer and one or more hidden layers connecting them. The stacked autoencoder is an autoencoder working on multi autoencoders with multi

hidden layers. A stacked autoencoder neural network is an unsupervised learning algorithm that implements backpropagation by adjusting the target values to be equal to the inputs. In the federated learning setting, these algorithms learn from the new observations and update the local and global models in order to identify new trends.

3.2 Logistic Regression

Logistic regression [26] is a machine learning algorithm used to analyze a dataset with one or more independent variables that determine a result. The result is measured by a binary variable (there are only two possible results; 0 or 1). The purpose of logistic regression is to find the most appropriate model to describe the relationship between a series of independent variables and its bidirectional characteristics. Logistic regression generates the coefficients of a formula to estimate the probability of the presence of interest characteristics in the logit transformation. Logistic regression is like the regression problem in which the dependent variable is a categorical variable. It is widely used in linear classification problems. Although it is called regression, it is a classification method. Furthermore, for this research, we used federated averaging (FedAvg) algorithm to make calculation on the dataset.

3.3 FedAvg Algorithm

Federated Averaging (FedAvg) is an approach that uses three parameters over rounds; fraction of clients performing computations, number of training passes on respective local datasets, local minibatch size [12]. This helps to share the load of gradient descent over clients, then the server averages the resulting models, saving time and providing a dynamic adjustancy. In this approach, the server computes

distributed sums, averages and other aggregations; then broadcasts updated models and parameters for the next round.

3.4 I.i.d vs Non-i.i.d

The independent and identically distributed (i.i.d) [27] term is used to describe a particular type of relation between various random variables in probability theory. Each data point is likely to be sampled in the same way. It means, if all of the variables are independent of each other and all variables have the same probability distribution, the set of variables is independent and distributed identically. On the other hand, for Non-i.i.d, the distribution of the variables is related with the raw data. We call this non-i.i.d since the data distribution varies variable by variable.

3.5 LEAF Approach

To evaluate the federated intrusion detection approach we used performance metrics introduced by Caldas [19] for the benchmarking framework LEAF. To capture and analyze the distribution of training and testing performance across devices, the accuracy performance at the 10th and 90th percentiles are recorded for inspection and visualization. Another important metric for federated learning accounts for the total amount of computing resources and communication needs from the edge devices in terms of number of computer operations and number of bytes downloaded/uploaded.

4 IDS Datasets

To validate the proposed method, the Aegean Wi-Fi Intrusion Dataset (AWID) [8] was used. This dataset, published in 2015, contains records labeled "normal" and multiple types of attacks ("attack"). The number of normal and three main attack categories is shown in Table 1. This dataset is currently the largest and most recent Wi-Fi network data publicly available. The data were captured using Wireshark [28] in a small wireless network environment comprised of 11 clients. There are training and testing subsets available.

Table 1: AWID Dataset Distribution

Dataset	**Normal**	**Injection**	**Impers.**	**Flooding**
AWID-CLS-R-Trn	1,633,190	65,379	48,522	48,484
AWID-CLS-R-Tst	530,785	16,682	20,079	8,097
Total	2,163,975	82,061	68,601	56,581
Balanced	205,285	82,061	68,601	56,581

Before running any experiments, we follow the preprocessing, normalization and balancing procedure described by Ran [13]. The resulting balanced dataset has the same number of normal instances as all the attack instances combined. This is showed on the last row of Table 1. To perform federated learning experiments in a simulated environment, using dataset previously collected, we need to distribute the data among devices in a heterogeneous manner such that the number of records and the underlying data distribution varies. We use the LEAF approach [19] to create different number of samples and users using AWID federated learning dataset. Initially, the dataset is divided between devices in a stratified manner. The statistics between clients of different datasets are shown in Table 2, Table 3 and the distributions

are shown in Figure 3, and Figure 4. Each device's dataset is split into training and test datasets and each set of data contains both normal and attack data instances.

