Journal of Network and Systems Management (2024) 32:20
https://doi.org/10.1007/s10922-023-09782-9

ORIGINAL PAPER

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]

Received: 14 June 2023 / Revised: 8 October 2023 / Accepted: 13 October 2023 /
Published online: 4 January 2024

Abstract
The popularity of Internet of Things (IoT) devices in smart homes has raised significant concerns regarding data security and privacy. Traditional machine learning (ML) methods for anomaly detection often require sharing sensitive IoT data with a central server, posing security and efficiency challenges. In response, this paper introduces FedGroup, a novel Federated Learning (FL) method inspired by FedAvg. FedGroup revolutionizes the central model's learning process by updating it based on the learning patterns of distinct groups of IoT devices. Our experimental results demonstrate that FedGroup consistently achieves comparable or superior accuracy in anomaly detection when compared to both federated and non-federated learning methods. Additionally, Ensemble Learning (EL) collects intelligence from numerous contributing models, leading to enhanced prediction performance. Furthermore, FedGroup significantly improves the detection of attack types and their details, contributing to a more robust security framework for smart homes. Our approach demonstrates exceptional performance, achieving an accuracy rate of 99.64% with a minimal false positive rate (FPR) of 0.02% in attack type detection, and an impressive 99.89% accuracy in attack type detail detection.

Keywords Smart home environment · Cyber attack · Anomaly detection · Federated learning · Internet of things (IoT) · Machine learning

Mathematics Subject Classification 35A01 · 65L10 · 65L12 · 65L20 · 65L70

1 Introduction

IoT has revolutionized the way we bridge the virtual and physical realms, enabling data collection, analysis, and automation of business activities [15]. This transformation has simplified lives and improved the quality of life through

Extended author information available on the last page of the article

continuous and automatic data input [28]. By the year 2025, 478.2 million smart homes will be present across 150 countries worldwide [19]. As the digital economy continues to thrive, underpinned by countless online interactions among individuals, businesses, devices, and data, the need for robust security and privacy becomes paramount [15].

Despite the convenience offered by smart home security systems, they also introduce the risk of compromising personal data security [4]. Trust plays a pivotal role in users' acceptance and adoption of smart homes [29]. These homes are susceptible to various forms of attacks, stemming primarily from network security vulnerabilities and insecure IoT devices [4]. Cybercrime expenses are projected to surge annually by 15%, reaching a staggering USD 10.5 trillion by 2025 [25]. This alarming trend underscores the imperative for enhanced cybersecurity measures and heightened awareness.

As a result, to safeguard the security and privacy of IoT devices in smart homes, maintaining the highest standards is essential. Anomaly detection methods have been extensively studied to identify abnormal behaviours and unexpected anomalies, often relying on traditional ML and deep learning (DL) models, which pose challenges to data privacy [34]. In response, researchers have turned to federated learning, an approach that ensures security and lightweight communication by aggregating updates from local models [22, 23]. Moreover, existing research on anomaly detection has largely overlooked attack-type identification using federated learning. Identifying unusual patterns is crucial in various domains, such as fraud detection in credit card transactions [26]. Effective cybersecurity requires not only detecting malicious behaviour but also categorizing the type of attack. This can be achieved through multi-class categorization procedures that describe the attack and pinpoint its source.

In the prior study [39], we introduced FedGroup, a model addressing anomaly detection by extending the principles of Federated Learning with a group master in the central server. FedGroup proved to be a fast, secure, and fairness-enhancing algorithm with minimal communication overhead. Our comparative analysis showed that FL-based models, including FedGroup, performed on par with or even outperformed standard ML models. Moreover, by integrating Ensemble Learning with FedGroup, we achieved an outstanding attack detection accuracy of 99.91% on the UNSW IoT dataset.

Building upon this foundation, this extended study focuses on attack type detection and attack type detection details, extending the original research scope. The primary contributions of this work are:

1. Addressing Attack Detection: Identifying whether an attack occurred.
2. Introducing Attack Type Detection: Identifying the specific type of attack.
3. Enhancing Attack Type Detection Details: Predicting aspects such as "direct or reflection," "type of attack," "rate of attack," and "layer of attack."
4. Evaluating the performance of Traditional ML, Federated Learning (FedAvg), and FedGroup algorithms in detecting anomalies within smart homes, using a real-world 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 Sect. 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 summarises the main findings of the study.

2 Literature Review

2.1 Traditional Machine Learning

The realm of cybersecurity has witnessed a significant reliance on traditional ML techniques for safeguarding internal networks against potential cyberattacks. These methodologies typically involve training algorithms using historical network traffic data to identify patterns and anomalies indicative of ongoing attacks. Notably, Tsai et al. [34] conducted an analysis spanning from 2000 to 2007, identifying 55 research papers dedicated to intrusion detection. The majority of these studies concentrated on the use of single classifiers, such as K-Nearest Neighbors (KNN) and logistic regression, with limited exploration of ensemble classifiers, which exhibit the potential to outperform single classifiers in terms of classification accuracy.

However, the effectiveness of traditional anomaly detection approaches has been questioned, especially in the context of high-dimensional data [32]. In 2017, Risteska Stojkoska and Trivodaliev [27] highlighted the shortcomings of existing architectures for IoT-based smart home systems, emphasizing the significant data storage and processing demands that prove far from efficient. They underscored the need for novel techniques addressing the challenges associated with managing vast volumes of data in the cloud. Moreover, the imperative of ensuring security in cloud-based solutions, which pose a significant risk of disclosing personal information and data, has become a pressing concern.

In 2021, Al-Haija et al. [2] introduced a pioneering approach, deploying deep learning to address the privacy concerns associated with data collection across various devices. Their Deep Convolutional Neural Network-based system effectively detected IoT device attacks, boasting high classification accuracy and eliminating the need for a central data collection process. Building on this progress, in 2022, Al-Haija et al. [3] introduced Boost-Defence, a detection system tailored to the TON_IoT_2020 dataset. This solution harnessed machine learning techniques for cyberattack detection within 3-layer IoT networks, leveraging the AdaBoost framework, Decision Trees, and various optimizations to achieve remarkable accuracy in cyberattack detection.

