Privacy-Aware Anomaly Detection in IoT Environments using FedGroup: A Group-Based Federated Learning Approach

Yixuan Zhang[1], Basem Suleiman[1,2*], Muhammad Johan Alibasa[3], Farnaz Farid[4]

[1]The University of Sydney, Sydney, 2006, NSW, Australia.
[2*]The University of New South Wales, Sydney, 2052, NSW, Australia.
[3]School of Computing, Telkom University, Bandung, 40257, Indonesia.
[3]Western Sydney University, Penrith, 2751, NSW, Australia.

*Corresponding author(s). E-mail(s): b.suleiman@unsw.edu.au;
Contributing authors: yzha7679@uni.sydney.edu.au;
alibasa@telkomuniversity.ac.id; farnaz.farid@westernsydney.edu.au;

Abstract

Concerns on the data security and privacy of smart home users have been growing popularity due to the rising usage of IoT devices. Many traditional machine learning techniques have been used to perform anomaly detections. However, these models need to send private IoT data to a central model for validation and training, raising security and efficiency issues. We propose a new Federated Learning (FL) method called FedGroup, which adopts the FedAvg method, but it updates the learning of the central model based on the learning changes brought by each group of IoT devices. Our experimental results showed that FedGroup achieved same or better anomaly detection accuracy compared to other federated and non-federated learning methods. Furthermore, we showed how ensemble learning may be used to connect many contributing models for superior average prediction performance. FedGroup also improve the detection of attack type detection and attack type detail detection. By comparing our new models with baseline models, our models performed better showing an accuracy of 99.64% accuracy with 0.02% FPR on attack type detection and 99.89% accuracy on attack type detail detection.

Keywords: Smart Home Environment, Cyber Attack, Anomaly Detection, Federated Learning, Internet of Things (IoT), Machine Learning

1 Introduction

Internet of Things (IoT) bridges the gap between the virtual and real worlds by gathering, analysing, and measuring data to forecast and automate business activities [1]. Additionally, it enables continuous and automatic data input, simplifying people's lives and improving their quality of life [2]. By the year 2025, it is anticipated that 150 countries worldwide will collectively reach 478.2 million smart homes [3]. What can be expected in the long term is that the digital economy will continue to grow as the result of millions of daily online interactions between individuals, businesses, gadgets, and data continuing to transform the economy [1].

Despite smart home security systems making it easy to safeguard houses from burglary, damage, and accidents, they also run the danger of compromising the security of personal data [4]. From the users' perspective, trust is an essential factor influencing the attitudes towards being willing to accept smart homes and having a further intention to use them [5]. Smart homes are vulnerable to various attacks. Two main reasons identified for this vulnerability are network security vulnerabilities and IoT devices with limited security features. These two flaws compound the problem, making smart homes particularly fragile and exposed to different types of attacks [4]. Cybercrime expenses are expected to rise by 15% yearly over the next five years. In 2015, these expenses totalled USD 3 trillion, and they are projected to reach USD 10.5 trillion annually by 2025 [6]. This alarming trend underscores the urgent need for increased cybersecurity measures and awareness..

As a result, it is essential to uphold the highest standards of security and privacy when these IoT devices are being utilised in smart homes. There have been a lot of studies done to find ways on anomaly detection to identify abnormal behaviours and unexpected anomalies. These methods place a strong emphasis on traditional machine learning models and deep learning models, which provide additional difficulties in data privacy [7]. To ensure the security and privacy of smart homes, employing the highest security and privacy standards is essential. Researchers have investigated anomaly detection broadly to identify abnormal behaviours and unexpected anomalies to achieve such standards. Generally, a strong emphasis was on traditional machine learning models and deep learning models, which provide additional difficulties in data privacy. To address these challenges, researchers have turned to federated learning, which uses a central model to aggregate updates from local models. This approach ensures high-security and lightweight communication, making it an effective solution for IoT devices in smart homes [8, 9]

Moreover, previous research on anomaly detection has largely overlooked attack-type identification using federated learning. Identifying unusual patterns is critical in various disciplines, including decision-making, business intelligence, and so forth [10]. For instance, monitoring unusual credit card transactions to avoid fraud occurs when thieves get a physical credit card or account login details [11]. To effectively combat cyber attacks, detecting malicious behaviour and categorising the type of attack are crucial. This can be achieved using a multi-class categorisation procedure that describes the attack and helps identify its source.

In the prior study [12], we introduced a model called FedGroup to address the issue of anomaly detection. FedGroup is based on the Federated Learning model principle

but with an additional group master in the central server to compute learning updates based on parameters from a collection of IoT devices. FedGroup's advantages have demonstrated that it is a fast-running, highly secure, and fairness-solving algorithm with lightweight communication overhead. Additionally, we conclude that FL-based learning models performed on par with or better than standard ML models after comparing the assessment results of the FedGroup against federated learning and non-federated learning models. Finally, by integrating Ensemble Learning as the local model with the FedGroup model on the central model to train, the best accuracy of attack detection of 99.91% was achieved on the UNSW IoT dataset.

Considering the target variable attack types of the case study data for anomaly detection using FL-based models have not been addressed in any work, this extending study focused on attack type detection and attack type detection details which can be seen as additions to the original subject of attack detection. The main contributions of this study are:

1. Fixing the issues with Attack Detection: Whether it is an attack, Attack Type Detection: What is the attack type? and Attack Type Detection Details: to foresee "direct or reflection", "type of attack", "rate of attack", and "layer of attack" respectively.
2. Performance evaluation of Traditional ML, Federated Learning (FedAvg) and FedGroup algorithms to detect anomalies in the smart home on the use case dataset.

This paper is divided into several sections, which are summarized below. Section 2 provides a brief review of related research and identifies gaps in the literature. In Section 3, we describe our use case research data and present new models. Sections 4 and 5 present the evaluation results and limitations, respectively. Finally, the conclusion summarizes the main findings of the study."

2 Literature Review

2.1 Traditional machine learning

Researchers have extensively utilised traditional machine learning and deep learning techniques to detect potential cyber-attacks and safeguard internal networks. These approaches involve training algorithms on historical network traffic data to identify patterns and anomalies that may indicate an ongoing attack. According to Tsai et al. [7], in the period between 2000 and 2007, there were 55 research papers related to intrusion detection, and the majority of these papers focused on the use of single classifiers, such as K-Nearest Neighbors (KNN) and logistic regression. These classifiers were the subject of the most extensive literature during this period. Ensemble classifiers, on the other hand, are seldom included in the research, despite the fact that they may surpass single classifiers in terms of classification accuracy.

In addition, the identification and exploitation of various vulnerabilities have become a crucial area of focus for researchers, given that cyber attackers are constantly seeking to exploit weaknesses in systems and networks. As a result, the different types of cyber-attacks have become a vital topic for investigation among researchers. Bhardwaj et al. [13] compared various machine learning algorithms such as CNN, SVM, and

KNN with very high accuracy on four cyber attacks: IDS, SQLI, XSS and Phishing detection.

As the amount of data collected from real-world applications continues to grow, the data's size and dimensionality have also increased. This increase in dimensionality can result in a scarcity of data items, making it more challenging to detect abnormalities effectively. Additionally, traditional approaches to anomaly detection have proven to be ineffective and inefficient in such high-dimensional data [14]. According to Robles and Kim [15], the cloud-centric and comprehensive IoT-based architecture for smart home settings needs significant data storage and processing infrastructure, which is far from efficient. As a result, they emphasised that the new techniques should address the issue of enormous data management in the cloud. Furthermore, studying various approaches to assure security is the next stage, as cloud-based solutions provide a significant danger of disclosing personal information and data, which is one of the most pressing difficulties.

2.2 Federated learning (FedAvg)

While past research focused on centralised anomaly detection in which the central model collects data from local models, decentralised models generally display the benefits of easy computation and lightweight communication [9]. Federated Learning was first mentioned by Google to improve the efficiency and security of users interacting with mobile devices [16]. A central model receives the parameter updates and is followed by averaging updates at the server which is the reason why name it as FedAvg. The advantages of collaborative learning, low communication cost and decoupling cloud storage had overcome the challenges [16] [8, 9].

Rahman et al. [17] examined the IoT anomaly detection methods on centralised, on-device and federated learning in the study. Based on the experiment results of the three methods, they evaluate the efficiency of federated learning, reaching the equivalent accuracy with global insights. What's more, the study insists that aggregate FL outperform the self-learning approach participant devices because it can take advantage of the knowledge from others. In addition, centralised ML models have severe downsides in that they are costly, computationally challenging, and cannot keep up with the fast-growing pace of IoT-connected devices.