Table 2: Statistics of AWID Dataset for 1,000 devices

Number of devices	Total samples	Samples/device	
		mean	std
1,000	107,553	1.98	213.22

Table 3: Statistics of AWID Dataset for 10,000 devices

Number of devices	Total samples	Samples/device	
		mean	std
10,000	78,750,910	0.26	2011.83

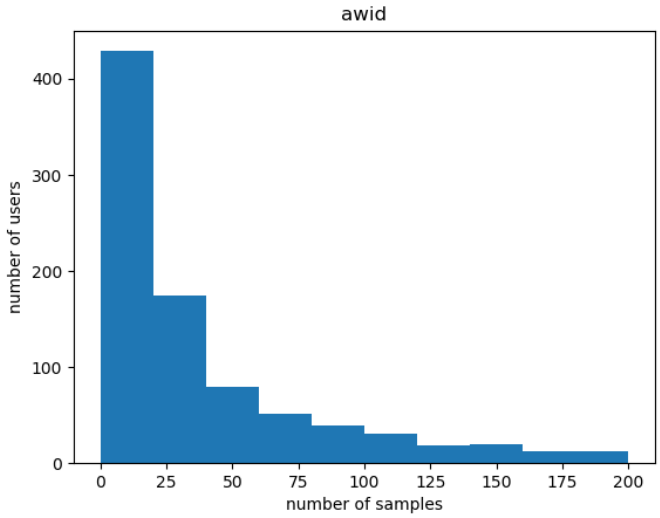

Figure 3: Number of samples histogram for 1000 devices

Results of the experiment carried out will be described thoroughly in the next section.

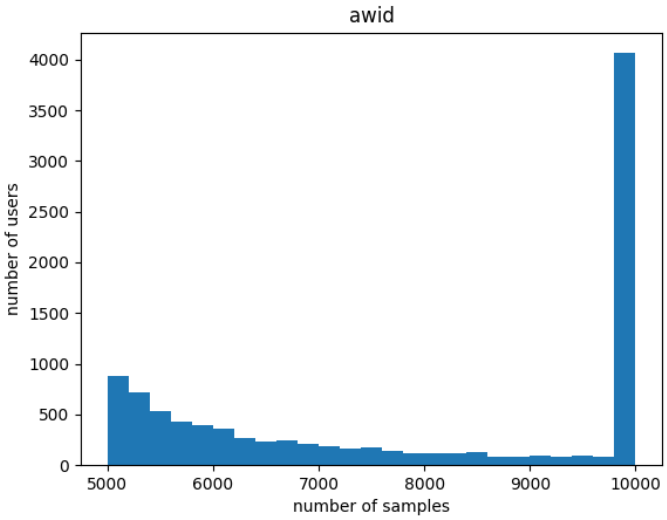

Figure 4: Number of samples histogram for 10000 devices

5 Experiments and Results

In our experiments, we prove the effectiveness of the FedAvg algorithm for the intrusion detection problem. Merging local models by averaging their weights on the central server works well even for a simple neural network model. In this case the input layer has a number of neurons equal to the number of attributes in the dataset (74), and the output layer has a number of neuron equal with the total number of classes (4). The neurons are equipped with the sigmoid activation function, and the loss is sparse softmax cross entropy. Experiments are made in a fashion that shows the difference between features and will be explained in respective subsections.

5.1 A Test of Different Training Rate

At this part, it is intended to show the effect of different frequencies of trainings made in rounds by changing the round numbers between each training. In the

experiment set shown in Figures 5-6, trainings are done in each round, whereas in the set shown in Figure 7-8 trainings are done in every 20 rounds.

5.1.1 Experiments Using 20 Rounds (Set 1)

We use a learning rate of 0.8, 10 devices per round and 20 rounds for all experiments. The convergence behavior of the FedAvg algorithm is shown in Figure 5. We also show the total communication cost in terms of bytes written and read during the training in Figure 6.

