Journal of Information Security and Applications 93 (2025) 104127

Contents lists available at ScienceDirect

Journal of Information Security and Applications

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

Collaborative intrusion detection in resource-constrained IoT environments: Challenges, methods, and future directions a review

Vasilis Ieropoulos [a],*, Eirini Anthi [a], Theodoros Spyridopoulos [a], Pete Burnap [a], Ioannis Mavromatis [b], Aftab Khan [b], Pietro Carnelli [b]

[a] Cardiff University, Computer Science Department, School of Computer Science and Informatics, Cardiff, CF244AG, UK
[b] Toshiba Europe Ltd., Bristol Research and Innovation Laboratory, 38-42 Temple Way, Bristol, BS1 6EZ, UK

ARTICLE INFO

Keywords:
IoT
Resource-constrained
Collaborative learning
Intrusion detection systems

ABSTRACT

The rapid growth of technology has increased interconnected large-scale systems, broadening the attack surface for malicious actors. Traditional security solutions often employ centralised management of components like firewalls and intrusion detection systems for consistent configuration. This centralisation introduces a "single point of failure," risking severe consequences if compromised. While redundancy can mitigate concerns in IT systems, it does not scale well for larger systems. Edge computing, which pushes computation closer to endpoint devices, has been explored to improve scalability. The research community has also explored distributing and decentralising cybersecurity operations, especially intrusion detection, using new machine learning methods that mix centralised and distributed approaches to scale effectively while preserving data privacy. However, challenges remain in implementing these methods in large-scale IoT systems due to resource constraints. This paper evaluates intrusion detection methods in large-scale, resource-limited IoT systems, exploring the benefits of low-powered devices for network security and discussing solutions to current implementation challenges.

1. Introduction

There is an explosive increase in the adoption of Internet of Things (IoT) devices in both home and industrial applications, as reported by Frost and Sullivan in 2022 [1]. This uptrend in the usage of IoT devices has revolutionised various sectors and transformed the way we live and work. According to IoT Analytics, the number of IoT devices deployed worldwide is expected to reach 30.9 billion by 2025, representing a compound annual growth rate (CAGR) of 23.9% between 2020 and 2025 [2]. Interconnected IoT devices can collect and share data and perform designated tasks; examples include alarm systems [3], HVAC systems [4], and pressure monitor systems [5]. They can control time-sensitive critical operations, such as monitoring equipment in Industrial Control Systems [6] and take health measurements such as blood pressure monitoring [7].

However, the massive adoption of IoT technologies has increased the cyberattack surface, rendering IoT systems susceptible to potential security breaches [8]. Although IoT devices provide numerous advantages by controlling various operations, they are also vulnerable to a range of attacks such as Denial of Service, Man in the Middle and sniffing attacks. Furthermore, most IoT deployments use standard devices to reduce cost, making them vulnerable to attacks due to poor security measures such as hard-coded passwords and unsecured communication protocols [9]. Due to the lack of security features, they are easy targets for threat actors who can take advantage of different communication protocols such as Wi-Fi [10], Bluetooth [11], LoRa [12], and Zigbee [13].

Existing intrusion detection approaches rely on their operation on signature-based, rule-based, anomaly-based, and machine-learning-based methods. While the former two offer a robust mechanism with centralised rule/signature creation and relevant security policies applied to the system, they fall short in detecting novel attacks. The adoption of ML techniques has emerged as a promising solution to overcome this challenge. By continuously monitoring the entirety of the network from a gateway, ML-based IDS can identify malicious traffic by matching patterns and detecting abnormal deviations from normal network traffic, without relying on predefined rules or signatures. This ability enables the system to learn and adapt continuously to the evolving threat landscape [14].

However, traditional ML solutions, deployed on a single device, still encounter issues such as the single point of failure. This centralised deployment can be problematic, as it makes the system vulnerable to disruptions. Moreover, accurately identifying rogue devices becomes

* Corresponding author.
E-mail address: ieropoulosv@cardiff.ac.uk (V. Ieropoulos).

https://doi.org/10.1016/j.jisa.2025.104127

Available online 17 June 2025

challenging due to the complex and heterogeneous data characteristics in IoT environments. Additionally, models trained to detect specific types of attacks may be susceptible to other forms of attack, limiting their overall effectiveness. Compounding these issues, the perpetually evolving threat landscape demands continuous learning to counter new attacks. However, gathering data for training across large-scale networks can negatively impact operations. Finally, the expensive and power-intensive hardware required for traditional ML implementations adds further barriers to widespread adoption and scalability [15,16].

These issues can be addressed through the use of various collaborative learning methods, such as Federated Learning, gossip learning, ensemble learning, and transfer learning. Collaborative learning methods enable the training of machine learning models directly on IoT devices themselves, avoiding the need to transmit sensitive data to a central server. By utilising the local data streams from multiple devices, these methods allow for the creation of more accurate representations of real network traffic. This approach takes into account the unique patterns and characteristics of the traffic sent and received by different entities on the network, resulting in improved detection of legitimate and malicious activities [17].

Among these collaborative learning methods, Federated Learning stands out as a particularly promising approach. Allows for the creation of a more customised model since the data is based on each node's unique traffic, which can help protect against insider threats such as rogue devices [18,19]. Federated Learning achieves this by training the model on local data from multiple IoT devices, without aggregating the sensitive data in a central server. By keeping data decentralised and localised, Federated Learning addresses privacy concerns and reduces the risk associated with transmitting sensitive information. Furthermore, this approach ensures that the model captures the various characteristics and patterns present in the network traffic of different entities, leading to greater accuracy in the detection of anomalies and attacks [17–19]. However, although various subsets and supersets of Federated Learning exist, they have not been implemented on resource-constrained devices to date. Furthermore, the complexity they introduce might render them unsuitable for such devices.

Research such as the study conducted by Enrique Mármol Campos et al. [20], extensively explores into the realm of FL for intrusion detection in the IoT, with a primary emphasis on commercial IoT devices. The evaluation presented in their work meticulously scrutinises the datasets employed within these scenarios. Additionally, noteworthy contributions from Shaashwat Agrawal et al. [21] explore the intricacies and viability of FL in the context of intrusion detection. Expanding on this idea, Sawsan Abdul Rahman et al. [22] highlight the importance of using collaborative learning in IoT environments for more robust and reliable intrusion detection.

What sets our work apart from these and similar studies is the focus on collaborative methods for intrusion detection, specifically tailored for resource-constrained devices. In particular, our research investigates the application of existing technologies to strengthen security measures on these devices. Moreover, our emphasis extends to the resource-constrained IoT, more precisely targeting microcontrollers that heavily depend on external devices for their security infrastructure. This unique perspective distinguishes our work, offering a novel approach to addressing security challenges in the context of constrained IoT environments. In the context of our work, we categorise devices as resource-constrained if they are capable of operating on bare metal or running a real-time operating system. Additionally, we clarify the distinction between a smart sensor and a microcontroller. While a smart sensor is a physical device that collects data, a microcontroller is a component that interfaces with the sensor to process this data. Thus, they are distinct entities that serve different roles in a system.

In this paper, we provide a review of existing literature on collaborative learning techniques for intrusion detection in IoT environments. This review is based on a broad survey of relevant research, aiming to capture the most significant developments and trends in this area. Rather than following a systematic literature review methodology, we focus on highlighting key studies that contribute to the understanding and advancement of collaborative intrusion detection systems in resource-constrained IoT settings. To the best of our knowledge, this is the first review paper that examines and captures the limitations of existing mechanisms, and demonstrates that collaborative methods can be a promising approach used on resource-constrained devices to enhance their security.

The contributions of this paper are as follows:

1. A review of current collaborative methods used in IoT.
2. Identification of current challenges and limitations of Collaborative IDS in IoT systems.
3. The cybersecurity challenges of Collaborative Learning.
4. A discussion of potential approaches towards collaborative-based security for resource-constrained IoT networks.

The rest of the paper is organised as follows, A Background 2 explaining the types of devices we shall be focusing on, what their vulnerabilities and challenges they face and how they can work in collaborative IDS environments. A Literature review Section 3 reviewing existing methodologies to addressing varying challenges in Collaborative learning environments. A look at the Cybersecurity Challenges 4 of collaborative learning and how they can be addressed. A review of Federated Aggregation methods 5 and how these can be applied in resource-constrained environments. A Review of existing implementations in FL 6 and finally a Conclusions and Future work section discussing the overall consensus 7

2. Background

IoT devices often have limited computing power and storage, making them vulnerable to security issues that can compromise data security [23]. With the increasing number of IoT devices, there is a corresponding rise in security threats like unauthorised access and denial-of-service attacks, highlighting the need for robust security measures [24]. Moreover, the decentralised nature of IoT networks and the resource constraints of IoT devices pose challenges to traditional centralised security approaches. Therefore, efficient IDSs capable of real-time detection and response are essential for protecting IoT systems [25]. Employing a layered security approach, which combines centralised and decentralised elements, is key to effectively addressing these security challenges. Collaborative learning methods have emerged as a possible solution, offering resource optimisation in an environment where IoT devices are often resource-constrained. In contrast to centralised learning, where all deep neural network inference necessitates data transfer to a central cloud server and can impose significant resource demands [26], collaborative learning migrates Deep Neural Network (DNN) computations from the cloud centre to the IoT device side. This is particularly beneficial as centralised learning approaches require a reliable and stable network, which may not always be available in extreme or unstable environments. One such collaborative learning system is DeColla [26], a decentralised and collaborative deep learning inference system for IoT devices. DeColla completely migrates DNN computations from the cloud centre to the IoT device side, relying on the collaborative mechanism to accelerate the DNN inference that is difficult for an individual IoT device to accomplish. It uses a parallel acceleration strategy via a Deep Reinforcement Learning adaptive allocation for collaborative inference, which aims to improve inference efficiency and robustness. DeColla evaluates the MobileNet DNN network trained on the ImageNet dataset [26], and also recognises the object for a mobile web augmented reality application and conducts extensive experiments to analyse the latency, resource usage, and robustness against existing methods. Numerical results show that DeColla outperforms other methods in terms of latency and resource usage, which can reduce at least 2.5 times the latency than the hierarchical inference method when the collaboration is interrupted abnormally [26]. On the other hand, some methodologies like Federated Learning (FL) offer data privacy and geographical flexibility, features that are fundamentally lacking in centralised approaches [27]. In this section, we investigate into the concept of collaborative learning and its evolution over the past three years, specifically in the context of Intrusion Detection Systems (IDS) and IoT.

2.1. Resource constrained devices in IoT

Resource-constrained devices in IoT are those that, by design, have limited processing and storage capabilities [28]. They are designed to provide the maximum data output possible with minimal power input while remaining cost-effective. These devices often operate in harsh conditions or hard-to-reach places and usually run on batteries to maintain the balance between the effective span of their lifetime and the potential costs of device replacement. They are frequently used in applications such as weather conditions monitoring at airports, marine cargo logistics, or agricultural automation [28].

In IoT applications, microcontrollers are crucial components, acting as compact computing units that enable the functionality of smart devices. They provide essential capabilities such as processing power, memory management, and interfacing with peripherals. Various microcontrollers are commonly used in IoT ecosystems, including those found in the Arduino series. Arduino, an open-source platform, offers a range of microcontroller boards like Arduino Uno, Arduino Mega, and Arduino Nano [29,30].

Another notable example is the ESP8266/ESP32 series, known for being cost-effective and low-power Wi-Fi modules that combine microcontrollers with Wi-Fi communication capabilities. These modules are widely employed in smart home applications, powering devices such as smart sockets, bulbs, and switches [30].

STMicroelectronics also provides several series of microcontroller products, including the STM32 series and STM8 series. These microcontrollers are valued for their high performance, low power consumption, and extensive peripheral interfaces, making them suitable for controlling and connecting smart home systems [30].

Additionally, Texas Instruments contributes to the microcontroller landscape with products like the MSP430 series and Tiva C series, known for their reliability and performance, enriching IoT applications with their robust features [30].

2.2. Attacks on resource constrained IoT devices

In the realm of IoT, devices are often exposed to a variety of cyber threats due to their inherent connectivity. These threats are dynamic, with cyber attackers constantly innovating and devising new strategies and techniques [31].

This implies that the AI-based Intrusion Detection Models developed at a certain point in time may lose their effectiveness in detecting new types of attack. For example, a model trained to detect a specific malware might fail to recognise a new variant of the malware that exhibits slightly different behaviour. Similarly, a model trained on network traffic patterns might not identify an intrusion if the attacker uses a unique method to conceal their activity [32].

However, the continuous evolution of IoT models presents its own set of challenges. For instance, it requires efficient and secure methods for sharing updates between the central server and the edge devices. It also necessitates careful management to ensure that the privacy benefits of IoT are not compromised during the update process.

Jamming attacks aim to disrupt the wireless communication between the IoT devices and the central server by emitting radio signals that interfere with legitimate signals. This can prevent the IoT devices from sending or receiving data, updates, or commands, and thus affect their functionality and performance [31]. For example, in [33], the authors propose a game-theoretic anti-jamming strategy for OFDM-based IoT systems, which enables an IoT controller to protect the IoT devices against a malicious radio jammer. Research done by Vlad Ionescu et al. [34], studies the impact of jamming attacks on NB-IoT devices using interference and shows that the battery lifetime can be reduced from 17 years to as low as four months. **Battery depletion attacks** attempt to drain the battery of the IoT devices by forcing them to perform unnecessary or excessive tasks, such as sending or receiving large amounts of data, running intensive computations, or scanning for signals. This can reduce the lifetime and availability of IoT devices and their AI models [31]. For example, Mottola et al. [35], investigate the energy attacks on battery-less IoT devices, which rely on ambient energy harvesting to power their operation. They show that by exerting limited control on the ambient supply of energy to the system, one can create situations of livelock, denial of service, and priority inversion. **Botnet attacks** exploit the vulnerabilities of the IoT devices, such as default credentials, unpatched software, or weak security, to infect them with malware that allows the attackers to remotely control them and form a network of compromised devices, also known as zombies [36]. The attackers then use the zombies to launch distributed DDoS attacks on target entities, such as websites, servers, or networks, by overwhelming them with traffic and disrupting their normal operations. IoT botnets can also be used to perform other malicious activities, such as stealing data, spreading malware, or executing commands [36]. **Adversarial attacks** manipulate the input data or the parameters of the AI models to cause them to produce incorrect or misleading outputs. For example, an attacker can craft a malicious packet that looks benign to the AI model but triggers a malicious payload on the IoT device. Alternatively, an attacker can modify the weights or the architecture of the AI model to degrade its accuracy or functionality [32]. These attacks pose serious threats to the security and reliability of IoT devices and their AI-based intrusion detection models. Therefore, it is imperative to develop robust and resilient AI models that can detect and mitigate these attacks and ensure the safety and functionality of IoT devices.

2.3. IDS's in IoT

Intrusion Detection Systems (IDS) play a crucial role in safeguarding IoT ecosystems by monitoring and analysing network traffic and system activities to detect and prevent malicious attacks [37]. These systems exhibit diverse classifications based on detection techniques, deployment strategies, and validation strategies [25]. Detection techniques include signature-based, anomaly-based, specification-based, and hybrid-based methods [37], while deployment strategies encompass host-based, network-based, and distributed-based approaches [25]. Validation strategies range from rule-based to machine learning-based and deep learning-based methodologies [25].

According to Elrawy et al.'s survey [23], the most widely used IDS for IoT include:

1. **SNORT**: A signature-based, network-based IDS employs predefined rules to identify patterns in network packets.
2. **ZEEK**: A network-based hybrid IDS that combines signature-based and anomaly-based techniques to analyse network events and detect attacks.
3. **Suricata**: A network-based, hybrid IDS utilising multi-threading and hardware acceleration to enhance the performance and scalability of SNORT.
4. **Kismet**: A network-based, anomaly-based IDS utilising wireless sniffing and packet analysis to detect rogue access points and unauthorised devices.
5. **OSSEC**: A signature-based host-based IDS monitoring file integrity, log files, and system processes of IoT devices to generate alerts for suspicious activities.

Despite the effectiveness of IDS, IoT encounters several challenges. The limited computing and storage capabilities of IoT devices create hurdles in the execution of complex IDS algorithms or the management of large data [23]. The diverse protocols, standards, and architectures among IoT devices complicate the development of a unified IDS solution [23]. Furthermore, the dynamic nature of IoT environments, characterised by frequent changes in network topology, traffic patterns, and device behaviour, makes establishing baselines for anomaly detection or updating signature databases a challenging task [23]. Moreover, as IoT devices handle sensitive data, privacy concerns arise, which require IDS solutions to ensure data confidentiality and integrity during intrusion detection [23]. Hence, methods which can mitigate these issues, such as collaborative learning approaches, have become more popular.