2.2 Federated Learning (FL or FedAvg)

Previous research has primarily focused on centralized anomaly detection, where a central model collects data from local models. However, decentralized models offer advantages in terms of computational ease and lightweight communication [22].

The concept of Federated Learning (FL) was introduced by Google in 2016, aiming to enhance the efficiency and security of users interacting with mobile devices [37]. FL involves a central model receiving parameter updates and performing averaging updates at the server which is the reason why name it as FedAvg. This approach has shown benefits in collaborative learning, low communication costs, and decoupling cloud storage, effectively addressing challenges in FL [22, 23, 37].

The literature reveals various attack types, including data poisoning, model poisoning, backdoor attacks, inference attacks, and membership inference attacks [36]. Researchers have proposed several methods for attack detection and prevention within FL, including differential privacy, encryption techniques, secure aggregation, and anomaly detection methods [33, 36]. However, recent work in 2022 highlighted the critical impact of non-iid and highly skewed data distributions on FL performance, underscoring the need for improved solutions in this context [12].

To tackle non-iid data distribution issues, a study by Li et al. (2020) outlined three pathways: (1) addressing high communication costs by reducing model update times and communication rounds; (2) managing statistical heterogeneity through local training model modifications and global model focus; (3) handling structural heterogeneity, encompassing fault tolerance and resource allocation strategies [20]. Another study by Li et al. (2020) [21] emphasized the importance of equitable device distribution and overall accuracy, introducing the q-FFL model to address model bias toward devices with extensive data. In a separate 2022 study on intrusion detection [12], the Fed+ [38] model was introduced, demonstrating improved accuracy compared to FedAvg when dealing with heterogeneous data distributions on the ToN_IoT dataset [5].

In 2023, our previous work [39] introduced FedGroup, an algorithm designed to address the highly skewed distribution challenge of FedAvg. FedGroup departs from computing the average learning of each device and instead adjusts the central model's learning based on the learning patterns observed in distinct groups of IoT devices. Our empirical study, conducted using a real-world IoT dataset, demonstrated that FedGroup achieves anomaly detection accuracy comparable to or better than both FL and non-FL methods. Moreover, FedGroup enhances security by keeping all IoT data localized for model training and updates.

2.3 Ensemble Learning

In their analysis, Vanerio and Casas (2017) demonstrated the effectiveness of Ensemble Learning in anomaly detection, utilizing a Super Learner that incorporated diverse first-level learners and opted for logistic regression for binary classification evaluation in two distinct scenarios [35]. EL, known for its integration of multiple learning models, has proven its capability to enhance predictive performance, particularly in handling challenging training data [35]. In a recent study by Abu Al-Haija et al., EL showcased its reliability in profiling behavioural features of IoT network traffic and detecting anomalous network traffic through their ELBA-IoT model, which achieved an impressive accuracy of 99.6% with minimal inference

overhead [1]. These findings serve as inspiration for amalgamating the advantages of ensemble learning with those of the federated learning model for anomaly detection.

2.4 Summary

This study seeks to explore the realm of attack-type detection within the framework of Federated Learning, taking into consideration not only accuracy but also the false positive rate as critical performance metrics. Additionally, the study addresses the potential bias introduced by the aggregation of distributed models in creating the final global model. FedGroup, the proposed solution, incorporates the functionality and structural insights from a variety of models to effectively tackle these challenges.

3 Methodology

The research plan for this study is structured according to the outline depicted in Fig. 1. This investigation comprises three primary objectives: Firstly, the development of an anomaly detection model to identify potential attacks (Attack Detection); Secondly, the classification of the attack type (Attack Type Detection); and thirdly, a detailed exploration of Attack Type Detection Details. While our prior study primarily centred around the first objective, this extended research effort is dedicated to addressing the second and third objectives. The initial section of this study, titled "Research Data," introduces the network traffic flow data and the attack data. Subsequently, the "Research Method" section details the specifics of the model design. Finally, the "Experiment and Analysis" section outlines the strategic planning and evaluation methodology.

3.1 Research Data

The UNSW laboratory hosts a diverse set of 28 distinct IoT devices organized into various groups, alongside numerous non-IoT devices within the smart environment. This dataset encompasses both malicious and benign data, each spanning two distinct periods captured in 30 PACP files. The initial set of PCAPs covers the timeframe from 28/05/2018 to 17/06/2018, while the subsequent stage extends from 24/09/2018 to 26/10/2018. This research leverages the dataset provided by the UNSW IoT analytics team [17, 18, 30, 31], focusing on a curated selection of 10 IoT devices with wireless internet connectivity. These devices encompass both benign and attack traffic datasets, categorizing them into four distinct groups: Energy management, Camera, Appliances, and Controllers/Hubs, as detailed in Table 1.

3.1.1 Network Traffic Flow Data

Every minute, data pertaining to the network traffic flows of 10 IoT devices is collected, annotated with activity indicators, and stored in ten distinct Excel files

