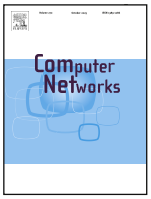
Computer Networks 277 (2026) 112040

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet



FedHome: A federated learning framework for smart home device classification and attack detection by broadband service providers

Md Mizanur Rahman [a,*], Faycal Bouhafs [a], Sayed Amir Hoseini [a], Frank den Hartog [a,b]

[a] School of Systems and Computing, University of New South Wales, 2610, Canberra, ACT, Australia
[b] Network Engineering and Cybersecurity, University of Canberra, 2617, Bruce, ACT, Australia

ARTICLE INFO

Keywords:
Cyber attack detection
Broadband service providers
Device classification
Federated learning
Smart home network
Machine learning

ABSTRACT

The rise of the Internet of Things (IoT) has led to the integration of various devices into smart homes, significantly increasing the complexity and vulnerability of home networks. Consequent network performance issues often lead to complaints directed at Broadband Service Providers (BSPs), which may arise from either legitimate usage or malicious cyber attacks. BSPs, however, lack visibility into client-side networks, which is partly due to privacy concerns. This makes it hard to identify the true cause of performance problems. While previous research has tackled these challenges using Machine Learning (ML) techniques, few studies have approached the problem from the perspective of BSPs. They need a solution that is scalable, accurate, and privacy-preserving. Existing centralized ML models fail to generalize across these heterogeneous environments and provide low accuracy. We address this gap by introducing a novel Federated Learning (FL) framework for smart home device classification and attack detection. The proposed approach offers a privacy-preserving, scalable framework that can achieve accuracies of more than 80%. This framework can be installed inside the existing resource-constrained home gateways, making it suitable for large-scale deployment by BSPs.

1. Introduction

The Internet of Things (IoT) is transforming industries by seamlessly connecting a vast array of devices to the Internet. Among its most impactful applications is the evolution of traditional homes into smart homes, where interconnected devices enhance convenience, efficiency, and security for their users. For instance, smart lighting and automated door systems optimize energy consumption. Artificial Intelligence (AI)-powered cameras, alarms, and motion sensors bolster home security [1]. Moreover, the integration of AI into these smart home systems further elevates the living experience by offering personalized and adaptive services [2]. According to IoT Analytics [3], the number of IoT devices worldwide is projected to reach approximately 50 billion devices by 2030, with the market size estimated to surge to USD 755.98 billion [4]. In the United States of America alone, the average household currently possesses around 21 IoT devices. This figure is expected to rise significantly in the coming years due to the rapid adoption of smart technologies [5].

Despite this rapid proliferation of smart home technology, security remains a major challenge. IoT device manufacturers often prioritize functionality, performance, and time-to-market over security. This often results in devices with a weak security posture [6]. This weakness is exacerbated by the limited technical knowledge of many smart home users, who may struggle to optimize device performance and secure their networks [7]. Consequently, Broadband Service Providers (BSPs) are inundated with customer complaints and troubleshooting requests related to smart home network performance. This is because customers often mistakenly attribute these issues to poor Internet connection quality. The root causes, however, may vary, including internal network issues, device-specific problems, and cyber attacks [8].

To efficiently assist customers, BSPs must be able to quickly and accurately identify and classify devices and detect cyber attacks within smart home networks. The use of automatic device classification and attack detection to address network performance disruptions can significantly enhance the quality of network management and improve customer satisfaction [1]. A promising approach to device classification and cyber attack detection is to leverage Machine Learning (ML) techniques. Traditionally, centralized ML schemes have been employed, where data from end devices is transferred to a central server for training. However, this approach raises privacy concerns and may not be feasible with the vast amounts of data distributed across millions of client networks [9]. Moreover, centralized ML techniques struggle with the

* Corresponding author.
E-mail addresses: md_mizanur.rahman@unsw.edu.au (M.M. Rahman), F.Bouhafs@unsw.edu.au (F. Bouhafs), s.a.hoseini@unsw.edu.au (S.A. Hoseini), frank.denhartog@canberra.edu.au (F. den Hartog).

https://doi.org/10.1016/j.comnet.2026.112040
Received 16 July 2025; Received in revised form 6 January 2026; Accepted 20 January 2026
Available online 22 January 2026

heterogeneity of client networks in BSPs, which consist of diverse devices and network architectures [10,11]. Consequently, distributed ML techniques have been explored to overcome the limitations of centralized approaches by enabling parallel model training at geographically dispersed locations. Nevertheless, distributed ML still faces challenges related to data and system heterogeneity, making it less practical for smart home environments [9].

Federated Learning (FL) emerges as a viable alternative, addressing these challenges by enabling model training directly on local devices, thus preserving privacy and minimizing bandwidth consumption [9]. While FL has primarily been utilized for scalability, security, and privacy enhancement, its potential to address the unique challenges posed by BSPs' heterogeneous client networks has not been fully explored. In particular, no prior studies have thoroughly investigated how FL could improve accuracy in device classification and attack detection scenarios within smart home networks.

In this paper, we evidence that existing ML-based techniques are inadequate for handling the heterogeneity inherent in BSP client networks, rendering them unsuitable for device classification and cyber attack detection. We then propose a novel framework that leverages FL to overcome the limitations of traditional approaches. Our framework is designed to accommodate the heterogeneity of BSPs' client networks in terms of IoT devices, communication protocols, and network topologies. The framework, then, effectively addresses these networks performance and security issues. We evaluate the proposed framework using multiple public datasets, and we assess its effectiveness in various scenarios.

The key contributions of this paper are outlined as follows:

- We introduce an innovative FL architecture that incorporates a hybrid model, merging initial gateway models with personalized client models. This combination effectively addresses the significant heterogeneity found in smart home environments.
- We validate the robustness of our framework using multiple public datasets obtained from various testbeds. This methodology ensures that our evaluation accurately represents real-world variability. Our model has been tested for device classification and cyber attack detection, consistently achieving accuracy rates exceeding 80% in both scenarios.

The remainder of this paper is structured as follows. In Section 2, we present a review of the existing literature and its limitations. Section 3 outlines the challenges BSPs face in deploying effective models due to client network heterogeneity and evaluates the limitations of cross-dataset performance. In Section 4, we present our proposed hybrid FL framework, tailored to BSP-managed smart home networks. Section 5 reports and analyzes the performance of the proposed model in various real-world scenarios, including device classification and Man-In-The-Middle (MITM) attack detection. Finally, Section 6 summarizes the key contributions and highlights avenues for future research and deployment.

2. Related work

In this section, we provide a review of the relevant literature related to device classification and cyber attack detection using ML, Deep Learning (DL), and FL techniques, with a particular focus on limitations in handling network heterogeneity. We categorize the related work into two main areas: Smart Home Device Classification and Cyber Attack Detection.

2.1. Smart home device classification

Device identification and fingerprinting have been pivotal in network security for authenticating devices. With the advent of smart home technologies, ML-based techniques have increasingly been adopted to address the challenge of device classification. These approaches can be broadly classified into two categories: physical hardware-based methods and network traffic-based methods. Physical hardware-based fingerprinting, often employed in Wireless Local Area Networks (WLANs), involves analyzing device configuration [12,13] or exploring radio frequency (RF) signals from the physical layer of the network [14,15].

Our focus is on network traffic-based fingerprinting, which aims to identify devices based on the characteristics of the traffic they transmit and receive. This is achieved by extracting features from packets or examining statistical information from traffic flows [1,2,16,17]. These techniques can be grouped into two categories based on the type of network traffic utilized for ML: device setup traffic (also referred to as network layer signatures) and traffic from the normal operation of the device (during interaction or idle states). Among these, normal operation traffic is more widely utilized than network layer signatures [18].

Traditional ML approaches have been extensively investigated for their use in device classification. Studies [1,19–23] employed Random Forest (RF) classifiers [24,25] and LogitBoost algorithms [26] and achieved high accuracy rates (90-99%) with their testbed datasets. However, these models are limited to specific devices and lack generalizability to future untested devices and diverse network configurations [1,2].

DL-based approaches, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) [27,28], offer enhanced feature learning capabilities with accuracy rates reaching 97–99%. However, these methods incur significant computational and memory overhead [29,30], rendering them impractical for deployment on resource-constrained home gateways where real-time processing is imperative.

FL has recently emerged as a promising paradigm for privacy-preserving device classification. However, existing implementations [31–33] predominantly evaluate performance on limited client configurations (3–5 clients) or through artificial dataset partitioning, thereby failing to adequately address the substantial device diversity encountered across millions of BSP-managed networks.

2.2. Cyber attack detection

Cyber attacks in smart home networks can be divided into two groups: (1) device-based cyber attacks and (2) attacks on smart home networks as a networked system. Most research on smart homes and IoT networks have focused on device-level attacks. In these studies, authors collected datasets from testbeds after applying various attacks to devices and then evaluated ML/DL algorithms for detection accuracy [34,35]. However, for BSPs, the primary concern lies with network-level attacks and overall network performance - an area that has received comparatively little attention from researchers. In this section, we discuss the literature related to the latter, i.e., attacks on smart homes as a networked system.

The performance of home networks can be significantly degraded by cyber attacks. For example, attacks may flood the network with unwanted packets, thus, increasing packet delivery latency and reducing throughput [36,37]. Traditional ML approaches targeted at identifying network-level attacks demonstrate strong performance on standardized datasets, with ensemble methods [38] and feature selection techniques [39] achieving 96–99.9% accuracy on benchmarks including CICIDS2017 and NSL-KDD. Specialized detection systems for botnets [40], wormhole attacks [41], and MITM threats [16,42] further illustrate ML's capabilities. These approaches still face generalization challenges across diverse network environments [10,11].

DL approaches have advanced detection capabilities through sophisticated neural architectures. For example, hybrid models that mix CNN and Gated Recurrent Units (GRUs) [43] can reach about 99% accuracy. In addition, systems for Software-Defined Wireless Networks [44] go beyond detection by also automating the mitigation of spoofing attacks. These methods offer enhanced feature learning but require substantial computational resources.

FL-based detection systems [45–49] represent significant advancements in privacy-preserving security. However, current systems use

general models that do not adapt to the specific needs and behaviors of different smart homes. As a result, they often miss important operational issues when deployed on a large scale across various types of building systems.

2.3. Discussion

The existing literature highlights a persistent gap between home network security solutions evaluated on curated datasets and those needed for deployment in large, diverse real-world environments. Conventional ML and DL models typically assume access to extensive, centrally collected traffic data from many homes. In practice, this assumption rarely holds as large-scale data collection is costly and raises significant privacy concerns. It also does not guarantee good generalization across households with different device mixes, usage patterns, and network configurations. Results based on training and testing on the same dataset often overestimate real-life performance. Even large centralized datasets tend to work well only in environments that closely resemble the original data source.

FL has emerged as a promising alternative because it allows models to be trained locally while keeping raw traffic within each home. However, most FL-based studies in this area still rely on artificial client simulations, created by partitioning a single centralized dataset into multiple subsets. Such setups fail to capture the true heterogeneity of real deployments, including variations in protocols, hardware, topology, and background traffic. As a result, reported accuracies may not translate to unseen environments, and the scalability of these approaches in operational networks remains unclear. Table 1 summarizes these limitations of the related works.

This paper addresses these gaps by proposing a novel hybrid FL framework specifically designed for BSP-managed smart home networks. Our approach demonstrates consistent performance across diverse network environments and introduces the first known application of FL to address the performance limitations of traditional ML, DL, and FL-based approaches in classifying smart home devices and detecting cyber attacks in remote client networks.