2.4. What is collaborative learning

Collaborative learning is a technique that involves multiple learners working together to achieve a common goal. In the context of IoT intrusion detection, collaborative learning can enhance the security and performance of IoT devices by sharing information and knowledge among devices [38]. It can be implemented in different ways, such as centralised, decentralised, or federated architectures, depending on the network topology, communication constraints, and privacy requirements, while also offering several benefits for IoT intrusion detection. These benefits include, but are not limited to:

- Improved accuracy and robustness of intrusion detection models by leveraging the diversity of data from different devices.
- Reduced computational and storage overhead of individual devices by offloading some tasks to other devices or a central server, depending on the selected topology.
- Enhanced scalability and adaptability of intrusion detection systems by enabling dynamic and flexible collaboration among devices.
- Preserving the privacy and security of data by using encryption, anonymisation, or differential privacy techniques, which are part of the fundamental attributes of Federated Learning.

However, collaborative learning also poses some challenges for IoT intrusion detection, such as:

- Coordinating the communication and synchronisation among devices with heterogeneous capabilities and resources. The communication cost can be influenced by several factors such as the network topology, heterogeneity of devices, privacy requirements, presence of malicious or faulty devices, and the need for reliability assessment. These factors necessitate careful design and optimisation of the communication protocols and system architecture.
- Dealing with the trade-off between collaboration and privacy, as sharing more information may improve detection performance while exposing sensitive data.
- Handling the malicious or faulty behaviour of some devices that may compromise the integrity or availability of the collaborative system.
- Evaluating the trustworthiness and reliability of different devices and their contributions to the collaborative system.

In conclusion, collaborative learning presents a promising approach to enhancing IoT intrusion detection. By allowing devices to work together and share information, it improves the accuracy of detection models, reduces computational load, and enhances scalability. It also offers flexibility in implementation, with options for centralised, decentralised, or federated architectures. Privacy and security are also considered, with techniques like encryption and anonymisation used to protect data. However, challenges exist, such as coordinating communication among diverse devices, balancing collaboration and privacy, managing potential device malfunctions, and assessing device reliability. Therefore, careful design and optimisation are crucial to fully harness the benefits of collaborative learning for IoT intrusion detection. As we shall explore in subsequent sections, literature has attempted to solve some of these issues by implementing different techniques and protocols.

2.5. Variations of collaborative learning

Various approaches in the field of machine learning, such as Centralised, Decentralised, and Federated Learning, have given rise to innovative techniques. These approaches each have their strengths and weaknesses, but when applied to resource-constrained devices, their limitations in covering the CIA triad (confidentiality, integrity, and availability) become [39].

One promising candidate for collaborative learning in this context is Deep Ensemble Learning (DEL). DEL combines multiple deep neural networks to enhance performance and robustness, handling heterogeneity and supporting collaborative learning scenarios [40]. However, current DEL implementations face challenges in feasibility for resource-constrained devices due to computational and memory requirements, motivating researchers to seek more efficient ensemble learning methods [40].

Cooperative Co-evolution, another approach, has been employed for anomaly detection, as seen in the work of A.N.M. Bazlur Rashid et al. [41]. Their ADUFS approach, integrating Feature Selection (FS) to improve accuracy and scalability, showed promising results in cybersecurity datasets. Despite its potential for collaborative learning, Cooperative Co-evolution faces challenges related to increased complexity and communication overhead among agents, as well as difficulties in effectively coordinating contributions to achieve an optimal global solution.

Moving to decentralised learning, Gossip learning has emerged as a popular solution aiming to replace Federated Learning. While its performance is comparable to Federated Learning approaches, deploying it on resource-constrained devices is impractical due to significant computational power requirements and a lack of available implementations [42].

In the realm of federated and decentralised intrusion detection systems, spin-offs such as Cluster-based [43], Swarm-based [44], and Multi-agent [45] aim to use multiple reference points for a holistic learning and detection approach. However, to the best of our knowledge, these approaches have not been implemented in resource-constrained devices. While some implementations can leverage a combination of different technologies [46], there is currently no implementation known to work specifically for resource-constrained devices. The following subsections provide more information on the identified methods.

2.5.1. Deep ensemble

Deep Ensembles are a method used in machine learning, particularly with neural networks. They are an effective way to increase accuracy and often match the performance of individual larger models. The method is simple to implement, readily parallelisable, requires minimal hyperparameter tuning and yields high-quality predictive uncertainty estimates [47]. Deep Ensembles work by ensembling or combining multiple neural networks. This approach has been shown to offer distinct benefits beyond predictive power, such as uncertainty quantification and robustness to dataset shift. Fig. 1 depicts the architecture of a typical deep ensemble. Each learner in the ensemble is trained on the same dataset and makes predictions. The final prediction is a combination of the predictions from all learners. Each learner makes a prediction, and these are combined, typically through averaging or voting, to yield a more accurate final prediction.

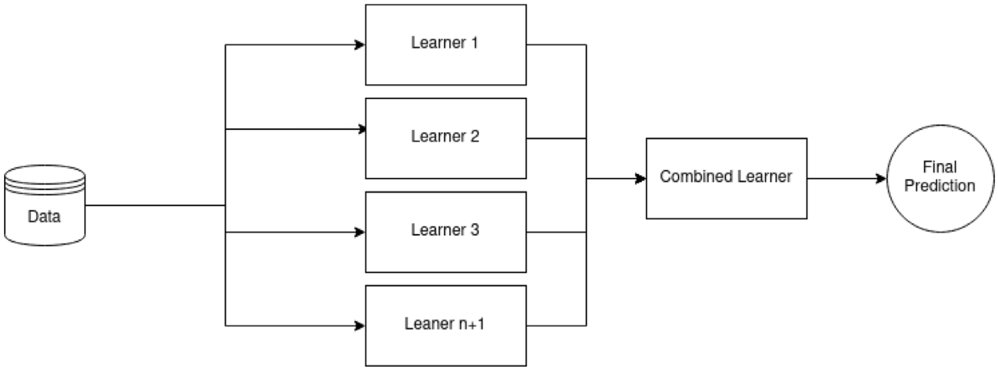

Fig. 1. Ensemble learning architecture.

Recent work suggests that these improvements can also be replicated by a single (but larger) neural network [48]. A key aspect of Deep Ensembles is their ability to handle out-of-distribution data while also being able to handle higher uncertainty on Out of Distribution (OOD) examples, benefiting many applications [49]. In recent years, there has been a shift towards using Deep Ensembles for low-data transfer learning, and they have been empirically shown to improve the accuracy, uncertainty, and out-of-distribution robustness of deep learning models. Research done by Selim Yılmaz et al. [50] explores the use of transfer learning to detect RPL-specific attacks, marking the first application of transfer learning in IoT security. The approach focuses on detecting different attacks and generating suitable intrusion detection algorithms for new devices. The performance of the proposed approach is evaluated based on the accuracy of the detection algorithms and their energy consumption. Experimental results show that the transfer learning-based approach, in most cases, outperforms the traditional learning approach, offering higher convergence speed and shorter training time. Although the approach targets RPL attacks, it can be adapted to other protocols or general attacks in IoT.

Despite their effectiveness, whether Deep Ensembles are necessary, given the choice between them and a single neural network with similar accuracy, is still a topic of ongoing research [47]. In addition, as depicted in Fig. 1, deep ensemble methods rely on the use of the same dataset across different devices. The collection of the data from IoT devices and its frequent update for continuous learning come in contrast with the need to reduce resource consumption in the IoT network and pose a threat to data privacy.

2.5.2. Gossip

Gossip Learning is a decentralised machine learning method that provides a unique approach to data analysis. Unlike traditional methods that rely on a central server for data aggregation, Gossip Learning allows for direct exchange and aggregation of models between nodes. In Gossip Learning, each device, or node, operates in cycles. During each cycle, a node combines its model parameters with those of its neighbours, updates the combined model with its data, and then shares the updated model with its neighbours [51]. This process allows for direct model aggregation between nodes. This eliminates the need for a central server, making the method highly scalable and robust as no existing infrastructure is required [52]. The performance of Gossip Learning shines when training data is distributed uniformly across nodes. In such cases, it has been found to outperform other methods like Federated Learning [53]. This makes it an attractive choice for scenarios where data is collected by edge devices, such as mobile phones [52]. However, the performance can be significantly affected by the distribution of the data. This is especially prevalent in cases where the IoT devices are in different geographical locations. If the training data are not uniformly distributed across nodes, it can lead to slow convergence of the protocol or even unfair bias in the produced models [54]. Gossip Learning's performance is reliant on communication speeds and bandwidth between devices, which could negatively impact the performance of the protocol [54] and result in lossy messages being sent between nodes, harming the model's performance. Despite these limitations, extensions have been introduced to mitigate some of these issues and improve their applicability to real-world scenarios [54].

2.5.3. Multi agent

Multi-agent Intrusion Detection Systems use multiple autonomous agents to detect anomalies and potential threats in a network [55]. These agents can be trained using various machine learning algorithms, including deep learning [56]. In one approach, a deep learning-based multi-agent system for intrusion detection combines the desired features of a multi-agent system approach with the precision of deep learning algorithms [56]. Autonomous, intelligent, and adaptive agents are created that implement algorithms like an autoencoder, multilayer perceptron, and k-nearest neighbours. In another approach, an Adaptive Rule-Based Multiagent Intrusion Detection System (ARMA-IDS) is proposed to detect anomalies in real-time datasets [55]. A feedback loop provides the necessary update of attacks in the database, leading to an improvement in detection accuracy. Ensuring the technology is correctly installed and optimised can be difficult due to budgetary and monitoring constraints. Managing the vast quantity of alerts generated by intrusion detection can be a significant burden for internal teams [57]. IDS alerts consist of base-level security information, which, when viewed in isolation, may mean very little. Advances in computer-agent representations, decision-making, reasoning, and learning methods pose technical challenges. As AI systems become more interconnected, ethical challenges arise in ensuring these systems function effectively and safely for people and societies [57]. Multi-agent IDSs for resource-constrained devices, such as those in IoT environments, face several challenges. They require significant computational power and storage [58], which may not be readily available in such devices. They are resource-intensive, especially network-based IDS that provide comprehensive visibility. While some approaches like lightweight encryption and authentication mechanisms can help overcome these constraints, they require regular updates [59]. These systems can quickly identify known threats but may struggle to detect new ones. Anomaly-based IDS can detect novel attacks but may generate more false positives [58].

2.5.4. Swarm

Swarm Intelligence (SI) based IDS works by using the principles of SI to detect anomalies and potential threats in a network [51]. In one approach, an Artificial Bee Colony (ABC) algorithm is used to train a Random Neural Network (RNN) based system. The ABC algorithm mimics the food-foraging behaviour of honey bee swarms, and it is used to optimise the RNN. This system can protect sensitive information and detect novel cyber-attacks with an accuracy of 91.65% [51]. In another approach, a Whale optimisation Algorithm (WOA), inspired by the behaviour of humpback whales, is used for feature selection in the IDS model [60]. This method reduces irrelevancy in the IDS model by choosing the most relevant features from the data. The model trained with these selected features produces an accuracy of 80%, which is 8% greater than the accuracy produced by the original dataset [60]. Swarm-based intrusion detection systems (IDS), despite their advantages, have several drawbacks. They suffer from partial optimism, which can affect their speed and direction [61]. They also face challenges with scattering and optimisation, and may not be suitable for use in the energy field. Like other IDS, they require constant updating of the signature database, which can be resource-intensive [61]. Furthermore, identifying zero-day attacks can be difficult as these attacks will not have a matching signature in the database until it is retrieved and saved [61]. These systems are essential for tackling cyberattacks reliably, especially in environments like cloud infrastructure and IoT devices, where network intrusion and cyberattacks are severe concerns [51,60,61].

2.5.5. *Split learning*

Split Learning is a novel approach to machine learning that aims to address the challenges of data privacy and computational efficiency in distributed systems [62]. It does this by dividing the learning process between a device-side model and a server-side model. This division is made at a predetermined layer in the model, known as the cut layer [63]. The device-side model processes the initial layers of the model, reducing the dimensionality of the input data and extracting preliminary features. This processed data, which is much smaller in size than the original raw data, is then sent to the server-side model for further processing. The server-side model completes the remaining layers of the model, using its superior computational resources to handle the more complex aspects of the learning process [63]. This approach has several advantages. By allowing the training of machine learning models without the need to share raw data, it preserves data privacy [62,63]. This is particularly beneficial in sectors such as healthcare and finance, where data privacy is of paramount importance [62,63]. By offloading the computationally intensive parts of the model to a server, Split Learning enables resource-constrained devices to participate in the training of complex models [63]. However, the performance of the model depends on the choice of the cut layer, and finding the optimal cut layer is a non-trivial task [62]. Furthermore, the current research on Split Learning is mainly focused on convolutional neural networks. More studies are needed to explore its applicability to other types of data, such as voice, image, and string data [63]. In the context of intrusion detection systems, Split Learning can be used to train models that can detect malicious activities in a network [63]. However, these systems can generate numerous false alarms, especially in networks with many users [63]. They may also struggle to detect new types of attacks [63]. The accuracy of the model is often inversely proportional to its interpretability; the more accurate the model, the more complex it becomes, and the harder it is to interpret its predictions [62]. Moreover, the optimisation algorithms used in these systems have their limitations [62,63]. They may not work well if applied alone, without tuning the parameters of the classifiers [63]. There is also a risk of data theft during the transmission of data between the client and the server. If an attacker knows the structure of the client's model and has access to some input/output data, they can potentially reconstruct the model and calculate the original data [62,63]. In conclusion, while Split Learning presents a promising solution to the challenges of data privacy and computational efficiency in distributed machine learning, more research is needed to address its limitations and potential security risks [62,63].

2.5.6. *Federated learning*

Federated learning is a machine learning technique that enables collaborative model training across decentralised devices or servers, each with its local data samples. A central server sends an initial model to participating devices. Each device independently updates the model using its local data. Devices send back only the model updates (not raw data) to the central server, which then aggregates these updates to refine the global model. This iterative process allows continuous model improvement, with the model learning from diverse data points without compromising data privacy [64]. The advantages of FL are significant. First, it ensures data privacy by keeping data local, eliminating the need for centralising sensitive information. Second, it reduces data transfer between clients and servers, which is essential in bandwidth-constrained environments or where data transfer costs matter [64]. Third, FL leads to more robust and accurate models by leveraging diverse data sources without centralisation. Finally, it ensures compliance with privacy laws and mitigates the risks of data breaches [65]. However, FL also has its challenges. Frequent communication between devices and the central server introduces communication overhead, impacting latency and network usage. Additionally, devices may have varying data quality and distributions, posing challenges for model convergence. Moreover, resource-constrained devices, such as edge devices (e.g. IoT devices), may lack computational resources, making on-device training challenging. For resource-constrained devices, FL offers several benefits. It promotes energy efficiency by conserving energy through on-device training [65]. It reduces latency by shifting computation closer to data sources, ideal for real-time applications. It enhances privacy and security by allowing localised model updates that protect sensitive data [65]. In summary, FL empowers collaborative learning while safeguarding privacy and accommodating devices with limited resources. Its impact extends across various domains, making it a powerful paradigm in the evolving landscape of machine learning.

2.6. *Summary of methods*

Table 1 summarises the various Collaborative AI-based IDSs. Some methods suffer from similar issues, such as bandwidth limitations, which is a big factor in how well they operate. Whilst there is not a one-size-fits-all collaborative defence mechanism to guard against cyber attacks, the benefits of using collaborative methods often surpass the drawbacks, especially when compared to a standard centralised approach. This is particularly true for situations where devices are required to operate independently, yet contribute to the overall network security. An example of this would be IoT devices spread across various geolocations. These devices need to function autonomously but also play a part in enhancing the network's security.

3. Literature review

Over the years, several review articles have examined the use of collaborative learning in intrusion detection systems (IDS) for the Internet of Things (IoT). For instance, Aitor Belenguer et al. [64] studied federated learning IDS in IoT, acknowledging its limitations but primarily focusing on large-scale, powerful IoT systems. Similarly, papers by Jose L. Hernandez-Ramos et al. [66] and Shaashwat Agrawal et al. [67] mostly discussed the pros and cons of implementing IDS in IoT, but they did not consider resource-constrained devices. Elrawy Awad Hamed et al. [23] also highlighted the advantages of using such collaborative approaches in IoT. From these reviews, it is clear that collaborative learning approaches are beneficial for IoT environments and are an effective way to address the reliability and stability issues associated with centralised approaches. However, many of these reviews have focused on large-scale, powerful hardware capable of running complex models. The mention of resource-constrained IoT in these reviews is often vague, typically referring to devices like Raspberry Pi. However, it has been demonstrated multiple times that such devices can run complex machine learning models for IDS [68–70]. Our paper sets itself apart from these reviews by exploring the feasibility of implementing IDS on resource-constrained devices, which we define as those capable of running real-time operating systems or on bare metal. These devices are ubiquitous and not only vulnerable to cyber threats but can also be the weakest link in network security [71].

Table 1
Advantages and disadvantages of collaborative AI-based IDSs.