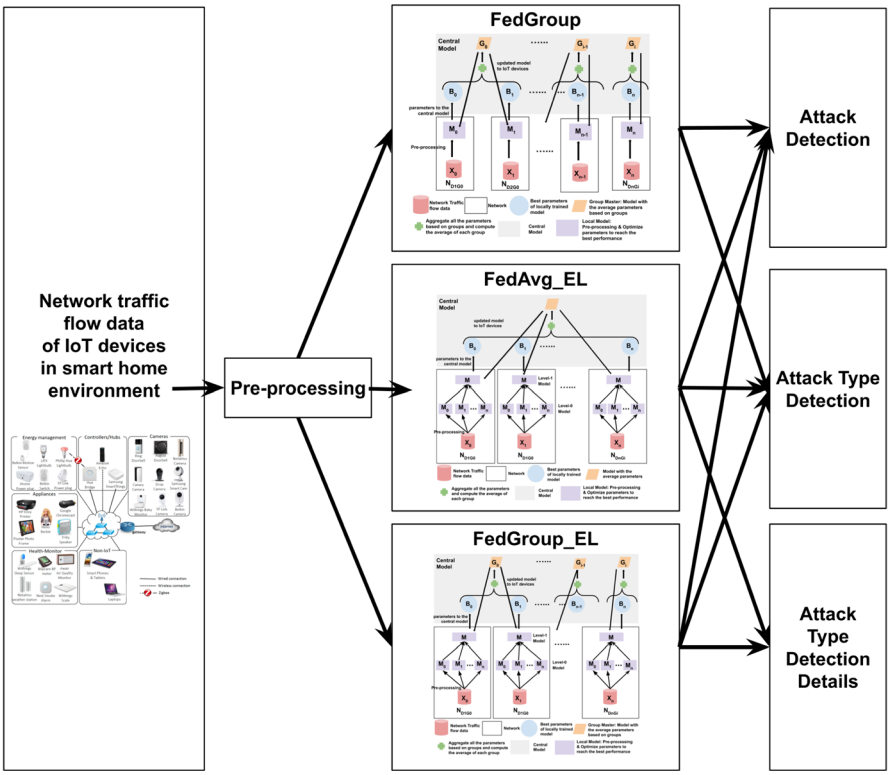

Fig. 1 Outline of the study

Table 1 Ten IoT devices

Ten 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	Controllers/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	Chromecast Ultra	Appliances

dedicated to network traffic flow data. The files include "Timestamp" and a sizable number of pattern characteristics: "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 the byte are not closely related because the size of the packets in this dataset varies. According to the statistics on network traffic flow, it is unknown 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), and ICMP (port number 0). Consequently, extracting direct insights from network flow data proves to be a formidable challenge. In this study, we employ network traffic flow data as the input for forecasting the model's capability to detect attacks and identify their specific attack types.

3.1.2 Attack Data

The UNSW IoT analytics team designed a set of attacks mirroring real-world scenarios 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.

Attack Detection: The determination of normal behavior and the identification of attacks are contingent on a rule-based criterion that evaluates whether a given flow time falls within the specified attack time window. In this context, the condition "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.

Attack Type Detection: There are 45 different types of attack, each attack lasting 10 min 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 about 45 attack types and type of attacks. The attack types mean 45 different attack types, such as ArpSpoof100L2D, and the attack types focus on the varieties such as ArpSpoof.

1. **Attack categories**: Reflection and direct attack 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 (packets per second).

Table 2 45 Attack Types

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

Table 2 (continued)

45 attack types					
No	Attack types	Attack categories	Types of attack	Rates of attack	The layer of attack
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 3 Proportion of attack and attack types

IoT Devices No	0	1	2	3	4	5	6	7	8	9	Total
Proportion of attack											
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											
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

Table 3 (continued)

IoT Devices No	0	1	2	3	4	5	6	7	8	9	Total
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

4. **The layer of attack**: L2D, L2D2L, L2D2W, W2D2W, W2D are the five types of layer scenario 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 FL or FedAvg

FedAvg operates by accepting an initial model from the central server, training decentralized models on local device servers, and subsequently transmitting the best performance parameters back to the central model [37]. The system design, depicted in Fig. 2: FedAvg Protocol, aligns with the principles outlined in Fig. 1: Federated Learning Protocol from Bonawiz's work, "Towards Federated Learning At Scale: System Design" [7]. FedAvg serves as a collaborative model for training data without central data storage, offering several key advantages:

1. FedAvg facilitates the utilization of extensive datasets distributed across various servers, thereby minimizing data transmission while upholding data privacy and security.
2. Distributed servers autonomously train global models on their local data and consolidate these changes into updates sent to the cloud, resulting in more efficient and secure communication.

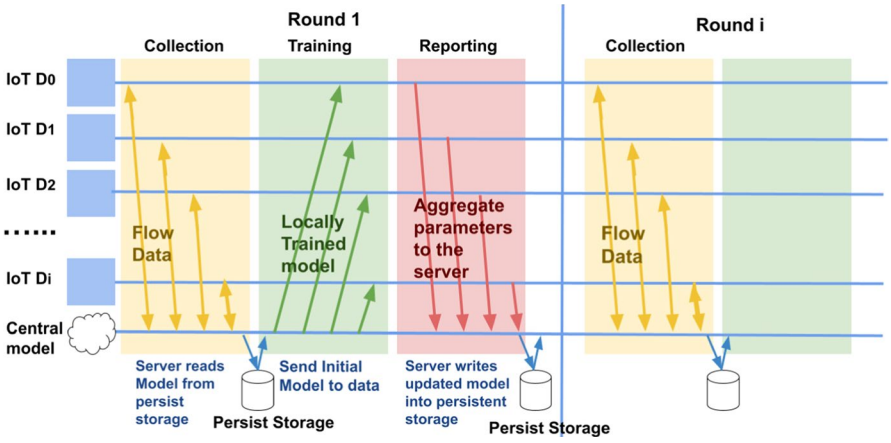

Fig. 2 Federated Learning

3. The cloud server updates the global model by computing a weighted average of parameters. This approach not only supports fault tolerance but also enables scalable computation.

3.2.2 FedGroup

While FedAvg is proficient at aggregating parameters from local servers and determining the mean for the subsequent round, it falls short in effectively managing fairness concerns. The algorithm overlooks a critical factor: the unequal distribution of smart home devices among various groups [24] [21]. Devices within the same category exhibit similar functionalities and face comparable risks. The discrepancy in the training updates, which can vary significantly among participants, is easily treated as an average. While the aggregate accuracy may appear satisfactory, individual accuracy remains obscure, potentially leading to skewed performance distribution [21].