3. Background and problem description

BSPs typically manage millions of client networks and face significant challenges in maintaining optimal network performance within client networks. One of the primary requirements for BSPs is the ability to accurately identify devices and detect cyber attacks. This allows them to diagnose the root causes of network performance disruptions. To be operationally useful, any analytics-based solution BSPs deploy must: (i) give them timely visibility into device types and cyber attacks inside each home; (ii) preserve subscriber privacy by keeping raw traffic inside the home; (iii) be generalizable across highly heterogeneous device mixes, vendors and usage patterns; and (iv) scale to millions of households, which practically means that part of the solution will have to run on existing, resource-constrained routers or gateways.

Conventional ML and DL approaches typically use the same dataset for both training and testing to assess the accuracy of algorithms. In practice, this centralized training paradigm conflicts with the above architectural concerns. More specifically, it requires the creation of separate datasets for each home, all of which have to be sent to the BSP network to be processed (training and testing) on a central server. Implementing this technique in large-scale BSP networks requires collecting large amounts of data from each client network, which is impractical due to high costs and scalability issues.

An alternative approach involves developing a single large-scale dataset that can be universally applied to classify devices and detect cyber attacks across all client networks. However, this method does not guarantee consistent performance in varying smart home networks. Authors in [10,11] have demonstrated that training on one dataset fails to achieve high accuracy when tested on a different dataset. Their research has shown classification accuracies falling to 20%-30% for various attacks, despite achieving 99.99% accuracy when both training and testing are performed on the same dataset. Table 2 is from [11] and highlights these findings, where different data collection strategies and feature extraction tools are used.

To investigate this challenge within the context of our work as described in this paper, we have selected the UNSW HomeNet dataset [50] and the ARP Spoofing Based MITM Attack Dataset [51]. These datasets are notable for being generated through multiple testbeds involving real smart home IoT and non-IoT devices. These properties make them the largest labeled datasets available in this domain. These datasets are collected using the same methodology, with a single tool employed to extract features from PCAP files. This mitigates concerns that the drop in classification accuracy observed in [10,11] may have been due to differences in data collection strategies between the datasets being compared, or errors during data labeling. Using the UNSW HomeNet and ARP Spoofing Based MITM Attack Datasets instead is likely to reflect real-world network implementations more accurately.

The UNSW HomeNet dataset is specifically designed for smart home device classification. It includes data from four distinct testbeds: CIC IoT [55], UNSW IoT Traces [21], UNSW's "Lab1" [50], and UNSW's "Lab2" [50]. In total, the dataset covers information from 105 devices. For our analysis, we use the same classes for the training and testing. We train the model on one testbed and evaluate its performance on others using the RF algorithm, as recommended in [1].

Table 3 summarizes the results. These results confirm the known challenge of poor generalization across heterogeneous environments. The model has performed well when trained and tested on the same dataset, but has dropped sharply when tested on different testbeds (Fig. 1). For example, training on Lab 1 and testing on Lab 2 has reduced accuracy, precision, recall, and F1 score to 0.232, 0.262, 0.232, and 0.172, respectively. Training on Lab 2 and testing on Lab 1 has given slightly better results (0.302, 0.382, 0.302, and 0.281). In contrast, the CIC IoT dataset has achieved higher accuracy when tested on Lab 1 (0.632) and Lab 2 (0.641), likely because these datasets share common devices from the same manufacturers.

These findings underscore the inherent challenge in building a general-purpose large-scale dataset for reliable device classification in BSP client networks. Differences in device types, manufacturers, and network configurations introduce significant variability that hinders model generalization. This situation highlights the need for adaptive, context-aware approaches to smart home device identification.

In another earlier study, we have also evaluated cross-dataset performance on ARP spoofing-based MITM attacks using the XGBoost algorithm [56]. The results, shown in Table 4, indicate a similar trend: while the CIC IoT dataset [55] has performed moderately well due to its larger volume and variety of device data, the other three datasets, being relatively smaller, have yielded inconsistent results. Notably, the UQ IoT IDS [57] and IoT Network Intrusion datasets [58] have demonstrated high mutual accuracy, and the ARP PCAP Files [59] dataset has performed well on both. These results highlight a critical uncertainty in cyber attack detection in smart home environments. They show that well-known scenarios present in the training dataset are detected accurately, while unknown scenarios fail to be identified, as shown in Fig. 2.

In summary, the conventional approaches of training on a single dataset and testing on others fail to meet the needs of BSPs. This is mainly due to the inherent variability in client networks and device types. As such, there is a need for an alternative approach to device classification and cyber attack detection, which can adapt to diverse and dynamic network environments. FL directly addresses these fundamental limitations as it: (1) eliminates the need for centralized data collection by training models locally on client devices, preserving privacy and reducing bandwidth. (2) enables the global model to learn from the diverse data distributions across all participating households, naturally improving generalization across heterogeneous environments.

Table 1
Summary of related work in smart home device classification and cyber attack detection.

Ref.	Task	Device/Attack Type	Method	Dataset/Testbed	Acc.	Limitation
[1]	Device classification	IoT & non-IoT devices	RF	4 testbeds (105 devices); UNSW HomeNet	0.906	Most ML/DL studies report strong accuracy on public or private testbeds, but the results do not reliably generalize across heterogeneous smart-home networks. These approaches typically assume centralized access to traffic (often raw flows/packets), which is costly and raises privacy concerns for BSP deployments; DL/hybrid methods are often too compute/memory heavy for real-time deployment on home gateways.
[19]			RF	31-device testbed; IoT Sentinel	0.95	
[20]			RF	UNSW IoT Traces, S-IoT, D-IoT, L-NonIoT	0.998	
[21]			Naive Bayes, RF	28-device testbed; UNSW IoT Traces	0.97	
[22]			RF	26-device testbed; IoT IPFIX Records	0.96	
[23]			LogitBoost	41-device testbed	0.998	
[27]			CNN + RNN	RedIRIS	0.996	
[28]			CNN, LSTM	UNSW IoT Traffic	0.975	
[16]	Cyber attack detection	ARP spoofing	XGBoost	CIC IoT, UQ IoT IDS, IoT Network Intrusion, ARP PCAP	0.967	
[38]		Multiple	XGBoost	CICIDS2017	0.999	
[42]		MITM	Gaussian Process Regression	Simulated MITM dataset	0.89	
[43]		Multiple	CNN + GRU	Kaggle dataset	0.99	
[52]		Ping flood	K-Means clustering	Private dataset	0.999	
[53]		Various	Decision Tree, RF, XGBoost	TON_IoT	0.96	
[54]		Various	SVM	NSL-KDD	0.99	
[31]	Device classification	IoT & non-IoT devices	FL + PCA	N-BaIoT, IoTSentinel, UNSW BoT-IoT	0.99	Mostly evaluated with few clients and/or simulated clients via artificial partitioning of centralized datasets, so real non-IID household heterogeneity and large-scale BSP deployment realism are not captured.
[32]			FL + DNN	Aalto; UNSW ACM SOSR; UNSW IEEE TMC	0.90	
[33]			FL + LSTM + 1D-CNN	LwHBench	0.97	
[45]	Cyber attack detection	Various	FL + KNN	IoT dataset	0.94	
[46]		DDoS	FL + DNN	CICIoT2023	0.998	
[47]		Various	FL + NN	CICIDS2017	0.58–1.00	
[48]		Various	FL + TCN + GAN	UNSW-NB15, EdgeIIoT, BoT-IoT	0.99	

Here, LSTM = Long Short-Term Memory; SVM = Support Vector Machine; KNN = K-Nearest Neighbors; NN = Neural Network; DNN = Deep Neural Network; TCN = Temporal Convolutional Network; GAN = Generative Adversarial Network; PCA = Principal Component Analysis.

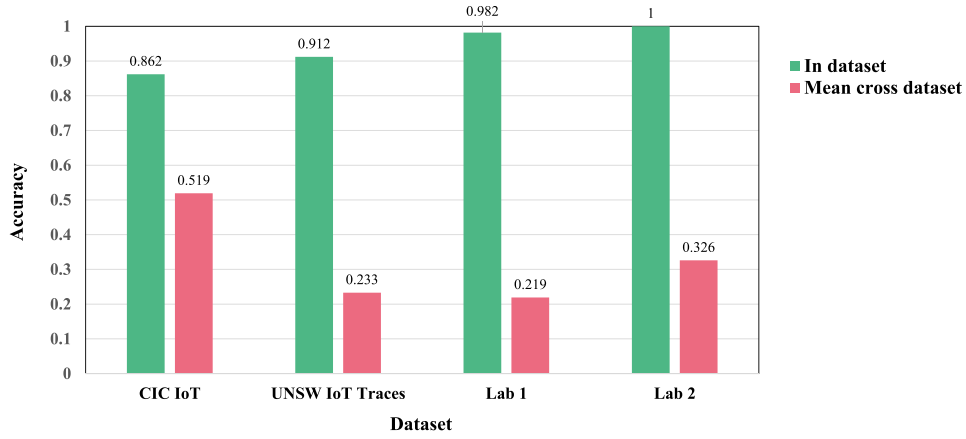


Fig. 1. Random Forest performance metrics for device classification: in-dataset vs mean cross-dataset accuracy.

Table 2
Model performance on Aposemat IoT-23 and Ton_IoT datasets [11].

Train	Test					
	Aposemat IoT-23			Ton_IoT		
	GBM	RF	NN	GBM	RF	NN
Aposemat IoT-23	0.999	0.999	0.994	0.634	0.605	0.439
Ton_IoT	0.307	0.308	0.209	1.000	1.000	0.951
Combined	0.999	0.999	0.990	1.000	1.000	0.996

Here, GBM = Gradient Boosting Machine and NN = Neural Network

In the following section, we present our FL architecture, specifically designed to overcome these identified challenges. This architecture enables BSPs to deploy accurate device classification and attack detection models. These models adapt to the unique characteristics of each client network while maintaining privacy and scalability.

4. Proposed FedHome framework

Fig. 3 shows the overall architecture of the proposed framework, and Fig. 4 presents the implementation workflow. The framework consists of three main components: an initial gateway model, a client model, and a global model. To deploy the system, BSPs first set up a centralized server and provide gateways to customers. The initial gateway and client models run on the customer's router or gateway, while the global model is maintained on the BSP's server.

The proposed framework works as follows.

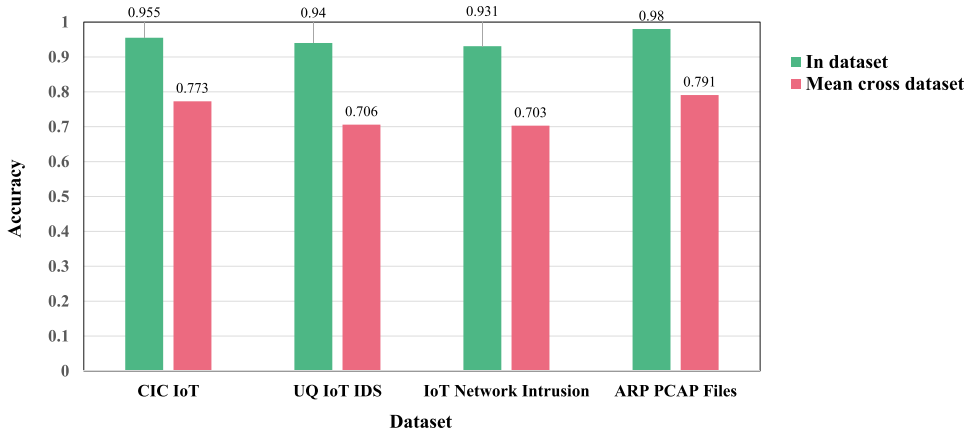


Fig. 2. XGBoost performance metrics for ARP spoofing-based MITM Attack detection: in-dataset vs mean cross-dataset Accuracy.

