FedGroup: A Federated Learning Approach for Anomaly Detection in IoT Environments

Yixuan Zhang[1], Basem Suleiman[1][0000−0003−2674−0253], and Muhammad Johan Alibasa[2][0000−0002−2335−0404]

[1] School of Computer Science, University of Sydney, Australia
{nikki.zhang,basem.suleiman}@sydney.edu.au
[2] School of Computing, Telkom University, Indonesia
alibasa@telkomuniversity.ac.id

Abstract. The increasing adoption and use of IoT devices in smart home environments have raised concerns around the data security or privacy of smart home users. Several studies employed traditional machine learning to address the key security challenge, namely anomaly detection in IoT devices. Such models, however, require transmitting sensitive IoT data to a central model for training and validation which introduces security and performance concerns. In this paper, we propose a federated learning approach for detecting anomalies in IoT devices. We present our FedGroup model and algorithms that train and validate local models based on data from a group of IoT devices. FedGroup also updates the learning of the central model based on the learning changes that result from each group of IoT devices, rather than computing the average learning of each device. Our empirical evaluation of the real IoT dataset demonstrates the capability of our FedGroup model and anomaly detection accuracy as the same or better than federated and non-federated learning models. FedGroup is also more secure and performs well given all the IoT data are used to train and update the models locally.

Keywords: Internet of Things (IoT) · Anomaly Detection · Federated Learning · Machine Learning · Privacy · Smart Home.

1 Introduction

With the advancement of the Internet technologies, it was predicted that the number of Internet of Things (IoT) devices in the smart home environment in 2020 would be 7 devices per person [1]. In smart home environments devices such as sensors, smartphones, and smart TVs are connected to the Internet so that they can be accessed and monitored remotely. To achieve this, data will be continuously sensed so that these IoT devices can perform certain functions. For example, turning on the air conditioner when the temperature reaches a certain degree and switching off the TV when there is no one detected watching TV [2]. The significant growth of IoT devices in smart homes has also brought forward research interests in the very large amount of data that are collected and used to support different types of intelligent services for smart homes [3]. The collected data can be used to develop intelligent data-driven models for enhancing the user's experience of smart homes.

The connectivity of smart home devices to the Internet and continuous data sensing brought a number of key challenges to smart home users, including data privacy and malicious access and control of the sensitive IoT devices. The settings of IoT devices do not take users' privacy and security as a priority. Such IoT devices are often vulnerable to network attacks given it's connected to the Internet, and these attacks can be pervasive [4]. Such attacks may include access and transfer of data being sensed by these devices, remotely switching off security and monitoring cameras, and opening the doors remotely for unauthorised home residents. Recent research reported that around 59% of users are concerned about smart devices listening to them without permission and gathering data without their knowledge [5]. Therefore, it becomes crucial to maintain the highest levels of privacy and security while these IoT devices are used in smart homes. To address this challenge, a large amount of research work that employs AI-based approaches to detect anomalous behaviour in such IoT devices [6][7][8][9]. These approaches heavily focus on traditional machine learning models which also bring new challenges. Such machine learning approaches require transmitting all data sensed from all IoT devices to an external server to train and validate central anomaly detection models. This can be very expensive and could exhaust the bandwidth of the network. It also can expose the sensitive data collected from the IoT devices over the network which makes it vulnerable to cyber-attacks. The situation would not get better even with encrypting and decrypting the data as it could add performance overhead.

In this study, we address the anomaly detection problem on IoT devices by employing a federated learning approach. Federated learning allows the training of local models based on data sensed from a group of IoT devices within a smart home. The local models do not need to transmit the raw IoT data, but only model updates that result from local training. The model (parameter) updates are then shared with a central model which aggregates the values of the learning parameters and then sends them back to all local models, so they can update their learning. Although this federated learning approach addresses the above data security and performance overhead posed by traditional machine learning models, it has its unique challenges. In this paper, we address the research question: How to design local and central models that work in federated learning settings, given various IoT devices in smart homes?

