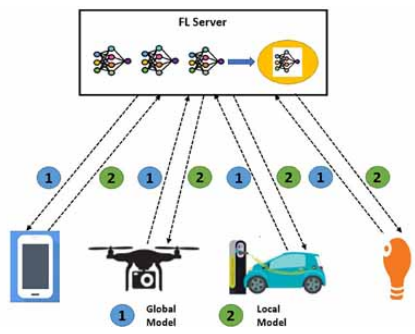
IoT Malware Analysis Using Federated Learning: A Comprehensive Survey

Publisher: IEEE

Cite This PDF

Madumitha Venkatasubramanian ; Arash Habibi Lashkari ; Saqib Hakak All Authors



Topology of Federated Learning.

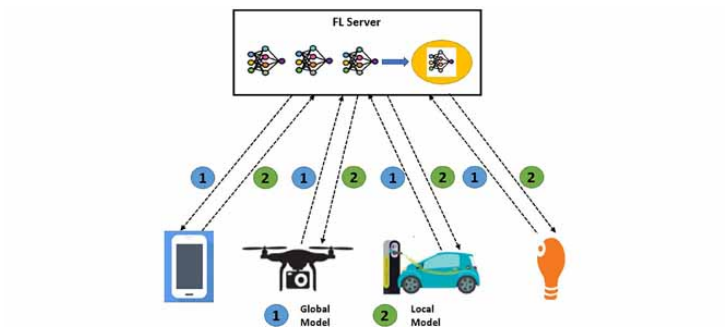
Abstract: The Internet of Things (IoT) has paved the way to a highly connected society where all things are interconnected and exchanging information has become more accessible thr... **View more**

Metadata

Abstract:

The Internet of Things (IoT) has paved the way to a highly connected society where all things are interconnected and exchanging information has become more accessible through the internet. With the use of IoT devices, the threat of malware has increased rapidly. The increased number of existing and new malware variants has made protecting IoT devices and networks challenging. The malware can hide in the systems and disables its activity when there are attempts to discover and detect them. With technological advances, there are various emerging techniques to address this problem. However, they still encounter issues concerning the privacy and security of the user's data and suffer from a single point of failure. To address this issue, there are recent research developments conducted to use Federated Learning (FL). FL is a decentralized technique that trains the user's data on-device and exchanges the parameters without sharing the user's data. FL is implemented to secure the user's data, provide safe and accurate models, and prevent the single point of failure in the centralized models. This paper provides an overview of different approaches

that integrate FL with IoT. Finally, we discuss the applications of FL, the research challenges, and future research directions.



Topology of Federated Learning.

Published in: IEEE Access (Volume: 11)

Page(s): 5004 - 5018

INSPEC Accession Number: 22538165

Date of Publication: 09 January 2023

DOI: 10.1109/ACCESS.2023.3235389

Electronic ISSN: 2169-3536

Publisher: IEEE

Funding Agency:

Contents

SECTION I.
Introduction

The Internet of Things (IoT) is an emerging technology rapidly evolving communication. IoT permits exchanging of meaningful information and knowledge across IoT devices and systems to create value for humans [1]. IoT involves billions of various devices connected, generating vast amounts of data [2]. These devices may include sensors, smartphones, computers, or home appliances. These devices are connected to the internet and each other through heterogeneous access networks [2]. It can be described as connecting devices or things across the internet to send or receive data [3]. IoT devices are used in several industries and have proven helpful for remote health monitoring, early diagnosis, and elderly care for the healthcare sector [4] and reducing the necessity to meet with doctors in person [5]. The research on agriculture using IoT has also risen in the last two decades around crop, soil, and microclimate monitoring [6]. In addition, several industries utilize IoT devices, including the finance sector [7], retail [8], vehicle monitoring systems [9] and the manufacturing industry [10].

A. Motivation

Today there are various industries reliant on IoT devices, and there is an increase in security issues faced by the industries. With the design complexity and lack of security due to outdated firmware and weak authentication, IoT devices are targeted by cybercriminals, and malware compromises IoT devices [11]. Therefore, it is critical to improve the security and privacy aspects of IoT devices and protect them against malware. Many types of research and studies are conducted to protect IoT devices against malware, and one such method is to use centralized techniques such as Machine Learning [1], [12] and Deep Learning [13], [14]. However, these centralized learning techniques share the user's private data with a centralized server to train the models. Therefore, these techniques affect the user's privacy. Recent advancements using Federated Learning (FL) to protect the user's data and provide a secure system are on the rise. The main idea behind FL is to train the user data on-device without sharing its private data to a central server, as present in centralized learning techniques. In this paper, we discuss the applications of FL and provide an overview of

different approaches to integrate FL and IoT for malware analysis. The inclusion and exclusion criteria for the paper used for analysis and comparisons are given in Table 1.