Various attack types have been proposed in the literature, such as data poisoning, model poisoning, backdoor attacks, inference attacks, and membership inference attacks, as mentioned by Zhao et al. [18]. Despite this, Ghimire [19] notes that Federated Learning (FL) has the potential to be an efficient solution to tackle different types of attacks and enhance cybersecurity. However, most research in this area primarily focuses on model accuracy and neglects other performance metrics.

In a study by Zhang et al. [20], a new algorithm called FedDetect was proposed, which utilizes an adaptive optimizer and cross-round learning rate scheduler instead of the original FedAvg algorithm. The authors demonstrated the efficacy of FL in detecting a wide range of attack types, and the results were comparable to the upper-bound performance achieved in centralised training. Additionally, numerous researchers have addressed the issue of attack detection and prevention in FL, proposing solutions

such as differential privacy, encryption techniques, secure aggregation, and anomaly detection methods [18, 21].

2.3 FedGroup

Our previous study [12] proposed a new algorithm called FedGroup to address the unfairness problem of FedAvg. Previous research has overlooked the fact that local models have varying functionality and structures. In smart home environments, FedAvg was unable to consider the similarity of network traffic flow data patterns among device types within the same category. When subjected to comparable attacks, the same type of IoT device may have similar vulnerability architectures. Furthermore, an unfair model distribution may result in disproportionate performance, as the aggregate accuracy may be high, but individual accuracy is unknown [22].

To address these issues, IoT devices within the same group should use identical parameters to detect anomalies. Instead of computing the average learning of each device, FedGroup adjusts the central model's learning based on the learning changes that emerge from each group of IoT devices. In our empirical study using a real-world IoT dataset, we demonstrated that FedGroup has comparable or superior anomaly detection accuracy to both FL and non-FL methods. Additionally, FedGroup is more secure and performs well since all IoT data is used locally to train and update the models.

2.4 Ensemble Learning

In the analysis of J. Vanerio [23], Ensemble Learning (EL) achieve good results in anomaly detection because a Super Learner was developed with different first-level learners and choose logistic regression as the solution of binary classification evaluation on two scenarios. EL integrates various learning models to improve prediction outcomes. In addition, an ensemble of models is more robust in the face of ambiguous training data.

Based on the study of Xu et al. [10], they explored common anomaly detection challenges associated with high-dimensional and mixed-type data. Firstly, they classified detection techniques into three groups, namely, neighbour-based, subspace-based, and ensemble-based approaches. They ran extensive tests on publicly available datasets to evaluate common and popular anomaly detection methods. When the variety of the subspaces or base learners is high, the subspace-based and ensemble-based approaches perform pretty well. It is yet unclear how to select the appropriate subspaces or base learners, as well as their amounts and combining tactics, for these types of anomaly detection algorithms. However, because of the equidistant properties, typical distance measures in neighbour-based algorithms (for example, KNN) cannot perform effectively for high-dimensional data.

According to Tsai et al. [7], the concept of collaboration among multiple classifiers, instead of competition, is a promising approach for designing a complex model by combining hybrid and ensemble learning for anomaly detection. This idea inspires the combination of the benefits of ensemble learning and the advantages of the federated learning model for anomaly detection.

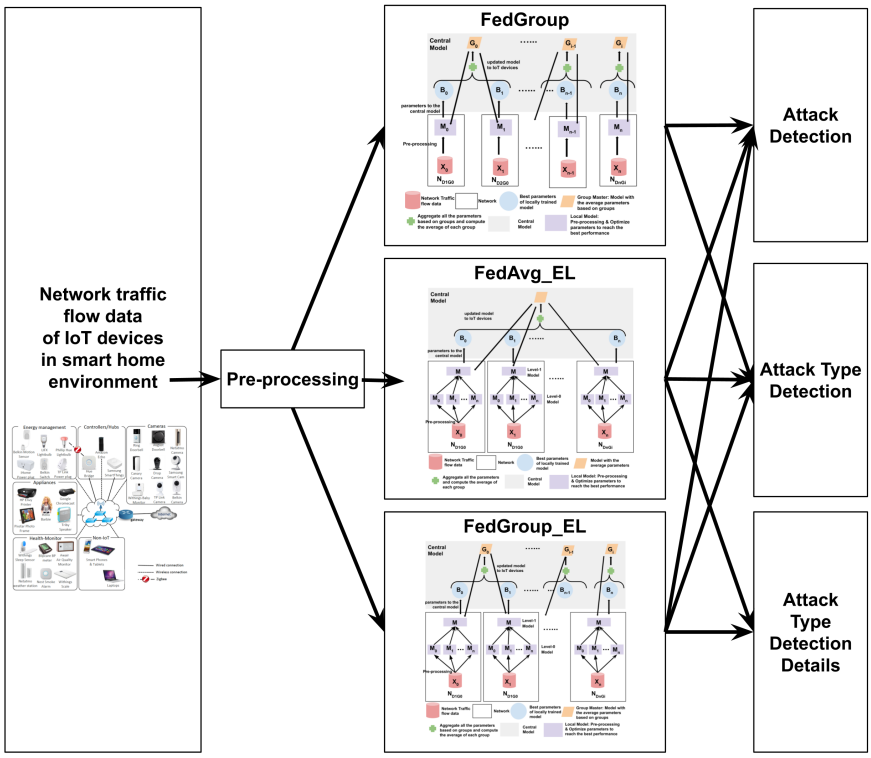

Fig. 1 Outline of the study

2.5 Summary

Many researchers have shown that federated learning (FL) outperforms classical machine learning in terms of security and privacy, collaborative learning, and low request on cloud storage, making it a promising solution for the issue of anomaly detection. However, research towards using FL to recognize attack types has not received sufficient attention. No existing work has addressed the problem of detecting attack types in intrusion detection using FL-related models for the target variable of the case study data. This study aims to investigate the problem of attack type detection using FL by considering not only accuracy as a single performance metric but also false positive rate (FPR) as an important metric. Furthermore, the bias of distributed models is typically averaged to produce the final global model, leading to potential unfairness. FedGroup takes into account the functionality and structure of various models to address these challenges. The FedGroup algorithm is used to solve these problems and compare the results with other models.

3 Methodology

The research plan for this study is based on the study's outline illustrated in Fig. 1. The study has three primary objectives: the first objective is to develop an anomaly

IoT devices			
Device	MAC Addresses	IoT devices	Category
0	00:16:6c:ab:6b:88	Samsung Smart Cam	Camera
1	00:17:88:2b:9a:25	Phillip Hue Lightbulb	Energy management
2	44:65:0d:56:cc:d3	Amazon Echo	Contollers/Hubs
3	50:c7:bf:00:56:39	TP-Link Plug	Energy management
4	70:ee:50:18:34:43	Netatmo Camera	Camera
5	74:c6:3b:29:d7:1d	iHome PowerPlug	Energy management
6	d0:73:d5:01:83:08	LiFX Bulb	Energy management
7	ec:1a:59:79:f4:89	Belkin Switch	Energy management
8	ec:1a:59:83:28:11	Belkin Motion Sensor	Energy management
9	F4:F5:D8:8F:0A:3C	Chromcast Ultra	Appliances

Table 1 Ten IoT devices

detection model to identify any attack attempts. The second objective is to determine the type of attack (Attack Type Detection), followed by investigating the Attack Type Detection Details. While the first objective was the main focus of our previous study, this extended study concentrates on solving the second and third objectives. The first section of the study, "Research Data," presents the network traffic flow data and the attack data. The "Research Method" section details the specifics of the model design. Lastly, the "Experiment and Analysis" section outlines the planning and assessment procedure.

3.1 Research Data

This project is based on the dataset from the UNSW IoT analytics team [24] [25] [26] [27] that centres on the selected 10 IoT devices that are the wireless connection to the Internet and contain both the benign and attack traffic datasets in four categories of Energy management, Camera, Appliances, and Controllers/Hubs listed in the table 1.