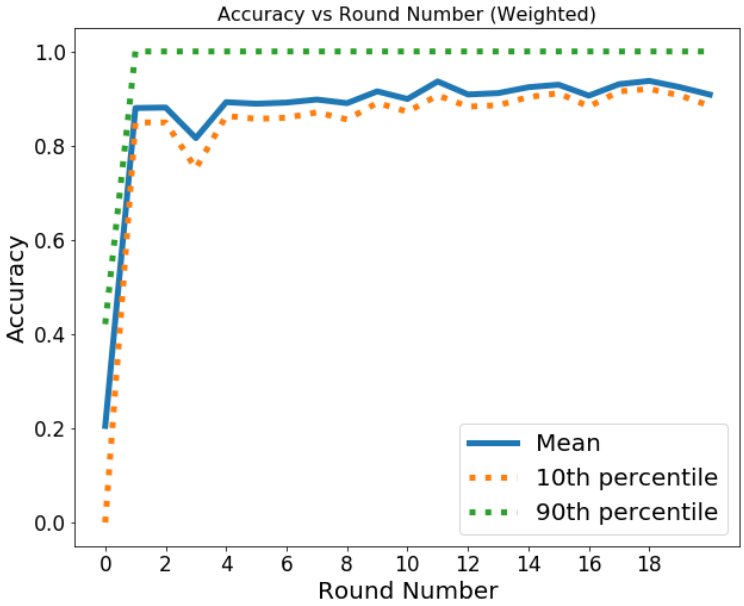

Figure 5: Accuracy vs Round Number for the subset number 1

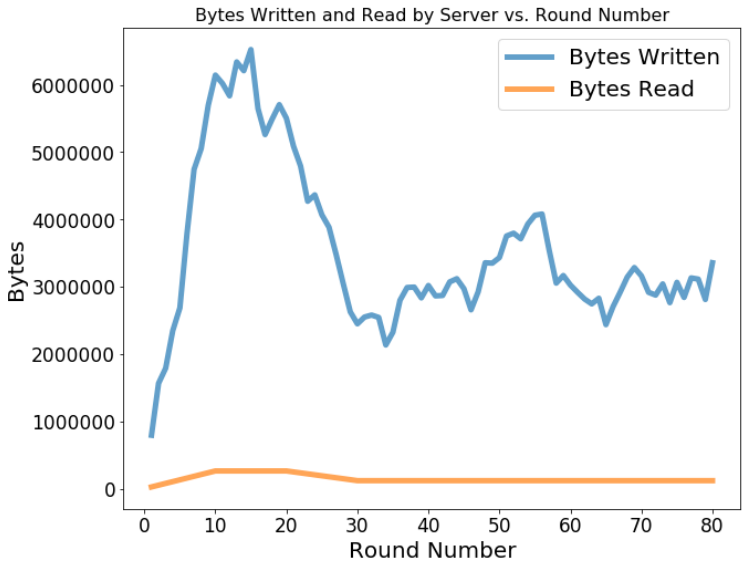

Figure 6: Bytes Written/Read by Server vs Round Number for the subset number 1

Train and test accuracies in Experiment Set 1 are comparable to the results reported by Wang in [9].

5.1.2 Experiment Using 60 Rounds (Set 2)

Experimental results are shown below at Figures 7 and 8 for a learning rate of 0.8, 60 devices per round and 60 rounds. It should be noted that a training is done by 20 round iterations in this experiment while in other cases, the data has been trained at each round to show the difference.