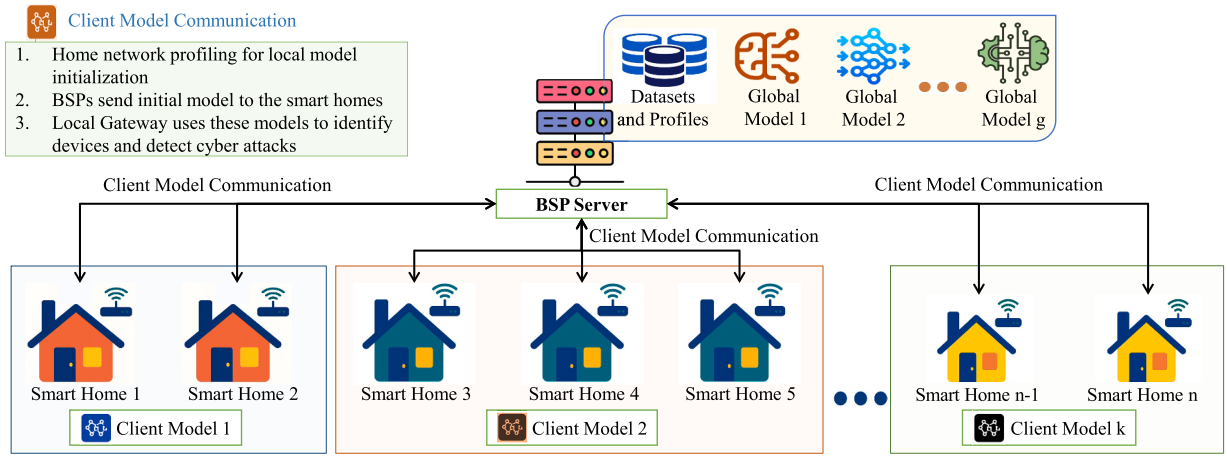


Fig. 3. Architecture of the proposed federated learning framework for BSP-managed smart home networks.

Table 3
Random Forest performance metrics for device classification: in-dataset and cross-dataset testing.

Training Dataset	Metrics	Training Datasets			
		CIC IoT	UNSW IoT Traces	Lab 1	Lab 2
CIC IoT	Accuracy	0.862	0.283	0.632	0.641
	Precision	0.864	0.243	0.703	0.792
	Recall	0.892	0.281	0.632	0.641
	F1 Score	0.881	0.251	0.603	0.682
UNSW IoT Traces	Accuracy	0.203	0.912	0.283	0.212
	Precision	0.373	0.894	0.603	0.381
	Recall	0.203	0.892	0.283	0.212
	F1 Score	0.224	0.894	0.303	0.203
Lab 1	Accuracy	0.153	0.273	0.982	0.232
	Precision	0.532	0.423	0.931	0.262
	Recall	0.153	0.273	0.952	0.232
	F1 Score	0.153	0.241	0.931	0.172
Lab 2	Accuracy	0.432	0.243	0.302	1.000
	Precision	0.543	0.283	0.382	0.972
	Recall	0.432	0.243	0.302	0.992
	F1 Score	0.423	0.173	0.281	0.981

- **BSPs gateway deployment with an initial gateway model:** Initially, a gateway is deployed at each customer's smart home premise. This gateway, which is connected to the BSP's main network, is equipped with a lightweight module capable of collecting device-specific information from the customer's network. The module profiles the home network by implementing ARP spoofing-based data collection techniques. These techniques are combined with ML or custom-built algorithms that help to identify and classify connected devices.

The use of ARP spoofing for these purposes comes with various security and privacy concerns. These concerns include the potential abuse of spoofed ARP messages. They also include the exposure of detailed IP-MAC mappings and sensitive network metadata. In our implementation, the active scanning occurs only once during the first connection to the gateway, minimizing network disruption. Crucially, this process requires explicit customer consent before booting up the gateway, ensuring compliance with privacy regulations. All data processing occurs locally on the gateway, with no raw network data transmitted to the BSP server.

Authors in [60] implemented an active network data extraction method based on ARP spoofing in real-world smart home environments. Their study successfully collected data from 7942 devices across 53 vendors and 13 device categories, achieving an identification accuracy of 85%. Using the same method, our gateway system is capable of extracting comparable information, including device names, port numbers, and communication protocols. This information is initially processed using a default built-in initial gateway model to classify devices. Based on this initial identification, the gateway requests an appropriate client model from the BSP server for accurate classification and security monitoring.

To build the initial gateway model, BSPs collect labeled device data from their testbeds and selected volunteer client networks. This data is used to train a device classification model using supervised ML algorithms. Each gateway, embedded with this model,

Table 4
XGBoost performance metrics for ARP Spoofing-based MITM Attack detection: in-dataset and cross-dataset testing [56].

Training Dataset	Metric	Testing Datasets			
		CIC IoT	UQ IoT IDS	IoT Network Intrusion	ARP PCAP Files
CIC IoT	Accuracy	0.955	0.789	0.885	0.646
	Precision	0.955	0.916	0.908	0.435
	Recall	0.955	0.789	0.885	0.646
	F1 Score	0.955	0.840	0.896	0.520
UQ IoT IDS	Accuracy	0.503	0.940	0.951	0.663
	Precision	0.750	0.949	0.905	0.439
	Recall	0.503	0.940	0.951	0.663
	F1 Score	0.336	0.944	0.927	0.528
IoT Network Intrusion	Accuracy	0.503	0.942	0.931	0.663
	Precision	0.253	0.888	0.934	0.439
	Recall	0.503	0.942	0.931	0.663
	F1 Score	0.336	0.914	0.932	0.528
ARP PCAP Files	Accuracy	0.484	0.939	0.951	0.980
	Precision	0.306	0.888	0.905	0.981
	Recall	0.484	0.939	0.951	0.980
	F1 Score	0.333	0.912	0.927	0.980

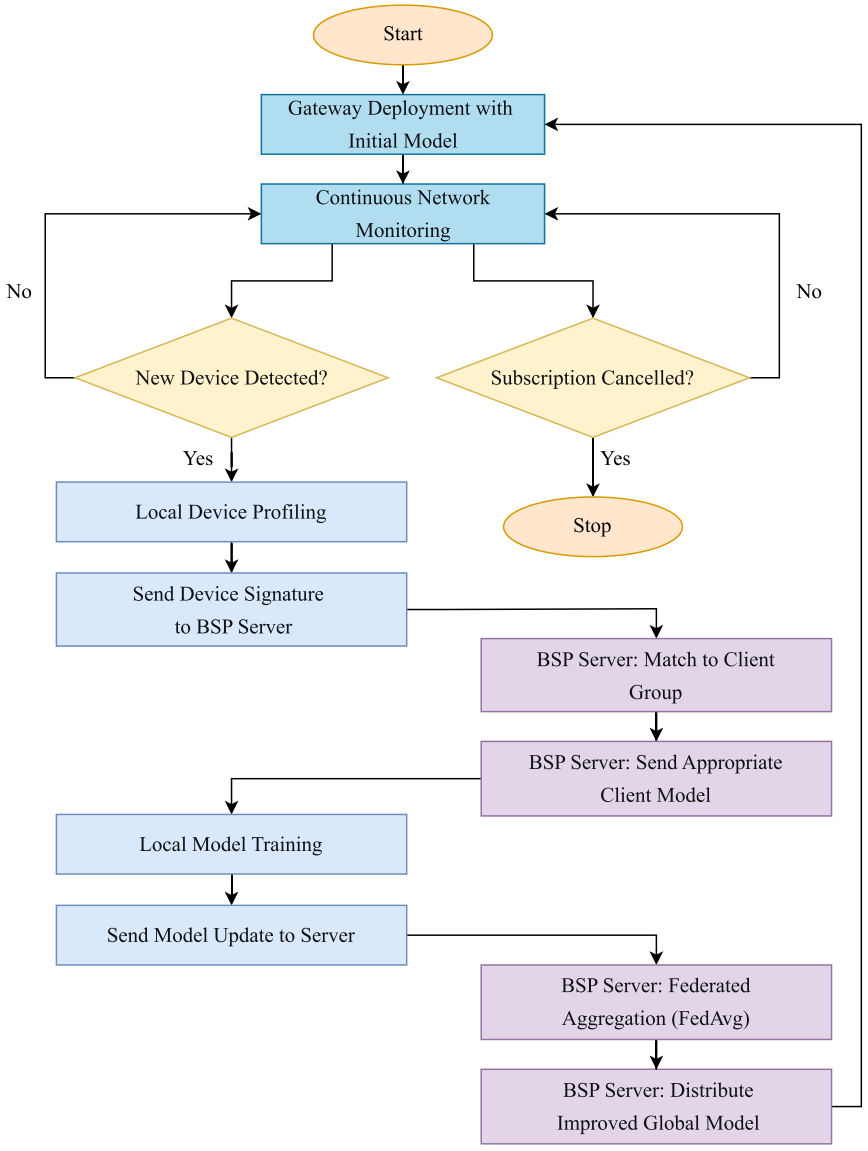


Fig. 4. Operational workflow of the federated learning framework.

Table 5
List of selected features in UNSW HomeNet dataset.

Feature Name	Description
SrcPort	Source port number
DstPort	Destination port number
InitBwdWinByts	Initial backward window size
FlowIATMin	Min time gap between flow packets
FlowIATMax	Max time gap between flow packets
BwdPktLenMax	Max backward packet size
FlowDuration	Total duration of the flow
FlowByts/s	Bytes transferred per second
BwdPkts/s	Backward packets per second
FwdPktLenMax	Max forward packet size
PktLenMin	Smallest packet length
PktLenMax	Largest packet length

continuously monitors and collects real-time device data from smart home environments. When a new or unknown device joins the network, the gateway identifies it and sends a request to the BSP server. The request asks for a new suitable client model tailored to the specific characteristics of that smart home.

- **Client Model Distribution and Local Training:** The BSP server acts as a central coordinator, managing a large-scale dataset of detailed device profiles, client network characteristics, and pre-trained global models. Upon receiving a gateway's request, the server classifies the network into a group with similar device configurations and sends the most appropriate client model. Thousands of client models are maintained in the BSP server to address the diversity of smart home environments. All clients with the same client model are considered as a group. Once the gateway receives the client model, it deploys it for advanced device classification and real-time cyber attack detection. The client model is trained on the local gateway using the smart home's data, ensuring privacy and minimizing raw data transmission to the server.
- **Federated Learning and Global Model Aggregation:** After a predefined training interval, the gateway sends only the updated model parameters back to the BSP server. These updates are then aggregated to refine and enhance the global model. The aggregation model may vary for different client groups and will depend on the ML/DL algorithms used in client networks. This process follows the standard FL paradigm, where multiple rounds of training are conducted to incrementally improve the performance and generalizability of the global model across diverse client environments.

5. Evaluation

To evaluate our framework, we used Flower [61] to train ML models and to implement FL with the above-mentioned scenarios. We chose the Flower framework for its agnostic design, scalability, and efficient communication, enabling seamless integration with TensorFlow and PyTorch. It supports heterogeneous clients and non-Independent-and-Identically-Distributed (non-IID) data, making it ideal for diverse IoT environments.