System	Advantages	Disadvantages
Deep ensemble	High detection rate for known attacks	Low detection rate for zero-day attacks
Gossip	Supports collaboration between entities that cannot share their dataset for confidential or other reasons	Bandwidth limitations heavily impact performance
Federated	Supports collaboration between entities that cannot share their dataset for confidential or other reasons	Can cause network congestion when implemented in large-scale networks
Swarm	Constantly monitor a given computer network for invasion or abnormal activity	Can generate a large number of false positives
Multi-agent	Mobile agents in IDS can detect attacks that are hidden from an ordinary firewall using versatile technology	Can generate a large number of false positives.
Split	Constantly monitors a given computer network for invasion or abnormal activity	Split Learning might perform slower than federated learning due to relay-based training across multiple clients

3.1. Operational challenges of collaborative learning in resource-constrained IoT environments

Collaborative Learning methods are poised to revolutionise IoT, but they bring forth several formidable challenges. A major concern lies in the intricacies of communication between IoT devices, leading to increased network traffic, potential bottlenecks, and the risk of packet loss [72]. Researchers are actively working to address this issue by devising strategies to reduce packet sizes and optimise translation intervals, thus minimising network overhead [73]. In Collaborative Learning systems, there is a mix of devices with different computational, storage, and communication capacities, leading to inherent diversity within IoT environments [74,75]. The data distributed among these devices may also exhibit disparities. Researchers have put forward innovative solutions, including the integration of proximal terms, to accommodate this data heterogeneity while mitigating any domain shift concerns [76,77]. Proximal terms in optimisation refer to functions or operators that measure the distance between a point and a set, typically used to encourage certain properties in solutions, especially in convex optimisation problems. Furthermore, Collaborative Learning for IoT presents unique technical challenges stemming from hardware and software diversity, intellectual property rights across geolocations [78–81], and network connectivity issues, such as latency spikes and dropouts. Implementations such as fibre optic networks for latency reduction [82], the introduction of quality of service to manage network congestion, and the use of various connectivity methods, such as concurrent Wi-Fi and 5G [83], offer robust solutions to this issue, ensuring the reliable and efficient operation of Collaborative Learning within IoT ecosystems.

3.1.1. Hyper parameter tuning

Collaborative learning introduces intricate challenges in hyperparameter tuning, which are exacerbated by the inherent variability of the device [84]. Models trained on diverse devices require unique hyperparameters to achieve optimal results due to this device heterogeneity. To address this issue, researchers have explored solutions such as employing online versions of the REINFORCE algorithm to mitigate training losses and improve accuracy [85]. The REINFORCE algorithm is a type of reinforcement learning algorithm that updates the neural network weights after every trial. The algorithm is based on a Monte Carlo version of a policy gradient algorithm, which means it uses random paths to approximate the gradient of the expected reward with respect to the policy parameters. The algorithm tries to achieve the highest expected reward by boosting the likelihood of actions that result in greater rewards [85].

Compression techniques enable more frequent model synchronisation, facilitating hyperparameter tuning by mitigating the effects of heterogeneity. Another approach involves mixing data between devices to create a more homogeneous data distribution, although it raises privacy concerns [74]. Effective communication between devices is a central challenge in Collaborative Learning. Multiple methods exist which try to combat this issue, such as SOLO [86], FedAvg [87], and FedProx [88], which performed below 70%. They improve communication efficiency by allowing local updates and low participation, minimising bandwidth use. Collaborative Learning has the potential to match or surpass centralised models when configured appropriately with algorithm-specific considerations [84].

3.1.2. Privacy

Privacy concerns in collaborative learning have long been a significant issue, with traditional methods often involving the centralisation of data, raising questions about the security and confidentiality of sensitive information. Over the last decade, stricter data privacy regulations have been implemented, prompting manufacturers and developers to seek more secure, privacy-oriented data gathering and processing approaches. Federated Learning (FL) emerged as a response to these concerns. FL operates by conducting model training directly on user devices, thereby avoiding the need to centralise raw data. Instead, only model updates or aggregated results are shared with a central server or among devices, preserving the anonymity of individual user data. This decentralised approach inherently safeguards user privacy, as highlighted by recent studies [78–80]. Moreover, FL offers additional benefits for privacy in learning methods. For instance, it facilitates the implementation of encryption techniques without compromising data security. By encrypting data and performing computations locally, FL ensures that sensitive information remains confidential, even from the central server or master node. Notably, encryption keys are not shared with the central entity, further enhancing the security of the process [89]. However, despite these advantages, FL alone may not fully mitigate all privacy risks, particularly concerning data inference attacks. These attacks exploit statistical patterns in aggregated data to infer sensitive information about individual users. To address this challenge, complementary techniques such as differential privacy have been introduced. Differential privacy introduces noise or randomness to the data during aggregation, making it more challenging for adversaries to extract meaningful insights. By combining FL with techniques like differential privacy, developers can enhance the overall privacy protection of collaborative learning systems [90–92]. In summary, while FL represents a significant advancement in preserving user privacy in collaborative learning, it is essential to recognise its limitations and supplement it with additional privacy-enhancing measures like differential privacy to ensure comprehensive data protection.

3.1.3. Dealing with biases

Biases in ML datasets can skew results, resulting in less-than-ideal models. Therefore, careful manipulation of these biases need to be made in a way which would benefit the model [93]. Since CL deals with multiple data streams from different and unique devices, dealing with unique data is one of the fundamental challenges. Hence, the problem of model memorisation has become a challenging research topic with very promising results [94,95]. If model memorisation is not addressed, then there is the possibility that an attacker exploits the model by feeding it arbitrary data [96]. Limiting the amount of data a device can contribute to the network and adding noise to the data to obscure it allows the algorithm to deal with model memorisation. This method is called differential privacy [97]. Implementing DP in deep learning can increase software complexity and affect training efficiency [98]. However, it is possible to train deep neural networks with non-convex objectives in a modest privacy budget without significantly increasing complexity or reducing efficiency [98]. The main trade-off when using DP is between privacy and accuracy [99]. Adding noise to protect privacy can affect the accuracy of the model [98]. The challenge is to find a balance where privacy is protected without significantly compromising the model's performance [98].

3.1.4. Performance

Collaborative learning methods face significant challenges, including data heterogeneity [87]. Work done by Liangqiong Qu et al. [100] introduces a generative replay strategy, employing a generative adversarial network (GAN) to craft a synthetic dataset that encapsulates the collective knowledge from all participating institutions [100]. This synthetic dataset is subsequently utilised to train a primary model for the desired task, such as classification or regression. According to the paper, this approach offers superior handling of data heterogeneity compared to existing methods, while also lowering the communication and privacy expenses associated with collaborative learning. The paper evaluates this method using two medical imaging datasets: one for diabetic retinopathy classification and another for bone age prediction. It reports significant enhancements over state-of-the-art methods for both datasets [100]. The traditional approach, exemplified by fedAVG, aggregates data from all participants, averaging their contributions [101]. However, fedAVG eliminates lag nodes, leading to the omission of valuable device-specific data [101]. Furthermore, it struggles when confronted with non-identical data distributions across devices, hampering theoretical analysis and result reliability [102]. Innovative solutions like FedProx introduce proximal terms to stabilise results and improve overall performance [88]. Split Learning reduces communication overhead by splitting model update computation between devices and a central server [103]. Devices handle the initial layer of the model and the server handles the rest [103]. Communication costs in FL depend on factors such as communication rounds, utilisation, learning rates, and data isolation strategies [104]. To enhance user privacy and security, adaptive gradient descent and differential privacy mechanisms are employed [104]. This approach effectively facilitates multi-party collaborative modelling, improving efficiency and performance in constrained communication scenarios [104].

3.1.5. Scalability

Collaborative learning allows for a scalable approach that enables the training of machine learning models on distributed devices without relying solely on central data transmission [105]. It enables local training on each device, transmitting only updated model parameters to a central server for aggregation [105]. This distributed method is particularly beneficial for large datasets or sensitive, non-transmittable data, reducing communication overhead and central server workload [106]. Collaborative Learning can support parallelised training on multiple devices, accelerating model convergence and reducing training time. It adapts to heterogeneous devices with varied resources, ensuring flexibility and scalability [106]. Solutions like FEDn facilitate seamless transitions from local development to horizontally scalable distributed deployments [106]. This distributed, parallelised approach is ideal for faster model convergence and reduced training time in a multi-node, large-scale Collaborative Learning network. Heterogeneous devices with varying resources make it adaptable and flexible for scaling to numerous devices. The FEDless solution [107] extends this scalability to various Function as a Service (FaaS) providers, offering security features and Differential Privacy across a diverse range of platforms, including the cloud, data centres, and edge devices. A summary of the challenges is outlined in Table 2.

Table 2
Challenges of collaborative learning for resource-constrained IoT.

Challenge	Description
Communication overhead	High traffic between IoT devices may cause bottlenecks and packet loss. Efforts focus on minimising network overhead.
Heterogeneity	Varied capabilities of IoT devices lead to data heterogeneity. Proximal terms are integrated to address disparities.
Technical challenges	Hardware and software complexities pose technical hurdles. Legal issues across geolocations also impede progress.
Privacy concerns	User privacy is paramount. Federated Learning (FL) and differential privacy techniques safeguard sensitive data.
Dealing with biases	Biases in ML datasets can skew results. Strategies include limiting device contributions and adding noise to data.
Performance	Data heterogeneity and communication efficiency affect performance. Innovations like FedProx and split learning improve efficiency.
Scalability	Scalable training of ML models on distributed IoT devices is vital. Solutions like FEDn ensure seamless transitions to distributed deployments.

Collaborative learning in IoT environments poses various challenges. Communication overhead can lead to significant traffic between devices, potentially causing bottlenecks and packet loss. Researchers focus on optimising communication protocols to minimise this issue. The heterogeneity of IoT devices, with variations in computational and communication capabilities, results in data heterogeneity. Addressing these differences involves integrating proximal terms. Technical challenges encompass hardware, software complexities, and legal concerns across different geolocations. Preserving user privacy is crucial, prompting the use of techniques like Federated Learning and differential privacy. Biases in ML datasets require strategies such as limiting device contributions and adding noise to the data. Performance issues arise from data heterogeneity and communication efficiency, addressed by innovative solutions like FedProx and Split Learning. Ensuring scalability for scalable training of ML models on distributed IoT devices is essential, facilitated by solutions like FEDn for seamless transitions to distributed deployments. In the next Section 4, we shall also look at the Cybersecurity challenges which are faced by Collaborative learning systems and how some of these have been or could be addressed.

3.2. Performance comparison between centralised and collaborative IDS

A direct comparison of accuracy and F-score values for centralised and collaborative IoT NIDS reveals that the optimal approach is context-dependent. Table 3 summarises the performance metrics from various academic studies.

The table presents a comparison of accuracy and F-score metrics for both centralised and collaborative IDS in the context of the IoT, drawing from various academic studies. The findings indicate a mixed landscape. Some studies, like [108,109], show that collaborative IDS achieve higher accuracy and F-scores on datasets such as NSL-KDD and NSW-NB15 compared to centralised approaches. For instance, in [109], the collaborative method achieved 99.89% accuracy on NSLKDD, while the non-cooperative (centralised/local) method reached 97.79%. Similarly, on NSW-NB15, the collaborative approach had 97.72% accuracy compared to 94.39% for the non-cooperative one. Furthermore, [108] explicitly states that their collaborative FedPPID model demonstrated superior performance in both accuracy and F1-score compared to traditional centralised IDS across NSL-KDD, BoT-IoT, and UNSW-NB15 datasets.

Table 3
Performance comparison of centralised, collaborative, and non-cooperative IoT NIDS.

Study (Ref.)	NIDS type	Dataset	Accuracy (%)	F-score
[108]	Collaborative	NSL-KDD	92.78	0.927
	Collaborative	BoT-IoT	91.47	0.914
	Collaborative	UNSW-NB15	92.05	0.921
	Centralised	NSL-KDD	88.45	–
	Centralised	BoT-IoT	85.12	–
	Centralised	UNSW-NB15	86.73	–
[109]	Collaborative	NSL-KDD	99.89	0.999
	Collaborative	UNSW-NB15	97.72	0.976
	Non-cooperative	NSL-KDD	97.79	0.978
	Non-cooperative	UNSW-NB15	94.39	0.940
	Centralised	BoT-IoT	99.20	–
[110,111]	Collaborative	ToN-IoT (Binary)	97.59	96.6
	Collaborative	ToN-IoT (Multi)	97.86	0.979
	Collaborative	CICIDS2017 (Bin)	98.20	0.955
	Collaborative	CICIDS2017 (Mul)	98.15	0.982
	Centralised	ToN-IoT (Binary)	98.40	0.977
	Centralised	ToN-IoT (Multi)	99.40	0.994
	Centralised	CICIDS2017 (Bin)	98.40	0.961
	Centralised	CICIDS2017 (Mul)	98.20	0.982
[112]	Centralised	IoT-23	99.15	–
	Centralised	ToN-IoT	99.78	–
	Centralised	UNSW-NB15	99.88	–
	Centralised	Edge-IIoT	99.45	–
[113]	Centralised	NSL-KDD	94.09–98.84	–
	Centralised	CICIDS2017	94.09–98.84	–
	Centralised	UNSW-NB15	94.09–98.84	–
[114]	Centralised	ToN-IoT	99.98	0.9998

However, other studies, such as [110], suggest that centralised models can achieve comparable or even slightly better performance on datasets like ToN_IoT and CICIDS2017. The centralised model showed slightly higher accuracy in binary and multiclass classification on the ToN_IoT dataset. Despite these instances, the collaborative Federated Learning model in [111] still maintained high performance, generally above 97% accuracy.

3.2.1. Accuracy advantages of collaborative approaches

Collaborative IoT NIDS often achieve superior accuracy compared to centralised systems due to their ability to share threat intelligence across distributed nodes, enabling comprehensive detection of complex attacks such as DDoS through coordinated insights. Their decentralised architecture enhances resilience, as the failure of individual nodes minimally impacts overall accuracy. Privacy-preserving techniques such as Federated Learning further strengthen collaborative systems by training models on decentralised data without compromising sensitive information, with studies like [109] demonstrating competitive or even higher accuracy than centralised approaches. Additionally, collective intelligence from multiple nodes allows these systems to identify sophisticated attack patterns that isolated systems might overlook.

3.2.2. Accuracy variations in centralised systems

Even when using identical datasets, centralised NIDS exhibit accuracy variations due to differences in algorithm selection (e.g., decision trees vs. neural networks), hyperparameter tuning, and feature engineering strategies. Data preprocessing methods such as normalisation or handling missing values alongside training-validation splits (e.g., 80–20 vs. k-fold) and class imbalance mitigation techniques (e.g. SMOTE)further contribute to discrepancies. Implementation nuances, including software libraries or code optimisations, and varying generalisation capabilities across models also play critical roles in performance outcomes.

3.2.3. Collaborative approaches in resource constrained IoT

The Table 4 presents a selection of academic papers that demonstrate the feasibility of implementing collaborative and on-device machine learning on resource-constrained devices. These studies showcase the successful application of techniques like Split Learning and Federated Learning on various platforms, including FPGA-based IoT devices and microcontrollers such as the Arduino Nano 33 BLE Sense and ESP32. The applications range from water quality monitoring and keyword spotting to enterprise management and healthcare, indicating the versatility of these approaches. The fact that researchers have been able to deploy and train machine learning models, even in a collaborative manner, on devices with limited processing power, memory, and energy budgets suggests a promising future for embedding intelligence directly into resource-constrained IoT devices. This opens up possibilities for enhanced privacy, reduced latency, and more efficient use of network resources in a wide array of applications.

4. Cybersecurity challenges and protection approaches in collaborative systems

Collaborative Learning faces several cybersecurity challenges that need to be addressed to ensure the security and privacy of the data used in the training process. As we shall explore in subsequent subsections, some of these include **data poisoning**, **model poisoning**, and **malicious participants**. Addressing these challenges requires a combination of technical solutions, such as encryption, and authentication mechanisms, as well as robust governance frameworks to ensure compliance with data privacy regulations and ethical guidelines.

4.1. Relevance of collaborative models

One of the hurdles in collaborative learning, especially in cybersecurity applications, is the ever-changing landscape of cyber threats. As attackers constantly devise new strategies and techniques, the effectiveness of data sets and models established at a certain point may diminish over time [119].

For example, a model trained to identify a specific type of malware may fail to recognise a new variant that exhibits slightly different

Table 4
Collaborative/On-device ML for resource-constrained devices.

Paper description	Method	Utilisation in resource-constrained devices	Specific devices
Efficient split learning LSTM Models for FPGA-based edge IoT devices [115]	Split learning (LSTM)	Deployment of split learning with compressed LSTM models on FPGA edge platforms. Balances performance with limited processing, memory, and energy.	FPGA-based edge IoT platform (Basys 3 dev board w/ Artix-7 FPGA)
Accuracy-guaranteed collaborative DNN inference in industrial IoT via deep reinforcement learning [116]	Collaborative inference (DNN)	Minimises service delay while guaranteeing accuracy for DNN inference in industrial IoT-edge networks.	Industrial IoT devices, edge networks
TinyFL: On-Device training, Communication and aggregation on a microcontroller [117]	Federated learning (TinyFL)	Resource-efficient FL framework for microcontrollers. Optimises training, communication, and aggregation.	Microcontrollers
Secure android IoT mobile and collaborative ML for enterprise management [118]	Collaborative ML (ECDH/RSA)	Enhances security in android IoT devices via cryptographic protocols and collaborative ML algorithms.	Android IoT mobile devices

behaviour [119]. Likewise, a model trained on network traffic patterns may not detect an intrusion if the attacker employs a unique method to conceal their activity [120].