In contrast, FedGroup introduces a novel approach [39]. It advocates computing the average of updates on a group basis rather than opting for a one-size-fits-all averaging strategy (refer to Fig. 3 and Fig. 4). This model comprises multiple local models, a central model, and several group masters within the central model. Local models operate on local servers deployed on IoT devices. Each IoT device collects network traffic data to train a local model and forwards learning updates to the respective group master within the central model. Importantly, this process does not involve data sharing or transmission, maintaining data security and privacy. Each group master aggregates learning parameters within its designated group using a predefined function (e.g., averaging) to fine-tune the learning process. The updated learning is subsequently relayed to all client servers within the group for the next round of training, optimizing the local model's focus on group-specific information. To ensure data security and privacy, information remains localized and is not transmitted over the internet or shared with other devices. Furthermore, to mitigate

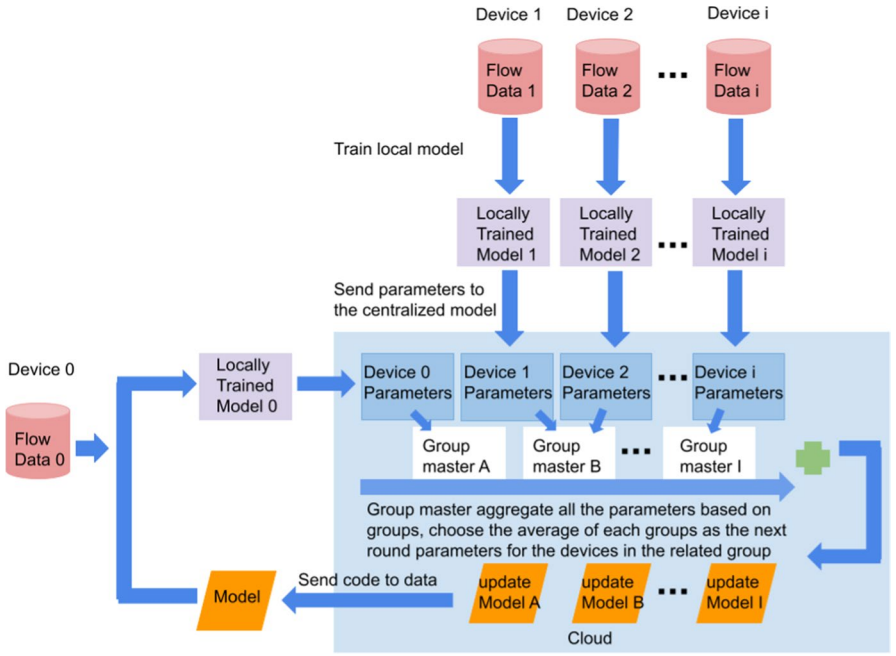

Fig. 3 FedGroup

accuracy disparities stemming from bias, IoT device parameters are determined on a group-specific basis rather than relying on an overall average.

In our study, the IoT devices in the smart home primarily consist of energy management applications such as plugs and bulbs, as indicated by the dataset. Given the substantial disparity in the number of such devices compared to other groups, the cloud server's parameters may exhibit bias towards energy management devices. Specifically, the four IoT devices utilized 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 fall under the Group Energy Management category, encompassing a Belkin Motion Sensor, an iHome PowerPlug, a LIFX Bulb, a Philips Hue lightbulb, a TP-Link Plug, and a Belkin Switch.

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.

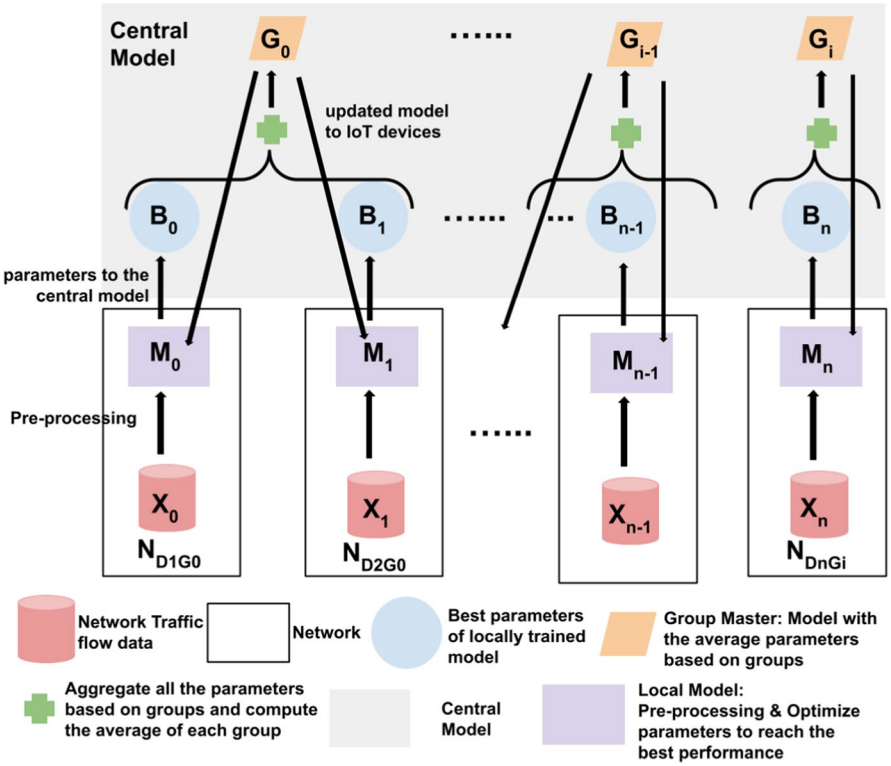

Fig. 4 FedGroup

3.2.3 FedAvg_EL

FedAvg_EL adheres to the FedAvg workflow but introduces a novel approach by replacing the local models with ensemble learning techniques. This adaptation is applied to the task of attack detection and attack type detection, following the procedural steps established by FedGroup, as illustrated in Fig. 5.

Traditionally, local models in federated learning have often employed ML techniques, which can yield inconsistent results due to their specialization in addressing specific types of questions or issues. In contrast, EL harnesses the collective intelligence of various contributing models, offering the advantage of robust and uninterrupted operation even in the presence of individual model failures.