We evaluated the performance of our proposed framework according to three main criteria: i) device classification accuracy, ii) MITM attack detection accuracy, and iii) computational resource usage for our client networks and the amount of data to be sent to the BSP's server. For that, we chose two datasets to which we applied our framework, namely, the UNSW HomeNet dataset [50] and the ARP Spoofing Based MITM Attack Dataset [51].

In addition, we used the top 12 features of the UNSW HomeNet dataset [1] and 25 features of the ARP Spoofing Based MITM Attack Dataset [16], which were selected based on feature importance rankings as described in the respective papers. These features are listed in Tables 5 and 6 respectively.

Table 6
List of selected features in ARP Spoofing Based MITM Attack dataset.

Feature	Description
src_port	Port number used by the source
dst_port	Port number used by the destination
protocol	Protocol used in the network flow
ip_version	Internet Protocol version (IPv4/IPv6)
bidirectional_last_seen_ms	Timestamp of the last packet in bidirectional flow
bidirectional_bytes	Total bytes transferred in bidirectional flow
src2dst_bytes	Byte count from source to destination
dst2src_bytes	Byte count from destination to source
bidirectional_min_ps	Smallest packet size in bidirectional flow
bidirectional_max_ps	Largest packet size in bidirectional flow
src2dst_min_ps	Smallest packet size from source to destination
src2dst_mean_ps	Average packet size from source to destination
src2dst_max_ps	Largest packet size from source to destination
dst2src_min_piat_ms	Smallest inter-arrival time (ms) from destination to source
dst2src_max_piat_ms	Largest inter-arrival time (ms) from destination to source
bidirectional_syn_packets	Total SYN packets in bidirectional flow
bidirectional_cwr_packets	Total CWR packets in bidirectional flow
src2dst_syn_packets	SYN packets sent from source to destination
src2dst_psh_packets	PSH packets sent from source to destination
src2dst_packets	Packet count from source to destination
application_name	Traffic-generating application name
application_is_guessed	Indicator if the application is guessed
application_confidence	Confidence level in the application's classification
requested_server_name	Server name requested by the client

First, we split each lab testbed dataset into 10 subsets to maintain statistical consistency. We achieved this by comparing mean, median, and variance and validating distributions using histograms, box plots, and the Kolmogorov-Smirnov test [62]. Our FL model was trained over 10 rounds to evaluate accuracy improvements. We utilized RF for device classification and XGBoost for MITM attack detection on the client node. This choice was justified by the demonstrated ability of these algorithms to provide the best performance in classifying devices and detecting attacks for the chosen datasets [1,16].

More importantly, our primary goal was to demonstrate how FedHome can address shortcomings in centralized ML, particularly in real-world routers and gateways. We could have used DL models like CNNs or LSTMs, but they require much more memory and processing power. That would not be practical for typical home routers. Instead, we focused on models that balance accuracy with real-world deployability. This ensures our FL approach can scale across millions of homes without needing hardware upgrades.

In addition, we used the FedAvg aggregation technique on the server. FedAvg updates the global model by computing a weighted average of locally trained models from distributed clients. Eq. (1) shows the FedAvg algorithm, where the global model w is computed as a weighted average of the local models w_k, based on the proportion of data samples contributed by each client:

$$w = \sum_{k=1}^{K} \frac{n_k}{n} \cdot w_k \tag{1}$$

Where:

- K is the total number of participating clients (or devices) in the FL round.
- $k \in \{1, 2, \ldots, K\}$ indexes each individual client.
- n_k represents the number of data samples held by client k.
- $n = \sum_{k=1}^{K} n_k$ denotes the total number of data samples across all clients.

The term "data size" in this context refers specifically to the number of training samples available on each client device, rather than memory size or data collection duration.

In our study, the number of clients K varied depending on the specific task. For device classification, we set $K = 2$, corresponding to the

two datasets Lab 1 and Lab 2, and $K = 4$ for Lab 1, Lab 2, Lab 3, and Lab 4 in scenario 5. Each lab acted as a separate client in the FL framework, and their respective local models w_k are aggregated proportionally based on the number of training samples n_k they contain. For MITM attack detection, we set $K = 4$, where each dataset, CIC IoT, UQ IoT IDS, IoT Network Intrusion, and ARP PCAP Files, is treated as an individual client.

For each criterion, we describe the evaluation approach and present the corresponding results.

5.1. Evaluation of device classification

Smart home networks vary in terms of device types, models, and manufacturers. To assess our model's adaptability, we designed multiple scenarios by changing device types, models, and manufacturers that reflected different smart home configurations. Since the UNSW HomeNet dataset does not contain ARP-based information for grouping testbeds based on similar devices, we assumed that the method proposed in [60] can accurately identify initial devices in the network. We used Lab 1 and Lab 2 data from the UNSW HomeNet Dataset for this analysis, as both testbeds have similar device types, with the exception of Zigbee devices. This assumption allowed our global model to correctly assign the appropriate client model in the gateways. Instead of applying our framework to the entire dataset, we designed distinct smart home scenarios that reflect realistic and diverse real-world environments. These scenarios were selected to evaluate how well our framework generalizes to practical deployment settings with varying network conditions and device profiles. These scenarios are described below:

1. **Smart homes with similar device types but different manufacturers and at least one uncommon device type:** In this scenario, smart homes have a different number of device types, but the smart home with the largest number of device types includes all device types present in the other homes. We used Lab 1 and Lab 2 as testbeds, which contained similar device types. The primary difference was that Lab 2's devices were a subset of Lab 1, with Lab 1 including an additional Zigbee hub. Additionally, all devices in both testbeds originated from different manufacturers. This setup simulated a common real-world scenario in which similar sets of devices from various brands coexist within smart homes.
2. **Smart homes with similar device types but different manufacturers:** In this scenario, we removed the Zigbee hub from Lab 1 to evaluate the model's performance when all smart homes contain the same device types but from different manufacturers. This setup simulated a common real-world scenario where similar devices from various brands exist within smart homes.
3. **Smart homes with the same device types and manufacturers but different models:** To create this scenario, we selected devices that appear multiple times in the testbeds (Lab 1 and Lab 2). We then split these devices into two groups, representing different smart homes, and analyzed the model's performance in distinguishing devices of the same type and manufacturer but with different models. This setup simulated a common real-world scenario where similar devices but different models from the same brands exist within smart homes. This scenario evaluated how well our FL model generalizes across devices with minimal variations in hardware or firmware.
4. **Smart homes with identical device types, manufacturers, and models:** Similar to the previous scenario, we split Lab 1 and Lab 2 devices into two groups, ensuring that both testbeds contained identical devices from the same manufacturer and model. This setup simulated a common real-world scenario where the same devices from the same brands exist within smart homes. This scenario served as a baseline to measure the highest expected performance, as all devices should have near-identical network characteristics.
5. **Smart homes with identical device types distributed across different physical testbeds:** In this scenario, we created four synthetic smart homes by randomly selecting devices of the same five types (Audio, Camera, Bulb, Smart Plug, and PC) from all four testbeds (Lab 1, Lab 2, CIC IoT, and UNSW IoT Traces). We named these synthetic homes as Lab 1, Lab 2, Lab 3, and Lab 4. Each synthetic home has the same set of device types, but the actual devices come from different original testbeds. As a result, they reflect different environments and deployment conditions. This scenario tested how well our FL model performs when device types are consistent across homes, but the physical devices come from diverse locations and contexts, similar to real deployments in different households.

A detailed comparison of performance across the five scenarios is presented in Table 7. Recall that our evaluations of the Lab 1 and Lab 2 datasets (which contain similar device types but varied manufacturers) using traditional ML methods resulted in low cross-dataset accuracy (below 20%, see Table 3). However, Table 7 shows that when our FL approach was applied to Scenario 1, accuracy improved dramatically, with an overall server accuracy of 0.860 ± 0.052 (95% Confidence Interval (CI) $\approx$ [0.814, 0.906]), i.e., a mean accuracy in the mid-80% range across rounds. This is a remarkable result. FL is particularly known for its superior performance in scalability and privacy preservation. But here, we also achieved substantial and statistically significant gains in accuracy. We attribute this to the ability of FL to deal much more effectively with local models as heterogeneous as smart homes than traditional ML methods.

In the second scenario, after removing the Zigbee hub to harmonize device types, the model maintained strong performance. Across 10 FL rounds, the server accuracy averaged 0.832 ± 0.062, with most rounds above 80%, demonstrating its resilience in environments with brand-level variations.

For the third scenario, we isolated devices of the same type and manufacturer but different models. After 10 rounds, the model achieved an average server accuracy of 0.872 ± 0.064 (high-80% range), indicating strong sensitivity to intra-brand variation and highlighting the benefits of FL in learning fine-grained device distinctions.

In the fourth scenario, we expanded the test using devices from the same manufacturer and similar models across a broader range of types. Even in this more generalized setting, the model maintained an overall server accuracy of 0.842 ± 0.032 (mid-80% range), confirming its robustness and scalability across diverse configurations.

In the fifth Scenario, where identical device types are distributed across four physical testbeds, the framework achieved its best performance. The server reached an overall accuracy of 0.932 ± 0.012 with a very tight 95% CI of approximately [0.921, 0.943]. Individual labs also performed strongly, with mean accuracies ranging from 0.902 (Lab 1) to 0.962 (Lab 4), showing that our FL model generalized well when the same device types were deployed in different environments.

All metrics were computed over five independent runs per scenario, and the reported means, standard deviations, and 95% CIs were averaged over these five repetitions. The statistical analysis confirmed strong model stability across all five scenarios. Overall server accuracy exhibited low standard deviations, ranging from 0.012 (Scenario 5) to 0.064 (Scenario 3). The corresponding 95% CIs for server accuracy lie approximately between 0.814 and 0.943 across all scenarios, providing tight and reliable performance bounds. Notably, Scenario 5 combined the highest accuracy (0.932) with the lowest variability (SD = 0.012). Scenario 4 also showed low variability (SD = 0.032). These results underscore the stability of our FL approach in handling diverse smart home configurations. Apart from a few early rounds in Scenarios 2 and 3, the per-round server standard deviations remained below 0.15, indicating stable convergence behavior throughout the federated training process.

We also evaluated three additional aggregation models: FedProx, FedNova, and Scaffold, to assess the accuracy of device classification. These models displayed lower accuracy compared to FedAvg. Fig. 5 illustrates the device classification accuracy for all four aggregation models across different rounds for scenarios 1–5.

Table 7
Device classification performance across federated learning rounds (FedAvg) – 5 runs average accuracy.