Existing federated learning approaches suggest using the overall average to update the learning parameters shared by all local models. This might not be practical as it ignores the bias of the local models (each might have different data with or without anomalies). We propose a new federated learning model called FedGroup for anomaly detection in smart home environments. We present algorithms that detail the training process on data collected from a group of IoT devices and the process of updating learning parameters in the central model based on the learning results from each group of IoT devices. Our FedGroup addresses the bias resulting from averaging all updates from each device.

The main contributions of this paper are:

1. A new federated learning model and algorithms called FedGroup for anomaly detection on IoT devices. The FedGroup model computes the learning updates based on parameters from a group of IoT devices.
2. Empirical evaluation of FedGroup on real data collected from various IoT devices [6][7][8][9]. The evaluation also includes a performance analysis of the FedGroup against federated learning and non-federated learning models.

The rest of this paper is organised as follows. Section 2 describes related work in the field of anomaly detection for IoTs. Section 3 presents the dataset used in this study and the FedGroup model and algorithms proposed in this paper. Experiments and Results are then presented in Section 4. Section 5 presents key conclusions and future work.

2 Literature Review

Various traditional machine learning and deep learning approaches have been utilised to identify attacks on IoT devices. One study by Stojkoska et al. (2017) [10] suggests that the cloud-centric of holistic IoT-based framework for smart home environments requires substantial data storage and processing infrastructure, and the current state is far from efficient. They highlighted that the new approaches should comprehend the issue of massive data management on the cloud. Furthermore, future studies also have to investigate different methods to ensure security since the cloud-based techniques pose an enormous risk of revealing personal information and data, which are considered as urgent issues.

Past research tended to focus on centralised anomaly detection in which the cloud collects data from various sources. This raises several issues including high communication load and data privacy. Federated Learning (FL) was proposed with the characteristic of lightweight communicating updates, and it was proven to successfully predict text input on mobile devices [11]. FL merges the updates from all the distributed devices thus the calculation on the cloud was significantly reduced, resulting in improved scalability and lightweight communication [12]. Another study [13] compared the IoT intrusion detection using different approaches, including centralised, on-device and FL. The efficiency of FL reached a similar accuracy to the centralised approach. Besides, the study suggests that FL outperformed the on-device approach as it could take advantage of the knowledge from others. FL answers significant drawbacks of centralised ML models that are expensive, computationally difficult, and have low scalability support.

Mohri et al. (2019) [14] indicated that different clients might be weighted differently by FL resulting in unfairness. Fairness in this context refers to both the training data and the training procedures. The study indicates that the uniform distribution is not the common distribution in many cases. Therefore, minimising the anticipated loss concerning the specific distribution is harmful and might lead to a mismatch with the target. Consequently, the study presented an agnostic FL framework in which the centralised model is optimised for any target distribution produced by a mixture of client distributions by utilising data-dependent Rademacher complexity. However, the optimisation of the single worst device is limited for a smart environment with numerous IoT devices. In separate research, Li et al. (2020) [15] concur that unfair distribution of model could bring disproportionate performance since overall accuracy is high but individual accuracy is uncertain. The generated model may be biased towards devices with massive data. Their study developed An enhanced model configured at a more granular scale to ensure equitable device distribution and maintain the same overall accuracy.

A study analysis [16] is crucial for understanding ensemble learning (EL) for network security, and anomaly detection can perform well in results. EL combines multiple learning models and achieves better prediction results. Furthermore, an ensemble of models has a stronger resilience in the face of training data uncertainty. The EL concept has similarity to how FL aggregate the training results. This opens up an opportunity to incorporate ensemble learning with FL.

As previously shown, many studies showed FL showed better performance than traditional ML and confirmed the high privacy level of FL. However, research identifying the attack using FL is still scarce and has not been explored in depth. Besides, there are issues, for instance, the bias of the distributed models is averaged to produce the final global model that will cause unfairness. The past studies neglected the reality that various local models have distinct functionality and structure. The research gaps in the smart home environments are that past research failed to address the similarity of network traffic flow data patterns of device models in the same category. The same type of IoT devices have similar vulnerability structures under similar attacks. Therefore, IoT devices within the same group should use similar parameters for anomaly detection. While it seems like a straightforward method to aggregate updates together, the bias in the training phase arises from the updating of participants' parameters that differ from one another and the selection of the average.