3.1.1 Network Traffic Flow Data

Every minute, the 10 IoT devices' network traffic flow data is gathered, marked with activity, and recorded to ten different excel network traffic flow data files. The files include "Timestamp," and a sizable number of pattern characteristics, including "From###Port###Byte,", "To###Port###Byte.", "From###Port###Packet", and "To###Port###Packet". The contents after "From" and "To" are "InternetTcp", "InternetUdp", "LocalTcp", "LocalUdp", and so forth, whereas the contents after "Port" are port numbers. We choose to anticipate assaults by using both since the packet and byte are not closely related because the size of the packets in this dataset varied. According to the statistics on network traffic flow, it is uncertain which network flow is en route to or emanating from which IoT devices. The reasons are different IoT devices using the same port number and the same device using different port numbers at the same time. For example, both the Amazon Echo and the LIFX lightbulb use DNS (port number 53) and NTP (port number 123). Amazon Echo uses HTTP (port number 80), HTTPS (port number 443),

45 Attack Types					
No.	Attack types	Attack categories	Types of attack	Rates of attack	The layer of attack
0	ArpSpoof1L2D	Direct	ArpSpoof	1	Local to Device
1	ArpSpoof10L2D	Direct	ArpSpoof	10	Local to Device
2	ArpSpoof10L2D	Direct	ArpSpoof	100	Local to Device
3	TcpSynDevice1L2D	Direct	TcpSynDevice	1	Local to Device
4	TcpSynDevice10L2D	Direct	TcpSynDevice	10	Local to Device
5	TcpSynDevice100L2D	Direct	TcpSynDevice	100	Local to Device
6	PingOfDeath1L2D	Direct	PingOfDeath	1	Local to Device
7	PingOfDeath10L2D	Direct	PingOfDeath	10	Local to Device
8	PingOfDeath100L2D	Direct	PingOfDeath	100	Local to Device
9	UdpDevice1L2D	Direct	UdpDevice	1	Local to Device
10	UdpDevice10L2D	Direct	UdpDevice	10	Local to Device
11	UdpDevice10L2D	Direct	UdpDevice	100	Local to Device
12	TcpSynReflection1L2D2L	Reflection	TcpSynReflection	1	Local to Device to Local
13	TcpSynReflection10L2D2L	Reflection	TcpSynReflection	10	Local to Device to Local
14	TcpSynReflection100L2D2L	Reflection	TcpSynReflection	100	Local to Device to Local
15	Snmp1L2D2W	Reflection	Snmp	1	Local to Device to Internet
16	Snmp10L2D2W	Reflection	Snmp	10	Local to Device to Internet
17	Snmp100L2D2W	Reflection	Snmp	100	Local to Device to Internet
18	TcpSynReflection1W2D2W	Reflection	TcpSynReflection	1	Internet to Device to Internet
19	TcpSynReflection10W2D2W	Reflection	TcpSynReflection	10	Internet to Device to Internet
20	TcpSynReflection100W2D2W	Reflection	TcpSynReflection	100	Internet to Device to Internet
21	Snmp1W2D2W	Reflection	Snmp	1	Internet to Device to Internet
22	Snmp10W2D2W	Reflection	Snmp	10	Internet to Device to Internet
23	Snmp1002D2W	Reflection	Snmp	100	Internet to Device to Internet
24	UdpDevice1W2D	Direct	UdpDevice	1	Internet to Device
25	UdpDevice10W2D	Direct	UdpDevice	10	Internet to Device
26	UdpDevice100W2D	Direct	UdpDevice	100	Internet to Device
27	TcpSynDevice1W2D	Direct	TcpSynDevice	1	Internet to Device
28	TcpSynDevice10W2D	Direct	TcpSynDevice	10	Internet to Device
29	TcpSynDevice100W2D	Direct	TcpSynDevice	100	Internet to Device
30	Ssdp1W2D2W	Reflection	Ssdp	1	Internet to Device to Internet
31	Ssdp10W2D2W	Reflection	Ssdp	10	Internet to Device to Internet
32	Ssdp100W2D2W	Reflection	Ssdp	100	Internet to Device to Internet
33	Smurf1L2D2L	Reflection	Smurf	1	Local to Device to Local
34	Smurf10L2D2L	Reflection	Smurf	10	Local to Device to Local
35	Smurf100L2D2L	Reflection	Smurf	100	Local to Device to Local
36	Snmp1L2D2L	Reflection	Snmp	1	Local to Device to Local
37	Snmp10L2D2L	Reflection	Snmp	10	Local to Device to Local
38	Snmp100L2D2L	Reflection	Snmp	100	Local to Device to Local
39	Ssdp1L2D2WL	Reflection	Ssdp	1	Local to Device to Internet
40	Ssdp10L2D2WL	Reflection	Ssdp	10	Local to Device to Internet
41	Ssdp100L2D2WL	Reflection	Ssdp	100	Local to Device to Internet
42	Ssdp1L2D2L	Reflection	Ssdp	1	Local to Device to Local
43	Ssdp10L2D2L	Reflection	Ssdp	10	Local to Device to Local
44	Ssdp100L2D2L	Reflection	Ssdp	100	Local to Device to Local

Table 2 45 Attack Types

and ICMP (port number 0). As a result, we can't derive any information directly from network flow data. In this study, the network traffic flow data is set as the input to forecast if the model will be able to measure the attacks and what attack types they will be.

3.1.2 Attack data

The UNSW IoT analytics team designed a set of attacks comparable to real-world attacks and are particular to a number of 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.

Attack Detection: When determining the normal behaviour or under the attacks, it relies on the rules of "if the flow time within the start time and end time of the attack, then the attack is true.".

Attack Type Detection: There are 45 different kinds of attack types, with each attack lasting for 10 minutes each time with 200 attacks in total (see Table. 2). In Table. 3, the proportion of attack and attack types on the ten IoT devices are listed.

Attack Type Detection Details: The detection details continue to work on "direct or reflection", "type of attack", "rate of attack", and "layer of attack" respectively. Please note that to prevent confusion of 45 attack types and type of attack. The attack types mean 45 different attack types, such as ArpSpoof100L2D, and the types of attack focus on the varieties such as ArpSpoof.

1. **Attack categories**: Reflection and direct are two types of attack.
2. **Types of attack**: ArpSpoof, TcpSynDevice, UdpDevice, and PingofDeath are direct attacks. SNMP, Ssdp, TcpSynReflection, and Smurf are reflective attacks.
3. **Rates of attack**: 100 PPS, 10 PPS, and 1 PPS which PPS means packets per second.
4. **The layer of attack**: L2D, L2D2L, L2D2W, W2D2W, W2D are the five types of layer scenarios which L: Local, 2: to, D: Device, and W: Internet. L2D represents Local to Device.

Set one of the attack conditions of the Samsung smart camera as an example: "1527838552, 1527839153, Localfeatures|Arpfeatures, ArpSpoof100L2D" represents a direct attack named Arpspoof launched with the attack from local to device with the rate of 100 packets per second started at 1527838552 and ended at 1527839153 (time in milliseconds) was influence both the local communication and ARP protocol.

3.2 Research Method

3.2.1 FedAvg

Federated learning accepts the initial model from the central server, training models on decentralized local device servers, and reports the best performance parameters to the central model [16]. The system design Fig. 2: FedAcg Protocol can be seen below that following the ideas of Figure 1: Federated Learning Protocol from Bonawiz [28] "Towards Federated Learning At Scale: System Design".

Federated learning (FL) is a sharing model to train data without centrally storing it:

1. Federated learning allows enormous datasets stored in various distributed servers, reducing data transmission and ensuring data privacy and security.
2. The distributed servers trained global models on local data and compiled the changes as an update sent to the cloud in a less communicative, efficient and secure way.
3. The cloud server renews the global model with the weighted average of parameters, so the approach is also named Federated Average (FedAvg). It allows fault-tolerant and scalable computation.