(a) Scenario 1: Smart homes with similar device types but different manufacturers and at least one uncommon device type

Class/Device Selection	Round	Lab 1	Lab 2	Server
		Mean ± SD (95% CI)	Mean ± SD (95% CI)	Mean ± SD (95% CI)
Lab 1: Google Home Mini, Philips Hue Bridge, LIFX Bulb 1, Raspberry Pi4 1, Windows Razor, TP Link Router, TP Link Security Camera 1, TP Link Security Camera 2, Aiwit Video Doorbell 1, Aiwit Video Doorbell 2 **Lab 2**: Amazon Alexa Echo Dot, EVIZ Security Camera, IP LAN Security Camera, MI Security Camera, Raspberry Pi4 2, OpenWRT Router, Ring Video Doorbell	1	0.940±0.011 (±0.010)	0.940±0.012 (±0.011)	0.940±0.012 (±0.011)
	2	0.772±0.160 (±0.140)	1.000±0.003 (±0.003)	0.848±0.113 (±0.099)
	3	0.872±0.014 (±0.012)	0.880±0.013 (±0.011)	0.876±0.012 (±0.011)
	4	0.742±0.064 (±0.056)	0.820±0.064 (±0.056)	0.772±0.044 (±0.039)
	5	0.842±0.031 (±0.027)	0.880±0.032 (±0.028)	0.852±0.022 (±0.019)
	6	0.812±0.091 (±0.080)	0.942±0.093 (±0.081)	0.852±0.072 (±0.063)
	7	0.772±0.120 (±0.105)	0.942±0.122 (±0.107)	0.832±0.084 (±0.074)
	8	0.842±0.113 (±0.099)	1.000±0.112 (±0.098)	0.902±0.082 (±0.072)
	9	0.812±0.054 (±0.047)	0.880±0.052 (±0.046)	0.832±0.034 (±0.030)
	10	0.842±0.110 (±0.096)	1.000±0.112 (±0.098)	0.902±0.082 (±0.072)
	Overall	**0.812±0.064 (±0.056)**	**0.928±0.070 (±0.061)**	**0.860±0.052 (±0.046)**

(b) Scenario 2: Smart homes with similar device types but different manufacturers

Class/Device Selection	**Round**	**Lab 1**	**Lab 2**	**Server**
		Mean ± SD (95% CI)	**Mean ± SD (95% CI)**	**Mean ± SD (95% CI)**
We removed all rows for from Lab 1. Now, device types are the same in Lab 1 and Lab 2, but devices are different except for one bulb. **Lab 1**: Google Home Mini, LIFX Bulb 1, Raspberry Pi4 1, Windows Razor, TP Link Router, TP Link Security Camera 1, TP Link Security Camera 2, Aiwit Video Doorbell 1, Aiwit Video Doorbell 2 **Lab 2**: Amazon Alexa Echo Dot, EVIZ Security Camera, IP LAN Security Camera, MI Security Camera, Raspberry Pi4 2, OpenWRT Router, Ring Video Doorbell	1	0.782±0.112 (±0.098)	0.942±0.112 (±0.098)	0.842±0.083 (±0.073)
	2	0.742±0.182 (±0.159)	1.000±0.181 (±0.158)	0.842±0.132 (±0.115)
	3	0.742±0.102 (±0.089)	0.882±0.102 (±0.089)	0.802±0.073 (±0.064)
	4	0.812±0.011 (±0.010)	0.822±0.011 (±0.010)	0.822±0.011 (±0.010)
	5	0.852±0.021 (±0.018)	0.882±0.022 (±0.019)	0.862±0.015 (±0.013)
	6	0.742±0.142 (±0.124)	0.942±0.141 (±0.124)	0.822±0.102 (±0.089)
	7	0.932±0.013 (±0.011)	0.942±0.012 (±0.011)	0.932±0.011 (±0.010)
	8	0.482±0.372 (±0.326)	1.000±0.372 (±0.326)	0.682±0.262 (±0.230)
	9	0.812±0.052 (±0.046)	0.882±0.052 (±0.046)	0.842±0.034 (±0.030)
	10	0.742±0.182 (±0.159)	1.000±0.181 (±0.158)	0.842±0.132 (±0.115)
	Overall	**0.762±0.123 (±0.107)**	**0.932±0.064 (±0.056)**	**0.832±0.062 (±0.054)**

(c) Scenario 3: Smart homes with the same device types and manufacturers but different models

Class/Device Selection	**Round**	**Lab 1**	**Lab 2**	**Server**
		Mean ± SD (95% CI)	**Mean ± SD (95% CI)**	**Mean ± SD (95% CI)**
Lab 1: Pi_1, Aiwit_VideoDoorbell_1, TP_Link_SecurityCamera_1, WiFiBulb_1 **Lab 2**: Aiwit_VideoDoorbell_2, TP_Link_SecurityCamera_2, Pi_2, Lifx_bulb_2	1	0.942±0.012 (±0.011)	1.000±0.003 (±0.003)	0.972±0.010 (±0.009)
	2	0.892±0.023 (±0.020)	0.922±0.023 (±0.020)	0.902±0.016 (±0.014)
	3	1.000±0.003 (±0.003)	0.922±0.064 (±0.056)	0.972±0.044 (±0.039)
	4	0.782±0.102 (±0.089)	0.922±0.102 (±0.089)	0.832±0.073 (±0.064)
	5	0.832±0.122 (±0.107)	1.000±0.122 (±0.107)	0.902±0.084 (±0.074)
	6	0.942±0.132 (±0.116)	0.752±0.132 (±0.116)	0.872±0.092 (±0.081)
	7	0.892±0.221 (±0.194)	0.582±0.221 (±0.194)	0.772±0.153 (±0.134)
	8	0.832±0.011 (±0.010)	0.832±0.012 (±0.011)	0.832±0.011 (±0.010)
	9	0.892±0.162 (±0.142)	0.672±0.162 (±0.142)	0.802±0.112 (±0.098)
	10	0.892±0.082 (±0.072)	1.000±0.082 (±0.072)	0.932±0.062 (±0.054)
	Overall	**0.882±0.064 (±0.056)**	**0.862±0.142 (±0.124)**	**0.872±0.064 (±0.056)**

(d) Scenario 4: Smart homes with identical device types, manufacturers, and models

Class/Device Selection	**Round**	**Lab 1**	**Lab 2**	**Server**
		Mean ± SD (95% CI)	**Mean ± SD (95% CI)**	**Mean ± SD (95% CI)**
Lab 1: Amazon Alexa Echo Dot 1, Yi Indoor Camera, Philips Hue Bridge, LIFX Bulb 1, Raspberry Pi4 2GB, Teckin Plug1, Android Phone, Aiwit Video Doorbell 1 **Lab 2**: Amazon Alexa Echo Dot 2, Yi Indoor2 Camera, Philips Hue Bridge, LIFX Bulb 2, Raspberry Pi4 2GB, Teckin Plug2, Samsung Galaxy Tab, Aiwit Video Doorbell 2	1	0.922±0.082 (±0.072)	0.802±0.082 (±0.072)	0.862±0.064 (±0.056)
	2	0.812±0.082 (±0.072)	0.932±0.082 (±0.072)	0.862±0.064 (±0.056)
	3	0.862±0.064 (±0.056)	0.772±0.064 (±0.056)	0.822±0.044 (±0.039)
	4	0.692±0.132 (±0.116)	0.872±0.132 (±0.116)	0.772±0.092 (±0.081)
	5	0.832±0.102 (±0.089)	0.972±0.102 (±0.089)	0.892±0.072 (±0.063)
	6	0.832±0.072 (±0.063)	0.932±0.072 (±0.063)	0.882±0.052 (±0.046)
	7	0.812±0.042 (±0.037)	0.872±0.042 (±0.037)	0.832±0.032 (±0.028)
	8	0.782±0.112 (±0.098)	0.932±0.112 (±0.098)	0.852±0.082 (±0.072)
	9	0.862±0.021 (±0.018)	0.832±0.022 (±0.019)	0.852±0.015 (±0.013)
	10	0.812±0.013 (±0.011)	0.802±0.013 (±0.011)	0.802±0.012 (±0.011)
	Overall	**0.822±0.052 (±0.046)**	**0.872±0.072 (±0.063)**	**0.842±0.032 (±0.028)**

Overall, these results underscore the strength of our FL-based approach in accurately classifying devices under varying conditions, while maintaining narrow CIs and low variability across all five scenarios.

5.2. Cyber attack detection

For cyber attack detection, we evaluated our model using the ARP Spoofing Based MITM Attack Dataset, which includes data from four testbeds with varying device counts, locations, and network configurations. This diversity allowed us to assess the generalization capability of our approach across heterogeneous smart home networks.

We used the same FL setup as in the device classification experiments and repeated each attack-detection experiment 5 times. Table 8 reports the mean accuracy, standard deviation, and 95% CIs for each round, both per testbed and at the server. The FL framework achieved a mean

Table 7
Continued of Table 7

(e) Scenario 5: Smart homes with identical device types distributed across different physical testbeds

Class/Device Selection	Round	Lab 1	Lab 2	Lab 3	Lab 4	Server
		Mean ± SD (95% CI)	Mean ± SD (95% CI)	Mean ± SD (95% CI)	Mean ± SD (95% CI)	Mean ± SD (95% CI)
Lab 1: Amazon Echo Dot (Lab 1), Amazon Echo Show, TP-Link Security Camera 1, LIFX Bulb 1, Raspberry Pi4 2GB, GoSund Smart Plug WP2(1)	1	0.912±0.023 (±0.020)	0.922±0.043 (±0.038)	0.982±0.042 (±0.037)	0.982±0.023 (±0.020)	0.952±0.023 (±0.020)
	2	0.912±0.021 (±0.018)	0.962±0.032 (±0.028)	0.962±0.032 (±0.028)	0.962±0.012 (±0.011)	0.952±0.012 (±0.011)
	3	0.912±0.022 (±0.019)	0.982±0.043 (±0.038)	0.932±0.042 (±0.037)	0.952±0.023 (±0.020)	0.942±0.023 (±0.020)
Lab 2: Amazon Alexa Echo Dot 1, Amazon Alexa Echo Studio, TP-Link Security Camera 2, LIFX Bulb 2, Raspberry Pi4 2GB, GoSund Smart Plug WP2(2)	4	0.932±0.021 (±0.018)	0.892±0.032 (±0.028)	0.922±0.032 (±0.028)	0.942±0.012 (±0.011)	0.922±0.012 (±0.011)
	5	0.922±0.022 (±0.019)	0.912±0.043 (±0.038)	0.902±0.032 (±0.028)	0.972±0.023 (±0.020)	0.932±0.023 (±0.020)
	6	0.922±0.021 (±0.018)	0.962±0.032 (±0.028)	0.962±0.032 (±0.028)	0.942±0.023 (±0.020)	0.942±0.012 (±0.011)
Lab 3: Amazon Alexa Echo Dot 2, TP-Link Day Night Cloud Camera, LIFX Light Bulb, Raspberry Pi4 2GB, GoSund Smart Plug WP2(3), GoSund Smart Plug WP3(2)	7	0.902±0.022 (±0.019)	0.922±0.043 (±0.038)	0.972±0.042 (±0.037)	0.932±0.023 (±0.020)	0.932±0.023 (±0.020)
	8	0.892±0.021 (±0.018)	0.882±0.032 (±0.028)	0.892±0.032 (±0.028)	0.972±0.012 (±0.011)	0.912±0.012 (±0.011)
	9	0.872±0.022 (±0.019)	0.892±0.043 (±0.038)	0.952±0.042 (±0.037)	0.972±0.023 (±0.020)	0.922±0.023 (±0.020)
Lab 4: Amazon Echo, Amazon Alexa Echo Spot, TP-Link Tapo Camera, LIFX Smart Bulb, Raspberry Pi4 8GB, GoSund Smart Plug WP3(1)	10	0.882±0.022 (±0.019)	0.952±0.043 (±0.038)	0.902±0.032 (±0.028)	0.952±0.023 (±0.020)	0.922±0.012 (±0.011)
	Overall	**0.902±0.022 (±0.019)**	**0.932±0.043 (±0.038)**	**0.942±0.032 (±0.028)**	**0.962±0.023 (±0.020)**	**0.932±0.012 (±0.011)**

server accuracy of 85.4% (0.854±0.049, 95% CI ±0.043). Server accuracy stayed between 0.803 and 0.962 across the 10 rounds, indicating consistently strong detection performance.