3 Methodology

Our study aims to build an anomaly detection to detect whether there are any attack attempts (Attack Detection). The first section Research Data displays the network traffic flow data, and attack data are the original input data. Then, the Research Method shows the details of designing models, and the Experiment and analysis provide the preparation and evaluation process.

3.1 IoT Datasets

Our dataset was obtained from the UNSW IoT analytics team, consisting of real-world attacks to assess the privacy and security dangers of IoT devices [6][7][8][9]. The dataset was collected from 28 unique IoT devices in various categories and multiple non-IoT devices in the smart environment. There are 30 PACP files consisting of both attack and benign data in two separate stages. The dataset was split into two stages: the first stage is between 28/05/2018 and 17/06/2018, and the second stage is from 24/09/2018 to 26/10/2018. IoT devices are defined as devices linked to the Internet with application logic and executing TCP/IP connection. Ten IoT devices in this dataset contain benign and attack traffic datasets, whereas the others contain only benign data. The datasets used in our study can be found in Flow and Annotation data [3], the implementation of our algorithms and models, and supplementary results and materials can be accessed from the project repository [4]”. Therefore, this research focuses on the selected ten IoT devices with wireless connection to the Internet in four categories listed in Table. 1.

IoT devices			
IoT Devices No.	MAC Addresses	IoT devices	Category
IoT Device 0	00:16:6c:ab:6b:88	Samsung Smart Cam	Camera
IoT Device 1	00:17:88:2b:9a:25	Phillip Hue Lightbulb	Energy management
IoT Device 2	44:65:0d:56:cc:d3	Amazon Echo	Contollers/Hubs
IoT Device 3	50:c7:bf:00:56:39	TP-Link Plug	Energy management
IoT Device 4	70:ee:50:18:34:43	Netatmo Camera	Camera
IoT Device 5	74:c6:3b:29:d7:1d	iHome PowerPlug	Energy management
IoT Device 6	d0:73:d5:01:83:08	LiFX Bulb	Energy management
IoT Device 7	ec:1a:59:79:f4:89	Belkin Switch	Energy management
IoT Device 8	ec:1a:59:83:28:11	Belkin Motion Sensor	Energy management
IoT Device 9	F4:F5:D8:8F:0A:3C	Chromcast Ultra	Appliances

Table 1. IoT devices included in the dataset

The network traffic flow data of the ten IoT devices are collected every minute, marked with activity, and recorded to the ten separate excel network traffic flow data files. The files contain ”Timestamp”, ”NoOfFlows”, and a significant number of attributes of patterns. Several features such as ”InternetTcp”, ”InternetUdp”, ”LocalTcp”, and ”LocalUdp” are the contents of the following ”From” and ”To”, and the contents after ”Port” are port numbers (e.g., ”From###Port###Packet”). Since the packet and byte are not closely connected and the sizes of the packets in this dataset vary, we decided to forecast attacks by including them. Based on the network traffic flow data, it is unknown which network flow is going to which IoT devices or coming from which IoT

[3] https://iotanalytics.unsw.edu.au/attack-data

[4] https://github.com/BasemSuleiman/IoT_Anomaly_Detection_Smart_Homes

devices. The reasons are that different IoT devices use the same port number and use different port numbers simultaneously.

The UNSW IoT analytics team designed a set of attacks comparable to real-world attacks and are particular to several real-world consumer IoT devices. The tools were created in Python to find susceptible and vulnerable devices on the local network by running different tests against them. Then, the program performs targeted attacks on IoT devices that are susceptible. The attack condition includes the start and end time of the attacks, the impact of the attack, and attack types. When determining the normal behaviour or under the attacks, it relies on the rules ”if (flowtime >= startTime × 1000 and endTime × 1000 >= flowtime, then attack = true”. It is multiplied by 1000 since the times are recorded in different units: flow time in milliseconds while start time and end time are not.