Proportion of Attack											
IoT Devices No.	0	1	2	3	4	5	6	7	8	9	Total
Non-Attack (%)	99.42	99.32	99.80	99.62	99.71	99.93	99.66	99.66	99.49	99.42	99.60
Attack (%)	0.578	0.679	0.204	0.381	0.294	0.069	0.343	0.341	0.512	0.576	0.403
Proportion of Attack Types											
IoT Devices No.	0	1	2	3	4	5	6	7	8	9	Total
Attack Type 0 (%)	0.014	0.023	0.023	0.014	0.010	0.023	0.023	0.014	0.028	0.023	0.017
Attack Type 1 (%)	0.014	0.023	0.023	0.014	0.014	0.023	0.023	0.014	0.028	0.023	0.018
Attack Type 2 (%)	0.014	0.023	0.023	0.014	0.014	0.023	0.023	0.014	0.028	0.023	0.018
Attack Type 3 (%)	0.014	0.023	0.0	0.014	0.014	0.0	0.0	0.014	0.028	0.023	0.012
Attack Type 4 (%)	0.014	0.023	0.0	0.014	0.014	0.0	0.0	0.014	0.028	0.023	0.012
Attack Type 5 (%)	0.014	0.023	0.0	0.014	0.014	0.0	0.0	0.014	0.028	0.023	0.012
Attack Type 6 (%)	0.014	0.023	0.0	0.014	0.0	0.0	0.023	0.014	0.028	0.0	0.011
Attack Type 7 (%)	0.014	0.023	0.0	0.014	0.0	0.0	0.023	0.014	0.028	0.0	0.011
Attack Type 8 (%)	0.014	0.023	0.0	0.014	0.0	0.0	0.023	0.014	0.028	0.0	0.011
Attack Type 9 (%)	0.014	0.0	0.023	0.0	0.0	0.0	0.023	0.0	0.028	0.0	0.007
Attack Type 10 (%)	0.014	0.0	0.023	0.0	0.0	0.0	0.023	0.0	0.028	0.0	0.007
Attack Type 11 (%)	0.014	0.0	0.023	0.0	0.0	0.0	0.023	0.0	0.028	0.0	0.007
Attack Type 12 (%)	0.014	0.023	0.0	0.014	0.014	0.0	0.0	0.014	0.028	0.023	0.012
Attack Type 13 (%)	0.014	0.023	0.0	0.014	0.014	0.0	0.0	0.014	0.028	0.023	0.012
Attack Type 14 (%)	0.014	0.021	0.0	0.013	0.014	0.0	0.0	0.014	0.028	0.023	0.012
Attack Type 15 (%)	0.014	0.023	0.0	0.014	0.0	0.0	0.023	0.0	0.0	0.0	0.007
Attack Type 16 (%)	0.014	0.023	0.0	0.014	0.0	0.0	0.023	0.0	0.0	0.0	0.007
Attack Type 17 (%)	0.014	0.023	0.0	0.014	0.0	0.0	0.023	0.0	0.0	0.0	0.007
Attack Type 18 (%)	0.010	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.001
Attack Type 19 (%)	0.014	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.002
Attack Type 20 (%)	0.014	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.002
Attack Type 21 (%)	0.014	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.002
Attack Type 22 (%)	0.014	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.002
Attack Type 23 (%)	0.014	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.002
Attack Type 24 (%)	0.028	0.023	0.0	0.028	0.028	0.0	0.0	0.028	0.028	0.025	0.021
Attack Type 25 (%)	0.028	0.023	0.0	0.028	0.028	0.0	0.0	0.028	0.028	0.046	0.023
Attack Type 26 (%)	0.028	0.023	0.0	0.028	0.028	0.0	0.0	0.028	0.028	0.023	0.021
Attack Type 27 (%)	0.014	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.002
Attack Type 28 (%)	0.014	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.002
Attack Type 29 (%)	0.014	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.002
Attack Type 30 (%)	0.028	0.023	0.0	0.028	0.028	0.0	0.0	0.028	0.028	0.023	0.021
Attack Type 31 (%)	0.014	0.021	0.0	0.028	0.028	0.0	0.0	0.028	0.028	0.023	0.019
Attack Type 32 (%)	0.028	0.023	0.0	0.027	0.028	0.0	0.0	0.028	0.028	0.023	0.021
Attack Type 33 (%)	0.014	0.0	0.023	0.0	0.0	0.0	0.023	0.0	0.0	0.0	0.005
Attack Type 34 (%)	0.014	0.0	0.021	0.0	0.0	0.0	0.023	0.0	0.0	0.0	0.005
Attack Type 35 (%)	0.014	0.0	0.023	0.0	0.0	0.0	0.023	0.0	0.0	0.0	0.005
Attack Type 36 (%)	0.0	0.021	0.0	0.0	0.0	0.0	0.0	0.0	0.028	0.023	0.005
Attack Type 37 (%)	0.0	0.023	0.0	0.0	0.0	0.0	0.0	0.0	0.028	0.023	0.005
Attack Type 38 (%)	0.0	0.023	0.0	0.0	0.0	0.0	0.0	0.0	0.028	0.023	0.005
Attack Type 39 (%)	0.0	0.023	0.0	0.0	0.0	0.0	0.0	0.0	0.028	0.023	0.005
Attack Type 40 (%)	0.0	0.023	0.0	0.0	0.0	0.0	0.0	0.0	0.028	0.023	0.005
Attack Type 41 (%)	0.0	0.023	0.0	0.0	0.0	0.0	0.0	0.0	0.028	0.023	0.005
Attack Type 42 (%)	0.0	0.023	0.0	0.0	0.0	0.0	0.0	0.0	0.028	0.025	0.005
Attack Type 43 (%)	0.0	0.023	0.0	0.0	0.0	0.0	0.0	0.0	0.028	0.023	0.005
Attack Type 44 (%)	0.0	0.023	0.0	0.0	0.0	0.0	0.0	0.0	0.028	0.023	0.005
Non-Attack (%)	99.42	99.32	99.80	99.62	99.71	99.93	99.66	99.66	99.49	99.42	99.60

Table 3 Proportion of attack and attack types

3.2.2 FedGroup

Although FedAvg is capable of aggregating all parameters from local servers and selecting the mean as the next round parameter, it fails to handle unfairness effectively. This is because the algorithm does not take into account the fact that smart home devices may not be equally distributed among different groups [29]. Devices in the same categories have comparable functionality and are exposed to the same dangers. The bias in the training approach was caused by the updates of participant parameters that differed from one another and were readily chosen as the average. Since aggregate accuracy is high but individual accuracy is an unknown, unjust distribution of the model might result in disproportionate performance [22].

FedGroup [12] suggests calculating the average of updates based on groups rather than simply choosing a one-shot averaging of all updates. The model includes multiple local models, a central model and several group masters in the central model. Each IoT device collects network traffic data to train a local model (client-server, decentralised model, participant, distributed model) and sends the learning updates to the related group master in the central model (cloud server, cloud) to share their learning. During