Across individual testbeds, three datasets (CIC IoT, UQ IoT IDS, and IoT Network Intrusion) achieved overall mean accuracies close to 88% (0.882, 0.880, and 0.884, respectively). The ARP PCAP Files dataset performed lower at 66.0% on average (0.660±0.247, 95% CI ±0.216), reflecting its higher variability and more challenging distribution. Notably, in Round 10, all four testbeds reached accuracies above 83.3%. As expected, we saw larger fluctuations in the ARP PCAP Files dataset, where accuracy ranged from 25.3% in Round 6 to 100% in Rounds 9 and 10, and in the UQ IoT IDS dataset, which varied between 60.1% and 100%. These variations arose from the non-IID nature of the distributed smart home data, where the local data available in a given round may contain attack patterns or network conditions that differed from the global model's learned distribution.

Despite this, the FedAvg algorithm successfully mitigated these fluctuations through weighted aggregation of model updates, ensuring that temporary client-level performance dips did not compromise the overall system security. The server maintained consistent performance across all rounds, demonstrating the framework's resilience to localized data variations. This characteristic made the approach particularly suitable for real-world deployment scenarios where network heterogeneity and temporary anomalies commonly occur.

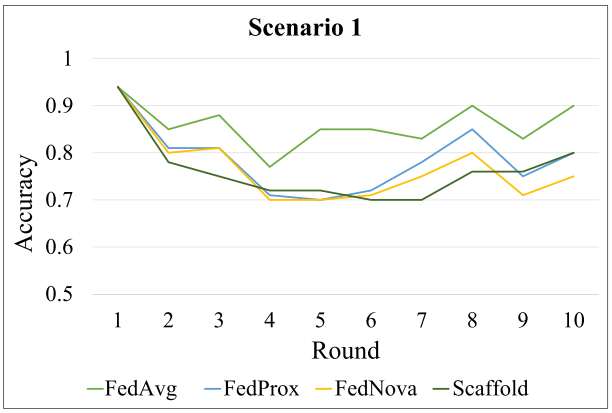
Similar to device classification, we also evaluated FedProx, FedNova, and Scaffold aggregation models to assess the accuracy of ARP spoofing-based MITM attack detection. These models displayed lower accuracy compared to FedAvg in this scenario as well. Fig. 6 illustrates the accuracy for all four aggregation models across different rounds.

5.3. Resource usage efficiency

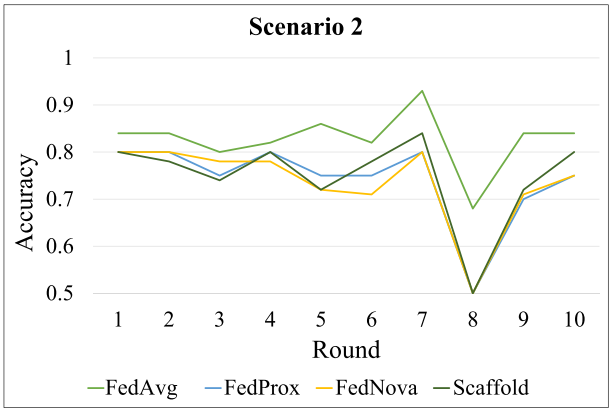
In addition to assessing the performance of device classification and cyber attack detection, it is essential to evaluate the resource requirements of our approach. We analyzed the Central Processing Unit (CPU) and memory requirements to determine if our method can be implemented on existing gateways. Furthermore, we also examined the amount of data that needs to be transferred to the central server of the BSPs.

According to the Australian Competition and Consumer Commission (ACCC), Australia hosts approximately 8.8 million smart home users, distributed among various BSPs. Telstra [63] accounts for the highest number of users, with 3,464,672 households. Aussie Broadband [64] has the lowest number of users, with 750,188 users [65]. The average monthly data usage per household is 450 GB, which is projected to increase to 1 TB per household by 2030 [66]. Given the current 450 GB monthly usage and an average packet size of 1500 bytes [67], it is estimated that approximately 11 million packets are transmitted per household per day, translating to 15 GB of daily data traffic. This underscores the immense volume of data processed through BSPs, rendering the storage of raw network data for each household impractical.

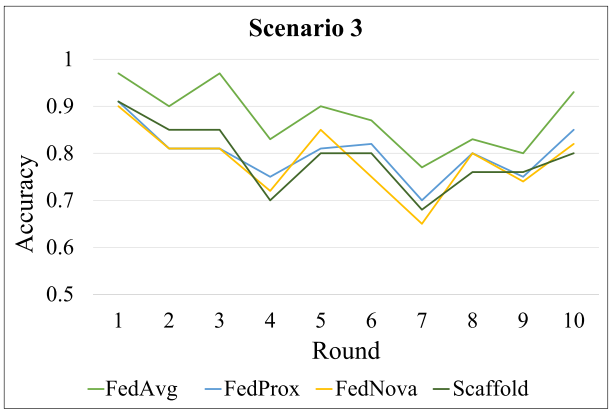
To evaluate the computational efficiency of our proposed framework, we divided the datasets into 10 subsets and generated synthetic data using the make_classification function [68] from the Scikit-learn library when the dataset contained fewer than 1,000,000 rows. This threshold was chosen to ensure a sufficient sample size for meaningful performance evaluation and statistical relevance of the results. This is particularly important when simulating large-scale, real-world IoT environments. These experiments were conducted on a Raspberry Pi 4 equipped with 8 GB RAM and a Quad-core ARM Cortex-A72 (64-bit) @ 1.5 GHz processor. We, then, measured CPU and memory usage by processing 100,000 packets per round for each testbed. For a 100,000-row dataset (1.3 MB of data), the proposed method consumed 330 MB of memory and required 3330 s for processing. The final model update sent to the BSP server was only 0.62 KB in size, meaning that each household would transmit just 66 KB per day.



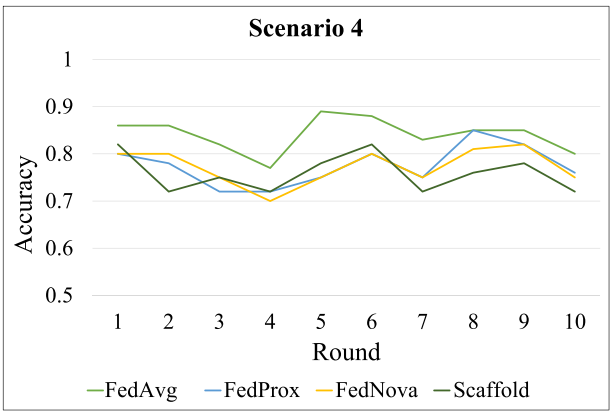
(a) Performance across different rounds for FedAvg, FedProx, FedNova, and Scaffold in scenario 1



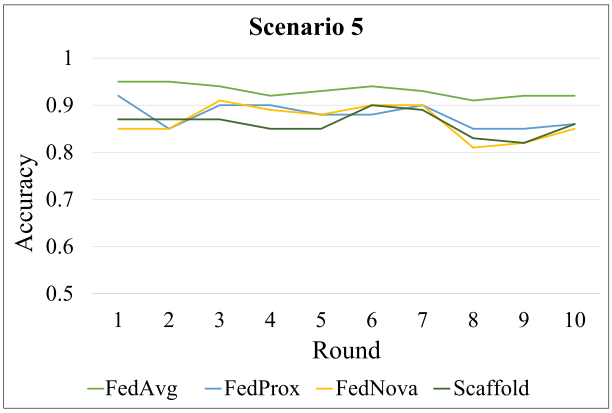
(b) Performance across different rounds for FedAvg, FedProx, FedNova, and Scaffold in scenario 2



(c) Performance across different rounds for FedAvg, FedProx, FedNova, and Scaffold in scenario 3



(d) Performance across different rounds for FedAvg, FedProx, FedNova, and Scaffold in scenario 4



(e) Performance across different rounds for FedAvg, FedProx, FedNova, and Scaffold in scenario 5

Fig. 5. Comparison of device classification Accuracies for FedAvg, FedProx, FedNova, and Scaffold across Scenarios 1–5 in the server. (a) Performance across different rounds for FedAvg, FedProx, FedNova, and Scaffold in scenario 1. (b) Performance across different rounds for FedAvg, FedProx, FedNova, and Scaffold in scenario 2. (c) Performance across different rounds for FedAvg, FedProx, FedNova, and Scaffold in scenario 3. (d) Performance across different rounds for FedAvg, FedProx, FedNova, and Scaffold in scenario 4.

Traditional centralized processing requires approximately 15 GB of daily data transmission per household [65,66]. Our federated approach reduces this by 99.99%, demonstrating superior communication efficiency. Computationally, our solution shows substantial improvement over ML alternatives that require 1–6 GB of memory for similar IoT classification tasks [1]. Our approach uses approximately half the memory compared to other FL implementations that report 520 MB memory usage in constrained environments [31]. Furthermore, existing FL approaches are not specifically designed for BSP-managed smart home deployments and require additional analysis before being deployable at scale in BSP smart home environments.

To assess feasibility within existing BSP infrastructure, we reviewed common smart home router hardware and grouped devices into three tiers. Low-end routers usually have a single-core CPU (600-900 MHz),

Table 8
ARP spoofing-based MITM attack detection performance across federated learning rounds (FedAvg) - 5 runs average accuracy.

Round	CIC IoT	UQ IoT IDS	IoT Network Intrusion	ARP PCAP Files	Server
	Mean ± SD (95% CI)	Mean ± SD (95% CI)	Mean ± SD (95% CI)	Mean ± SD (95% CI)	Mean ± SD (95% CI)
1	0.900 ± 0.004 (± 0.004)	0.602 ± 0.210 (± 0.184)	0.672 ± 0.192 (± 0.168)	0.751 ± 0.108 (± 0.095)	0.840 ± 0.131 (± 0.115)
2	0.900 ± 0.002 (± 0.002)	1.000 ± 0.003 (± 0.003)	1.000 ± 0.002 (± 0.002)	0.502 ± 0.353 (± 0.310)	0.882 ± 0.222 (± 0.195)
3	0.852 ± 0.038 (± 0.033)	1.000 ± 0.001 (± 0.001)	1.000 ± 0.001 (± 0.001)	0.752 ± 0.073 (± 0.064)	0.803 ± 0.110 (± 0.096)
4	0.872 ± 0.022 (± 0.019)	0.601 ± 0.283 (± 0.248)	1.000 ± 0.001 (± 0.001)	0.601 ± 0.142 (± 0.125)	0.822 ± 0.178 (± 0.156)
5	0.900 ± 0.003 (± 0.003)	1.000 ± 0.002 (± 0.002)	0.672 ± 0.243 (± 0.213)	0.503 ± 0.284 (± 0.249)	0.862 ± 0.200 (± 0.175)
6	0.923 ± 0.011 (± 0.010)	1.000 ± 0.002 (± 0.002)	0.671 ± 0.241 (± 0.212)	0.253 ± 0.352 (± 0.309)	0.861 ± 0.313 (± 0.275)
7	0.901 ± 0.003 (± 0.003)	1.000 ± 0.002 (± 0.002)	1.000 ± 0.002 (± 0.002)	0.752 ± 0.071 (± 0.062)	0.821 ± 0.108 (± 0.095)
8	0.921 ± 0.012 (± 0.011)	0.801 ± 0.143 (± 0.125)	1.000 ± 0.002 (± 0.002)	0.501 ± 0.281 (± 0.247)	0.881 ± 0.218 (± 0.191)
9	0.793 ± 0.092 (± 0.081)	0.800 ± 0.003 (± 0.003)	1.000 ± 0.003 (± 0.003)	1.000 ± 0.002 (± 0.002)	0.823 ± 0.102 (± 0.089)
10	0.872 ± 0.023 (± 0.020)	1.000 ± 0.001 (± 0.001)	0.833 ± 0.121 (± 0.106)	1.000 ± 0.002 (± 0.002)	0.962 ± 0.083 (± 0.073)
Overall	**0.882 ± 0.043 (± 0.038)**	**0.880 ± 0.183 (± 0.160)**	**0.884 ± 0.164 (± 0.144)**	**0.660 ± 0.247 (± 0.216)**	**0.854 ± 0.049 (± 0.043)**