3.2 Proposed Approach: FedGroup

FedAvg model accepts the initial model from the central server, training models on decentralised local device servers, and reports the best performance parameters to the central model [11]. For ML, there is only one step which is a client-to-server upload step. In contrast with Traditional Machine Learning, FedAvg sends code to data rather than send data to code. For FL-based learning, there are four steps in one iteration:

1. A server-to-client broadcast step
2. A local client update step
3. A client-to-server upload step
4. A server update step

While it is simple for FedAvg to summarise all the parameters from local servers and select the mean as the following round parameter, the main weakness is the failure to address the similarity of network traffic flow data patterns of the device models in the same category. Furthermore, the devices in smart homes are not assigned into different groups based on their similarity. The devices in the same groups should have similar functionalities and vulnerable risks. The bias in the training procedure was from the updates of device parameters that are different from each other and easily choose the average. Imaging the IoT devices in a smart home are mostly energy management applications such as plugs or blubs, the parameters of cloud server will bias to the energy management devices because the numbers of its are more remarkable than other groups.

Therefore, we present FedAvg with group masters called FedGroup that send parameters to the group master rather than the central server. The group masters have helped combine devices within the same group and aggregate parameters depending on the groups. In the central model, there are multiple group masters then send back the aggregated results to the corresponding participant local model, which is more efficient for the local model to focus on the information within the same group.

The product can be grouped into many categories based on features. The group can be defined based on the category of the IoT devices such as Camera or Energy management. Apart from that, the group can also be defined by other characteristics. If separate groups are based on features, for instance, the smart door product contains many features to open the door such as app control, fingerprint recognition, entering the password, scanning an intelligent card or simple using key unlock. If the smart door is under attack, the central model will tell which specific part is under attack.

The followings are the steps of FedGroup (Fig 1):

1. Every local model computes training network traffic flow data with all parameters and sends the parameters' best results to the central model;
2. Group master in the central model aggregates the parameters based on the group;
3. Group master sends back the aggregated results to devices in the corresponding group;
4. Local models update the models with the new parameters.

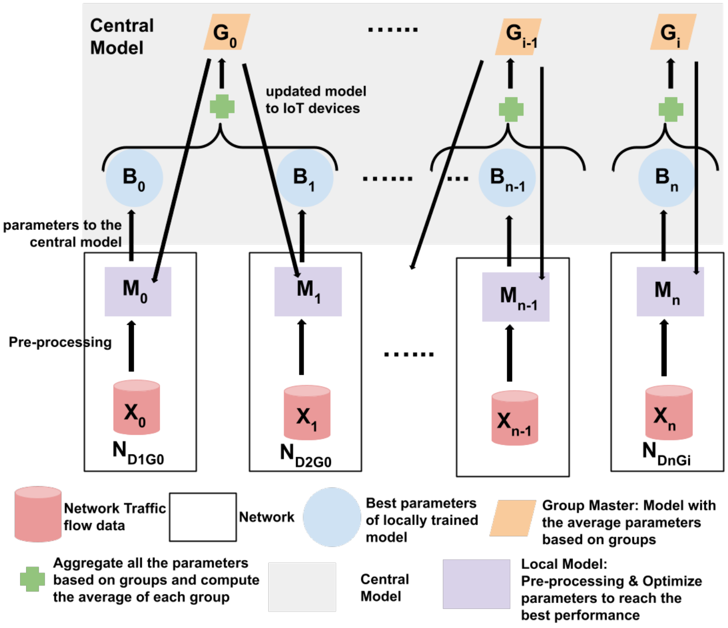
Central Model
G_0
......
G_{i-1}
G_i
updated model to IoT devices
B_0
B_1
......
...
B_{n-1}
B_n
parameters to the central model
M_0
M_1
M_{n-1}
M_n
Pre-processing
......
X_0
X_1
X_{n-1}
X_n
N_{D1G0}
N_{D2G0}
N_{DnGi}
Network Traffic flow data
Network
Best parameters of locally trained model
Group Master: Model with the average parameters based on groups
Aggregate all the parameters based on groups and compute the average of each group
Central Model
Local Model: Pre-processing & Optimize parameters to reach the best performance