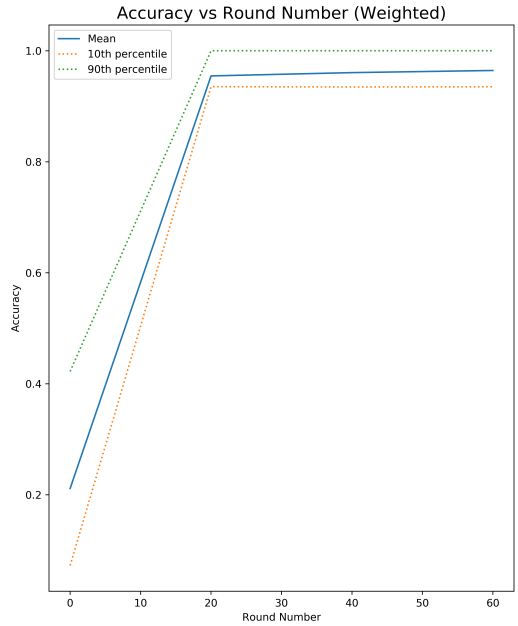

Figure 7: Accuracy vs Round Number for the subset number 2

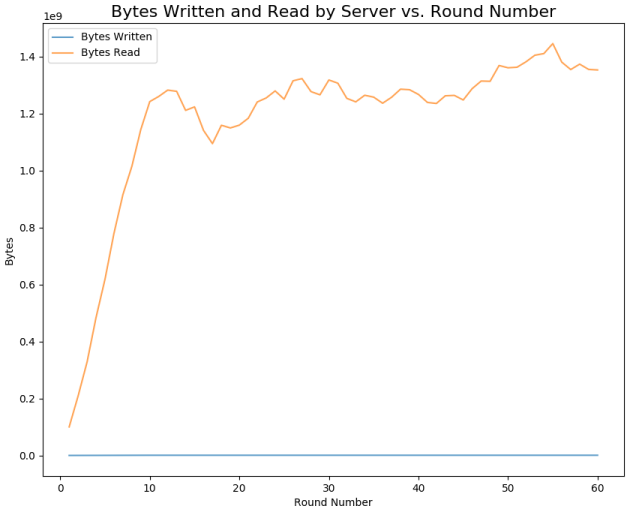

Figure 8: Bytes Written/Read by Server vs Round Number for the subset number 2

These parameters give an accuracy of 97.77%, reading a total of 133306400 bytes in the last round. Most notable change in these graphs is that the accuracy change by rounds is more severe, and some training is needed to achieve high accuracy percentages.

5.2 A Test of FedAvg vs. Local Epochs Method

At this part, it is intended to show the effect of different algorithms applied. On the same data two experiment sets have been created and run with same arguments, but with different algorithms. In the experiment set shown in Figures 9-10, FedAvg algorithm has been utilized, whereas in the set shown in Figure 11-12 AWID's integral experiment is used.