Given this, it is imperative for models used in collaborative learning for cybersecurity to be regularly updated and adapted to keep pace with the evolving threat environment [120]. This process involves the frequent retraining of models with fresh data, integrating the most recent threat intelligence, and possibly employing methods like transfer learning to leverage knowledge from one domain to another [120]. This ensures that models stay relevant and effective in the face of new and emerging cyber threats.

In today's era of machine learning and artificial intelligence, the security and integrity of these systems are paramount. FL models, which are becoming increasingly popular for their ability to utilise decentralised data, also introduce new vulnerabilities. These include the risk of **model poisoning**, where the trained FL model is maliciously altered to produce incorrect outputs, and **data poisoning**, where the training data is corrupted, leading to biased or inaccurate results. Additionally, safeguarding against external threats such as **rogue devices**, which can compromise the entire system by introducing arbitrary data, and **malicious participants**, who may manipulate the model's parameters or inject poisoned data, is critical. Understanding and addressing these distinct threats is essential to developing effective strategies for securing FL models and networks, thereby ensuring their reliability and trustworthiness in critical applications.

4.2. Data poisoning

CL models can be vulnerable to data poisoning attacks, where malicious participants can inject poisoned data into the model training set to manipulate the results [121]. Data poisoning attacks in CL can be particularly challenging to detect and prevent because the data is distributed across multiple participants and is often encrypted or anonymised. This makes it difficult to identify malicious data that has been injected into the training set [121]. One common approach to prevent data poisoning attacks is to use a validation dataset that is kept separate from the training dataset. This validation dataset can be used to detect anomalies or inconsistencies in the training data that may indicate the presence of poisoned data [122]. Another approach is to use cryptographic techniques such as homomorphic encryption or secure multi-party computation to allow participants to share their data without revealing their private information. These techniques can help prevent malicious participants from manipulating the data or model parameters [123]. Overall, preventing data poisoning attacks requires a combination of technical solutions and robust governance frameworks to ensure compliance with data privacy regulations and ethical guidelines. It is important to implement secure communication protocols, authentication mechanisms, and data protection measures to ensure the security and privacy of the data used in the CL training process.

4.3. Model poisoning

Attackers can also manipulate the model's parameters during the training phase, leading to the creation of a malicious model that can generate incorrect results [121]. Model poisoning can be difficult to detect and prevent because it does not involve the injection of false data, but rather the manipulation of the model's parameters. Malicious participants can alter the model in subtle ways that are not immediately apparent, leading to a model that appears to be accurate but produces incorrect results [124,125].

Lingchen Zhao et al. [126] proposed a novel defence scheme aimed at detecting anomalous updates in CL scenarios, where multiple clients collectively train a model without sharing their data. The scheme employs client-side cross-validation to assess each update using other clients' local data and adjusts update weights accordingly. Additionally, the scheme ensures client-level privacy protection by integrating differential privacy into its design. The paper demonstrates the scheme's robustness against two common poisoning attacks in both IID and non-IID settings.

One common approach to prevent model poisoning attacks in FL is to use a defence mechanism called "robust aggregation". In this approach, the server aggregates the model updates received from the participants while excluding any updates that are identified as outliers. This can help to detect and prevent malicious updates from affecting the model's parameters [127].

Another approach is to use cryptographic techniques such as secure multi-party computation or differential privacy, as mentioned throughout the paper, to prevent malicious participants from manipulating the model's parameters. These techniques can help to ensure that the participants' contributions are combined in a way that is resistant to tampering [99].

A notable toolset for adversarial attack mitigation, which is being actively maintained, is the Adversarial Robustness Toolbox by Trust-AI [128]. ART offers developers and researchers tools to evaluate and defend Machine Learning models and applications against adversarial threats like Evasion, Poisoning, Extraction, and Inference.

4.4. Malicious participants

CL models require the participation of multiple parties, and one malicious participant can jeopardise the entire process by manipulating

Table 5
Summary of cybersecurity attacks and defence mechanisms in collaborative learning.

Attack	Defence mechanisms
Data poisoning	• Use a validation dataset separate from the training dataset to detect anomalies. • Use cryptographic techniques like homomorphic encryption or secure multi-party computation to share data without revealing private information. • Implement secure communication protocols, authentication mechanisms, and data protection measures.
Model poisoning	• Use robust aggregation techniques (e.g., excluding outlier updates) to prevent malicious updates from affecting model parameters. • Employ cryptographic techniques such as secure multi-party computation or differential privacy to ensure tamper-resistant combination of participant contributions. • Detect anomalous updates using client-side cross-validation and adjust update weights. Integrate differential privacy for client-level privacy.
Malicious participants	• (Can manifest as data poisoning or model poisoning). Defence mechanisms for those attacks apply. • Rogue device detection using spectral anomaly detection models to differentiate between benign and malicious clients.
Rogue devices	• Collaborative learning can be trained to detect these attacks using a spectral anomaly detection model.

the data or the model parameters [129,130]. There are several ways that malicious participants can compromise the FL model. For example, they may intentionally inject poisoned data into the training set, manipulate the model's parameters during the training phase, or refuse to participate in the training process altogether. These actions can lead to inaccurate or biased results, undermining the accuracy and reliability of the FL model [131].

4.5. Rogue devices

When building a network, in many cases, the assumption is made that all devices on the inside of our network are trusted and do not pose a threat. Therefore, rogue devices are one of the most devastating kinds of threats that network administrators must deal with. Therefore, having a method of detecting these kinds of attacks is crucial to any secure network. Federated networks are still prone to rogue device attacks since there is always a possibility that a threat actor might take control of a node and try to inject arbitrary data into the network to manipulate the Federated Learning model [132]. However, research has shown that CL can be trained to detect these attacks by using a spectral anomaly detection model, allowing for the model to differentiate between benign and malicious clients [133].

4.6. Conclusion

In Table 5 we have summarised the previously explained challenges and protective measures.

In conclusion, collaborative learning methods face various cybersecurity challenges, including communication overhead, heterogeneity of devices, technical hurdles, privacy concerns, biases in datasets, and performance issues. While traditional centralised approaches struggle to address these challenges effectively, FL emerges as a promising solution for resource-constrained devices in the IoT ecosystem. FL's inherent design addresses privacy concerns by allowing model training directly on user devices, reducing the need for data transmission and ensuring sensitive data remains on the device. This unique combination of efficiency and privacy preservation makes Federated Learning a preferred choice for IoT devices in collaborative learning scenarios.

In the following Section 5, we shall explore into federated aggregation methods and explore how they can be applied in resource-constrained IoT environments. Additionally, we will examine the potential applications of Federated Learning in these scenarios, further illustrating its advantages over traditional centralised approaches.

5. Federated aggregation methods

Federated aggregation methods are techniques used in distributed machine learning to aggregate the model updates from multiple devices or nodes without directly transmitting the raw data. In a Federated Learning setting, each device performs local training using its data and sends the updated model weights or gradients to a central server for aggregation. The central server then combines these updates to create a new global model, which is sent back to the devices for further local training. The federated aggregation methods aim to enable efficient collaborative learning while maintaining data privacy and security, making it particularly useful in applications such as IoT. While not an extensive list these are some of the most used ones:

- **Federated Averaging**: Devices train models locally, send weights to a central server, which averages and updates the global model weights [134].
- **Federated SGD**: Similar to Federated Averaging, but devices send gradients instead of weights, and the server aggregates these gradients for model updates [134].
- **Federated Proximal Gradient Descent (PGD)**: Extends Federated SGD by adding a regularisation term on the server to keep global model weights near local ones [135].
- **Federated Boosting**: Combines models trained on individual devices to strengthen performance by creating a unified, more accurate model [136].
- **Bayesian Federated Aggregation**: Uses Bayesian inference in aggregation to improve global model accuracy and robustness, integrating device-specific knowledge and uncertainty estimation [137].
- **Federated Partial Aggregation (FPA)**: Allows devices to send only partial updates to the server, which aggregates them to create the global model [138].

A summary of the advantages and disadvantages of each is depicted in Table 6.

5.1. Federated learning methods for resource-constrained devices

The Federated Averaging method is often considered the best option for resource-constrained devices. It is simple to implement, computationally efficient and can handle numerous devices and datasets. Additionally, Federated Averaging does not require as much computation and communication resources compared to other Federated Learning algorithms. This makes it well-suited for devices with limited computational power or constrained network connectivity. However, the best method may vary depending on the specific constraints and requirements of the device or system. While Federated Averaging is a preferred choice for resource-constrained devices, the other Federated Learning methods may have limitations that make them less suitable for such devices. Federated Stochastic Gradient Descent (FSGD) and Federated Proximal Gradient Descent methods may require more computation and communication resources than Federated Averaging. This can pose challenges for devices with limited computational capabilities or limited network bandwidth. These methods are more computationally intensive and may suffer from slower convergence if the devices have low computational power. Similarly, Federated Boosting, while capable of handling heterogeneous devices and data, may require more computational and communication resources than Federated Averaging [139]. Its complexity in implementation can make

Table 6
Comparison of federated learning algorithms.

Algorithm	Pros	Cons
Federated averaging	• Simple and easy to implement. • Computationally efficient. • Handles numerous devices and datasets.	• Assumes that the local models perform similarly, which may not be the case in heterogeneous environments. • Can suffer from slow convergence if there is a significant difference in the local models.
Federated Stochastic Gradient Descent (FSGD)	• Can handle non-i.i.d. (independent and identically distributed) data. • Achieves faster convergence compared to Federated Averaging.	• Requires more computation and communication resources compared to Federated Averaging. • May suffer from slower convergence if the devices have low computation power.
Federated proximal gradient descent	• Can handle non-convex optimisation problems. • Can achieve faster convergence compared to federated averaging.	• Requires more computation and communication resources compared to Federated Averaging. • May suffer from slower convergence if the devices have low computation power.
Federated boosting	• Can handle heterogeneous devices and data. • Can improve the overall model performance by focusing on the weak learners.	• Requires more computation and communication resources compared to Federated Averaging. • Maybe more complex to implement compared to other federated aggregation methods.
Bayesian federated aggregation	• Enables principled uncertainty estimation and probability-based modelling. • Provides a more accurate estimation of the global model by incorporating uncertainty information. • Allows for robustness in handling diverse device-specific knowledge and uncertainty.	• Requires more computation and communication resources compared to Federated Averaging.
Federated Partial Aggregation (FedPA)	• Reduces communication between devices and the server. • Improves device privacy by not sending entire model updates to the server. • Improves accuracy by allowing devices with more data to contribute more to the aggregation process.	• Reduced precision without a sufficient update of the device model. • Increased communication overhead due to additional metadata sent with model updates. • Increased computational complexity for aggregating a larger number of partial updates. • Vulnerability to attacks if devices are not cautious in sending model updates, potentially exposing sensitive data.

it less suitable for resource-constrained devices, where simplicity and efficiency are crucial. Furthermore, Bayesian Federated Aggregation, despite its benefits in providing principled uncertainty estimation and more accurate estimation of the global model, may involve additional computational overhead and complexity. The Bayesian approach typically requires more computational resources for probabilistic modelling and uncertainty estimation, which can be challenging for devices with limited processing power. Taking into account these factors, Federated Averaging stands out as a favourable choice for resource-constrained devices due to its simplicity, computational efficiency, and ability to handle numerous devices and datasets without excessive resource requirements.

5.2. Challenges of independent and non-independent data

Cybersecurity datasets often exhibit significant variations in terms of the types of attacks, their frequencies, and the targeted systems or networks [140]. Each organisation's network infrastructure, security policies, and user behaviour can lead to unique patterns and characteristics in the data. This heterogeneity makes it challenging to assume that the data samples are independently and identically distributed. The non-i.i.d. nature of cybersecurity data poses several challenges for traditional machine learning approaches. Models trained on the data of one organisation may not generalise well to other organisations or new attack scenarios [141]. The presence of rare or evolving threats can lead to imbalanced datasets, where certain types of attacks are underrepresented. This imbalance can bias the model's performance and accuracy. Addressing the non-i.i.d. nature of cybersecurity data requires careful consideration. Furthermore, Federated Learning, a distributed learning approach, can enable collaborative model training while preserving data privacy, allowing organisations to collectively learn from their diverse datasets without the need to share sensitive information [142].

5.3. Types of federated learning

Federated Learning encompasses various approaches and techniques that enable collaborative model training without centrally aggregating sensitive data. In this section, we will explore different types of Federated Learning methods, including centralised Federated Learning, decentralised Federated Learning, and Heterogeneous Federated Learning. Each type offers unique benefits and addresses specific challenges in different use cases. Understanding these types of Federated Learning will provide insights into how organisations can leverage collaborative learning techniques while preserving data privacy and security in diverse scenarios.

5.3.1. Centralised federated learning

Centralised Federated Learning, as depicted in Fig. 2 is the simplest form of ML to conceptualise. In centralised Federated Learning, devices still learn on their own but rely on the central server to update them on any updates sent by any of the other devices [72]. Since the model exchange between the devices is aggregated by a central server, this removes the burden from the device itself and allows for a centralised form of networking. As mentioned above, one of the biggest disadvantages of centralised learning is the lack of redundancy and scalability of the system [72]. If the central server of the network goes down, the rest of the network will not be able to receive updates, but the devices themselves can still continuously learn on their own [72]. Furthermore, intermittent drops in network connectivity can also increase the risk of data breaches. If network connectivity is lost during transmission, data packets can be lost or intercepted by malicious actors, potentially leading to data breaches [143]. These discrepancies can make the system more susceptible to various types of cyber-attacks, such as man-in-the-middle attacks, where an attacker can intercept and manipulate data transmitted over the network [144].

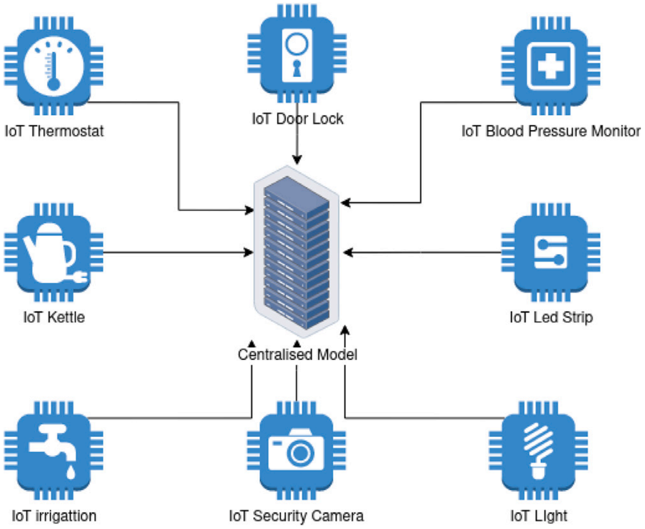

Fig. 2. Centralised FL system consisting of multiple IoT devices.

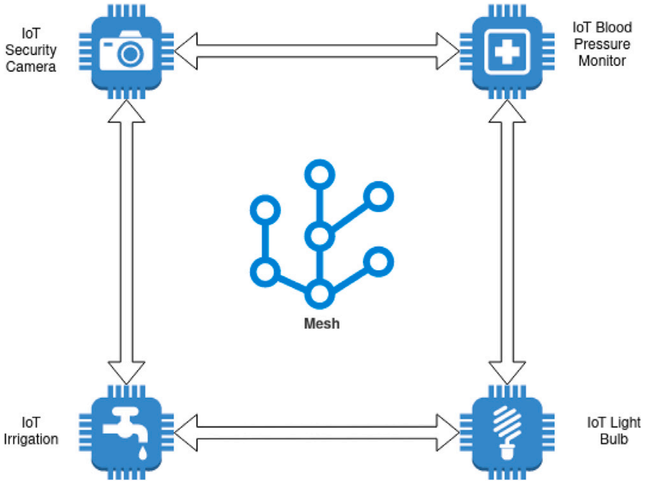

Fig. 3. Decentralised FL consisting of multiple independent devices.

5.3.2. *Decentralised learning*

Decentralised learning, as depicted in Fig. 3, allows devices to work independently while also being able to communicate with each other without a central node [145]. In the concept of decentralised learning, each device is only responsible for its data, effectively running in a mesh configuration, allowing all devices to communicate with each other. It does not need to consider other devices' data while learning if that data is irrelevant to it. Each device in the decentralised learning system functions as a node within the mesh-like network. These nodes possess their local datasets, which may vary in terms of size, content, or distribution. Instead of relying on a central server to aggregate all the data, each device performs its learning process using its locally stored data. This aspect of decentralised learning aligns with the principle of data privacy and security, as individual devices keep their data locally without sending it to a central authority. However, the decentralised learning model does not completely isolate each device. The devices in the network can communicate and exchange information with each other. This communication occurs through a coordination mechanism that enables the sharing of model updates rather than raw data.