When considering ensemble learning as a local model, there are three types of ensemble learning: Bagging, Stacking, and Boosting. Bagging divides the training dataset into multiple samples within the same model while Boosting

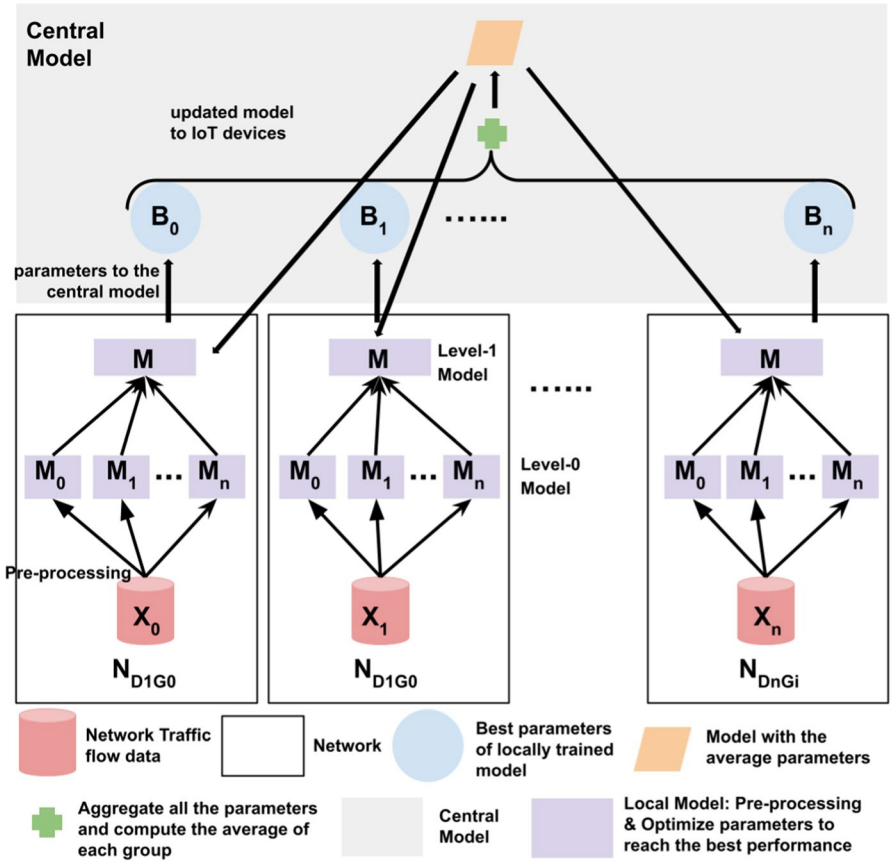

Fig. 5 FedAvg_EL

iteratively corrects predictions. For our specific scenario, Stacking ensemble learning is deemed most suitable. In the Stacking approach, a two-tier model structure is employed. The base models also referred to as Level-0 models, are trained on local devices using the network traffic data. Subsequently, a Level-1 classification model, such as logistic regression, combines the predictions generated by the Level-0 models [8, 9].

Regarding the prediction of attack type details, FedAvg_EL possesses the capability to locally integrate a variety of models within the ensemble learning framework. Figure 6 visually illustrates the model's proficiency in providing customers with what kind of attack rates it is, what type of attack it is, what layers are suffering attacks, and whether it is a direct attack or reflection attack. Armed

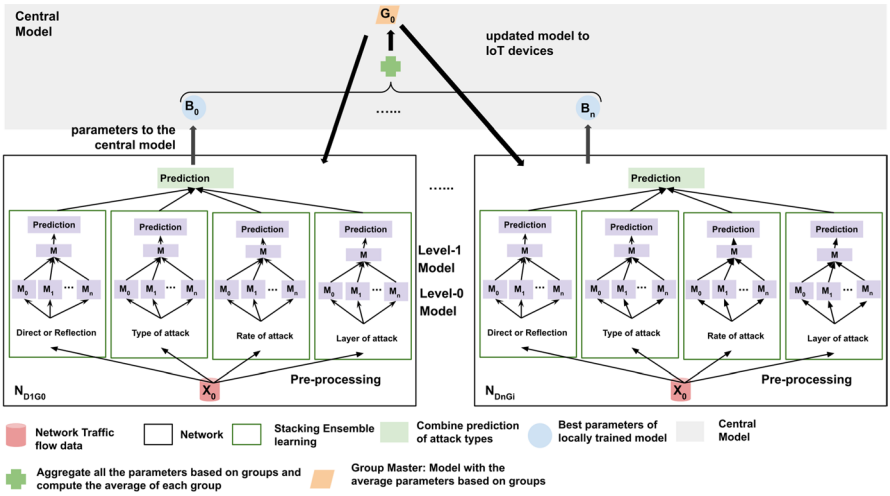

Fig. 6 FedAvg_EL on attack type detection details

with this valuable information, customers can make informed decisions and take appropriate defensive actions. The sequential steps of FedAvg_EL for attack type detection details include:

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 ∈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 ∈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 ensemble learning models, keeping the security and privacy of the data, and the fairness of the FedGroup training procedure 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 [16]. 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 the FedGroup in Fig. 7. Because 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.