TABLE 1 Inclusion and Exclusion Criteria

Inclusion	Exclusion
Studies that propose, analyze, design architectures and models for IoT, FL and malware analysis	Studies in language other than English are not considered.
Studies subject to peer review i.e., journal papers, conference proceedings, book chapters and workshop papers are included	Incomplete and duplicate studies are not included.
Studies written in English language and available online in full-text online.	Types of publications other than journal papers, conference proceedings, book chapters and workshop papers are not included.
Studies published recently in the last 5 years i.e., 2017 to 2022 are included.	Studies published before 2017 are not included for comparisons and applications of IoT and FL.

B. Organization

In this paper, we discuss in detail and provide an overview of integrating the different approaches of FL with IoT. The remainder of this paper is organized as follows. First, section II discusses the overview of existing studies on Federated Learning and IoT. Then, in section III, we present a detailed discussion on IoT malware along with their taxonomy and nature of IoT malware and types of IoT malware analysis. Later, in section IV, we discuss the recent advances in IoT malware. Next, section V provides a detailed description of malware analysis for Federated Learning. Finally, in section VI, we discuss the research challenges and future research directions, and in section VII, we conclude the paper.

SECTION II.
Overview of Existing Studies on Federated Learning and IoT

Numerous research works are focusing on FL and IoT. For example, the authors of [15] presented a survey focusing on security to protect the vulnerable IoT environment using FL. They also discuss several approaches to address the performance issues, such as accuracy, latency, and resource constraint associated with FL. Similarly, the authors of [16] survey other literature related to the application of FL in healthcare, smart transportation, Unmanned Aerial Vehicles (UAVs), and smart cities. Finally, the paper provides a taxonomy of the FL-IoT services.

Meanwhile, the study in [17] discusses the applications of FL on resource-constrained IoT devices and explores distributed implementations highlighting the drawbacks and their future research directions. The authors of [18] explore and discuss in detail the recent research work on FL-IoT based on criteria such as scalability, robustness, sparsification, security, and privacy. In [19], the authors present a comprehensive survey on Vehicular IoT systems, such as cooperative autonomous driving and intelligent transport systems, with many devices and privacy-sensitive data. The authors of [20] present a literature review on intrusion detection in IoT. They discuss the IoT ecosystem in communication, fog computing, and cloud computing layers. They provide a taxonomy of the potential attacks based on the layers targeted by the attackers.

Several research works have combined FL and blockchain technology to prevent privacy leakage by assigning training tasks to multiple clients. This method separates the central server from the local devices [21]. Another work in [22] presented a comprehensive survey on FL, blockchain, and IoT. It discusses the privacy issues related to blockchain and FL-enabled IoT and possible techniques to tackle threats. The applications of FL also extend to the Industrial Internet of Things (IIoT), where

[23] the authors discuss the aspects of IIoT and FL for privacy, resource constraints, and data management. In addition, there are also personalized FL techniques to tackle the device, data, and model heterogeneities in IoT environments [24]. The comparison of the related works is combined and presented in Table 2.

TABLE 2 Comparison of Existing Techniques for IoT Malware Detection and Contribution of the Paper

Reference	Year	Area	Contribution	Limitations
[19]	2020	FL for Vehicular IoT	A survey on recent advances in vehicular IoT and FL and provides challenges and future research directions.	The paper only focuses on discussing the roles of FL in vehicular network and transportation domain.
[16]	2021	FL-IoT	A survey on different applications of IoT including healthcare, transportation, aerial vehicles and smart transportation.	The paper focuses on only a few of the applications of FL and IoT networks and several other industries such as smart supply chain management are not discussed.
[18]	2021	FL-IoT	A survey on recent research work based on scalability, robustness, sparsification, security and privacy.	The benefits of FL and IoT services such as IoT data sharing, data offloading, localization have not been explored.
[22]	2021	FL-Blockchain-IoT	A survey on privacy issues related to blockchain and FL-enabled IoT and possible techniques to tackle threats.	The paper focuses only on the priacy isses and privacy preservation techniques using blockchain and IoT using FL.
[15]	2022	FL for ML and cybersecurity	A survey on cybersecurity for FL in IoT/CPS environment, where it compares the different datasets and evaluation metrics from different papers.	The paper only focuses on the security aspects while not on the privacy aspects and presents limited applications for FL for IoT.
[17]	2022	FL for Resource- Constrained devices	A survey on application of FL on resource-constrained IoT devices where the paper explores distributed implementations and highlights the drawbacks and its future research directions.	The paper only focuses on resource-constrained IoT devices but its advantages of application on industries are not discussed in detail.
[20]	2022	FL-IoT	A survey on the literature of intrusion detection in IoT environment and provides a taxonomy of potential attacks based on the layer targeted.	The paper only discusses the intrusion detection aspects of IoT and its integration with FL based on several criteria while it does not focus on any of the other aspects.
[23]	2022	FL-IIoT	A survey on IIoT and FL for privacy, resource, and data management.	The paper analysis the use of FL and IIoT integration but does not provide detailed applications of integration of IIoT and FL.
This paper		FL for IoT Malware Analysis	A survey on integrating FL and IoT for malware analysis and discusses the research challenges and future directions.	