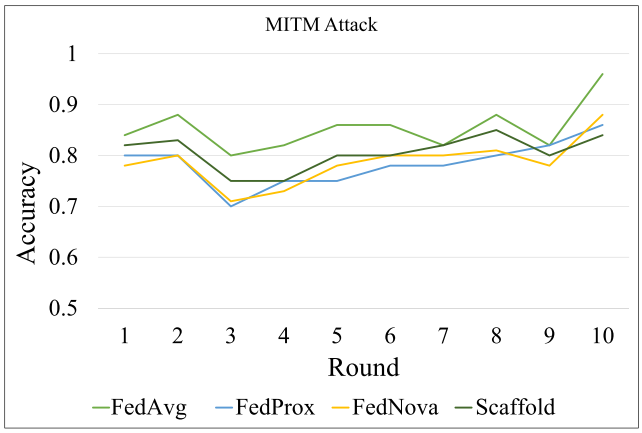


Fig. 6. Comparison of ARP spoofing-based MITM attack detection Accuracies for FedAvg, FedProx, FedNova, and Scaffold across rounds.

64-128 MB RAM, and 8-16 MB flash storage. Mid-range routers often provide dual-core CPUs (800-1400 MHz), 128-512 MB RAM, and 16-128 MB NAND/flash storage. High-end routers are more capable, with quad-core CPUs (1.4-2.2 GHz), 512 MB-2 GB DDR3/DDR4 RAM, and 256 MB-1 GB flash/eMMC.

Our Raspberry Pi tests show the system uses about 330 MB of memory when handling 100,000 packets per round. This level works well on high-end routers and upper mid-range models with 512 MB of RAM. For typical mid-range or budget routers with less memory, we can adjust by using smaller packet batches each round. For example, processing 25,000 packets at a time would lower the memory use to around 80-90 MB just for the data. Including system overhead, total memory would likely stay under 150 MB. This change may mean we run training rounds a bit more often to keep accuracy high. BSPs can also provision or customize gateway hardware when large-scale deployment is required.

5.4. Limitations

This study has several limitations, including:

1. The experiments were conducted on a limited number of lab testbeds and public datasets. As a result, the findings may not fully represent the diversity of real-world smart home deployments.
2. The security evaluation focused only on ARP spoofing-based MITM attacks. Other common smart home attack types (such as DoS, eavesdropping, spoofing, and wormhole) were not included and are left for future work.
3. In the evaluated scenarios, FedAvg emerged as the best-performing aggregation method. However, the optimal aggregation strategy may vary across testbeds, ML/DL models, datasets, and attack types. Therefore, conclusions about aggregation performance may not be generalizable to all settings.
4. We conducted practical experiments using a Raspberry Pi 4. However, we still need to evaluate our approach on actual residential routers or gateways.

6. Conclusion and future work

The rapid proliferation of IoT devices has fundamentally reshaped smart home networks, introducing both convenience and complexity. Performance disruptions have become a common issue, leading to a high volume of complaints to BSPs. However, BSPs often struggle to determine whether these disruptions stem from legitimate causes or are the result of cyber attacks. Accurate identification of devices and detection of malicious activity are, therefore, essential for diagnosing the root causes of such disruptions. Existing solutions in the literature, however, fall short due to the diverse and heterogeneous nature of smart home environments.

This paper proposes a novel framework based on FL for IoT device classification and attack detection, addressing critical limitations of centralized approaches. Traditional models send raw network data to a central server for training. In contrast, our FL approach trains models locally on client devices and shares only the updated parameters with the BSP server. This significantly reduces communication overhead while preserving user privacy.

Our experiments show that the proposed method achieves over 80% accuracy in device classification across diverse testbeds with different device types, vendors, and network setups. It also performs strongly in detecting MITM attacks, maintaining reliable results even in cross-dataset tests. This accuracy is much higher than traditional ML methods, proving that FL can effectively handle highly varied local models like those found in smart homes.

Furthermore, resource usage analysis confirms that the method is both lightweight and efficient. It successfully operates on constrained edge devices like the Raspberry Pi 4. With only 66 KB of daily data transmission per household, the proposed approach offers a scalable and deployable solution for BSPs aiming to secure and manage large-scale smart home infrastructures.

To overcome the limitations of our current study, future work will focus on:

1. **Expanding to additional testbeds:** We plan to conduct experiments across more diverse testbeds involving a wider variety of IoT device types, manufacturers, and network behaviors. We also aim to implement ARP-based techniques to group client networks. These implementations will enhance our model's acceptance for real networks.
2. **Broadening attack coverage:** In this study, we only evaluated ARP spoofing-based MITM attacks. Future work will extend the framework to multi-attack settings to assess how well the model generalizes across different threat categories.

3. **Adaptive model update scheduling:** We intend to introduce adaptive scheduling mechanisms that determine model update frequency based on client activity, data volume, or the detection of anomalies.
4. **Real-world deployment with BSPs:** Future work includes testing the system in live environments in collaboration with BSPs. We plan to run live trials with BSP partners using an OpenWRT-based router or gateway that has the FL framework preinstalled. As part of these pilots, we will also move beyond our current Raspberry Pi 4-based prototype and evaluate the framework on a range of routers or gateways, including high-, mid-, and low-tier devices, to understand resource usage and performance on more constrained hardware. Units will be distributed to volunteer smart-home users, installed in their homes, and exercised under normal conditions before any large-scale rollout. During this pilot, we will evaluate user impact, real-time responsiveness, model stability across rounds, and the operational feasibility of running and managing the system in production networks.

CRediT authorship contribution statement

Md Mizanur Rahman: Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Faycal Bouhafs:** Writing – review & editing, Validation, Supervision; **Sayed Amir Hoseini:** Writing – review & editing, Validation, Supervision; **Frank den Hartog:** Writing – review & editing, Validation, Supervision.

Data availability

The dataset we used in this article are public datasets.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M.M. Rahman, F. Bouhafs, S.A. Hoseini, F. den Hartog, UNSW HomeNet: a network traffic flow based dataset for IoT and non-IoT device classification, Comput. Ind. Eng. 204 (2025) 111041. https://doi.org/10.1016/j.cie.2025.111041

[2] M.M. Rahman, F. Bouhafs, S.A. Hoseini, F. den Hartog, The influence of device type aggregation on the classification of smart home devices using machine learning algorithms, in: Proceedings of the 27th International Conference on Computer and Information Technology, IEEE, 2024, pp. 345–350.

[3] S. Sinha, Number of connected IoT devices, 2023, Accessed: March 28, 2025, https://iot-analytics.com/number-connected-iot-devices/.

[4] SkyQuest, IoT smart homes market insights, 2025, Accessed: March 10, 2025, https://www.skyquestt.com/report/iot-smart-homes-market.

[5] C. Blinder, Average number of smart devices in a home, 2023, Accessed: March 10, 2025, https://www.consumeraffairs.com/homeowners/average-number-of-smart-devices-in-a-home.{html}.

[6] F. Shaikh, E. Bou-Harb, J. Crichigno, N. Ghani, A machine learning model for classifying unsolicited IoT devices by observing network telescopes, in: 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, 2018, pp. 938–943.

[7] W. Wang, Z. Lu, Cyber security in the smart grid: survey and challenges, Comput. Netw. 57 (5) (2013) 1344–1371.

[8] N. Apthorpe, D. Reisman, N. Feamster, Poster: a smart home is no castle: privacy vulnerabilities of encrypted IoT traffic, in: Proceedings of the Network and Distributed System Security Symposium (NDSS), Internet Society, 2017.

[9] L.U. Khan, W. Saad, Z. Han, E. Hossain, C.S. Hong, Federated learning for internet of things: recent advances, taxonomy, and open challenges, IEEE Commun. Surv. Tutorials 23 (3) (2021) 1759–1799.

[10] M.M. Rahman, F. Bouhafs, F. den Hartog, A survey on the effectiveness of existing smart home cyber attacks detection solution: a broadband service providers' perspective, IEEE Open J. Commun. Soc. (2025) 1. https://doi.org/10.1109/OJCOMS.2025.3563270

[11] T.M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, F.T.H.d. Hartog, ToN_IoT: the role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets, IEEE Internet Things J. 9 (1) (2022) 485–496. https://doi.org/10.1109/JIOT.2021.3085194

[12] Y. Cheng, X. Ji, J. Zhang, W. Xu, Y.-C. Chen, DeMiCPU: device fingerprinting with magnetic signals radiated by CPU, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, 2019, pp. 1149–1170.

[13] I. Sanchez-Rola, I. Santos, D. Balzarotti, Clock around the clock: time-based device fingerprinting, in: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 1502–1514.

[14] K. Sankhe, M. Belgiovine, F. Zhou, L. Angioloni, F. Restuccia, S. D'Oro, T. Melodia, S. Ioannidis, K. Chowdhury, No radio left behind: radio fingerprinting through deep learning of physical-layer hardware impairments, IEEE Trans. Cognit. Commun. Netw. 6 (1) (2019) 165–178.

[15] X. Zhou, A. Hu, G. Li, L. Peng, Y. Xing, J. Yu, Design of a robust RF fingerprint generation and classification scheme for practical device identification, in: 2019 IEEE Conference on Communications and Network Security (CNS), IEEE, 2019, pp. 196–204.

[16] M.M. Rahman, F. Bouhafs, S.A. Hoseini, F. den Hartog, Feature relevance for detecting address resolution protocol spoofing in smart homes with machine learning, in: Proceedings of the 51st International Conference on Computers and Industrial Engineering (CIE51), 2024, pp. 1–10.

[17] M.M. Rahman, M.M.H. Chayan, K. Mehrin, A. Sultana, M.M. Hamed, Explainable deep learning for cyber attack detection in electric vehicle charging stations, in: Proceedings of the 11th International Conference on Networking, Systems, and Security (NSysS '24), ACM, New York, NY, USA, 2024, pp. 1–7.

[18] H. Jmila, G. Blanc, M.R. Shahid, M. Lazrag, A survey of smart home IoT device classification using machine learning-based network traffic analysis, IEEE Access 10 (2022) 97117–97141.

[19] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, S. Tarkoma, IoT sentinel: automated device-type identification for security enforcement in IoT, in: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2017, pp. 2177–2184.

[20] R.R. Chowdhury, A.C. Idris, P.E. Abas, Identifying SH-IoT devices from network traffic characteristics using random forest classifier, Wirel. Netw. 30 (1) (2024) 405–419.

[21] A. Sivanathan, H.H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, V. Sivaraman, Classifying IoT devices in smart environments using network traffic characteristics, IEEE Trans. Mob. Comput. 18 (8) (2018) 1745–1759.