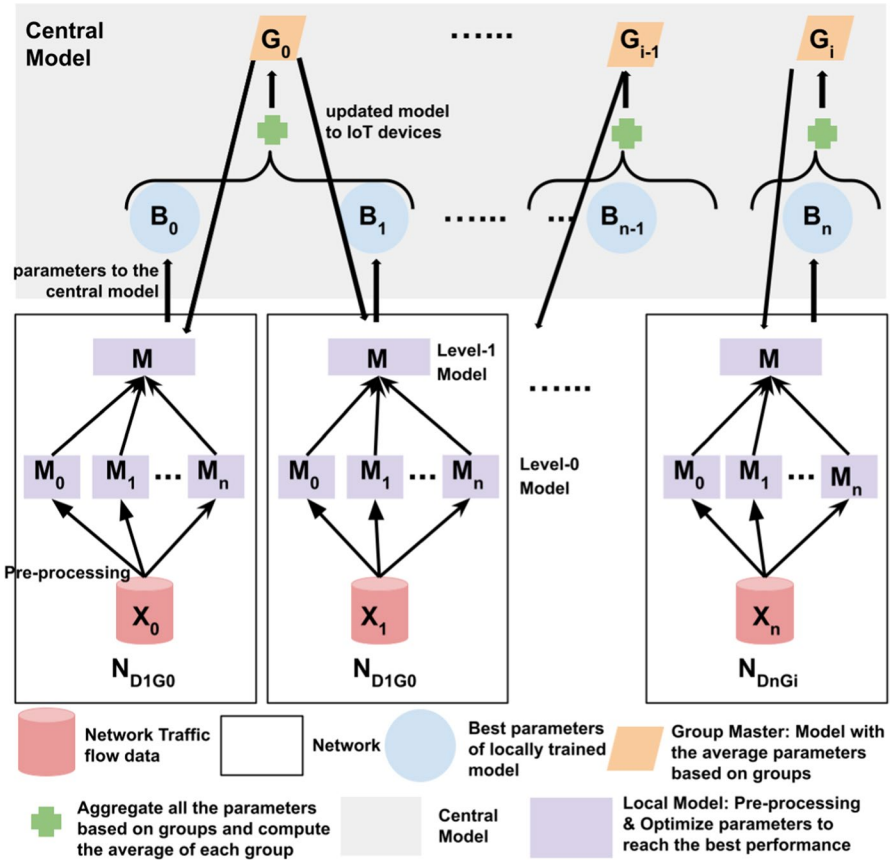

Fig. 7 FedGroup_EL

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

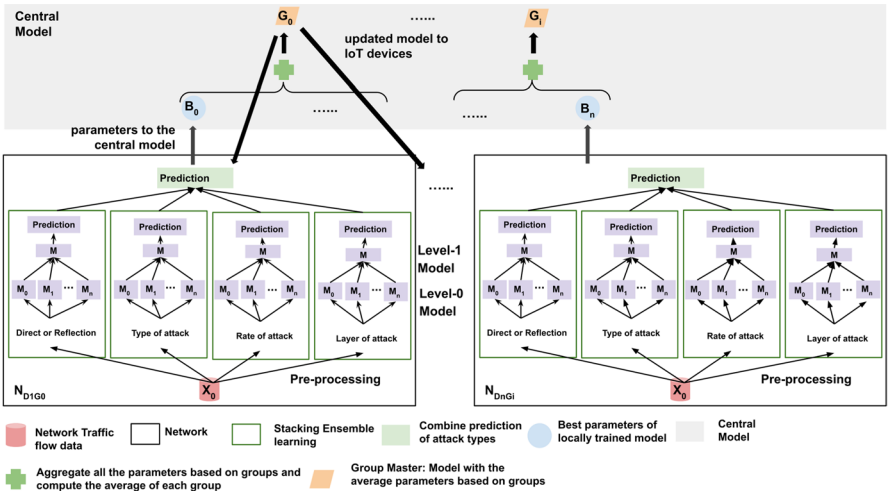

Fig. 8 FedGroup_EL on attack type detection details

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 preprocessing IoT network traffic flow data, "NoOfFlow" is removed since it counts closely related flows. There are 253 attributes related to bytes and packages of port numbers, which encompass various devices using the same port number, while a single device may employ different port numbers. Missing data with NaN

Table 4 Ensemble Learning-adjust level 0 models

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

values represent instances of no network activity for a matching port number, which we replace with a value of 0. This signifies zero packet-level and zero-byte-level network traffic flow data, indicating no network activity at that moment.

The dataset exhibits an imbalance, favoring certain labels. To address this, we employed StratifiedShuffleSplit to divide the data into an 80% training set and a 20% testing set, ensuring a consistent label distribution. We adopted Stratified 5-fold Cross-Validation for model training and evaluation, using an F1 score with weighted averaging on the 20% testing data.

In the context of Stacking Ensemble Learning for attack detection and attack type detection, we employed KNN and Decision Tree at Level-0 and Logistic Regression at Level-1. For attack type detection details, we adjusted four pattern models using the Samsung Device Smart Cam to enhance the initial ensemble learning. Based on the results in Table 4, we selected KNN, Decision Tree, and Naive Bayes for Level-0.

The accuracy classification score is a crucial metric for evaluating the multi-label classification performance, which requires an exact match to the actual data [10]. 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 ($N = TN + FP$) [11, 13]. In our case study, we utilise both accuracy and FPR to evaluate the models. Accuracy measures the correct predictions of abnormal and normal behaviours, and FPR, which quantifies the likelihood of misclassifying a cyber attack as normal behavior.

4 Results

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, and 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

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

Algorithms		Attack detection			Attack Type detection		
Local model	Central model	Accuracy	Running times (sec)	FPR	Accuracy	Running times (sec)	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 rree	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		
Local model	Central model	Accuracy	Running times (sec)	FPR
Ensemble learning	FedAvg	99.89%	4448	4.79%
Ensemble learning	FedGroup	99.89%	4431	5.23%

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. Furthermore, if we focus on differences in FPR that are larger than

1%, then the FPRs of the FL-based model are less than the FPRs of the traditional ML model. FL takes advantage of local training data to reduce running time as a result of lightweight communication and a decentralised learning model. Furthermore, data security is ensured when 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 in FPRs that are greater 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, parameter 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 be useful for experiment reproducibility and model extension and comparison.

5 Discussion

This study expanded on our previous work on attack detection by investigating attack types and their details, providing valuable information. 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 et al. [24] and Li et al. [21], who argue that uniform distribution may not always be the most suitable objective distribution. Given the significance of addressing bias in training data disclosure, it is essential to bridge this research gap by incorporating group-based update aggregation. Compared with the recent work of Campos et al. [12], we noticed the same problem and our state-of-the-art model provides another way to solve the problem of the various data distribution for the detection of different attacks in an IoT environment.

The study has several constraints. In order to defend the practicality of the proposed strategy, it is first necessary to consider the computational requirements for developing and executing models on the local servers since they are implemented on IoT devices. Incorporating embedded systems, which will be connected to IoT devices with constrained computing capabilities, is one potential solution. There are several papers that have examined how machine learning is implemented on

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

embedded devices [6, 14]. Second, our model did not include real-time detection, and the analysis was performed utilizing all available data in just two communication cycles. Future developments could consider spreading out this procedure over several iterations to increase accuracy. Thirdly, due to computational constraints, only a subset of hyperparameters was considered, which may limit the ability to fine-tune the models.

Moreover, our study is confined to a single smart home environment. As the IoT landscape continues to evolve, encompassing numerous smart homes, smart cities, and transportation systems, we anticipate the emergence of a multitude of diverse attacks occurring concurrently and across various locations. For instance, voice recognition sensors within smart homes serve various functions, from playing music to answering questions and controlling various devices. By studying the parameters of voice recognition devices, the central model can identify vulnerabilities and enhance security for all voice recognition devices within the city.