5.2.1 Experiments Using FedAvg (Set 3)

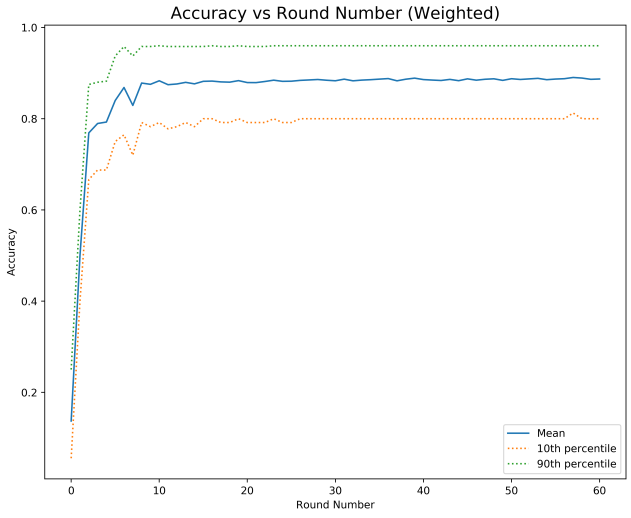

Figure 9: Accuracy vs Round Number for the subset number 3

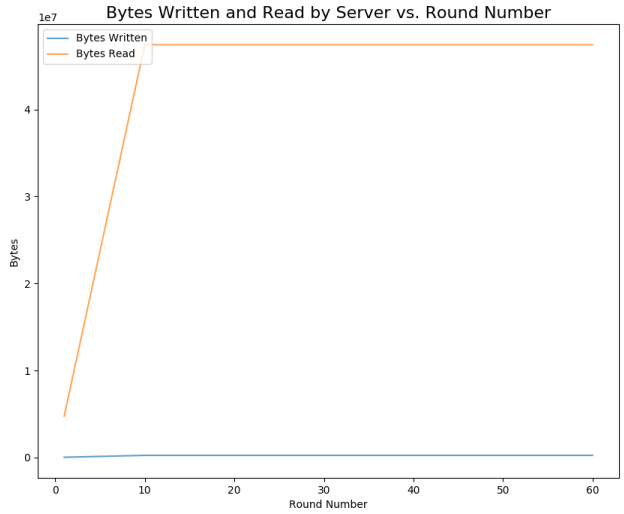

Figure 10: Bytes Written/Read by Server vs Round Number for the subset number 3

FedAvg algorithm gives an accuracy of 88.72%, reading a total of 4744000 bytes in the last round.

5.2.2 Experiment Using Local Epochs (Set 4)

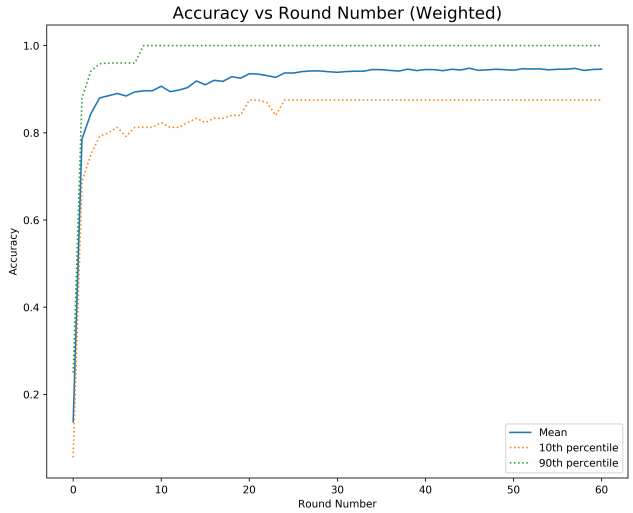

Figure 11: Accuracy vs Round Number for the subset number 4

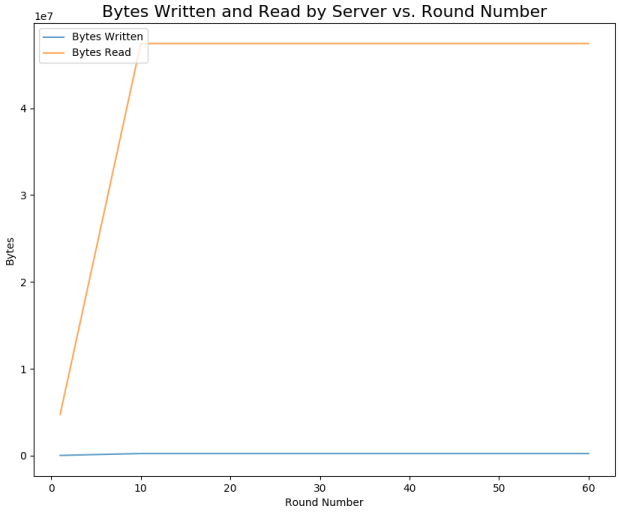

Figure 12: Bytes Written/Read by Server vs Round Number for the subset number 4

Local epochs method gives an accuracy of 94.59%, reading a total of 4744000 bytes in the last round.

We trained the data with same parameters for both FedAvg algorithm and another algorithm which uses local epochs to train data. The comparison between Experiment Sets 3 and 4 show that utilizing FedAvg results in a lower accuracy, but more privacy-aware.

5.3 A Test of Different Parameters

In the Experiment Set 3 shown in Figures 9-10 and Experiment Set 5 shown in Figures 13-14 different experiments have been done with the same sample set with different parameters to further demonstrate the effects on accuracy. The arguments have been changed are devices per round, round number and epoch.

5.3.1 Experiment Using FedAvg (Set 3)

Experimental results are shown below at Figures 9 and 10 for a learning rate of 0.8, 20 devices per round, 60 rounds and an epoch of 20. These parameters give an accuracy of 88.72%, reading a total of 4744000 bytes in the last round.