[22] A. Pashamokhtari, N. Okui, Y. Miyake, M. Nakahara, H.H. Gharakheili, Inferring connected IoT devices from IPFIX records in residential ISP networks, in: 2021 IEEE 46th Conference on Local Computer Networks (LCN), IEEE, 2021, pp. 57–64.

[23] I. Cvitić, D. Peraković, M. Periša, B. Gupta, Ensemble machine learning approach for classification of IoT devices in smart home, Int. J. Mach. Learn. Cybern. 12 (11) (2021) 3179–3202.

[24] M.M. Rahman, A. Shome, S. Chellappan, A.A.A. Islam, How smart your smartphone is in lie detection?, in: Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2019, pp. 338–347.

[25] A. Shome, M.M. Rahman, S. Chellappan, A.B.M. A.A. Islam, A generalized mechanism beyond NLP for real-time detection of cyber abuse through facial expression analytics, in: Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MobiQuitous '19, Association for Computing Machinery, New York, NY, USA, 2020, p. 348-357.

[26] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting (with discussion and a Rejoinder by the authors), 28 (2000) 337–407. https://doi.org/10.1214/AOS/1016218223

[27] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, J. Lloret, Network traffic classifier with convolutional and recurrent neural networks for internet of things, IEEE Access 5 (2017) 18042–18050.

[28] R. Kumar, M. Swarnkar, G. Singal, N. Kumar, IoT network traffic classification using machine learning algorithms: an experimental analysis, IEEE Internet Things J. 9 (2) (2021) 989–1008.

[29] Y. Lai, A comparison of traditional machine learning and deep learning in image recognition, J. Phys. Conf. Ser., 1314, IOP Publishing, 2019, p. 012148.

[30] C. Chen, P. Zhang, H. Zhang, J. Dai, Y. Yi, H. Zhang, Y. Zhang, Deep learning on computational-resource-limited platforms: a survey, Mob. Inf. Syst. 2020 (1) (2020) 8454327.

[31] H. Wang, D. Eklund, A. Oprea, S. Raza, FL4IoT: IoT device fingerprinting and identification using federated learning, ACM Trans. Internet Things 4 (3) (2023) 1–24.

[32] Sumitra, M.V. Shenoy, HFedDI: a novel privacy preserving horizontal federated learning based scheme for IoT device identification, J. Netw. Comput. Appl. 214 (2023) 103616. https://doi.org/10.1016/j.jnca.2023.103616

[33] P.M.S. Sánchez, A.H. Celdrán, G. Bovet, G.M. Pérez, B. Stiller, A trustworthy federated learning framework for individual device identification, in: 2023 JNIC Cybersecurity Conference (JNIC), IEEE, 2023, pp. 1–8.

[34] I. Cvitić, D. Perakovic, B.B. Gupta, K.-K.R. Choo, Boosting-based DDoS detection in internet of things systems, IEEE Internet Things J. 9 (3) (2021) 2109–2123.

[35] S. Shukla, H. Gupta, Identification and counting of hosts behind NAT using machine learning, SN Comput. Sci. 3 (2022) 1–9. https://doi.org/10.1007/S42979-022-01017-Z

[36] R. Bokka, T. Sadasivam, DIS flooding attack impact on the performance of RPL based internet of things networks: analysis, Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, ICESC 2021 (2021) 1017–1022. https://doi.org/10.1109/ICESC51422.2021.9532901

[37] E. Gamess, T.N. Ford, M. Trifas, Performance evaluation of a widely used implementation of the MQTT protocol with large payloads in normal operation and under a DoS attack, Proceedings of the 2021 ACMSE Conference - ACMSE 2021: The Annual ACM Southeast Conference (2021) 154–162. https://doi.org/10.1145/3409334.3452067

[38] C.S. Kalutharage, X. Liu, C. Chrysoulas, O. Bamgboye, Utilizing the ensemble learning and XAI for performance improvements in IoT network attack detection, in: European Symposium on Research in Computer Security, Springer, 2023, pp. 125–139.
[39] S. Krishnan, A. Neyaz, Q. Liu, IoT network attack detection using supervised machine learning, Int. J. Artif. Intell. Expert Syst. (IJAE) 10(2) (2021) 18–32.
[40] W.N. F. W.M. Zaki, R.S. Abdullah, W. Yassin, M.A. Faizal, M.S. Rosli, Constructing IoT botnets attack pattern for host-based and network-based platform, Int. J. Adv. Comput. Sci. Appl. 12 (8) (2021) 161–168.
[41] S.A. Bhosale, S.S. Sonavane, Wormhole attack detection system for IoT network: a hybrid approach, Wirel. Pers. Commun. 124 (2022) 1081–1108. https://doi.org/10.1007/S11277-021-09395-Y/TABLES/10
[42] N. Sivasankari, S. Kamalakkannan, Detection and prevention of man-in-the-middle attack in IoT network using regression modeling, Adv. Eng. Softw. 169 (2022) 103126. https://doi.org/10.1016/J.ADVENGSOFT.2022.103126
[43] B.B. Gupta, K.T. Chui, A. Gaurav, V. Arya, P. Chaurasia, A novel hybrid convolutional neural network- and gated recurrent unit-based paradigm for IoT network traffic attack detection in smart cities, Sensors 23 (21) (2023) 8686.
[44] H. Mohammadnia, S.B. Slimane, IoT-NETZ: practical spoofing attack mitigation approach in SDWN network, in: 2020 7th International Conference on Software Defined Systems, SDS 2020 (2020) 5–13. https://doi.org/10.1109/SDS49854.2020.9143903
[45] A. Thakur, R. Tyagi, H.K. Tripathy, T. Yang, R.S. Rathore, D. Mo, L. Wang, Detecting network attack using federated learning for IoT devices, in: 2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), IEEE, 2024, pp. 1–6.
[46] A. Almadhor, A. Altalbe, I. Bouazzi, A.A. Hejaili, N. Kryvinska, Strengthening network DDOS attack detection in heterogeneous IoT environment with federated XAI learning approach, Sci. Rep. 14 (1) (2024) 24322.
[47] M. Zang, C. Zheng, T. Koziak, N. Zilberman, L. Dittmann, Federated learning-based in-network traffic analysis on IoT edge, in: 2023 IFIP Networking Conference (IFIP Networking), IEEE, 2023, pp. 1–6.
[48] L. Vemulapalli, P.C. Sekhar, A customized temporal federated learning through adversarial networks for cyber attack detection in IoT, J. Rob. Control 6 (1) (2025) 366–384.
[49] B.B. Gupta, A. Gaurav, W. Alhalabi, V. Arya, E. Alharbi, K.T. Chui, Distributed optimization for IoT attack detection using federated learning and siberian tiger optimizer, ICT Express 11(3) (2025) 542–546.
[50] M.M. Rahman, F. Bouhafs, S.A. Hoseini, F. den Hartog, UNSW HomeNet dataset, Accessed: May 17, 2024, (). https://www.kaggle.com/datasets/mizanunswcyber/iot-and-non-iot-device-classification-dataset.
[51] M.M. Rahman, F. Bouhafs, S.A. Hoseini, F. den Hartog, ARP Spoofing Based MITM Attack Dataset, Accessed: May 17, 2024, (). https://www.kaggle.com/datasets/mizanunswcyber/arp-spoofing-based-mitm-attack-dataset.
[52] D. Stiawan, M.E. Suryani, Susanto, M.Y. Idris, M.N. Aldalaien, N. Alsharif, R. Budiarto, Ping flood attack pattern recognition using a K-Means algorithm in an internet of things (IoT) network, IEEE Access (2021). https://doi.org/10.1109/ACCESS.2021.3105517
[53] S. Tabassum, N. Parvin, N. Hossain, A. Tasnim, R. Rahman, M.I. Hossain, IoT network attack detection using XAI and reliability analysis, in: 2022 25th International Conference on Computer and Information Technology (ICCIT), IEEE, 2022, pp. 176–181.
[54] N. Ben Henda, A. Msolli, I. Haggui, A. Helali, H. Maaref, Attack detection in IoT network using support vector machine and improved feature selection technique, J. Netw. Syst. Manage. 32 (4) (2024) 92.
[55] E.C.P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, A.A. Ghorbani, CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment, Sensors 23 (13) (2023) 5941.
[56] M.M. Rahman, F. Bouhafs, S.A. Hoseini, F. den Hartog, ARProof: a cross-protocol approach to detect and mitigate ARP-spoofing attacks in smart home networks, J. Netw. Comput. Appl. 246 (2026) 104396. https://doi.org/10.1016/j.jnca.2025.104396
[57] E. Chong, J. Ma, K.T. Lwin, UQ IoT-IDS 2021, Accessed: May 17, 2024. https://espace.library.uq.edu.au/view/UQ:17b44bb.
[58] H. Kang, D. H. Ahn, G. M. Lee, J. D. Yoo, K. H. Park, H. K. Kim, IoT network intrusion dataset, 2019. https://doi.org/10.21227/q70p-q449
[59] Researcher111, ARP spoofing PCAP file, Accessed: May 17, 2024. https://github.com/researcher111/ARP-pcap-files/blob/master/arpspoof.pcap.
[60] D.Y. Huang, N. Apthorpe, F. Li, G. Acar, N. Feamster, IoT inspector: crowdsourcing labeled network traffic from smart home devices at scale, Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 4 (2) (2020). https://doi.org/10.1145/3397333
[61] D.J. Beutel, T. Topal, A. Mathur, X. Qiu, J. Fernandez-Marques, Y. Gao, L. Sani, K.H. Li, T. Parcollet, P.P.B. de Gusmão, N.D. Lane, FLOWER: a friendly federated learning framework, 2022. Open-Source, mobile-friendly Federated Learning framework, https://hal.science/hal-03601230.
[62] A. Kolmogorov, Sulla determinazione empirica di una legge di distribuzione, Izv. Akad. Nauk SSSR Seriya Matematicheskaya 4 (1933) 83–99. English translation available in “Selected Works of A.N. Kolmogorov”, Vol. I, 1991, Kluwer Academic Publishers.
[63] Telstra personal, Accessed: March 10, 2025. https://www.telstra.com.au/.
[64] Australia’s most trusted NBN internet provider, Accessed: March 10, 2025. https://www.aussiebroadband.com.au/.
[65] ACCC, NBN wholesale market indicators report, Accessed: March 25, 2025. https://www.accc.gov.au/by-industry/telecommunications-and-internet/national-broadband-network-nbn-access-regulation/nbn-wholesale-market-indicators-report/june-quarter-2024-report.
[66] M. Rowland, Faster NBN for hundreds of thousands more South Australians, Accessed: March 25, 2025. https://minister.infrastructure.gov.au/rowland/media-release/faster-nbn-hundreds-thousands-more-south-australians.
[67] E. Garsva, N. Paulauskas, G. Grazulevicius, Packet size distribution tendencies in computer network flows, in: 2015 Open Conference of Electrical, Electronic and Information Sciences (eStream), IEEE, 2015, pp. 1–6.
[68] N. Jakse, Classification techniques in machine learning, Machine learning in geomechanics 1: overview of machine learning, unervised learning, regression, classification and artificial neural networks (2024) 117-144. https://books.google.com.au/books?hl=en&lr=&id=_TspEQAAQBAJ&oi=fnd&pg=PR9&dq=Machine+Learning+in+Geomechanics+1:+Overview+of+Machine+Learning,+Unervised+Learning,+Regression,+Classification+and+Artificial+Neural+Networks&ots=jRgxLy-snQ&sig=Cayyvk2LGL4q_cdcuIzlJbrS0uQ#v=onepage&q&f=false