SECTION III.
IoT Malware

Most of the malware families are designed to target personal computers running on Microsoft Windows, the most popular operating system. IoT devices are built upon different CPU architectures and have become an attractive target for attackers. The IoT malware performs brute-force attacks to gain access to the devices. The Linux.Hydra was the first DDoS-capable IoT malware that appeared in 2008 [11]. The IoT malware developers developed several variants of Linux.Hydra, including Psybot, Chuck Norris, and Tsunami, emerged in the upcoming years. The Tsunami is the ancestor of Bashlite, and from Bashlite, the Mirai malware inherited and evolved more and more complex in 2016 [11]. In this paper, we discuss in detail the conducive environments for IoT malware execution, the types and nature of IoT malware, and the types of malware analysis in the following sub-sections.

A. Conducive Environment for the Execution of IoT Malware

IoT devices are prone to different attacks, including physical attacks, network-layer attacks, and application-layer attacks. The attacker exploits the vulnerabilities present in the targeted system. There are several reasons for the execution of malware: outdated firmware, weak authentication, connectivity, and resource-constrained devices. We will be discussing the outlined reasons in detail.

- Outdated Firmware - Firmware updates the functionality and features of a device. Usually, outdated firmware does not have security patches if new vulnerabilities are found [25]. Therefore, the attackers can exploit this vulnerability and gain access to the rest of the system.

- Weak Authentication - IoT devices usually have an easy installation procedure for the people to use the devices. The majority of the users either reuse their credentials or do not change the default credentials, which becomes an easy target for attackers [26].

- Connectivity - Many IoT devices are connected to the internet almost always. This creates open ports which attract attackers easily. In addition, most IoT devices are resource-constraint and have fewer security policies.

- Resource-constrained - Most IoT devices, such as smart watches, CCTV cameras, and Bluetooth-operated devices, are resource constrained and heterogeneous, making it easier for attackers to target the system [27].

B. Nature of IoT Malware

The different types of malware have different modes and natures of exploiting the vulnerabilities of the targeted system. For example, the malware can exploit network-based vulnerabilities or use operational business functionality through available network shares [28]. Incident response and malware eradication efforts are challenging when the malware propagates utilizing the infrastructure. Earlier in the section, we discussed the evolution of certain malware. To know more about the nature of the malware and their methods of propagation, in this section, we will discuss in detail one of the most popular IoT botnet, Mirai.

The Mirai botnet consists of five major components [29], and all of these work independently to compromise vulnerable devices and launch massive DDoS attacks. In addition, all these components are distributed geographically, which makes them difficult to track [29]. The following are each of the components explained in detail, along with their functionalities.

1. Bot - A bot is a malicious component in the network; a bot could be any IoT device connected to the network. It is a slave node and acts on the attackers' behalf, taking instructions from the attackers and executing them in the network. Each bot scans for the nearby vulnerable device and reports it to the report server. The bots attempt brute-force attacks using default usernames and passwords.

2. Command and Control Server (C&C) - The C&C server, as we discussed in the earlier section, is the attacker who controls the botnets and sends out instructions that are carried out by the botnets.

3. Report Server - The report server contains information about the vulnerable nodes and their stolen login credentials. The bot communicates this information to the report server when it locates a vulnerable node.

4. Loader - The loader obtains information from the report server and exploits these vulnerabilities to change the node into a bot.

5. Webserver - It hosts the precompiled bot binaries for multiple different architectures. The loader identifies the appropriate architecture and downloads the corresponding binary from the web server [29].