5.3.2 Experiment Using 50 Devices, 100 Rounds and 100 Epochs (Set 5)

Experimental results are shown below at Figures 13 and 14 for a learning rate of 0.8, 50 devices per round, 100 rounds and an epoch of 100.

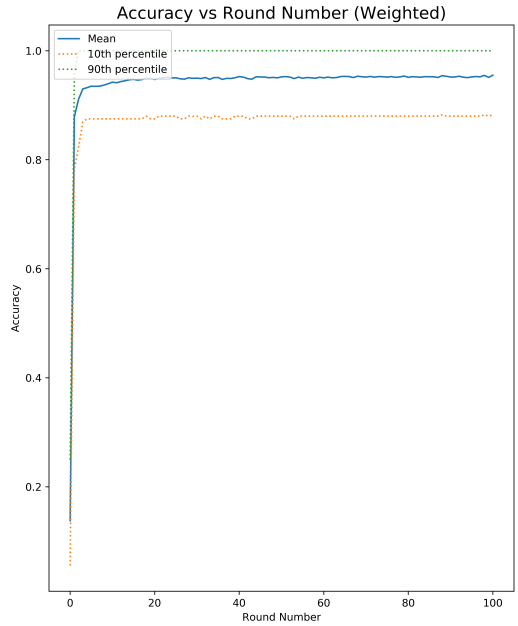

Figure 13: Accuracy vs Round Number for the subset number 5

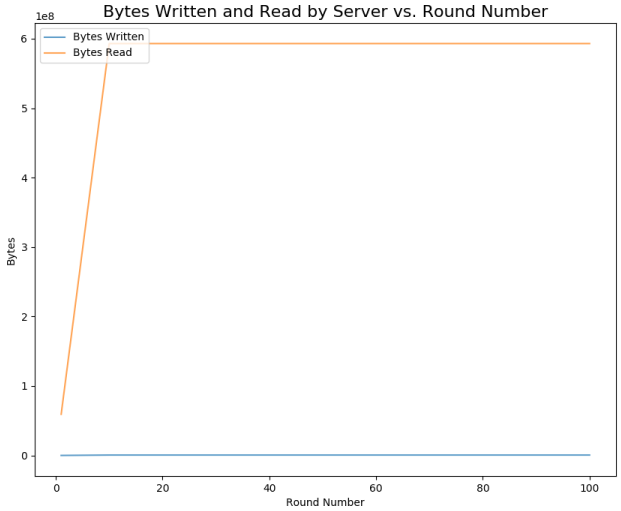

Figure 14: Bytes Written/Read by Server vs Round Number for the subset number 5

These parameters give an accuracy of 95.03%, reading a total of 59300000 bytes in the last round.

This comparison shows that increasing the parameters result in better accuracy, in exchange of reading more bytes in each round and in total. In most of the test cases, we have achieved a 88-95% accuracy in detecting malicious attacks if sample set is trained frequently. Our best test case is Experiment Set 5, shown at Figure 13-14. One other case that should be mentioned is Experiment Sets 1 and 2, where it can be seen that more tests in rounds provide a much more natural curve and stable results.

In this chapter, we have shown the experimental results and compared their different aspects in detail. A general discussion and points for possible future work is given in the last section.

6 Conclusion

In this work, we show how to build federated learning datasets using existing large datasets for performing federated learning experiments in simulated environments. The federated learning model built upon deep learning performs similar to the server-trained deep learning in terms of classification performance when applied to the wireless intrusion detection problem. Compared with the classical deep learning approach,the proposed model has the advantage that does not require moving the data to a central server, preserving user's privacy in this way that is very important for this particular problem. In the future we plan to design and run additional experiments to show the effectiveness of federated learning for the network intrusion detection problem. As a larger version of the AWID dataset is available, so the same experiments can be repeated in a larger setting with even more clients.

7 References

[1] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[2] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

[3] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018.

[4] Kashmir Hill. How target figured out a teen girl was pregnant before her father did. *Forbes, Inc*, 2012.

[5] Stefan Axelsson. Intrusion detection systems: A survey and taxonomy. 04 2000.