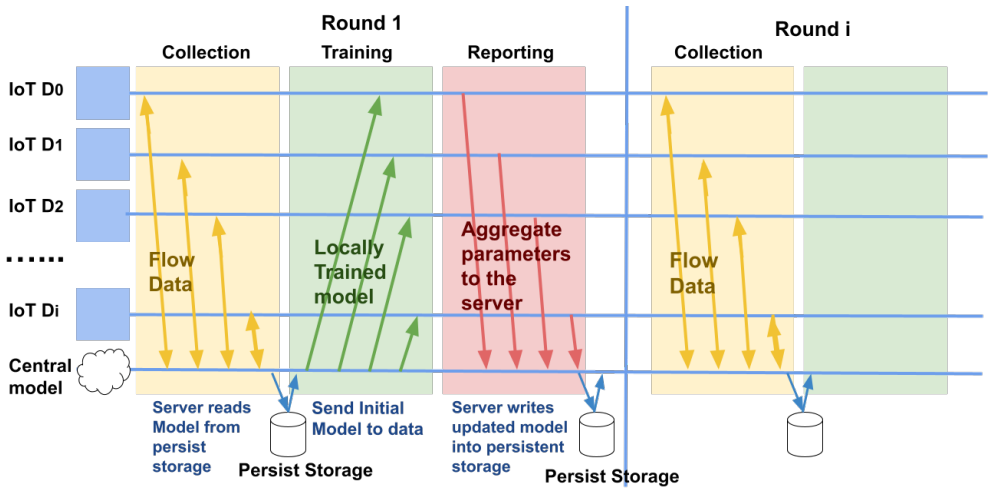

Fig. 2 Federated Learning

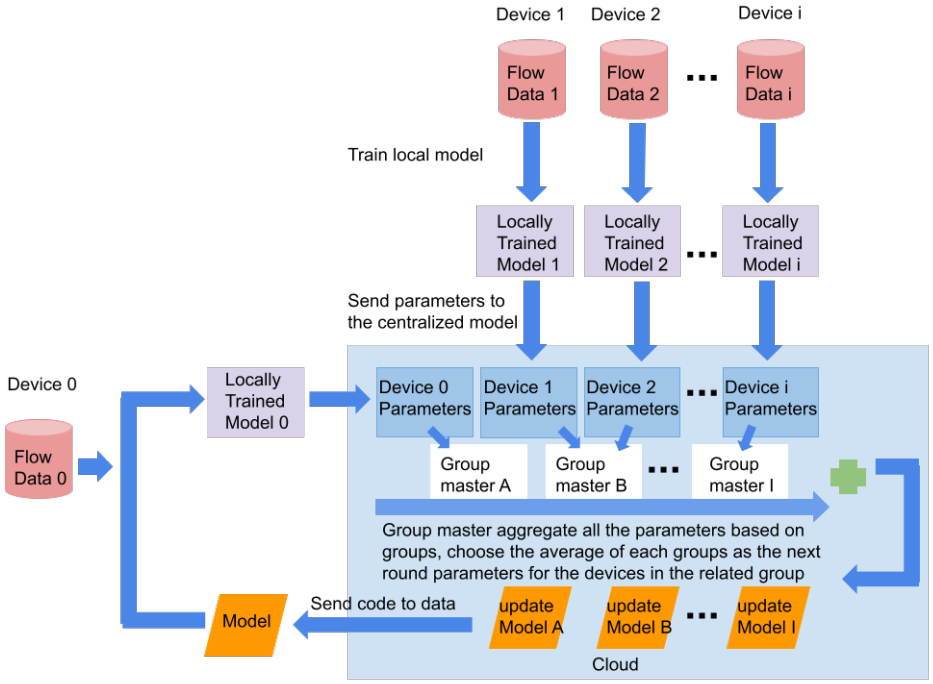

Fig. 3 FedGroup

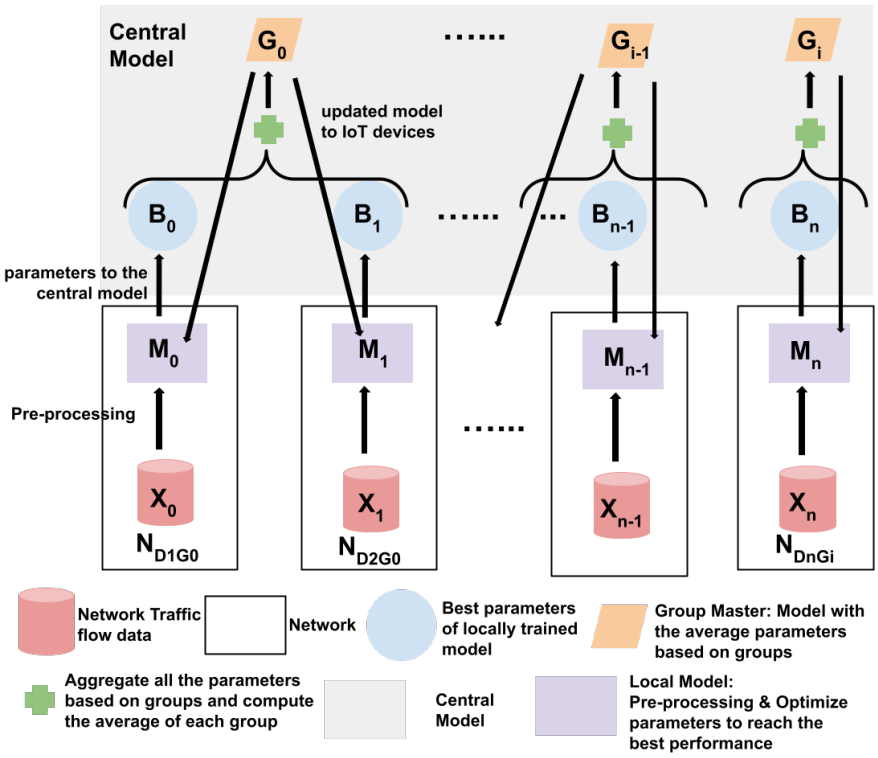

Fig. 4 FedGroup

this process, the data are not shared or transmitted as in traditional machine learning. Each group master aggregates the values of all the learning parameters within the same group using a defined function (e.g., average) to adjust the learning. Then, the group master sends the updated learning to all client servers in the group for their next round of training, which is more efficient for the local model to focus on the information within the same group.

In this study, it is assumed that the IoT devices present in a smart home primarily comprise energy management applications such as plugs or bulbs. Due to the disproportionate number of such devices compared to those in other groups, the parameters of the cloud server will have a bias towards energy management devices. The four IoT devices used in this research are categorized as follows: one device in the Group Controllers/Hubs, one device in the Group Appliances, and two devices in the Group Camera. The remaining six IoT devices belong to Group Energy Management, which comprises a Belkin Motion Sensor, an iHome PowerPlug, a LIFX Bulb, a Philips Hue lightbulb, a TP-Link Plug, and a Belkin Switch.

To ensure the security and privacy of the data, it is kept locally and not transmitted over the internet or communicated with other devices. Additionally, to prevent low accuracy due to bias, the parameters of IoT devices are identified by the group rather than the overall average.

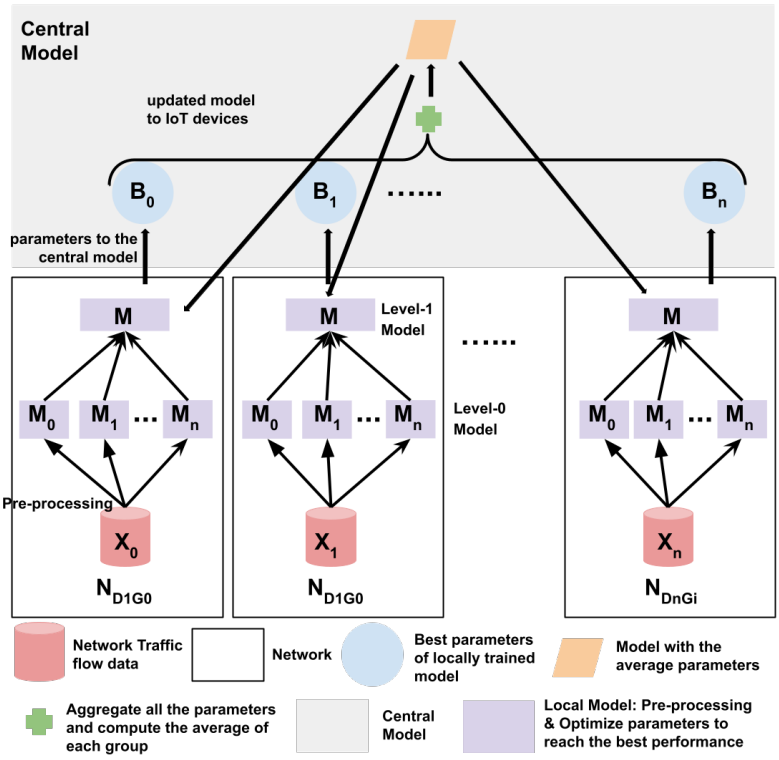

Fig. 5 FedAvg_EL

Definition: 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} where X_n represents the network traffic flow data of the IoT device n, and M_n means the local model of the IoT device n. 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 parameter grids in the local models after the update to the central model. y_n to represent the prediction target, for example, cyber attack types.

3.2.3 FedAvg_EL

FedAvg_EL follow the workflow of FedAvg but replace the local models with ensemble learning. We implement the FedAvg_EL on attack detection and attack type detection following the steps of FedGroup which can be seen in Fig. 5. In previous studies, using ML as the local model is common but not always performed as expected because they are used to solve a specific question or a type of question leading to various inconsistent performances. However, EL joins different contributing models to seek better forecasts

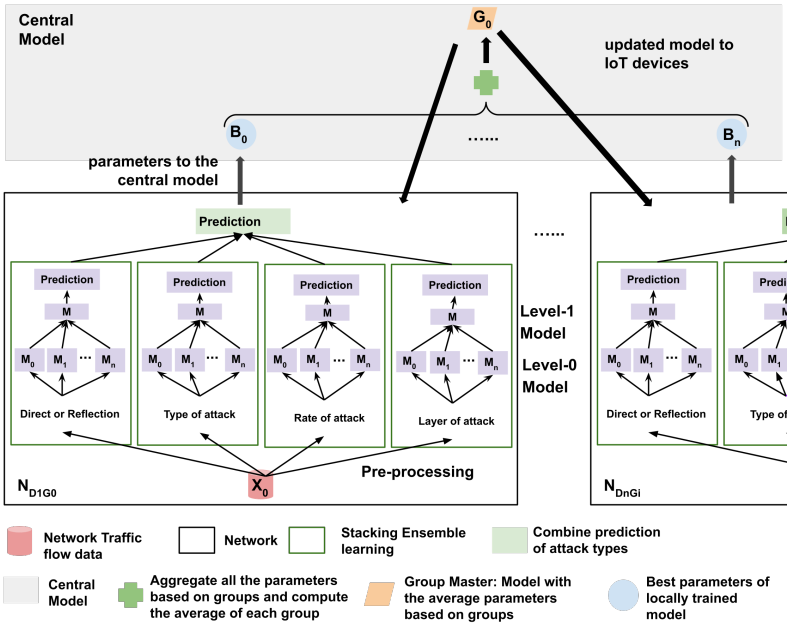

Fig. 6 FedAvg_EL on attack type detection details

and it allows continuous operation properly without interruption when facing one or more of its model failures.