C. Types of IoT Malware Analysis

Malware Analysis is the study of a malware sample's impact, functionalities, origin, and potential. It helps understand the behaviour and purpose of a suspicious file, reduces the false positives, and helps determines how extensive is a malware incident. There are three types of malware analysis: dynamic, static, and hybrid. Each of the techniques has its strengths and weakness compared to the others. The dynamic analysis uses a behaviour-based approach. Compared to static analysis, dynamic analysis is effective against all types of malware. Static analysis is ineffective against sophisticated malware but, compared to dynamic analysis, is cost and time efficient. The static analysis involves file fingerprinting and virus scanning, and it searches the body of the malware for strings [30]. The limitations of static and dynamic analysis inspired researchers to develop a hybrid analysis that involves the benefits of both static and dynamic analysis [31]. This section discusses each technique in detail and provides a taxonomy and its features.

1) Static Analysis

The static approach analyzes and detects malicious files without executing them. In static analysis, the analysts reverse the executable files into assembly code to better understand the malware activities. The significant advantage of using static analysis is that it can observe the malware's structure and scalability. Observing the malware's structure helps explore all the possible execution paths in the malware sample and makes the static approach effective in solving heterogeneous issues in IoT devices. The major drawback of static analysis is that it cannot detect complex and polymorphic malware [15]. The static analysis relies on extracting certain characteristics: Control Flow Graph (CFG), Function Call Graph (FCG), opcodes, strings, and file headers. Then, the assembly code is disassembled [32] using tools like Radare2 [33] and IDA Pro [34]. These characteristic features can be categorized as graph-based features and non-graph-based features. The two types of features are discussed in detail below:

- *Graph-based features:* The Control Flow Graph (CFG) is the most popular feature in graph-based features. CFG is a directed graph representing all the possible execution paths in a

program where each vertex represents a basic block, and each directed edge is the control flow between the blocks [11]. The experimental results in [35] have shown that the IoT malware contains fewer nodes and edges than android malware. The authors of the paper [36] build a detection mechanism for IoT malware using CFGs. The paper shows that the IoT malware has a more significant number of edges despite the smaller number of nodes. In [37], the authors propose preserving the malware's integrity by extracting the CFG of malware as feature information. Here, a packed malware's control flow graph consists of unpacked and local CFG. The paper [38] proposes a new algorithm in the CFG feature based on dynamic programming to efficiently detect the malware with fast processing time.

The next type of graph-based feature is the Function Call Graph (FCG). The FCG is also a directed graph constructed from programs where the vertices specify the functions and the edges define the caller-callee relationship between functions [39]. In the paper [11], there is another type of graph-based feature: the opcode sequence graph. The opcode sequence graph is a graph representation of an executable file as the opcodes have a suitable structure to be represented as a graph [40]. Representing an executable file as a graph allows graph-based implementation methods like graph compression and embedding to distinguish between malware and benign files.

- *Non-graph-based features:* There are several non-graph-based static features, such as opcodes, ELF headers [41], and strings. One of the functionalities most IoT malware supports is the Command & Control server connection [42]. Therefore, there is a high chance that the C&C server and IP address might be available in printable strings of IoT malware binary. In the paper [43], the authors have obtained the statistical, structural, and string features. The statistical features have been obtained using course-grained clustering, the structural features are obtained using fine-grained clustering, and the string features are obtained using N-grams. In one of the survey papers [11], the authors mentioned that the string features that include information such as the IP addresses and URL connect take the least time for feature extraction. For the opcodes, the malware file is decompiled to extract opcodes and utilized for malware classification [42]. The authors in [44] extracted opcode features for malware and benignware using the objdump tool. In the paper [45], the authors have extracted the opcode sequences using fuzzy and fast fuzzy pattern trees.

2) Dynamic Analysis

The dynamic approach monitors the executables during the run-time period and detects abnormal behaviour. It observes behaviour information such as network activities, system calls, file operations, and registry modification records [46]. The dynamic analysis monitors the execution process and is resource-intensive, time-consuming, and expensive for constructing a virtual environment. In some cases, the malware could infect real environments. Although they are resource intensive, the dynamic analysis is effective against all types of malware. The main advantage is that it analyses the run-time behaviour of a program which is hard to obfuscate [47]. Some examples of the features in the dynamic analysis include memory, network, system call sequences, process ID, and parent process ID [46].

In [48], the authors design and implement an automatic, virtual machine-based profiling system to collect IoT malware behaviour, such as API calls and system calls. The method converts multiple sequential data into a family behaviour graph for analysis. The paper [49] proposes a dynamic analysis methodology by preparing an analysis tool and running the malicious samples in a controlled environment to investigate them. Meanwhile, the authors of [50] propose a method for malware classification based on analyzing the sequences of system calls and using an attention-based LSTM model for malware classification. In [51], the paper discusses the techniques performed by malware to evade detection in a dynamic analysis environment.