Fig. 1. The architecture of our proposed approach FedGroup

Definition: Using X_n to represent the network traffic flow data of the IoT devices and y_n to represent the prediction target. Network N_{DnGi}: N represents network, $_{Dn}$ means Device n and $_{Gi}$ represents Group i. The X_n and M_n are included in N_{DnGi}. During the training, setting the best score S, the best parameter B, the average score of the entire model C, and the average parameters of the entire model A. For each Model M, parameters $P = \{a, b, ...\}$ means parameters such as weights, n_estimator and so on with all possible parameters grid $p = \{a_0, a_1, ...\}, \{b_0, ...\}, ...$ such as n_estimator have parameters 1, 2 and so on. E represents the selected parameters grids in the local models after the update to the central model.

Algorithm 1 FedGroup: Client Side Learning Algorithm

```
INPUT: $P, E$
REQUIRE: $X_n, y_n, M$
OUTPUT: $B$ and $S$ to $Central\,Server\,Side\,Learning : FedAvg$ with the related Group Master
SET: Local Model $M$
/* Fit possible parameters grids and return the best parameters and the best score*/
for $e \in$E do
    Fit $P$ in $M$ with different grid $e$ to train $X_n$ and $y_n$
    Test the $M$ to get the accuracy
    CALCULATE $B$ and $S$
end for
```

Algorithm 2 FedGroup: Group Master Algorithm

```
INPUT: $B$ and $S$ of each $d$
DISPLAY: scores of each groups
OUTPUT: $A$ and $C$ to $Central\,Server\,Side\,Learning : FedGroup$
CALCULATE $A$ and $C$ based on $B$ and $S$
```

Algorithm 3 FedGroup: Central Server Side Learning Algorithm

```
INPUT: $M, P, p$
OUTPUT: $A, C$
/* 1st round: receive the best parameters and best devices from every model, and
calculate the average parameters of each group*/
for $g$ ∈G do
    for $n$ ∈N do
        Initial: $M$
        $Client\,Side\,Learning$ : $FedGroup$ (P, p)
        Return $B$ and $S$ of each $N$
    end for
    $Group\,Master$ : $FedGroup$ ($B$ and $S$ of each $N$)
    Return $A$ and $C$
end for
/* 2nd round: send mean parameter to Client Server and return the mean score
and average parameter of mode*/
for $g$ ∈G do
    for $n$ ∈N do
        $Client\,Side\,Learning$ : $FedGroup$ (P, A)
        Return $B$ and $S$ of each $N$
    end for
    $Group\,Master$ : $FedGroup$ ($B$ and $S$ of each $N$)
    Return $A$ and $C$
end for
```

3.3 Experiment and analysis

For network traffic flow data of the IoT devices, remove "NoOfFlow" because it counts the numbers of flow, which is highly correlated to all the other attributes. Due to the fact that different devices use the same port number and the same device use different port numbers, there are 253 attributes about bytes of port number and packages of the port number in total. When capturing the network behaviour of one device by milliseconds, various port numbers have not been used. In other words, the NaN data means there is no network behaviour for the corresponding port number. That is the reason we have missing data. To fill in the missing data, we assign the most likely value and the global constant to a particular value of 0. It signifies no network behaviour with zeros packet-level and zeroes byte-level network traffic flow data at that time point.

To predict whether it is an attack means that there are two options: attack or non-attack. Implementing Decision Tree, Logistic Regression and Ensemble Learning as local models on ML, FedAvg, and FedGroup as central models to attack detection. To avoid overfitting, StratifiedShuffleSplit split the dataset 80% training and 20% testing dataset. For training data, Stratified 5-Fold Cross-Validation randomly divides the entire data into five folds, fits four folds to the model, and validates the model using the remaining fold. Evaluate the 20% testing data to compute accuracy with an F1 score with a weighted average.

To evaluate the model, the False Positive Rate (FPR) is used to calculate the probability of falsely rejecting the null hypothesis to measure the accuracy of the test.