[6] Margaret Rouse. What is an intrusion detection system (ids) and how does it work?, Feb 2020.

[7] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*, 2(1):20, 2019.

[8] C Kolias, G Kambourakis, A Stavrou, and S Gritzalis. Intrusion detection in

802.11 networks: Empirical evaluation of threats and a public dataset. *IEEE Communications Surveys Tutorials*, 18(1):184–208, 2016.

[9] Shaoqian Wang, Bo Li, Mao Yang, and Zhongjiang Yan. Intrusion detection for WiFi network: A deep learning approach. In *Wireless Internet*, pages 95–104. Springer International Publishing, 2019.

[10] Shikha Agrawal and Jitendra Agrawal. Survey on anomaly detection using data mining techniques. *Procedia Comput. Sci.*, 60:708–713, January 2015.

[11] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, January 2016.

[12] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient learning of deep networks from decentralized data. February 2016.

[13] J Ran, Y Ji, and B Tang. A Semi-Supervised learning approach to IEEE 802.11 network anomaly detection. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–5, April 2019.

[14] V L L Thing. IEEE 802.11 network anomaly detection and attack classification: A deep learning approach. In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. ieeexplore.ieee.org, March 2017.

[15] S Ramamoorthy, V Shanthi, Srinivas Mukkamala, and Andrew Sung. Knowledge required for detecting and defending against denial of service attacks 1 2 3 4. *International Journal on Intelligent Electronic Systems*, 1, 01 2007.

[16] M. O. Depren, M. Topallar, E. Anarim, and K. Ciliz. Network-based anomaly intrusion detection system using soms. In *Proceedings of the IEEE 12th Signal Processing and Communications Applications Conference, 2004.*, pages 76–79, 2004.

[17] Gang Wang, Jinxing Hao, Jian Ma, and Lihua Huang. A new approach to intrusion detection using artificial neural networks and fuzzy clustering. *Expert Systems with Applications*, 37(9):6225 – 6232, 2010.

[18] Florian Hartmann, Sunah Suh, Arkadiusz Komarzewski, Tim D Smith, and Ilana Segall. Federated learning for ranking browser history suggestions. *arXiv preprint arXiv:1911.11807*, 2019.

[19] Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečný, H Brendan McMahan, Virginia Smith, and Ameet Talwalkar. LEAF: A benchmark for federated settings. December 2018.

[20] Thien Duc Nguyen, Samuel Marchal, Markus Miettinen, Hossein Fereidooni, N. Asokan, and Ahmad-Reza Sadeghi. Dot: A federated self-learning anomaly detection system for iot, 2018.

[21] J Schneible and A Lu. Anomaly detection on the edge. In *MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM)*, pages 678–682, October 2017.

[22] S Xu, Y Qian, and R Q Hu. Data-driven edge intelligence for robust network anomaly detection. *IEEE Transactions on Network Science and Engineering*, pages 1–1, 2019.

[23] Davy Preuveneers, Vera Rimmer, Ilias Tsingenopoulos, Jan Spooren, Wouter Joosen, and Elisabeth Ilie-Zudor. Chained anomaly detection models for federated learning: An intrusion detection case study. *NATO Adv. Sci. Inst. Ser. E Appl. Sci.*, 8(12):2663, December 2018.

[24] Tiffany Tuor, Shiqiang Wang, Bong Jun Ko, Changchang Liu, and Kin K. Leung. Data selection for federated learning with relevant and irrelevant data at clients, 2020.

[25] Guifang Liu, Huaiqian Bao, and Baokun Han. A stacked autoencoder-based deep neural network for achieving gearbox fault diagnosis. *Mathematical Problems in Engineering*, 2018, 2018.

[26] Bfortuner. bfortuner/ml-glossary, 2020.

[27] Longbing Cao. Non-iidness learning in behavioral and social data. *The Computer Journal*, 57(9):1358–1370, 2014.

[28] Anish Nath. *Packet Analysis with Wireshark*. Packt Publishing Ltd, 2015.