Decentralised learning is a new concept that is being implemented, which has even been adapted to use blockchain technology [145–147].

5.3.3. *Heterogeneous learning*

Heterogeneous learning allows the use of different hardware and software configurations, which would be optimal in most cases. It is the most technically challenging to achieve compared to the previous models as it needs to handle the differences between nodes [148,149]. As depicted in Fig. 4, it is made up of different devices with different characteristics, such as different system configurations (Storage space, network controllers, sensors), depending on the use case (air quality, temperature, motion monitoring, etc.). There have been attempts to achieve heterogeneity using Stochastic Controlled Averaging, as presented by Sai Praneeth Karimireddy et al. [150]. The SCAFFOLD algorithm uses control variates to correct for the client drift phenomenon. Client drift is a phenomenon observed in Federated Learning where the local updates of clients diverge from the global model, leading to unstable and slow convergence [151]. This is often caused by non-IID (Independent and Identically Distributed) data, where data distributions vary across clients [152]. SCAFFOLD has been compared to other federated learning methods. For example, a study compared the performance of SCAFFOLD with SGD (Stochastic Gradient Descent), Adam, and Adabelief [150]. Another study compared SCAFFOLD with FedProx, another federated learning algorithm [153]. These comparisons help to highlight the strengths and weaknesses of SCAFFOLD and provide insight into its performance under different conditions. The study shows that SCAFFOLD requires fewer communication rounds compared to other methods and is not affected by data heterogeneity or client sampling. Additionally, the research demonstrates that, for quadratics, SCAFFOLD can exploit similarities in client data to achieve even faster convergence. This is the first study to quantify the benefits of local steps in distributed optimisation. Works like those of Nishio et al. [154] introduce a novel Federated Learning (FL) protocol named FedCS. This protocol is designed to perform FL efficiently by actively managing client devices based on their resource conditions. Specifically, FedCS addresses the issue of resource constraints by solving a client selection problem, which enables the server to aggregate the maximum number of client updates and improve the performance of ML models. Furthermore, the research done by Zhuangdi Zhu et al. [155] proposes a data-free knowledge distillation approach to address the issue of heterogeneity in FL. The process includes learning a simple model that generates information without using actual user data. This generated information is then shared with the nodes, who use it as a guiding principle during their training.

6. Review of existing implementations in resource constraint IoT devices

Cybersecurity is in a perpetual state of flux, where new hazards arise daily. As a result, advanced tools and techniques are needed to combat these threats. As we shall explore in the following section, current research has focused on using Federated Learning in the context of cybersecurity and intrusion detection. Specifically, studies have investigated the practicality and effectiveness of using FL in IoT networks and compared it with traditional centralised approaches. Additionally, research has examined the potential of integrating blockchain technology into FL to improve permissions and auditing of the model data.

6.1. *Federated learning frameworks*

However, one significant challenge in the adoption and implementation of Federated Learning is the lack of standardisation. This has resulted in the development of multiple frameworks, each trying to achieve Federated Learning in its unique way. At the time of writing, some of the most popular open-source frameworks for Federated Learning include TensorFlow Federated (TFF), NVIDIA Clara, FedLab [156], FATE [157], Flower [158], OpenFL [159], Plato [157], Substra [160], IBMFL [161], ConcreteML [162] and Syft [163]. These frameworks have the highest engagement on GitHub and are the most contributed

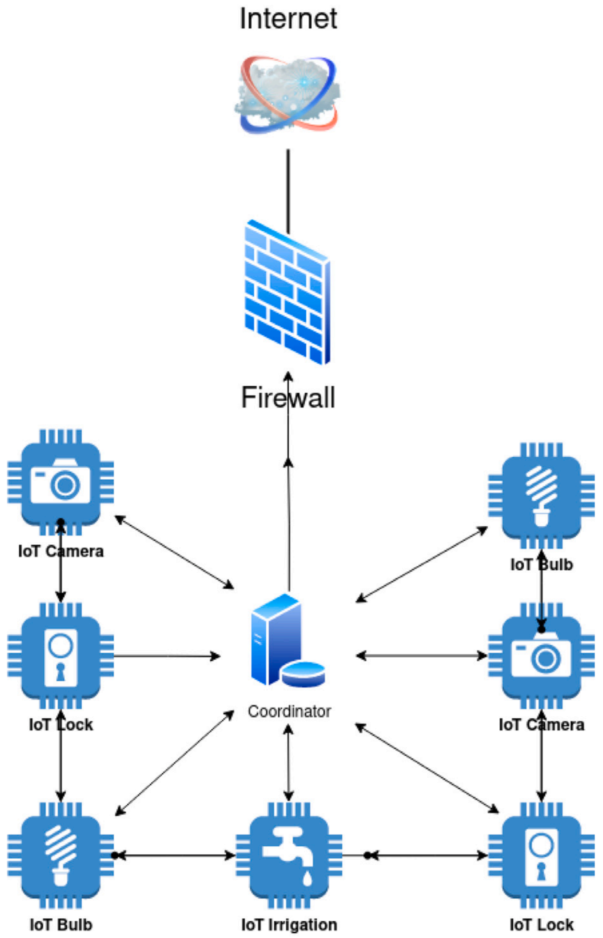

Fig. 4. Heterogeneous learning.

to. The absence of standardisation across Federated Learning frameworks has caused fragmentation in the field, making it difficult for developers to choose a uniform approach. Each framework comes with its own set of features, capabilities, and compatibility with different machine-learning libraries, adding complexity to the adoption and integration process. One of the critical concerns resulting from the lack of standardisation is the lack of standardised authentication and encryption mechanisms for exchanging models and authenticating between devices during the Federated Learning process. Authentication and encryption are crucial to ensure the security and privacy of the data and models shared in a Federated Learning setup. Without a standardised approach, developers must individually implement these security features, leading to potential inconsistencies and vulnerabilities in different frameworks. However, there are some frameworks among the mentioned ones that have recognised the importance of security and have included authentication and encryption features to address these concerns. In particular, NVIDIA Clara [164], OpenFL [157], FATE, Flower, IBMFL, and Syft are among the frameworks that have incorporated these security measures. An outlier is Substra, which, although on the surface has implemented authentication and encryption, this is only the case for its web management interface. To evaluate this, the documentation provided by the framework was investigated and if authentication or encryption was mentioned as one of the features, the code specifically implementing this would be analysed to check if this was true. Another noteworthy aspect is the integration of FL with IoT devices. FL is particularly useful for IoT scenarios, as it allows for on-device training without compromising user data privacy. TFFED [165], NVIDIA Clara, FedLab, FATE, Flower, and other frameworks have been designed with IoT in mind and offer implementations optimised for such use cases. Concerning Table 7, it is apparent that many of these frameworks lack these important features, creating another attack vector for malicious threat actors. Expanding, it should be noted that this is not an extensive list, as research continues in the field of FL it is extremely likely that more frameworks and IoT implementations will be created.

Table 7
Federated learning frameworks with authentication, encryption, and IoT support.

Framework	Authentication	Encryption	IoT
TFF	✗	✗	✓
Nvidia Clara	✓	✓	✓
FedLab	✗	✗	✓
Flower	✓	✓	✓
FATE	✓	✓	✓
OpenFL	✓	✓	✗
Plato	✗	✓	✗
Substra	Only for web Interface	Only for web Interface	✗
IBMFL	✓	✓	✗
Syft	✓	✗	✗
ConcreteML	✓	✓	✗

As the field continues to evolve, the community needs to work towards creating standards and best practices to ensure better interoperability, security, and wider adoption of Federated Learning across various applications and industries.

6.2. FL implementations in cybersecurity

The applications of FL in cybersecurity are diverse, with notable implementations, including hierarchical FL, which facilitates collaborative material interaction in IoT applications [166]. Although it is a relatively new machine learning technique, FL has been extensively adopted in cybersecurity, particularly in industrial settings [134]. Research findings suggest that global models aggregated from data across devices outperform localised models, resulting in a significant 7.07% performance improvement with only ten queries [167]. Additionally, FL has been effectively utilised with deep neural networks, such as the deep-fed framework for intrusion detection in industrial cyber–physical systems, achieving remarkable accuracy levels exceeding 97% [168]. Furthermore, the implementation of binary neural networks (BNNs) within the Federated framework has demonstrated reduced communication overhead and memory utilisation, with comparable performance to Support Vector Machines (SVMs) but the added benefit of lower memory overhead [169]. FL offers versatile implementations in Cybersecurity, showcasing its capabilities in hierarchical FL, deep neural network integration, and binary neural networks. These findings highlight FL's potential to significantly improve intrusion detection accuracy while reducing communication and memory overhead. Future investigations could explore the integration of FL with more advanced neural network architectures and other ML techniques, aiming to optimise intrusion detection accuracy further.

6.3. FL and blockchain integration: Enhancing immutability and efficiency

The decentralised nature of Federated Learning (FL) aligns seamlessly with blockchain technology, leading to its integration for auditable and immutable model updates. This integration is exemplified by the introduction of FED-IDS, a pioneering framework that enables the creation of blockchain-managed Federated training networks specifically for intrusion detection, thereby paving the way for blockchain-based advancements in cybersecurity [170]. Although still in its early stages, FED-IDS has achieved impressive accuracy rates exceeding 97%, with only a marginal 2% difference compared to centralised training models. This demonstrates the potential of FL as a viable and efficient alternative to traditional centralised approaches [170]. The integration of FL with blockchain technology not only allows for auditable model updates but also enhances the immutability aspect of the blockchain. Extensive testing using the CICIDS2017 dataset demonstrates transparent distribution training, with performance impacts ranging between 5% and 15%. Aligning Federated averaging with blockchain block creation intervals minimises latency between nodes, thereby improving model auditability and ensuring the integrity of the data [171]. The integration of FL with blockchain technology through FED-IDS presents a promising avenue for auditable and decentralised model updates in intrusion detection. This combination offers the potential to improve accuracy while maintaining data privacy and security. However, in its current state, the resource requirements of such an approach would not be applicable for a device such as a microcontroller due to the tremendous computational power needed for the blockchain. To address these computational power requirements and make the technology more accessible, future work could focus on optimising the integration of FL and blockchain. This would aim to reduce resource consumption, making it more feasible for resource-constrained IoT devices, and further improving the scalability and efficiency of the system. In this way, the immutability of the blockchain can be fully leveraged to improve the efficiency and effectiveness of FL.

6.4. FL advantages for IoT networks

In the context of large-scale IoT networks utilising Wi-Fi communication, security challenges arise [172]. While ML and deep learning methods can address malicious attacks, the collection of data from such networks raises privacy concerns that may conflict with data protection legislation [172]. FL's local data processing offers a viable solution to combat these challenges, presenting promising results with at least 90% accuracy [172]. FL's advantage lies in local data processing, addressing privacy concerns in large-scale IoT networks. This approach offers substantial accuracy in intrusion detection, making it a promising solution to improve the security of IoT environments. Further research could explore the integration of FL with additional privacy-preserving techniques to enhance data protection in IoT networks.

6.5. FL vs. Centralised approach for intrusion detection

Comparative research between the centralised on-device and FL approaches for intrusion detection using the NSL-KDD data set indicates that FL's 83% precision is comparable to the centralised approach. However, certain limitations of FL, such as device dropouts and restricted update rates, need further consideration for optimal performance [22]. Furthermore, research by Riccardo Lazarrini et al. [173] evaluates FL using a shallow artificial neural network (ANN) and federated averaging (FedAvg) as the aggregation algorithm on the ToN_IoT and CICIDS2017 datasets. FL proves to be an efficient alternative, maintaining data privacy and achieving comparable accuracy, precision, recall, and F1 score to the centralised approach. Additionally, alternative aggregation algorithms are evaluated, with FedAvg and FedAvgM outperforming FedAdam and FedAdagrad in this scenario. The comparison between centralised and FL approaches highlights FL's competitive accuracy in detecting intrusions. However, addressing the limitations is essential to unlock its full potential for intrusion detection tasks. To overcome the limitations of FL, future research could investigate strategies for minimising device dropout and explore adaptive aggregation algorithms that adapt to varying network conditions for better performance. Additionally, the development of more efficient update mechanisms could be explored to minimise response times in the model aggregation process.

6.6. Limitations of current federated learning implementations and opportunities for improvement

After reviewing the above papers, it can be deduced that the majority of them implement FL using high-end machines in ideal scenarios. Very few of them attempt to test the models in a realistic scenario where each machine has large differences in network latency and different hardware specifications. Research done by Marc Monfort Grau et al. [178] and also Llisterri Giménez et al. [179] have proven that FL is possible on resource-constrained devices but suffers heavy losses in accuracy when met with networking issues. Although many of these articles have covered IoT-related topics, the term IoT is used loosely to describe devices on the edge with large amounts of resources and seems to have no consequences for being ’on the edge’ of the network. In many cases, these edge devices are just connected in the same network, which does not create a realistic test case scenario which would consist of a plethora of heterogeneous devices. Although the research is valid in itself and many of these papers have attempted to address the heterogeneity problem, they have not tested these models in segmented networks, which is a fundamental security mechanic implemented within such environments [180]. Furthermore, many of these articles mentioned in the literature review have not used a variety of devices, such as small sensors and resource-constrained IoT devices, to test their models, which would be a realistic scenario, since most homes and industrial networks consist of a variety of devices. Even though there is great difficulty in implementing such a solution in resource-constrained devices [174] because of the network overhead and computational needs of the model, the benefits of using a large plethora of devices with minimal network footprint to not only monitor themselves but also provide valuable telemetry data to the rest of the network allows the network to scale up while having a holistic view of the network.

With the ever-expanding number of IoT devices in both domestic and industrial situations, there has been a large amount of research done to implement intrusion detection systems in these devices. Furthermore, the expandability and robustness of FL make it an excellent candidate in this case, since it allows all of these devices to work together. Most of this research has been done using existing data and implementing them in a federated manner. Many of these implemented models use Federated averaging to try to combat the heterogeneity problem of FL, with some being more effective than others. A large majority of these implementations use decision trees, support vector machines, random forest, and multilayer perceptron in the Federated environments. Regarding Table 8 an observation can be made that many papers are looking into IoT devices, but since IoT is such a vague term, in the case of these research papers, it mostly consists of high-performance machines connected at the edge. The table is categorised as follows:

- Application Domain – domain to which the study can be applied to
- Study - What is being investigated by the specific research paper
- IoT Related—If the study attempts to implement the proposed solution in IoT
- Suitable for Resource-constrained devices —if the study has made optimisation or implemented features which could be used in such a device

Many of them failed to address the potential of investigating these solutions on resource-constrained devices. By adopting methods such as SCAFFOLD [174], which has been proven to work in devices with limited resources, there is potential for an influx of research to be explored in these types of scenarios. Expanding, recent advancements in resource-constrained FL have illuminated the emergence of Federated Dropout (FedDrop) [181]. FedDrop is a proposed scheme that extends the conventional dropout method for random model pruning. The scheme involves the independent generation of multiple subnets in each

Table 8
Comparison of research papers on collaborative learning in cybersecurity.

Research	Application domain	Study	IoT related	Suitable for resource-constrained devices
[166]	Intrusion detection system	Hierarchical federated learning	✗	✗
[167]	Intrusion detection system	Federated learning approach with active personalisation	✓	✗
[170]	Blockchain intrusion detection system	BlockChain intrusion detection on IoT attacks	✓	✓
[172]	Wireless network intrusion detection	Federated wireless network intrusion	✓	✓
[169]	Edge intrusion detection system	BNN based federated learning on the edge	✓	✗
[168]	Intrusion detection system	Federated deep learning in cyber–physical systems	✓	✓
[22]	Intrusion detection system	Comparison between centralised, On-device, and Federated learning	✓	✓
[171]	Blockchain intrusion detection system	Compressed blockchain blocked transfer	✓	✗
[89]	Encryption	Privacy-Preserving federated modelling	✗	✗
[90]	Secure segregation	Secure model aggregation	✗	✗
[91]	Secure segregation	Lightweight secure model aggregation	✗	✓
[92]	Secure segregation	Scalable secure aggregation	✗	✗
[94]	Model bias	Dealing with model biases through specific model extraction	✗	✗
[95]	Privacy risks	Evaluation of privacy risks of machine learning models	✗	✗
[96]	Adversarial attacks	Leveraging model memorisation for calibrated membership inference	✗	✗
[97]	Differential privacy	Foundations of differential privacy	✗	✗
[87]	Federated averaging	Bidirectional federated learning	✗	✗
[88]	Federated proximal	Federated averaging algorithm	✗	✗
[85]	Hyper-Parameter tuning	Representation matching and adaptive Hyper-Parameters	✗	✗
[74]	Hyper-Parameter tuning	Federated learning with Non-IID data	✗	✗
[86]	Federated learning method	Model-contrastive federated learning	✗	✗
[174]	Federated learning method	Stochastic controlled averaging for federated learning	✗	✗
[175]	Federated learning method	Advances in neural information processing systems	✗	✗
[176]	Federated learning method	Bayesian nonparametric federated learning	✗	✗
[173]	Federated learning method	Federated learning for IoT intrusion detection	✓	✓
[177]	Hetero-geneous networks	Federated optimisation	✗	✓
[145]	Decentral-ised learning	Federated deployment environments	✓	✓

iteration of the FL algorithm, derived from the global model at the server using dropout, with heterogeneous dropout rates or parameter-pruning probabilities, each of which is customised to the state of an assigned channel. Subsequently, the subnets are downloaded to connected devices for updating. FedDrop effectively mitigates both communication overhead and device computational burden in comparison to traditional FL approaches. Furthermore, it outperforms conventional FL techniques in instances of over-fitting, as well as the FL scheme utilising uniform dropout.