Future research endeavours should extend their scope to encompass multiple smart home environments and adapt to the evolving landscape of IoT devices. Rather than merely categorizing IoT devices by functionality, such as cameras and appliances, a more nuanced approach could involve dividing them into numerous groups based on various attributes. Consider a smart door product, which offers multiple methods of access, including app control, fingerprint recognition, password entry, card scanning, and key unlocking. By segmenting these attributes, the central model can pinpoint the precise element under attack in the event of a security breach, thereby improving overall security.

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% in attack detection and 99.50% in attack type detection. FedGroup gets 99.91% in 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 and deals with the concern of fairness of the 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 outperforms the traditional machine learning model.

In summary, 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 categories of IoT devices. Moreover, the study investigates the use of ensemble learning to improve the accuracy of attack type detection, specifically for direct or reflection attacks, 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 study sheds light on model comparisons, further empirical investigations are necessary to delve into continuous real-time learning and other fairness strategies in the realm of federated learning. Other options for future study include extending the model to other frameworks on anomaly detection, determining the system cost, and examining how wireless network link instability impacts model updating.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s10922-023-09782-9.

Author Contributions BS has led the conceptual design of the study including identifying the problem, the conceptual design of the proposed approach, and the design of the evaluation and experiments. YZ has led the work in terms of investigating and implementing the planned work with detailed supervision and guidance by BS. YZ has also contributed to the conceptual design of the approach and conducted the experiments. She also led the writing of the paper with ongoing and detailed feedback from BS and MJA. MJA has reviewed the research work, edited the paper and provided feedback to improve the technical aspects of the paper. FF has reviewed and edited the paper and provided feedback.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

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.

Declarations

Conflict of Interest The authors declare they have no financial interests.

Ethical Approval Not Applicable.

References

1. Abu Al-Haija, Q., Al Badawi, A., Bojja, G.R.: Boost-defence for resilient iot networks: a head-to-toe approach. Expert Systems **39**(10), e12934 (2022). https://doi.org/10.1111/exsy.12934

2. Abu Al-Haija, Q., Al-Dala'ien, M.: Elba-iot: An ensemble learning model for botnet attack detection in iot networks. Journal of Sensor and Actuator Networks **11**(1) (2022). https://doi.org/10.3390/jsan11010018, https://www.mdpi.com/2224-2708/11/1/18
3. Al-Haija, Q.A., McCurry, C.D., Zein-Sabatto, S.: Intelligent self-reliant cyber-attacks detection and classification system for IoT communication using deep convolutional neural network. In: Selected Papers from the 12th International Networking Conference, pp. 100–116. Springer International Publishing (2021). https://doi.org/10.1007/978-3-030-64758-2_8,
4. Ali, M.H.: Smart home security: Security and vulnerabilities. Wevolver (2021), https://www.wevolver.com/article/smart-home-security-security-and-vulnerabilities
5. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., Anwar, A.: Ton_iot telemetry dataset: a new generation dataset of iot and iiot for data-driven intrusion detection systems. IEEE Access **8**, 165130–165150 (2020)
6. Andrade, L., Prost-Boucle, A., Pétrot, F.: Overview of the state of the art in embedded machine learning. In: 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE). pp. 1033–1038. IEEE (2018)
7. Bonawitz, K.e.a.: Towards federated learning at scale: System design. (2019), http://arxiv.org/abs/1902.01046
8. Breiman, L.: Bagging predictors. Machine Learning. **24** (1996). https://doi.org/10.1007/BF00058655,
9. Brownlee, J.: Failure of classification accuracy for imbalanced class distributions. (2021), https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/
10. Brownlee, J.: A gentle introduction to ensemble learning algorithms. machine learning mastery. (2021), https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/
11. 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 (1988). https://doi.org/10.1056/NEJM198810133191501
12. Campos, E.M., Saura, P.F., González-Vidal, A., Hernández-Ramos, J.L., Bernabé, J.B., Baldini, G., Skarmeta, A.: Evaluating federated learning for intrusion detection in internet of things: Review and challenges. Computer Networks **203**, 108661 (2022). https://doi.org/10.1016/j.comnet.2021.108661. www.sciencedirect.com/science/article/pii/S1389128621005405
13. 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
14. David, R., Duke, J., Jain, A., Janapa Reddi, V., Jeffries, N., Li, J., Kreeger, N., Nappier, I., Natraj, M., Wang, T., et al.: Tensorflow lite micro: Embedded machine learning for tinyml systems. Proceedings of Machine Learning and Systems **3**, 800–811 (2021)
15. Deloitte: What is digital economy? | unicorns, transformation and the internet of things: Deloitte malta. Deloitte (2021), https://www2.deloitte.com/mt/en/pages/technology/articles/mt-what-is-digital-economy.html
16. 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
17. Habibi Gharakheili, H., Sivanathan, A., Hamza, A., Sivaraman, V.: Network-level security for the internet of things: Opportunities and challenges. Computer **52**(8), 58–62 (2019). https://doi.org/10.1109/MC.2019.2917972
18. Hamza, A., Gharakheili, H.H., Benson, T.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. p. 36–48. SOSR '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3314148.3314352
19. 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. pp. 1–6. Los Angeles CA USA (2017). https://doi.org/10.1145/3098593.3098594
20. Lasquety-Reyes, J.: Number of smart homes forecast in the world from 2017 to 2025. Statista (June 2021), https://www.statista.com/forecasts/887613/number-of-smart-homes-in-the-smart-home-market-in-the-world
21. Li, L., Fan, Y., Tse, M., Lin, K.Y.: A review of applications in federated learning. Computers & Industrial Engineering **149**, 106854 (2020). https://doi.org/10.1016/j.cie.2020.106854. www.sciencedirect.com/science/article/pii/S0360835220305532