When considering ensemble learning as a local model, there are three types of ensemble learning: Bagging, Stacking and Boosting. However, Bagging Ensemble Learning splits the training dataset into samples in the same model and Boosting Ensemble Learning continuous correct the predictions that are not appropriate for our study. Therefore, employing Stacking Ensemble Learning that combines multiple models and learning the models in parallel with the same data is suitable for our situation. In stacking ensemble learning, there are two levels of the model. The base models, also called Level-0 models, fit the training network traffic data in local devices. Then, using the classification level-1 model logistic regression to combine the predictions from level-0 [30] [31].

As for predicting the attack type details, FedAvg_EL can combine various models in the ensemble learning locally. Fig. 6 illustrates that the model helps customers understand what kind of attack rates it is, what type of attack it is, what layers are suffering attacks, and whether is it a direct attack or reflection attack. Then, these information pieces will help customers take proper action to defend against attacks.

The steps of FedAvg_EL on attack type detection details:

1. Every local model uses the network traffic flow data to train models. The models predict "direct or reflection", "type of attack", "rate of attack" and "layer of attack" in four stacking EL, respectively;

2. The prediction accuracy is the mean of the four aspects. Local models send the best parameters of the model to the central model;
3. The central model secure aggregates all the parameters;
4. The central model sends back the new global model with the average parameters to participants;
5. Local models update the models with the new parameters.

Algorithm 1 FedAvg_EL: Client Side LearningAlgorithm

INPUT: P, E
REQUIRE: X_n, y_n, M
OUTPUT: B and S to $Central\,Server\,Side\,Learning: FedAvg\_EL$
SET: Level 0 models and level 1 model of Stacking EL
/* Fit possible parameters grids and return the best parameters and the best score*/
for $e \in$E **do**
    P in Stacking EL 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 FedAvg_EL: Central Server Side Learning Algorithm

INPUT: M, P, p
OUTPUT: C
/* 1st round: receive the best parameters and best devices from every device, and calculate the mean */
for $n \in$N **do**
    Initial: M
    $Client\,Side\,Learning: FedAvg\_EL$ (P, p)
    Return B and S of each N
end for
Return A and C
/* 2nd round: send mean parameter to client server and return the mean score of model */
for $n \in$N **do**
    $Client\,Side\,Learning: FedAvg\_EL$ (P, A)
    Return B and S of each N
end for
Return A and C

3.2.4 FedGroup_EL

FedGroup_EL combines FedGroup and EL: using ensemble learning as the local model and FedGroup as the central model with the group master for group updates. The advantages of learning from a mixture of models from ensemble learning, keeping the security and privacy of data, and the fairness of training procedure from FedGroup are involved in the new model. Most importantly, the fault-tolerant can be seen as the biggest advantage of FedGroup_EL. FedGroup is available to tolerate adversarial attacks and resolve faults since it is deployed on multiple edge devices [32]. Besides, the structure of ensemble learning allows it to take benefits from many models without worrying about causing system failures. We implement the FedGroup_EL on attack detection and attack type detection following the steps of FedGroup in Fig. 7.

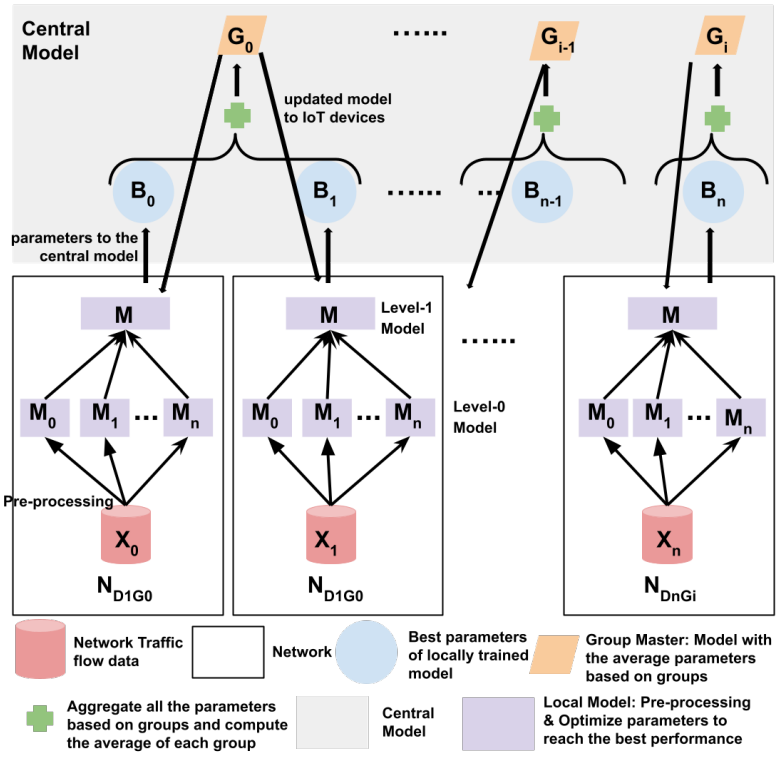

Fig. 7 FedGroup_EL

Due to the fact that the 45 attack types can be excavated to the four perspectives, which are meaningful and worth learning to predict the attack type detection details. Therefore, the local model is the aggregate of four stacking EL (see Fig. 8).

The steps of FedGroup_EL on attack type detection details:

1. Every local model uses the network traffic flow data to train. The models predict "direct or reflection", "type of attack", "rate of attack" and "layer of attack" in four stacking EL, respectively;
2. The prediction accuracy is the mean of the four aspects. Local models send the best parameters of the model to the central model;
3. Group master in the central model secure aggregate the parameters based on groups;
4. The central model sends back the new global model with the average parameters to participants in the related group;
5. Local models update the models with the new parameters.

Algorithm 3 FedGroup_EL: Group Master Algorithm

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

Algorithm 4 FedGroup_EL: Client Side Learning Algorithm

INPUT: P, E
REQUIRE: X_n, y_n, M
OUTPUT: B and S to $Central\ Server\ Side\ Learning:\ FedAvg\_EL$ with the related Group master
SET: Level 0 models and level 1 model of Stacking EL
/* Fit possible parameters grids and return the best parameters and the best score*/
for $e \in$E **do**
    Fit P in Stacking EL 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 5 FedGroup_EL: 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 $g \in$G **do**
    for $n \in$N **do**
        Initial: M
        $Client\ Side\ Learning:\ FedGroup\_EL$ (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 $g \in$G **do**
    for $n \in$N **do**
        $Client\ Side\ Learning:\ FedGroup\_EL$ (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

In data pre-processing of network traffic flow data for IoT devices, remove "NoOfFlow" since it counts flows, which are closely associated with all the other variables. There are 253 qualities concerning bytes of port number and packages of the port number in all because various devices use the same port number while the same device uses different port numbers. Different port numbers have not been utilised to capture the network behaviour of one device per millisecond. In other words, NaN data indicates that the matching port number has no network activity. We assign a specific value of 0 to the missing data and replace it with the most probable value and the global constant. It denotes no network activity at that moment with zero packet-level and zero byte-level network traffic flow data.