6.7. Research gaps and challenges

By analysing collaborative learning, this paper provides substantial information on its application in the field of cybersecurity, particularly in the context of intrusion detection for IoT devices. FL is a widely used and potentially viable method for implementing intrusion detection in resource-constrained devices using collaborative methods.

However, the lack of standardised frameworks is a significant issue that impedes the smooth progression of existing systems. The study identifies several areas that require further exploration. In addition, as we have seen throughout this paper, the majority of research in this field focuses solely on large-scale IoT systems with large amounts of resources. This leaves out a large portion of IoT devices which are not only vulnerable but also widely distributed throughout businesses and homes.

This presents a big challenge, where creating this standardisation of methods and implementations is difficult, knowing that devices differ in specifications, inevitably creating a heterogeneity problem. In essence, the challenge lies not only in developing standardisation but in doing so in a way that is adaptable to the heterogeneity of IoT devices. This intricate task underscores the need for future research to dive into this nuanced landscape, seek solutions that can be universally applied, and provide a comprehensive approach to cybersecurity in the expansive realm of the IoT.

Additionally, reducing model size while maintaining performance is a key challenge in resource-constrained environments. Model compression techniques such as pruning, quantisation, or knowledge distillation offer effective ways to shrink model size. Energy-efficient training algorithms are essential for IoT devices to utilise energy resources efficiently. Prioritising updates from devices with higher battery levels or adapting the training schedule based on energy availability are potential strategies to address this challenge.

IoT devices typically operate with limited memory and processing power, necessitating the development of lightweight machine-learning models suitable for on-device training. Techniques like model quantisation and knowledge distillation offer promising solutions to reduce the memory footprint without compromising accuracy. Expanding, achieving effective model convergence amidst the heterogeneity of devices poses a significant challenge. Different devices may have varying data distributions and capabilities, demanding adaptive aggregation methods such as weighted averaging or differential privacy to ensure model consistency.

Moreover, the diverse capabilities of IoT devices present a challenge in handling client heterogeneity. Adaptive algorithms are needed to adjust learning rates or model architectures based on device characteristics, with federated optimisation emerging as a viable approach to manage device diversity. While there have been papers such as Nil Llisterri Giménez et al. [179] which have used federated learning on resource-constrained devices, it should be noted that only three features had been used in the training process, severely limiting the precision of the model, regardless of whether the training was successful.

Privacy preservation is another critical concern, as transmitting raw data to a central server raises privacy risks. Techniques like secure aggregation, homomorphic encryption, FL or local differential privacy can help protect sensitive data during model aggregation.

6.8. Future directions

Subsequent investigation concerning collaborative intrusion detection systems for resource-constrained IoT devices should concentrate on several key areas. A primary difficulty is the absence of uniform frameworks for FL, impeding consistent application across varied devices. Developing adaptable standards is a priority. Optimisation for devices with constrained processing, memory, and power is needed, requiring the creation of lightweight machine learning models, potentially using Binary Neural Networks BNNs, and refinement of model compression methods like pruning, quantisation, knowledge distillation, or FedDrop [181].

Developing less demanding cryptographic methods could compensate for limited flash memory and compute power, improving security without overburdening devices. Energy-conserving training algorithms also require development, perhaps by adjusting operations based on available power. Addressing the mix of device capabilities and data distributions, known as non-IID data, within IoT networks necessitates better adaptive aggregation methods, like weighted averaging, and adaptive algorithms such as Stochastic Controlled Averaging for Federated Learning SCAFFOLD [174] or FedProx [153], alongside testing within realistic, varied network conditions, moving beyond idealised laboratory setups. Improvements to security involve deeper exploration of cryptographic techniques such as secure aggregation, homomorphic encryption, and differential privacy to protect against data and model manipulation, alongside methods like spectral anomaly detection to identify malicious participants or compromised devices.

The resource constraints in these devices vary between devices, but to the best of our knowledge, there is a lack of research looking into the microcontroller space comparing power consumption and viability of machine learning models on these devices. Therefore researchers could look into this topic for the development of viable algorithms and methods for streamlining the process.

Scaling these collaborative systems for extensive IoT deployments presents another obstacle needing attention. Moreover, ongoing progress in connectivity technologies, such as Wi-Fi and Bluetooth, may help lessen communication delays between devices participating in collaborative learning. Finally, examining hybrid systems combining FL with edge computing or transfer learning offers a promising route to improved efficiency and security. Progress in these directions will aid the development of more dependable security for interconnected systems.

7. Conclusion and future work

In conclusion, this study highlights the potential of collaborative learning, particularly Federated Learning, to enhance cybersecurity for Internet of Things (IoT) devices. Despite its promise, challenges such as the absence of standardised frameworks, device heterogeneity, and issues related to security, privacy, and scalability persist. Future research should prioritise developing custom optimisation techniques tailored for resource-constrained devices and explore advanced cryptographic methods to safeguard FLs against potential threats. Additionally, addressing scalability challenges in expansive environments is crucial for the widespread adoption of FL. Furthermore, investigating hybrid approaches that integrate Federated Learning with edge computing and transfer learning presents a promising avenue for enhancing efficiency and security. By combining the strengths of FL with these complementary technologies, it may be possible to overcome some of the limitations inherent in collaborative learning paradigms. A commitment to these research directions is essential for unlocking the full potential of Federated Learning and advancing intrusion detection in the ever-evolving IoT landscape. By addressing these challenges and exploring innovative solutions, we can ensure the continued growth and security of IoT ecosystems.

CRediT authorship contribution statement

Vasilis Ieropoulos: Writing – review & editing, Writing – original draft, Investigation, Conceptualization. **Eirini Anthi:** Supervision, Conceptualization. **Theodoros Spyridopoulos:** Supervision. **Pete Burnap:** Supervision. **Ioannis Mavromatis:** Supervision. **Aftab Khan:** Supervision. **Pietro Carnelli:** Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Vasilis Ieropoulos reports financial support was provided by Engineering and Physical Sciences Research Council. Vasilis Ieropoulos reports financial support was provided by Toshiba Europe Limited Bristol Research and Innovation Laboratory. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by UKRI EPSRC and Toshiba Europe Ltd [EP/T517951/1].

Data availability

No data was used for the research described in the article.

References

[1] Frost, Sullivan. 2022 update-total internet of things (IOT) device forecast, 2018–2027. 2022, URL: https://store.frost.com/2022-update-total-internet-of-things-iot-device-forecast-2018-2027.html.

[2] Vailshery LS. Global IOT and non-IoT connections 2010–2025. 2022, URL: https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/.

[3] Mahgoub A, Tarrad N, Elsherif R, Al-Ali A, Ismail L. IoT-based fire alarm system. In: 2019 third world conference on smart trends in systems security and sustainablity (worldS4). IEEE; 2019, p. 162–6.

[4] Islam FB, Ifeanyi Nwakanma C, Kim D-S, Lee J-M. IoT-based HVAC monitoring system for smart factory. In: 2020 international conference on information and communication technology convergence. ICTC, 2020, p. 701–4. http://dx.doi.org/10.1109/ICTC49870.2020.9289249.

[5] Pérez-Padillo J, García Morillo J, Ramirez-Faz J, Torres Roldán M, Montesinos P. Design and implementation of a pressure monitoring system based on IoT for water supply networks. Sensors 2020;20(15):4247.

[6] Xavier TCS, Delicato FC, Pires PF, Amorim CL, Li W, Zomaya A. Managing heterogeneous and time-sensitive IoT applications through collaborative and energy-aware resource allocation. ACM Trans Internet Things 2022;3(2). http://dx.doi.org/10.1145/3488248.

[7] Hashim N, Norddin N, Idris F, Yusoff S, Zahari M. IoT blood pressure monitoring system. Indones J Electr Eng Comput Sci 2020;19(3):1384–90.

[8] Salvi N, Doctor G. Identification of barriers in adoption of IoT: Commercial complexes in India. In: International conference on soft computing and its engineering applications. Springer; 2022, p. 181–93.

[9] Neshenko N, Bou-Harb E, Crichigno J, Kaddoum G, Ghani N. Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations. IEEE Commun Surv & Tutor 2019;21(3):2702–33. http://dx.doi.org/10.1109/COMST.2019.2910750.

[10] Stanislav M, Beardsley T. Hacking iot: A case study on baby monitor exposures and vulnerabilities. Rapid7 Rep 2015.

[11] Zubair M, Unal D, Al-Ali A, Shikfa A. Exploiting bluetooth vulnerabilities in E-health IoT devices. In: Proceedings of the 3rd international conference on future networks and distributed systems. ICFNDS '19, New York, NY, USA: Association for Computing Machinery; 2019, http://dx.doi.org/10.1145/3341325.3342000.

[12] Bor M, Vidler JE, Roedig U. Lora for the internet of things. Lancaster 2016.

[13] Adi PDP, Sihombing V, Siregar VMM, Yanris GJ, Sianturi FA, Purba W, Tamba SP, Simatupang J, Arifuddin R, Subairi, Prasetya DA. A performance evaluation of ZigBee mesh communication on the internet of things (IoT). In: 2021 3rd east Indonesia conference on computer and information technology. eIConCIT, 2021, p. 7–13. http://dx.doi.org/10.1109/EIConCIT50028.2021.9431875.

[14] Ganie SM, Malik MB, Arif T. Machine learning techniques for big data analytics in healthcare: Current scenario and future prospects. In: Telemedicine: the computer transformation of healthcare. Springer; 2022, p. 103–23.

[15] Yuan S, Wu X. Deep learning for insider threat detection: Review, challenges and opportunities. Comput Secur 2021;104:102221. http://dx.doi.org/10.1016/j.cose.2021.102221, URL: https://www.sciencedirect.com/science/article/pii/S0167404821000456.

[16] Mishra P, Varadharajan V, Tupakula U, Pilli ES. A detailed investigation and analysis of using machine learning techniques for intrusion detection. IEEE Commun Surv & Tutor. 2019;21(1):686–728. http://dx.doi.org/10.1109/COMST.2018.2847722.

[17] Vasilomanolakis E, Karuppayah S, Mühlhäuser M, Fischer M. Taxonomy and survey of collaborative intrusion detection. ACM Comput Surv 2015;47(4). http://dx.doi.org/10.1145/2716260.

[18] Zhou CV, Leckie C, Karunasekera S. A survey of coordinated attacks and collaborative intrusion detection. Comput Secur 2010;29(1):124–40. http://dx.doi.org/10.1016/j.cose.2009.06.008, URL: https://www.sciencedirect.com/science/article/pii/S016740480900073X.

[19] Miller P, Inoue A. Collaborative intrusion detection system. In: 22nd international conference of the North American fuzzy information processing society. NAFIPS 2003, 2003, p. 519–24. http://dx.doi.org/10.1109/NAFIPS.2003.1226839.

[20] Campos EM, Saura PF, González-Vidal A, Hernández-Ramos JL, Bernabé JB, Baldini G, Skarmeta A. Evaluating federated learning for intrusion detection in internet of things: Review and challenges. Comput Netw 2022;203:108661.

[21] Agrawal S, Sarkar S, Aouedi O, Yenduri G, Piamrat K, Alazab M, Bhattacharya S, Maddikunta PKR, Gadekallu TR. Federated learning for intrusion detection system: Concepts, challenges and future directions. Comput Commun 2022;195:346–61. http://dx.doi.org/10.1016/j.comcom.2022.09.012, URL: https://www.sciencedirect.com/science/article/pii/S0140366422003516.

[22] Rahman SA, Tout H, Talhi C, Mourad A. Internet of things intrusion detection: Centralized, on-device, or federated learning? IEEE Netw 2020;34(6):310–7. http://dx.doi.org/10.1109/MNET.011.2000286.

[23] Elrawy MF, Awad AI, Hamed HF. Intrusion detection systems for IOT-based smart environments: A survey. J Cloud Comput 2018;7(1). http://dx.doi.org/10.1186/s13677-018-0123-6.

[24] Kaushik A, Al-Raweshidy H. A novel intrusion detection system for internet of things devices and data. Wirel Netw 2023;30(1):285–94. http://dx.doi.org/10.1007/s11276-023-03435-0.

[25] Khraisat A, Alazab A. A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity 2021;4(1). http://dx.doi.org/10.1186/s42400-021-00077-7.

[26] Huang Y, Qiao X, Dustdar S, Zhang J, Li J. Toward decentralized and collaborative deep learning inference for intelligent IoT devices. IEEE Netw 2022;36(1):59–68. http://dx.doi.org/10.1109/MNET.011.2000639.

[27] Jalali NA, Chen H. Security issues and solutions in federate learning under IOT critical infrastructure. Wirel Pers Commun 2022;129(1):475–500. http://dx.doi.org/10.1007/s11277-022-10107-3.

[28] AVSystem A. How to manage resource-constrained IOT devices? 2020, URL: https://www.avsystem.com/blog/iot/what-is-resource-constrained-device/.

[29] Gregersen C. A complete guide to microcontrollers for IOT. 2023, URL: https://www.nabto.com/iot-microcontroller-guide/.

[30] Utmel. Using microcontrollers in the internet of things (IOT) applications. 2023, URL: https://www.utmel.com/blog/categories/microcontrollers/using-microcontrollers-in-the-internet-of-things-iot-applications.

[31] Haseeb J, Mansoori M, Al-Sahaf H, Welch I. IoT attacks: Features identification and clustering. In: 2020 IEEE 19th international conference on trust, security and privacy in computing and communications (trustCom). 2020, p. 353–60. http://dx.doi.org/10.1109/TrustCom50675.2020.00056.

[32] Shukla P, Krishna CR, Patil NV. IOT traffic-based ddos attacks detection mechanisms: A comprehensive review. J Supercomput 2023. http://dx.doi.org/10.1007/s11227-023-05843-7.

[33] Tushir B, Sehgal H, Nair R, Dezfouli B, Liu Y. The impact of dos attacks onresource-constrained iot devices: A study on the mirai attack. 2021, arXiv preprint arXiv:2104.09041.

[34] Ionescu V, Roedig U. Battery depletion attacks on NB-IoT devices using interference. In: Katsikas S, Lambrinoudakis C, Cuppens N, Mylopoulos J, Kalloniatis C, Meng W, Furnell S, Pallas F, Pohle J, Sasse MA, Abie H, Ranise S, Verderame L, Cambiaso E, Maestre Vidal J, Sotelo Monge MA, editors. Computer security. ESORICS 2021 international workshops. Cham: Springer International Publishing; 2022, p. 276–95.

[35] Mottola L, Hameed A, Voigt T. Energy attacks in the battery-less internet of things. 2023, arXiv preprint arXiv:2304.08224.

[36] Daud M, Rasiah R, George M, Asirvatham D, Rahman AFA, Halim AA. Denial of service: (DoS) impact on sensors. In: 2018 4th international conference on information management. ICIM, 2018, p. 270–4. http://dx.doi.org/10.1109/INFOMAN.2018.8392848.

[37] Gautam SK, Om H, Dixit K. Intrusion detection system in internet of things. In: Das SK, Samanta S, Dey N, Kumar R, editors. Design frameworks for wireless networks. Singapore: Springer Singapore; 2020, p. 65–93. http://dx.doi.org/10.1007/978-981-13-9574-1_4.

[38] Khoa TV, Saputra YM, Hoang DT, Trung NL, Nguyen D, Ha NV, Dutkiewicz E. Collaborative learning model for cyberattack detection systems in iot industry 4.0. In: 2020 IEEE wireless communications and networking conference. WCNC, IEEE; 2020, p. 1–6.

[39] Fenrich K. Securing your control system: The" CIA triad" is a widely used benchmark for evaluating information system security effectiveness. Power Eng 2008;112(2):44–9.

[40] Tsogbaatar E, Bhuyan MH, Taenaka Y, Fall D, Gonchigsumlaa K, Elmroth E, Kadobayashi Y. DeL-IoT: A deep ensemble learning approach to uncover anomalies in IoT. Internet Things 2021;14:100391. http://dx.doi.org/10.1016/j.iot.2021.100391, URL: https://www.sciencedirect.com/science/article/pii/S2542660521000354.

[41] Rashid AB, Ahmed M, Sikos LF, Haskell-Dowland P. Anomaly detection in cybersecurity datasets via cooperative co-evolution-based feature selection. ACM Trans Manag Inf Syst (TMIS) 2022;13(3):1–39.