4 Results

This study developed an anomaly detection system by using our proposed model, called FedGroup as described in the previous section. Table 2 provides the results summary from various models including Decision Tree, Logistic Regression and Ensemble Learning as the local model on traditional ML, FedAvg and FedGroup to attack detection. EL can merge several models even if the individuals are weak, and we use it as an initial local training model. Using ML as an initial model for every IoT device is not always performed as expected because they are used to solve a specific question or a type of question. For example, logistic regression effectively classifies data into discrete classes by investigating the connection between a collection of labelled data. However, If the number of features is greater than the number of observations, Logistic Regression should not be utilised. Considering the various performances that sometimes perform good but sometimes perform not of machine learning models when solving a problem, ensemble learning joins various contributing models to seek better forecasts.

Algorithms		Attack Detection		
Local Model	Central Model	Accuracy	Running Times (seconds)	FPR
Decision Tree	Traditional ML	99.84%	8524	10.04%
Decision Tree	FedAvg	99.85%	154	9.57%
Decision Tree	FedGroup	99.87%	154	7.70%
Logistic regression	Traditional ML	99.76%	21376	24.48%
Logistic regression	FedAvg	99.77%	2912	20.28%
Logistic regression	FedGroup	99.77%	2999	20.18%
Ensemble learning	Traditional ML	99.85%	33940	9.60%
Ensemble learning	FedAvg	**99.91%**	2390	9.03%
Ensemble learning	FedGroup	**99.91%**	2143	9.43%

Table 2. The accuracy of FedGroup, FedAvg and traditional ML using different models

Firstly, anomaly detection can determine whether there is any attack attempt. The highest accuracy of 99.91% of attack detection was reached by the FedGroup model using Ensemble learning as the locally model to train. Secondly, FL-based learning models performed either similar or better than the traditional ML models. Considering the anomaly detection problem is a no binary classification, while the StratifiedShuffleSplit is used to try to solve the problem of overfitting, the accuracy of all models is more than 99%. Therefore, FPR is a more reliable evaluation metric since higher FPR scores indicates higher ratio of negative events are incorrectly categorised as positive. As shown in Table 2, the FPRs of

FL-based are less than the FPRs of the Traditional ML model indicating better performance with less overfitting issue.

The running time of FL-based is less than the traditional ML model where the client slide model spends $O(n)$ and central server takes $O(n^2)$. For example, using Ensemble learning as the local model and FedGroup as the Central Model spends 2143 seconds which is around 1/16 of time spend on Traditional ML (33940 seconds) and 0.9 of time spend on FedAvg (2390 seconds). As a result of lightweight communication, no central authority, and a decentralised learning model, FL uses the advantages of locally training data to reduce the running time. Besides, data safety is guaranteed without sending, communicating or sharing to other IoT devices or the Internet.

Each smart home has a large amount of IoT devices to make our life more efficient and easier. If we focus on the differences of FPRs that are larger than 1%, then FPRs of FedGroup are better than FPRs of FedAvg. Different IoT devices have different vulnerable functions and maybe under different attacks. Meanwhile, one similar attack may have similar functionality or patterns. When the central model learns attack types from the same category of IoT devices, FedGroup is useful to provide parameters of IoT devices within the same group. Besides combining all the smart environments to build smart cities or industries, FedGroup can learn all the attack detection and attack type detection based on group categorisation such as the traffic light group, subway group, and others.

5 Conclusion

In this paper, we introduce a new model called FedGroup model and algorithms which address the issue of IoT anomaly detection in the smart home environments. FedGroup allow training and detecting anomalies based on data collected from group of devices, and thus reduces the vulnerability of the IoT data transmitted and shared on a central server. We evaluate our FedGroup approach on real dataset collected from various IoT devices in smart-home settings to detect anomalous behaviour. Based on our experimental results, it can be concluded that the performance of FedGroup improved in terms of accuracy of anomaly detection compared to the traditional FedAvg. Furthermore, FedGroup can address the issue of fairness of the training procedure and can maintain data privacy, as the values of learning parameters need to be shared with the central model. Our results also demonstrated that Ensemble Learning as local models used in our FedGroup achieved the best accuracy, 99.91%.