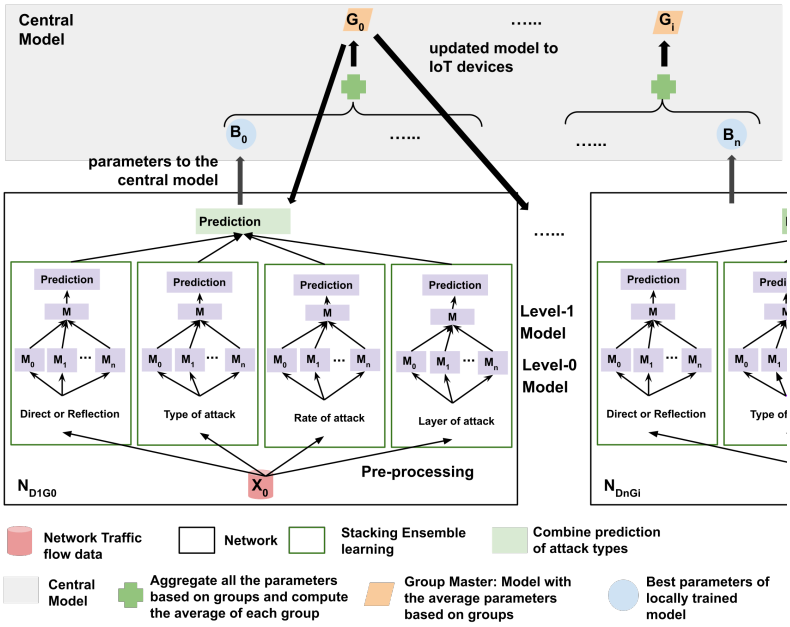

Fig. 8 FedGroup_EL on attack type detection details

Ensemble Learning - adjust level 0 models				
Level-0	KNN, DT, NB	KNN, DT, SVM	KNN, DT, NB, SVM	KNN, DT, NB, SVM, RF
Level-1	Logistic Regression			
Samsung Smart Cam Accuracy	0.998810	0.998756	0.998774	0.998721

Table 4 Ensemble Learning - adjust level 0 models

The given dataset is imbalanced, with the proportion of observations biased to particular labels. To avoid overfitting and ensure the same proportion of observations with a provided label, the dataset was divided into 80% training and 20% testing using StratifiedShuffleSplit. Stratified 5-Fold Cross-Validation randomly splits the full set of training data into five folds. In each of the 'five' iterations, fits the model to four of the folds, and then validate the model using the fifth fold [33]. To calculate accuracy using an F1 score with a weighted average, evaluate the 20% testing data.

Speaking of Stacking Ensemble Learning in attack detection and attack type detection, we utilise KNN and Decision Tree on the Level-0 model and Logistic Regression on the Level-1 model. As for attack type detection details, we use Device Samsung Smart Cam to adjust four patterns of models to adjust the level-0 models in the initial ensemble learning. From the results in table 4, we decide to use KNN, Decision Tree, Naive Bayes in level-0.

$$FPR = \frac{FP}{TN + FP}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

The accuracy classification score is a crucial metric for evaluating multilabel classification performance, requiring an exact match to actual data [34]. Another important metric is the False Positive Rate (FPR), which measures the ratio of negative events incorrectly classified as positive (False Positives) to the total number of ground truth negatives (N = TN + FP) [35] [36]. In our case study, we utilise both accuracy and FPR to evaluate the models. Accuracy measures the correct predictions of abnormal and normal behaviours, while FPR measures the likelihood of a cyber attack being incorrectly detected as normal behaviour.

4 Results

Algorithms		**Attack Detection**			**Attack Type Detection**		
Local Model	Central Model	Accuracy	Running Times (seconds)	FPR	Accuracy	Running Times (seconds)	FPR
Decision Tree	Traditional ML	99.84%	8524	10.04%	88.41%	35	0.27%
Decision Tree	FedAvg	99.85%	154	9.57%	93.90%	11	0.35%
Decision Tree	FedGroup	99.87%	154	7.70%	94.86%	10	0.27%
Logistic regression	Traditional ML	99.76%	21376	24.48%	39.73%	5443	1.37%
Logistic regression	FedAvg	99.77%	2912	20.28%	49.55%	183	2.76%
Logistic regression	FedGroup	99.77%	2999	20.18%	52.01%	199	2.63%
Ensemble learning	Traditional ML	99.85%	33940	9.60%	97.98%	1590	0.04%
Ensemble learning	FedAvg	**99.91%**	2390	**9.03%**	**99.50%**	371	**0.03%**
Ensemble learning	FedGroup	**99.91%**	2143	**9.43%**	**99.64%**	341	**0.02%**

Algorithms		**Attack Type Detection Details**		
Local Model	Central Model	Accuracy	Running Times (seconds)	FPR
Ensemble learning	FedAvg	**99.89%**	4448	**4.79%**
Ensemble learning	FedGroup	**99.89%**	4431	**5.23%**

Table 5 The accuracy of FedGroup, FedAvg and Traditional ML using different models

This study has analysed anomaly detection on three questions: 1. Attack Detection: Can we detect if there is an attack happening or not? 2. Attack Type Detection: If yes, can we identify its attack type? 3. Attack Type Detection Details: Can we further correctly predict the details of the attack?

Table.5 compares the performance of our schema. The first section displays the outcomes of a central model using Traditional ML, FedAvg, and FedGroup, a local model using Decision Tree, Logistic Regression, and Ensemble Learning for attack detection and attack type identification. The second section demonstrates the outcomes of using EL as the local model on both FedAvg and FedGroup to attack type detection details on "direct or reflection", "type of attack", "rate of attack", and "layer of attack".

To begin with, the analysis of anomaly detection focused on three aspects: 1. Detecting whether an attack is happening, 2. Identifying the type of attack if detected, and 3. Providing details of the attack type. The top-performing model achieved an accuracy of 99.91% in detecting attacks using a Federated Learning Based central model and Ensemble learning as the local model for training. In terms of attack type detection, the FedGroup model utilising EL as the local model achieved the highest accuracy of 99.64%. For attack type detection details, both FedAvg_EL and FedGroup_EL models achieved an overall accuracy of 99.89%, providing specific features of attack types to customers.

Secondly, FL-based learning models outperform conventional ML models, sometimes even better. The FL-based model runs faster than the traditional ML model, which requires an $O(n)$ for the client slide model and an $O(n^2)$ for the central server. Besides, If we focus on the differences in FPR that are larger than 1%, then the FPRs of the FL-based are less than the FPRs of the Traditional ML model. FL exploits the benefits of local training data to shorten the running time as a result of lightweight communication and a decentralised learning model. Additionally, data security is ensured when the raw data is not sent, communicated, or shared with other IoT devices or the Internet.

Besides, FedGroup performs equal to or better performance than FedAvg. If we focus on the differences of FPRs that are larger than 1%, then the FPRs of FedGroup are less than the FPRs of FedAvg. It is beneficial for FedGroup to offer parameters of IoT devices within the same group when the central model learns attack kinds from the same category of IoT devices.

Lastly, we developed the FedAvg_EL and FedGroup_EL and proved that employing EL as a local training model outperforms the traditional machine learning model. EL can merge several models even if the individuals are weak and show great tolerance for various models. Based on the results, FedAvg_EL and FedGroup_EL achieved the highest performance among the three questions.

The complete details about the experimental results can be found in the project repository [1]. This includes the results of attack detection with traditional ML and proposed federated learning models, paramater selection and hyper-parameter tuning, and the accuracy of each IoT device with FedAvg, FedGroup, FedAvg_EL, and FedGroup_EL models. Furthermore, the datasets, implementation of the models and detailed experimental results of the work presented in this paper are available in the project repository. This should useful for experiments reproducibility and models extension and comparison.

5 Discussion

This study expanded on our previous work on attack detection by investigating attack types and their details, providing valuable insights. Specifically, our focus was on examining the impact of bias in FedAvg and FedGroup models, and our findings are in line with those of Mohri and Li (2020), who argue that uniform distribution may not always be the most suitable objective distribution. Given the significance of

[1] https://github.com/BasemSuleiman/2023_Anomaly_Detection_IoT

addressing bias in training data disclosure, it is essential to bridge this research gap by incorporating group-based update aggregation.

The study has a few limitations. Firstly, real-time detection was not considered, and the model was built using all the available data with only two communication rounds. This could be extended to multiple iterations to improve accuracy. Secondly, due to computational constraints, only a subset of hyperparameters was considered, which may limit the ability to fine-tune the models.

Additionally, our study is limited by its implementation in only one smart home environment. In the future, IoT environments will consist of multiple smart homes, smart cities, smart transportation systems, and so on, with thousands or even billions of different types of attacks occurring at any time and place. For example, voice recognition sensors in smart homes can perform various functions, such as playing music, answering trivia questions, and controlling the TV or lights. By studying the parameters of voice recognition devices, the central model can identify the types of attacks that make the system vulnerable and improve the security of all voice recognition devices in the city.

Further studies should work on multiple smart home environments and update the IoT devices to group into account. Because we separate the IoT devices focus on functionality such as cameras, appliances, etc. To improve outcomes, the FedGroup will learn more specifics about assaults if various groups are based on numerous attributes. The smart door product, for instance, has a variety of functions to open the door, including app control, fingerprint recognition, password entry, intelligent card scanning, and key unlock. Based on its features, the product may be divided into a number of categories. The central model will identify the precise component being attacked if the smart door is attacked.

6 Conclusion

Addressing the issue of anomaly detection in IoT Anomaly detection in the smart home environment, we introduce a new method called FedGroup and two new frameworks using EL as a locally trained model called FedAvg_EL and FedGroup_EL, for which we present the detailed algorithms.