[42] Hegedűs I, Danner G, Jelasity M. Decentralized learning works: An empirical comparison of gossip learning and federated learning. J Parallel Distrib Comput 2021;148:109–24. http://dx.doi.org/10.1016/j.jpdc.2020.10.006, URL: https://www.sciencedirect.com/science/article/pii/S0743731520303890.

[43] Markiewicz RP, Sgandurra D. Clust-IT: Clustering-based intrusion detection in IoT environments. In: Proceedings of the 15th international conference on availability, reliability and security. ARES '20, New York, NY, USA: Association for Computing Machinery; 2020, http://dx.doi.org/10.1145/3407023.3409201.

[44] Mishra S, Sagban R, Yakoob A, Gandhi N. Swarm intelligence in anomaly detection systems: An overview. Int J Comput Appl 2021;43(2):109–18. http://dx.doi.org/10.1080/1206212X.2018.1521895, arXiv:https://doi.org/10.1080/1206212X.2018.1521895.

[45] Kisielewicz T, Stanek S, Zytniewski M. A multi-agent adaptive architecture for smart-grid-intrusion detection and prevention. Energies 2022;15(13). http://dx.doi.org/10.3390/en15134726, URL: https://www.mdpi.com/1996-1073/15/13/4726.

[46] Ashraf E, Areed NFF, Salem H, Abdelhay EH, Farouk A. FIDChain: Federated intrusion detection system for blockchain-enabled IoT healthcare applications. Healthcare 2022;10(6). http://dx.doi.org/10.3390/healthcare10061110, URL: https://www.mdpi.com/2227-9032/10/6/1110.

[47] Abe T, Buchanan EK, Pleiss G, Zemel R, Cunningham JP. Deep ensembles work, but are they necessary?. 2022, arXiv:2202.06985.

[48] Lakshminarayanan B, Pritzel A, Blundell C. Simple and scalable predictive uncertainty estimation using deep ensembles. 2017, arXiv:1612.01474.

[49] Mustafa B, Riquelme C, Puigcerver J, Pinto AS, Keysers D, Houlsby N. Deep ensembles for low-data transfer learning. 2020, arXiv:2010.06866.

[50] Yılmaz S, Aydogan E, Sen S. A transfer learning approach for securing resource-constrained IoT devices. IEEE Trans Inf Forensics Secur 2021;16:4405–18. http://dx.doi.org/10.1109/TIFS.2021.3096029.

[51] Qureshi A-U-H, Larijani H, Javed A, Mtetwa N, Ahmad J. Intrusion detection using swarm intelligence. In: 2019 UK/ China emerging technologies. UCET, 2019, p. 1–5. http://dx.doi.org/10.1109/UCET.2019.8881840.

[52] Hegedűs I, Danner G, Jelasity M. Gossip learning as a decentralized alternative to federated learning. In: Pereira J, Ricci L, editors. Distributed applications and interoperable systems. Cham: Springer International Publishing; 2019, p. 74–90.

[53] Tang Z, Shi S, Li B, Chu X. GossipFL: A decentralized federated learning framework with sparsified and adaptive communication. IEEE Trans Parallel Distrib Syst 2023;34(3):909–22. http://dx.doi.org/10.1109/TPDS.2022.3230938.

[54] Giaretta L, Girdzijauskas S. Gossip learning: Off the beaten path. In: 2019 IEEE international conference on big data (big data). 2019, p. 1117–24. http://dx.doi.org/10.1109/BigData47090.2019.9006216.

[55] Krishnan Sadhasivan D, Balasubramanian K. A fusion of multiagent functionalities for effective intrusion detection system. Secur Commun Netw 2017;2017:1–15. http://dx.doi.org/10.1155/2017/6216078.

[56] Louati F, Ktata FB. A deep learning-based multi-agent system for intrusion detection. SN Appl Sci 2020;2(4). http://dx.doi.org/10.1007/s42452-020-2414-z.

[57] Gal K, Grosz BJ. Multi-agent systems: Technical & ethical challenges of functioning in a mixed group. Daedalus 2022;151(2):114–26. http://dx.doi.org/10.1162/daed_a_01904.

[58] Liang C, Shanmugam B, Azam S, Karim A, Islam A, Zamani M, Kavianpour S, Idris NB. Intrusion detection system for the internet of things based on blockchain and multi-agent systems. Electronics 2020;9(7). http://dx.doi.org/10.3390/electronics9071120, URL: https://www.mdpi.com/2079-9292/9/7/1120.

[59] Durga Bhavani A, Mangla N. A review on intrusion detection approaches in resource-constrained IoT environment. In: Shakya S, Bestak R, Palanisamy R, Kamel KA, editors. Mobile computing and sustainable informatics. Singapore: Springer Nature Singapore; 2022, p. 171–83.

[60] Mirjalili S, Lewis A. The whale optimization algorithm. Adv Eng Softw 2016;95:51–67. http://dx.doi.org/10.1016/j.advengsoft.2016.01.008, URL: https://www.sciencedirect.com/science/article/pii/S0965997816300163.

[61] Ahmad Z, Shahid Khan A, Wai Shiang C, Abdullah J, Ahmad F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Trans Emerg Telecommun Technol 2021;32(1):e4150. http://dx.doi.org/10.1002/ett.4150, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.4150. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/ett.4150.

[62] Ryu J, Won D, Lee Y. A study of split learning model. In: 2022 16th international conference on ubiquitous information management and communication. IMCOM, 2022, p. 1–4. http://dx.doi.org/10.1109/IMCOM53663.2022.9721798.

[63] Vepakomma P, Raskar R. Split learning: A resource efficient model and data parallel approach for distributed deep learning. Fed Learn 2022;439–51. http://dx.doi.org/10.1007/978-3-030-96896-0_19.

[64] Belenguer A, Navaridas J, Pascual JA. A review of federated learning in intrusion detection systems for IoT. 2022, arXiv:2204.12443.

[65] Imteaj A, Mamun Ahmed K, Thakker U, Wang S, Li J, Amini MH. Federated learning for resource-constrained IoT devices: Panoramas and state of the art. In: Razavi-Far R, Wang B, Taylor ME, Yang Q, editors. Federated and transfer learning. Cham: Springer International Publishing; 2023, p. 7–27. http://dx.doi.org/10.1007/978-3-031-11748-0_2.

[66] Hernandez-Ramos JL, Karopoulos G, Chatzoglou E, Kouliaridis V, Marmol E, Gonzalez-Vidal A, Kambourakis G. Intrusion detection based on federated learning: A systematic review. 2023, arXiv:2308.09522.

[67] Agrawal S, Sarkar S, Aouedi O, Yenduri G, Piamrat K, Bhattacharya S, Maddikunta PKR, Gadekallu TR. Federated learning for intrusion detection system: Concepts, challenges and future directions. 2021, arXiv:2106.09527.

[68] Soe YN, Feng Y, Santosa PI, Hartanto R, Sakurai K. Implementing lightweight iot-ids on raspberry pi using correlation-based feature selection and its performance evaluation. In: Advanced information networking and applications: proceedings of the 33rd international conference on advanced information networking and applications (AINA-2019) 33. Springer; 2020, p. 458–69.

[69] Sforzin A, Mármol FG, Conti M, Bohli J-M. Rpids: Raspberry pi ids—a fruitful intrusion detection system for iot. In: 2016 intl IEEE conferences on ubiquitous intelligence & computing, advanced and trusted computing, scalable computing and communications, cloud and big data computing, internet of people, and smart world congress (UIC/aTC/scalCom/cBDCom/ioP/smartWorld). IEEE; 2016, p. 440–8.

[70] Sumanth R, Bhanu K. Raspberry pi based intrusion detection system using k-means clustering algorithm. In: 2020 second international conference on inventive research in computing applications. ICIRCA, IEEE; 2020, p. 221–9.

[71] Alkhazendar I, Zubair M, Qidwai U. Smart hardware trojan detection system. In: Proceedings of SAI intelligent systems conference. Springer; 2022, p. 791–806.

[72] Asad M, Moustafa A, Ito T, Aslam M. Evaluating the communication efficiency in federated learning algorithms. In: 2021 IEEE 24th international conference on computer supported cooperative work in design. CSCWD, 2021, p. 552–7. http://dx.doi.org/10.1109/CSCWD49262.2021.9437738.

[73] Google G. Federated learning: Collaborative machine learning without centralized training data. 2017, URL: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

[74] Zhao Y, Li M, Lai L, Suda N, Civin D, Chandra V. Federated learning with non-IID data. 2018, arXiv:1806.00582.

[75] Li Q, Diao Y, Chen Q, He B. Federated learning on non-IID data silos: An experimental study. 2021, arXiv:2102.02079.

[76] Li X, Jiang M, Zhang X, Kamp M, Dou Q. FedBN: Federated learning on non-IID features via local batch normalization. 2021, arXiv:2102.07623.

[77] T. Dinh C, Tran N, Nguyen J. Personalized federated learning with moreau envelopes. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H, editors. Advances in neural information processing systems. vol. 33, Curran Associates, Inc.; 2020, p. 21394–405.

[78] Voigt P, Von dem Bussche A. The eu general data protection regulation (gdpr). A Pr Guid 1st Ed Cham: Springer Int Publ 2017;10(3152676):10–5555.

[79] Watts D, Casanovas P. Privacy and data protection in Australia: A critical overview. In: W3C workshop on privacy and linked data. World Wide Web Consortium; 2019.

[80] Udoh V, Olajide DA. Data protection in Nigeria: A review of the Nigerian data protection regulation 2019 and the need for Nigeria to sign the african convention on data protection. 2021, Available At SSRN 3766819.

[81] May C, Sell S. Intellectual property rights: a critical history. Lynne Rienner Publishers; 2006.

[82] Qu S, Wang Z, Qin Z, Xu Y, Cong Z, Liu Z. Internet of things infrastructure based on fast, high spatial resolution, and wide measurement range distributed optic-fiber sensors. IEEE Internet Things J 2022;9(4):2882–9. http://dx.doi.org/10.1109/JIOT.2021.3094272.

[83] Cook J, Kirkby R, Booth M, Foster K, Clarke D, Young G. The noise and crosstalk environment for ADSL and VDSL systems. IEEE Commun Mag 1999;37(5):73–8. http://dx.doi.org/10.1109/35.762859.

[84] Alazab M, RM SP, Parimala M, Maddikunta PKR, Gadekallu TR, Pham Q-V. Federated learning for cybersecurity: Concepts, challenges, and future directions. IEEE Trans Ind Inform 2021;18(5):3501–9.

[85] Mostafa H. Robust federated learning through representation matching and adaptive hyper-parameters. 2019, arXiv E-Prints arXiv:1912.13075. arXiv:1912.13075.

[86] Li Q, He B, Song D. Model-contrastive federated learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021, p. 10713–22.

[87] Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: Concept and applications. ACM Trans Intell Syst Technol 2019;10(2). http://dx.doi.org/10.1145/3298981.

[88] Predd J, Kulkarni S, Poor H. Distributed learning in wireless sensor networks. IEEE Signal Process Mag 2006;23(4):56–69. http://dx.doi.org/10.1109/MSP.2006.1657817.

[89] Ju C, Zhao R, Sun J, Wei X, Zhao B, Liu Y, Li H, Chen T, Zhang X, Gao D, Tan B, Yu H, Jin Y. Privacy-preserving technology to help millions of people: Federated prediction model for stroke prevention. 2020, CoRR abs/2006.10517. URL: https://arxiv.org/abs/2006.10517. arXiv:2006.10517.

[90] Bonawitz K, Ivanov V, Kreuter B, Marcedone A, McMahan HB, Patel S, Ramage D, Segal A, Seth K. Practical secure aggregation for federated learning on user-held data. 2016, arXiv preprint arXiv:1611.04482.

[91] Fereidooni H, Marchal S, Miettinen M, Mirhoseini A, Möllering H, Nguyen TD, Rieger P, Sadeghi A-R, Schneider T, Yalame H, et al. SAFELearn: Secure aggregation for private federated learning. In: 2021 IEEE security and privacy workshops. SPW, IEEE; 2021, p. 56–62.

[92] Kadhe S, Rajaraman N, Koyluoglu OO, Ramchandran K. Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning. 2020, arXiv preprint arXiv:2009.11248.

[93] Baer T, Kamalnath V. Controlling machine-learning algorithms and their biases. McKinsey Insights 2017.

[94] Song C, Ristenpart T, Shmatikov V. Machine learning models that remember too much. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. CCS '17, New York, NY, USA: Association for Computing Machinery; 2017, p. 587–601. http://dx.doi.org/10.1145/3133956.3134077.

[95] Song L, Mittal P. Systematic evaluation of privacy risks of machine learning models. In: 30th USeNIX security symposium (USeNIX security 21). USENIX Association; 2021, p. 2615–32, URL: https://www.usenix.org/conference/usenixsecurity21/presentation/song.

[96] Leino K, Fredrikson M. Stolen memories: Leveraging model memorization for calibrated White-Box membership inference. In: 29th USeNIX security symposium (USeNIX security 20). USENIX Association; 2020, p. 1605–22.

[97] Dwork C, Roth A, et al. The algorithmic foundations of differential privacy. Found Trends® Theor Comput Sci 2014;9(3–4):211–407.

[98] Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. CCS'16,, ACM; 2016, http://dx.doi.org/10.1145/2976749.2978318.

[99] Ouadrhiri AE, Abdelhadi A. Differential privacy for deep and federated learning: A survey. IEEE Access 2022;10:22359–80. http://dx.doi.org/10.1109/ACCESS.2022.3151670.

[100] Qu L, Balachandar N, Zhang M, Rubin D. Handling data heterogeneity with generative replay in collaborative learning for medical imaging. Med Image Anal 2022;78:102424. http://dx.doi.org/10.1016/j.media.2022.102424, URL: https://www.sciencedirect.com/science/article/pii/S1361841522000755.

[101] LLC G. Federated learning: Collaborative machine learning without centralized training data. 2017, URL: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html.

[102] Mills J, Hu J, Min G. Communication-efficient federated learning for wireless edge intelligence in IoT. IEEE Internet Things J 2019;7(7):5986–94.

[103] Thapa C, Arachchige PCM, Camtepe S, Sun L. Splitfed: When federated learning meets split learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 36, 2022, p. 8485–93, 8.

[104] Wu X, Zhang Y, Shi M, Li P, Li R, Xiong NN. An adaptive federated learning scheme with differential privacy preserving. Future Gener Comput Syst 2022;127:362–72. http://dx.doi.org/10.1016/j.future.2021.09.015, URL: https://www.sciencedirect.com/science/article/pii/S0167739X21003617.

[105] Nguyen DC, Ding M, Pathirana PN, Seneviratne A, Li J, Vincent Poor H. Federated learning for internet of things: A comprehensive survey. IEEE Commun Surv & Tutor 2021;23(3):1622–58. http://dx.doi.org/10.1109/COMST.2021.3075439.

[106] Huang H, Zeng C, Zhao Y, Min G, Zhu Y, Miao W, Hu J. Scalable orchestration of service function chains in NFV-enabled networks: A federated reinforcement learning approach. IEEE J Sel Areas Commun 2021;39(8):2558–71. http://dx.doi.org/10.1109/JSAC.2021.3087227.

[107] Grafberger A, Chadha M, Jindal A, Gu J, Gerndt M. FedLess: Secure and scalable federated learning using serverless computing. In: 2021 IEEE international conference on big data (big data). 2021, p. 164–73. http://dx.doi.org/10.1109/BigData52589.2021.9672067.

[108] Amro SA. Securing internet of things devices with federated learning: A privacy-preserving approach for distributed intrusion detection. Comput Mater Contin http://dx.doi.org/10.32604/cmc.2025.063734, URL: http://www.techscience.com/cmc/online/detail/23058.

[109] Li P, Wang H, Tian G, Fan Z. A cooperative intrusion detection system for the internet of things using convolutional neural networks and black hole optimization. Sensors 2024;24(15). http://dx.doi.org/10.3390/s24154766, URL: https://www.mdpi.com/1424-8220/24/15/4766.

[110] Mahalingam A, Perumal G, Subburayalu G, Albathan M, Altameem A, Almakki RS, Hussain A, Abbas Q. ROAST-IoT: A novel range-optimized attention convolutional scattered technique for intrusion detection in IoT networks. Sensors 2023;23(19). http://dx.doi.org/10.3390/s23198044, URL: https://www.mdpi.com/1424-8220/23/19/8044.

[111] Lazzarini R, Tianfield H, Charissis V. Federated learning for IoT intrusion detection. AI 2023;4(3):509–30. http://dx.doi.org/10.3390/ai4030028, URL: https://www.mdpi.com/2673-2688/4/3/28.

[112] Bella K, Guezzaz A, Benkirane S, Azrour M, Fouad Y, S. Benyeogor M, Innab N. An efficient intrusion detection system for IoT security using CNN decision forest. PeerJ Comput Sci 2024;10:e2290. http://dx.doi.org/10.7717/peerj-cs.2290.