While our finding has provided the comparison results of different models, more empirical studies on continuous real-time learning and alternative ways to ensure the fairness of federated learning need to be conducted to test further and refine our findings. Besides, expanding the model to other frameworks not limited to anomaly detection, finding the system cost and how the link instability of wireless networks affects the model updating are several opportunities for in future work.

References

1. Evans, D. (2011). *How the Next Evolution of the Internet Is Changing Everything.* 11.
2. Robles, R. J., & Kim, T. (2010). *Applications, Systems and Methods in Smart Home Technology: A Review.* International Journal of Advanced Science and Technology, 15, 13.
3. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). *Internet of Things (IoT): A vision, architectural elements, and future directions.* Future Generation Computer Systems, 29(7), 1645–1660. https://doi.org/10.1016/j.future.2013.01.010
4. Abomhara, M., & Køien, G. M. (n.d.). *Security and privacy in the Internet of Things: Current status and open issues.* 8.
5. *59 per cent of smart speaker users have privacy concerns – report — Mobile Marketing Magazine.* Mobilemarketingmagazine.com. (2021). Retrieved 23 October 2021, from https://mobilemarketingmagazine.com/59-per-cent-of-smart-speaker-users-have-privacy-concerns-report-.
6. Habibi Gharakheili, H., Sivanathan, A., Hamza, A., & Sivaraman, V. (2019). *Network-Level Security for the Internet of Things: Opportunities and Challenges.* Computer, 52(8), 58–62. https://doi.org/10.1109/MC.2019.2917972
7. Hamza, A., Gharakheili, H. H., Benson, T. A., & Sivaraman, V. (2019). *Detecting Volumetric Attacks on loT Devices via SDN-Based Monitoring of MUD Activity.* Proceedings of the 2019 ACM Symposium on SDN Research, 36–48. https://doi.org/10.1145/3314148.3314352
8. Sivanathan, A., Gharakheili, H. H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A., & Sivaraman, V. (2019). *Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics.* IEEE Transactions on Mobile Computing, 18(8), 1745–1759. https://doi.org/10.1109/TMC.2018.2866249
9. Sivaraman, V., Gharakheili, H. H., Fernandes, C., Clark, N., & Karliychuk, T. (2018). *Smart IoT Devices in the Home: Security and Privacy Implications.* IEEE Technology and Society Magazine, 37(2), 71–79. https://doi.org/10.1109/MTS.2018.2826079
10. Stojkoska, B. L. R., & Trivodaliev, K. V. (2017). *A review of Internet of Things for smart home: Challenges and solutions.* Journal of Cleaner Production, 140, 1454-1464.
11. Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). *Federated Machine Learning: Concept and Applications.* ArXiv:1902.04885 [Cs]. http://arxiv.org/abs/1902.04885
12. McMahan, B., Moore, E., Ramage, D., Hampson, S., & Arcas, B. A. Y. (2017). *Communication-efficient learning of deep networks from decentralized data.* Retrieved from Artificial Intelligence and Statistics, 1273–1282 http://proceedings.mlr.press/v54/mcmahan17a.html.
13. Rahman, S. A., Tout, H., Talhi, C., & Mourad, A. (2020). *Internet of Things Intrusion Detection: Centralized, On-Device, or Federated Learning?* IEEE Network, 34(6), 310–317. https://doi.org/10.1109/MNET.011.2000286
14. Mohri, M., Sivek, G., & Suresh, A. T. (2019). *Agnostic Federated Learning.* 11.
15. Li, T., Sanjabi, M., Beirami, A., & Smith, V. (2020). *Fair Resource Allocation in Federated Learning.* ArXiv:1905.10497 [Cs, Stat]. http://arxiv.org/abs/1905.10497
16. Vanerio, J., & Casas, P. (2017). *Ensemble-learning Approaches for Network Security and Anomaly Detection.* Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks, 1–6. https://doi.org/10.1145/3098593.3098594