The study finds that:

1. FL-based algorithms perform equal or better performance than traditional machine learning: FedAvg reaches 99.91% on attack detection and 99.50% on attack type detection. FedGroup gets 99.91% on attack detection and 99.64% in attack type detection.
2. The analysis of FedGroup presents the fact that it slightly improves the performance of FedAvg deals with the concern of fairness of training procedure.
3. FedAvg_EL and FedGroup_EL model helps draw insight to help combine the four perspectives such as “direct or reflection”, “type of attack”, “rate of attack”, and “layer of attack” of attack types detection with the accuracy of 99.89%. Ensemble Learning brings the benefits of fault-tolerance which outperform the traditional machine learning model.

To summarize, this study demonstrates that FL-based models can effectively address the security and privacy challenges of decentralized local servers while achieving high accuracy. Additionally, FedGroup is proposed as a solution to address fairness issues in FL by aggregating updates based on IoT device categories. Moreover, the study investigates the use of ensemble learning to improve the accuracy of attack type detection, specifically for direct or reflection attack, type of attack, rate of attack, and the affected layers. As a result, two new models, FedAvg_EL and FedGroup_EL, are proposed.

While our findings have shown how different models compare, further empirical research on continuous real-time learning and other strategies for federated learning fairness must be done in order to evaluate and improve our conclusions. Other options for future study include extending the model to other frameworks other from anomaly detection, determining the system cost, and examining how wireless network link instability impacts model updating.

Declarations

Ethical Approval

Not Applicable

Availability of supporting data

The datasets, implementation of the models and detailed experimental results of the work presented in this paper are available in the following project repository: https://github.com/BasemSuleiman/2023_Anomaly_Detection_IoT.

Competing interests

Not Applicable

Funding

Not Applicable

Authors' contributions

Basem Suleiman has led the conceptual design of the study including identifying the problem, the conceptual design of the proposed approach, the design of the evaluation and experiments. Yixuan Zhang has led the work in terms of investigating and implementing the planned work with the detailed supervision and guide by Basem Suleiman. Yixuan has also contributed to the conceptual design of the approach and conducted the experiments. She also led the writting of the paper with on-going and detailed feedback from Basem Suleiman and Muhammad Johan Alibasa. Muhammad Johan Alibasa has reviewed the research work, editted the paper and provided feedback to improve the technical aspects of the paper. Farnaz Farid has reviewed and editted the paper and provided feedback.

Acknowledgments

Not Applicable

References

[1] Deloitte: What is digital economy? — unicorns, transformation and the internet of things: Deloitte malta. (2021)

[2] Sandro, N.: Internet of things (iot): Opportunities, issues and challenges towards a smart and sustainable future. Journal of Cleaner Production (2020) https://doi.org/10.1016/j.jclepro.2020.122877

[3] Lasquety-Reyes, J.: Number of smart homes forecast in the world from 2017 to 2025. (2021)

[4] Ali, M.-H.: Smart home security: Security and vulnerabilities. (2021)

[5] Shuhaiber, A., Mashal, B.: Understanding users' acceptance of smart homes. Technology in Society (2019) https://doi.org/10.1016/j.techsoc.2019.01.003

[6] Morgan, S.: Cybercrime to cost the world $10.5 trillion annually by 2025. Cybercrime Magazine (2020)

[7] Tsai, C.-F., Hsu, Y.-F., Lin, C.-Y., Lin, W.-Y.: Intrusion detection by machine learning: A review., 7 (2009)

[8] McMahan, H.-B., Ramage, D.: Federated learning: Collaborative machine learning without centralized training data. (2017)

[9] McMahan, H.-B., Moore, E., Ramage, D., Hampson, S.: Communication-efficient learning of deep networks from decentralized data., 10 (2017)

[10] Xu, X., Liu, H., Yao, M.: Recent progress of anomaly detection. Hindawi (2019)

[11] n.d.: Credit card fraud detection: Everything you need to know. (2023)

[12] Zhang, Y., Suleiman, B., Alibasa, M.-J.: Fedgroup: A federated learning approach for anomaly detection in iot environments. (2022)

[13] Bhardwaj, A., Chandok, S.-S., Bagnawar, A., Mishra, S., Uplaonkar, D.: Detection of cyber attacks: Xss, sqli, phishing attacks and detecting intrusion using machine learning algorithms. 2022 IEEE Global Conference on Computing, Power and Communication Technologies (GlobConPT), 1–6 (2022)

[14] Thudumu, S., Branch, P., Jin, J., Singh, J.: A comprehensive survey of anomaly detection techniques for high dimensional big data (2020) https://doi.org/10.1186/s40537-020-00320-x

[15] Robles, J.-R., Kim, T.: A review of internet of things for smart home: Challenges and solutions. Journal of Cleaner Production. **149**, 1454–1464 (2017) https://doi.org/10.1016/j.cie.2020.106854

[16] Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. (2019)

[17] Rahman, S.A., Talhi, H.T.C., Mourad, A.: Internet of things intrusion detection: Centralized, on-device, or federated learning? **34**, 310–317 (2020) https://doi.org/10.1109/MNET.011.2000286

[18] Zhao, B., Li, H., Gao, J., Lu, J.: Federated learning with differential privacy: Algorithms and performance analysis (2021)

[19] Ghimire, B.: Recent advances on federated learning for cybersecurity and cybersecurity for federated learning for internet of things. IEEE Internet of Things Journal **9**, 8229–8249 (2022) https://doi.org/10.1109/JIOT.2022.3150363

[20] Zhang, T., He, C., Ma, T., Gao, L., Ma, M., Avestimehr, S.: Federated learning for internet of things, 413–419 (2021) https://doi.org/10.1145/3485730.3493444

[21] Bagdasaryan, E., Shmatikov, V., Truex, S.: Privacy-preserving federated learning with adversarial attacks, 1661–1678 (2021)

[22] Li, T., Sanjabi, M., Beirami, A., Smith, V.: Fair resource allocation in federated learning. (2020)

[23] J. Vanerio, P.C.: Ensemble-learning approaches for network security and anomaly detection. In: Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks, Los Angeles CA USA, pp. 1–6 (2017). https://doi.org/10.1145/3098593.3098594 . https://dl.acm.org/doi/10.1145/3098593.3098594

[24] Gharakheili, H.H., Sivanathan, A., Hamza, A., Sivaraman, V.: Network-level security for the internet of things: Opportunities and challenges. **52**, 58–62 (2019) https://doi.org/10.1109/MC.2019.2917972

[25] Gharakheili, H.H., Sivanathan, A., Hamza, A., Sivaraman, V.: Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity. In: Proceedings of the 2019 ACM Symposium on SDN Research, pp. 36–48. ACM. https://doi.org/10.1145/3314148.3314352 . https://dl.acm.org/doi/10.1145/3314148.3314352

[26] Sivanathan, A., Gharakheili, H.H., Loi, F., Radford, A., Wijenayake, C., Vishwanath, A., Sivaraman, V.: Classifying iot devices in smart environments using network traffic characteristics. **18**, 15 (2019)

[27] Sivaraman, V., Gharakheili, H.H., Fernandes, C., Clark, N., Karliychuk, T.: Smart iot devices in the home: Security and privacy implications. **37**, 71–79 (2018) https://doi.org/10.1109/MTS.2018.2826079

[28] Bonawitz, K.e.a.: Towards federated learning at scale: System design. (2019)

[29] M. Mohri, A.T.S. G. Sivek: Agnostic federated learning., 11 (2019)

[30] Brownlee, J.: A gentle introduction to ensemble learning algorithms. machine learning mastery. (2021)

[31] Breiman, L.: Bagging predictors. Machine Learning. **24** (1996) https://doi.org/10.1007/BF00058655

[32] Gour, L., Waoo, A.-A.: Fault-tolerant framework with federated learning for reliable and robust distributed system. Proceedings of The International Conference on Emerging Trends in Artificial Intelligence and Smart Systems, THEETAS 2022, 16-17 April 2022, Jabalpur, India (2022) https://doi.org/10.4108/eai.16-4-2022.2318146

[33] Muralidhar, K.: What is stratified cross-validation in machine learning? (2021)

[34] Brownlee, J.: Failure of classification accuracy for imbalanced class distributions. (2021)

[35] Burke, D., J. Brundage, R.R.: Measurement of the false positive rate in a screening program for human immunodeficiency virus infections. The New England Journal of Medicine https://doi.org/10.1056/NEJM198810133191501

[36] Colquhoun, D.: An investigation of the false discovery rate and the misinterpretation of p values. **1**, 140216 (2014) https://doi.org/10.1098/rsos.140216