[113] Almotairi A, Atawneh S, Khashan OA, and NMK. Enhancing intrusion detection in IoT networks using machine learning-based feature selection and ensemble models. Syst Sci Control Eng 2024;12(1):2321381. http://dx.doi.org/10.1080/21642583.2024.2321381, arXiv:https://doi.org/10.1080/21642583.2024.2321381.

[114] Ashraf J, Raza GM, Kim B-S, Wahid A, Kim H-Y. Making a real-time IoT network intrusion-detection system (INIDS) using a realistic BoT–IoT dataset with multiple machine-learning classifiers. Appl Sci 2025;15(4). http://dx.doi.org/10.3390/app15042043, URL: https://www.mdpi.com/2076-3417/15/4/2043.

[115] Molina RS, Ninkovic V, Vukobratovic D, Crespo ML, Zennaro M. Efficient split learning LSTM models for FPGA-based edge IoT devices. 2025, arXiv preprint arXiv:2502.08692.

[116] Wu W, Yang P, Zhang W, Zhou C, Shen XS. Accuracy-guaranteed collaborative DNN inference in industrial IoT via deep reinforcement learning. IEEE Trans Ind Inform 2023;17(7):4988–98. http://dx.doi.org/10.1109/TII.2020.3046128.

[117] Wulfert L, Wiede C, Grabmaier A. Tinyfl: On-device training, communication and aggregation on a microcontroller for federated learning. In: NeurIPS 2020 workshop on tinyML. 2020.

[118] Subramanian S, Tamilselvan S. Secure android IoT mobile and collaborative machine learning for controlling the management of enterprise. J Control Decis 2022;11(1):15–25. http://dx.doi.org/10.1080/23307706.2022.2067253.

[119] Alazab M, RM SP, M P, Maddikunta PKR, Gadekallu TR, Pham Q-V. Federated learning for cybersecurity: Concepts, challenges, and future directions. IEEE Trans Ind Inform 2022;18(5):3501–9. http://dx.doi.org/10.1109/TII.2021.3119038.

[120] Liu P, Xu X, Wang W. Threats, attacks and defenses to federated learning: Issues, taxonomy and perspectives. Cybersecurity 2022;5(1). http://dx.doi.org/10.1186/s42400-021-00105-6.

[121] Tolpegin V, Truex S, Gursoy ME, Liu L. Data poisoning attacks against federated learning systems. In: Computer security–ESORICS 2020: 25th European symposium on research in computer security, ESORICS 2020, guildford, UK, September 14–18, 2020, proceedings, part i 25. Springer; 2020, p. 480–501.

[122] Huang H, Mu J, Gong NZ, Li Q, Liu B, Xu M. Data poisoning attacks to deep learning based recommender systems. In: Proceedings 2021 network and distributed system security symposium. Internet Society; 2021, http://dx.doi.org/10.14722/ndss.2021.24525.

[123] Wang J, Xia Z, Chen Y, Hu C, Yu F. Intrusion detection framework based on homomorphic encryption in AMI network. Front Phys 2022;10:1326.

[124] Cao D, Chang S, Lin Z, Liu G, Sun D. Understanding distributed poisoning attack in federated learning. In: 2019 IEEE 25th international conference on parallel and distributed systems. ICPADS, 2019, p. 233–9. http://dx.doi.org/10.1109/ICPADS47876.2019.00042.

[125] Zhou X, Xu M, Wu Y, Zheng N. Deep model poisoning attack on federated learning. Futur Internet 2021;13(3). http://dx.doi.org/10.3390/fi13030073, URL: https://www.mdpi.com/1999-5903/13/3/73.

[126] Zhao L, Hu S, Wang Q, Jiang J, Shen C, Luo X, Hu P. Shielding collaborative learning: Mitigating poisoning attacks through client-side detection. IEEE Trans Dependable Secur Comput 2021;18(5):2029–41. http://dx.doi.org/10.1109/TDSC.2020.2986205.

[127] Pillutla K, Kakade SM, Harchaoui Z. Robust aggregation for federated learning. IEEE Trans Signal Process 2022;70:1142–54.

[128] Trusted-AI. Trusted-AI/adversarial-robustness-toolbox. 2021, URL: https://github.com/Trusted-AI/adversarial-robustness-toolbox.

[129] Bagdasaryan E, Veit A, Hua Y, Estrin D, Shmatikov V. How to backdoor federated learning. In: International conference on artificial intelligence and statistics. PMLR; 2020, p. 2938–48.

[130] Wang J, Pal A, Yang Q, Kant K, Zhu K, Guo S. Collaborative machine learning: Schemes, robustness, and privacy. IEEE Trans Neural Netw Learn Syst 2023;34(12):9625–42. http://dx.doi.org/10.1109/TNNLS.2022.3169347.

[131] Lyu L, Yu H, Zhao J, Yang Q. Threats to federated learning. In: Yang Q, Fan L, Yu H, editors. Federated learning: privacy and incentive. Cham: Springer International Publishing; 2020, p. 3–16. http://dx.doi.org/10.1007/978-3-030-63076-8_1.

[132] Bagdasaryan E, Veit A, Hua Y, Estrin D, Shmatikov V. How to backdoor federated learning. 2018, http://dx.doi.org/10.48550/ARXIV.1807.00459, URL: https://arxiv.org/abs/1807.00459.

[133] Li S, Cheng Y, Wang W, Liu Y, Chen T. Learning to detect malicious clients for robust federated learning. 2020, ArXiv abs/2002.00211.

[134] McMahan HB, Moore E, Ramage D, Hampson S, Arcas BAy. Communication-efficient learning of deep networks from decentralized data. JMLR: W & CP 2017;54. http://dx.doi.org/10.48550/ARXIV.1602.05629, URL: https://arxiv.org/abs/1602.05629.

[135] Deng Y, Chen T, Xu Z, Zhang Y, Su W, Lan G. Adaptive federated optimization. 2019, arXiv preprint arXiv:1910.10033.

[136] Konečnỳ J, McMahan HB, Yu FX, Richtárik P, Suresh AT, Bacon D. Federated learning and meta learning: Application to distributed optimization. In: International conference on learning representations. 2016.

[137] Zhang X, Li Y, Li W, Guo K, Shao Y. Personalized federated learning via variational Bayesian inference. In: Chaudhuri K, Jegelka S, Song L, Szepesvari C, Niu G, Sabato S, editors. Proceedings of the 39th international conference on machine learning. Proceedings of machine learning research, vol. 162, PMLR; 2022, p. 26293–310, URL: https://proceedings.mlr.press/v162/zhang22o.html.

[138] Liu J, Wang JH, Rong C, Xu Y, Yu T, Wang J. FedPA: An adaptively partial model aggregation strategy in federated learning. Comput Netw 2021;199:108468. http://dx.doi.org/10.1016/j.comnet.2021.108468, URL: https://www.sciencedirect.com/science/article/pii/S1389128621004199.

[139] Casella B, Esposito R, Cavazzoni C, Aldinucci M. Benchmarking fedavg and fedcurv for image classification tasks. 2023, arXiv preprint arXiv:2303.17942.

[140] Yavanoglu O, Aydos M. A review on cyber security datasets for machine learning algorithms. In: 2017 IEEE international conference on big data (big data). IEEE; 2017, p. 2186–93.

[141] Tian P, Liao W, Yu W, Blasch E. WSCC: A weight-similarity-based client clustering approach for non-IID federated learning. IEEE Internet Things J 2022;9(20):20243–56. http://dx.doi.org/10.1109/JIOT.2022.3175149.

[142] Xiong Z, Cai Z, Takabi D, Li W. Privacy threat and defense for federated learning with non-i.i.d. Data in aIoT. IEEE Trans Ind Inform 2022;18(2):1310–21. http://dx.doi.org/10.1109/TII.2021.3073925.

[143] Hwang H, Jung G, Sohn K, Park S. A study on MITM (man in the middle) vulnerability in wireless network using 802.1X and EAP. In: 2008 International Conference on Information Science and Security (ICISS 2008). 2008, p. 164–70. http://dx.doi.org/10.1109/ICISS.2008.10.

[144] Stojmenovic I, Wen S. The fog computing paradigm: Scenarios and security issues. In: 2014 federated conference on computer science and information systems. IEEE; 2014, p. 1–8.

[145] Bellavista P, Foschini L, Mora A. Decentralised learning in federated deployment environments: A system-level survey. ACM Comput Surv 2021;54(1). http://dx.doi.org/10.1145/3429252.

[146] Lu Y, Huang X, Dai Y, Maharjan S, Zhang Y. Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. IEEE Trans Ind Inform 2019;16(6):4177–86.

[147] Shayan M, Fung C, Yoon CJM, Beschastnikh I. Biscotti: A blockchain system for private and secure federated learning. IEEE Trans Parallel Distrib Syst 2021;32(7):1513–25. http://dx.doi.org/10.1109/TPDS.2020.3044223.

[148] Sattler F, Wiedemann S, Müller K-R, Samek W. Robust and communication-efficient federated learning from non-iid data. IEEE Trans Neural Netw Learn Syst 2019;31(9):3400–13.

[149] McMahan HB, Moore E, Ramage D, y Arcas BA. Federated learning of deep networks using model averaging. 2016, CoRR abs/1602.05629. URL: http://arxiv.org/abs/1602.05629. arXiv:1602.05629.

[150] Karimireddy SP, Kale S, Mohri M, Reddi S, Stich S, Suresh AT. SCAFFOLD: Stochastic controlled averaging for federated learning. In: III HD, Singh A, editors. Proceedings of the 37th international conference on machine learning. Proceedings of machine learning research, vol. 119, PMLR; 2020, p. 5132–43, URL: https://proceedings.mlr.press/v119/karimireddy20a.html.

[151] Venkatesha Y, Kim Y, Park H, Li Y, Panda P. Addressing client drift in federated continual learning with adaptive optimization. 2022, arXiv:2203.13321.

[152] Yan Y, Feng C-M, Ye M, Zuo W, Li P, Goh RSM, Zhu L, Chen CLP. Rethinking client drift in federated learning: A logit perspective. 2023, arXiv:2308.10162.

[153] Yuan X, Li P. On convergence of FedProx: Local dissimilarity invariant bounds, non-smoothness and beyond. Adv Neural Inf Process Syst 2022;35:10752–65.

[154] Nishio T, Yonetani R. Client selection for federated learning with heterogeneous resources in mobile edge. In: ICC 2019 - 2019 IEEE international conference on communications. ICC, 2019, p. 1–7. http://dx.doi.org/10.1109/ICC.2019.8761315.

[155] Zhu Z, Hong J, Zhou J. Data-free knowledge distillation for heterogeneous federated learning. In: Meila M, Zhang T, editors. Proceedings of the 38th international conference on machine learning. Proceedings of machine learning research, vol. 139, PMLR; 2021, p. 12878–89, URL: https://proceedings.mlr.press/v139/zhu21b.html.

[156] Zeng D, Liang S, Hu X, Wang H, Xu Z. FedLab: A flexible federated learning framework. J Mach Learn Res 2023;24:100–1.

[157] Kholod I, Yanaki E, Fomichev D, Shalugin E, Novikova E, Filippov E, Nordlund M. Open-source federated learning frameworks for IoT: A comparative review and analysis. Sensors 2021;21(1). http://dx.doi.org/10.3390/s21010167, URL: https://www.mdpi.com/1424-8220/21/1/167.

[158] Beutel DJ, Topal T, Mathur A, Qiu X, Fernandez-Marques J, Gao Y, Sani L, Li KH, Parcollet T, de Gusmão PPB, et al. Flower: A friendly federated learning research framework. 2020, arXiv preprint arXiv:2007.14390.

[159] Foley P, Sheller MJ, Edwards B, Pati S, Riviera W, Sharma M, Moorthy PN, Wang S-h, Martin J, Mirhaji P, et al. OpenFL: The open federated learning library. Phys Med Biol 2022;67(21):214001.

[160] Substra O. Substra - open source federated learning software for healthcare research. 2023, URL: https://www.owkin.com/substra.

[161] Ludwig H, Baracaldo N, Thomas G, Zhou Y, Anwar A, Rajamoni S, Ong Y, Radhakrishnan J, Verma A, Sinn M, et al. Ibm federated learning: An enterprise framework white paper v0. 1. 2020, arXiv preprint arXiv:2007.10987.

[162] Zama. Concrete ML: A privacy-preserving machine learning library using fully homomorphic encryption for data scientists. 2022, https://github.com/zama-ai/concrete-ml.

[163] Ziller A, Trask A, Lopardo A, Szymkow B, Wagner B, Bluemke E, Nounahon J-M, Passerat-Palmbach J, Prakash K, Rose N, et al. Pysyft: A library for easy federated learning. Fed Learn Syst: Towar Next- Gener AI 2021;111–39.

[164] ltd N. Federated learning powered by nvidia clara. 2022, URL: https://developer.nvidia.com/blog/federated-learning-clara/,

[165] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. 2016, arXiv preprint arXiv:1603.04467.

[166] Saadat H, Aboumadi A, Mohamed A, Erbad A, Guizani M. Hierarchical federated learning for collaborative IDS in IoT applications. In: 2021 10th mediterranean conference on embedded computing. MECO, 2021, p. 1–6. http://dx.doi.org/10.1109/MECO52532.2021.9460304.

[167] Kelli V, Argyriou V, Lagkas T, Fragulis G, Grigoriou E, Sarigiannidis P. IDS for industrial applications: A federated learning approach with active personalization. Sensors 2021;21(20). http://dx.doi.org/10.3390/s21206743, URL: https://www.mdpi.com/1424-8220/21/20/6743.

[168] Li B, Wu Y, Song J, Lu R, Li T, Zhao L. DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems. IEEE Trans Ind Informatics 2021;17(8):5615–24. http://dx.doi.org/10.1109/TII.2020.3023430.

[169] Qin Q, Poularakis K, Leung KK, Tassiulas L. Line-speed and scalable intrusion detection at the network edge via federated learning. In: 2020 IFIP networking conference (networking). 2020, p. 352–60.

[170] Abdel-Basset M, Moustafa N, Hawash H, Razzak I, Sallam KM, Elkomy OM. Federated intrusion detection in blockchain-based smart transportation systems. IEEE Trans Intell Transp Syst 2022;23(3):2523–37. http://dx.doi.org/10.1109/TITS.2021.3119968.

[171] Preuveneers D, Rimmer V, Tsingenopoulos I, Spooren J, Joosen W, Ilie-Zudor E. Chained anomaly detection models for federated learning: An intrusion detection case study. Appl Sci 2018;8(12). http://dx.doi.org/10.3390/app8122663, URL: https://www.mdpi.com/2076-3417/8/12/2663.

[172] Cetin B, Lazar A, Kim J, Sim A, Wu K. Federated wireless network intrusion detection. In: 2019 IEEE international conference on big data (big data). 2019, p. 6004–6. http://dx.doi.org/10.1109/BigData47090.2019.9005507.

[173] Lazzarini R, Tianfield H, Charissis V. Federated learning for IoT intrusion detection. AI 2023;4(3):509–30.

[174] Karimireddy SP, Kale S, Mohri M, Reddi S, Stich S, Suresh AT. Scaffold: Stochastic controlled averaging for federated learning. In: International conference on machine learning. PMLR; 2020, p. 5132–43.

[175] Lin T, Kong L, Stich SU, Jaggi M. Ensemble distillation for robust model fusion in federated learning. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H, editors. Advances in neural information processing systems. vol. 33, Curran Associates, Inc.; 2020, p. 2351–63.

[176] Yurochkin M, Agarwal M, Ghosh S, Greenewald K, Hoang TN, Khazaeni Y. Bayesian nonparametric federated learning of neural networks. 2019, http://dx.doi.org/10.48550/ARXIV.1905.12022, URL: https://arxiv.org/abs/1905.12022.

[177] Sahu AK, Li T, Sanjabi M, Zaheer M, Talwalkar A, Smith V. On the convergence of federated optimization in heterogeneous networks. 2018, CoRR abs/1812.06127. URL: http://arxiv.org/abs/1812.06127. arXiv:1812.06127.

[178] Grau MM, Centelles RP, Freitag F. On-device training of machine learning models on microcontrollers with a look at federated learning. In: Proceedings of the conference on information technology for social good. GoodIT '21, New York, NY, USA: Association for Computing Machinery; 2021, p. 198–203. http://dx.doi.org/10.1145/3462203.3475896.

[179] Llisterri Giménez N, Monfort Grau M, Pueyo Centelles R, Freitag F. On-device training of machine learning models on microcontrollers with federated learning. Electronics 2022;11(4). http://dx.doi.org/10.3390/electronics11040573, URL: https://www.mdpi.com/2079-9292/11/4/573.

[180] Veluvarthi R, Rameswarapu A, Kalyan KS, Piri J, Acharya B. Security and privacy threats of IoT devices: A & short review. In: 2023 4th international conference on signal processing and communication. ICSPC, IEEE; 2023, p. 32–7.

[181] Wen D, Jeon K-J, Huang K. Federated dropout—A simple approach for enabling federated learning on resource constrained devices. IEEE Wirel Commun Lett 2022;11(5):923–7. http://dx.doi.org/10.1109/LWC.2022.3149783.