22. Li, T., Sanjabi, M., Beirami, A., Smith, V.: Fair resource allocation in federated learning. ICLR (2020), http://arxiv.org/abs/1905.10497
23. M. Mohri, G. Sivek, A.T.S.: Agnostic federated learning. arXiv p. 11 (2019)
24. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: International Conference on Artificial Intelligence and Statistics (2016), https://api.semanticscholar.org/CorpusID:14955348
25. McMahan, H.B., Ramage, D.: Federated learning: Collaborative machine learning without centralized training data. Google (2017), https://research.googleblog.com/2017/04/federated-learning-collaborative.html
26. Morgan, S.: Cybercrime to cost the world $10.5 trillion annually by 2025. Cybercrime Magazine (2020), https://cybersecurityventures.com/hackerpocalypse-cybercrime- report-2016/
27. n.d.: Credit card fraud detection: Everything you need to know. Inscribe (2023), https://www.inscribe.ai/fraud-detection/credit-fraud-detection
28. Risteska Stojkoska, B.L., Trivodaliev, K.V.: A review of internet of things for smart home: Challenges and solutions. Journal of Cleaner Production **140**, 1454–1464 (2017). https://doi.org/10.1016/j.jclepro.2016.10.006. www.sciencedirect.com/science/article/pii/S095965261631589X
29. 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. linkinghub.elsevier.com/retrieve/pii/S095965262032922X
30. Shuhaiber, A., Mashal, B.: Understanding users' acceptance of smart homes. Technology in Society (2019). https://doi.org/10.1016/j.techsoc.2019.01.003. linkinghub.elsevier.com/retrieve/pii/S0160791X18300484
31. 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. IEEE Transactions on Mobile Computing **18**(8), 1745–1759 (2019). https://doi.org/10.1109/TMC.2018.2866249
32. Sivaraman, V., Gharakheili, H.H., Fernandes, C., Clark, N., Karliychuk, T.: Smart iot devices in the home: Security and privacy implications. IEEE Technology and Society Magazine **37**(2), 71–79 (2018). https://doi.org/10.1109/MTS.2018.2826079
33. Thudumu, S., Branch, P., Jin, J., Singh, J.: A comprehensive survey of anomaly detection techniques for high dimensional big data. Journal of Big Data (2020). https://doi.org/10.1186/s40537-020-00320-x
34. Truong, N., Sun, K., Wang, S., Guitton, F., Guo, Y.: Privacy preservation in federated learning: An insightful survey from the gdpr perspective. Computers & Security **110**, 102402 (2021). https://doi.org/10.1016/j.cose.2021.102402. www.sciencedirect.com/science/article/pii/S0167404821002261
35. Tsai, C.F., Hsu, Y.F., Lin, C.Y., Lin, W.Y.: Intrusion detection by machine learning: A review. Expert Systems with Applications **36**(10), 11994–12000 (2009). https://doi.org/10.1016/j.eswa.2009.05.029. www.sciencedirect.com/science/article/pii/S0957417409004801
36. Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., Farokhi, F., Jin, S., Quek, T.Q.S., Vincent Poor, H.: Federated learning with differential privacy: Algorithms and performance analysis. IEEE Transactions on Information Forensics and Security **15**, 3454–3469 (2020). https://doi.org/10.1109/TIFS.2020.2988575
37. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: Concept and applications. ACM Trans. Intell. Syst. Technol. **10**(2) (jan 2019). https://doi.org/10.1145/3298981
38. Yu, P., Wynter, L., Lim, S.H.: Fed+: A family of fusion algorithms for federated learning. CoRR **abs/2009.06303** (2020), https://arxiv.org/abs/2009.06303
39. Zhang, Y., Suleiman, B., Alibasa, M.J.: Fedgroup: a federated learning approach for anomaly detection in iot environments. In: Longfei, S., Bodhi, P. (eds.) Mobile and Ubiquitous Systems: Computing, Networking and Services, pp. 121–132. Springer Nature Switzerland, Cham (2023)

Yixuan Zhang is a data science professional who received a Bachelor of Advanced Computing (Honours) degree and a Master of Data Science from the University of Sydney. Yixuan Zhang published papers in the area of Anomaly detection, IoT, Topic Modeling, and Sentiment Analysis. Yixuan Zhang is a data scientist in the Information System and Technology Department, at Rio Tinto as a data scientist.

Basem Suleiman earned his PhD in Computer Science from the University of New South Wales (UNSW), Australia. He currently serves as a Lecturer at UNSW's School of Computer Science and Engineering and holds an honorary Senior Lecturer position at the University of Sydney's School of Computer Science. His research focuses on applied Machine Learning, Software Systems, and Data Analytics, with a strong emphasis on interdisciplinary research to complex real-world problems. His novel research contributions and innovation AI-based software systems attracted recognition through publication in leading international conferences and journals, including IEEE Computer Society, IEEE Transactions, and ACM publications. Dr Suleiman is an active member of professional organizations, including the Australian Computer Society (ACS), Association for Computing Machinery (ACM), and the IEEE Computer Society.

Muhammad Johan Alibasa is a lecturer at the School of Computing, Telkom University, Indonesia. He completed his Ph.D. from the School of Computer Science at the University of Sydney, Australia, in 2020. With a diverse background, he has fostered research collaborations with various fields, including psychiatrists, psychologists, and software engineers. In addition to his research, he has been actively involved in numerous research conferences in Indonesia, serving as a committee member. His primary research interests revolve in the application of AI and machine learning in software engineering, human-computer interaction, and various interdisciplinary studies. He also holds M.Sc. and B.Sc. degrees from Institut Teknologi Bandung, Indonesia.

Farnaz Farid is working as a Lecturer in Cybersecurity and Behaviour at Western Sydney University. She specializes in cybersecurity, networks, data science, web and software systems. She has research experience in cyber security, networking and distributed systems, with more than ten years of experience in Academia and Industry. Her current research interest involves various aspects of cyber security. She is particularly interested in using machine learning techniques to analyze security aspects of IoT devices, focusing on data from agriculture, water monitoring systems, and healthcare applications.

Authors and Affiliations

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

✉ Basem Suleiman
basem.suleiman@sydney.edu.au

Yixuan Zhang
yzha7679@uni.sydney.edu.au

Muhammad Johan Alibasa
alibasa@telkomuniversity.ac.id

Farnaz Farid
farnaz.farid@westernsydney.edu.au

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

4 Western Sydney University, Penrith 2751, NSW, Australia