3) Hybrid Analysis

The hybrid malware analysis integrates both static and dynamic features. In the paper [47], the authors have combined the static and dynamic features to utilize the benefits of both techniques. It utilizes the string features for the static analysis and uses API call sequence extraction for the dynamic analysis. The combination of both features improved detection accuracy compared to the stand-alone techniques. In the paper [32], the authors have used hybrid analysis using an entropy-based method to differentiate packed malware samples from non-packed ones. The authors of [52] use two-stage hybrid malware detection to protect IoT devices from obfuscated malware. The method consists of two stages where after extracting opcode features using static analysis, the benign files are detected using a bidirectional long short-term memory model. In [53], the authors propose to use bidirectional long short-memory (Bi-LSTM) along with a spatial pyramid pooling network (SPP-Net) for smart IoT. The advantage of hybrid analysis is that certain actions that may be hidden in the run-time might be detected while unpacking the binary files or viewing them as assembly code. The detailed taxonomy of the IoT malware analysis can be seen in Figure 1.

SECTION IV.

Recent Advances in IoT Malware Analysis

Researchers have studied IoT Malware Analysis and contributed using several state-of-the-art machine learning and deep learning techniques to detect and classify IoT malware. In this section, we first discuss the research works focussing on machine learning for IoT malware analysis and classification. The authors of the paper [54] have proposed a method for IoT botnet detection using machine learning. In this paper, the authors have integrated static and dynamic features to distinguish IoT botnet samples from benign samples. The proposed method used a Support Vector Machine (SVM) as the machine learning classifier. The authors of the paper [55] proposed a new framework model and a hybrid algorithm for selecting the effective machine learning algorithm among the various algorithms available for effectively detecting IoT malware. The algorithms considered in the paper for the model selection are Bayes Net, Naive Bayes, Random Forest, and Random Tree. Among the other algorithms considered, the Naive Bayes algorithm gave the best results in terms of accuracy.

In the paper [56], the authors have provided a detailed survey of various technological advancements in Machine Learning and their applications for resolving several security issues in IoT. They have also discussed the different potential future research directions. In [57], the authors have proposed a distributed modular solution for IoT malware detection using machine learning. The authors have extracted four different features, including unique IP addresses and minimum, mean and maximum number of packets per destination IP address. The proposed method uses KNN (K nearest neighbours) for classification and obtains an accuracy of 94%.

Meanwhile, the authors in [58] use several machine learning algorithms such as Random Forest (RF), Decision Tree (DT), and KNN to predict attacks and anomalies in IoT network traffic. The system achieved an accuracy of 99%. In [59], the paper's authors attempt to detect unknown malware families using several machine learning techniques such as Naive Bayes, DT, and RF and achieve an accuracy of 98%.

The researchers have also focused on several Deep Learning techniques for IoT malware analysis and classification, such as in [60], [61], [62], and [63]. For example, the authors of [64] propose an approach for Linux IoT botnet detection based on a combination of PSI graphs and a Convolutional Neural Network (CNN). The DGCNN extracts the vertice's local substructure features and defines a vertex ordering. Furthermore, the authors of the paper [65] have compared three Convolutional Neural Network (CNN) approaches for IoT malware detection. In [14], the authors used CNN to detect and classify unknown malware and obtained an accuracy of 99%. Deep Learning has also been used along with visual representation techniques [66] for faster detection and classification of IoT malware. The proposed method in [66] used visual transformation with Binvis and achieved an accuracy of 94.5%. Finally, the paper [67] presents an end-to-end malware detection technique to reduce the time and effort for malware analysts to build static and dynamic features. The method uses CNN and achieves an accuracy of 95.5%. The list of studies using machine learning and deep learning, along with their techniques used and accuracies, have been presented in detail in Table 3.

TABLE 3 Various Studies on IoT Malware Analysis and Detection

Reference	Year	Technique Used	Machine Learning			Deep Learning		
			Acc > 90%	Prec >90%	F1 >90%	Acc > 90%	Prec >90%	F1 >90%
[35]	2018	Graph-Based approach using static analysis	√	×	×	×	×	×
[32]	2018	IoT malware analysis and classification	√	√	√	×	×	×
[36]	2019	Graph-Based approach using static analysis	×	×	×	√	√	√
[38]	2019	Control Flow Graph	√	√	√	×	×	×
[56]	2019	Feature extraction from IoT malware network traffic	√	√	√	×	×	×
[62]	2019	Control Flow Graph	×	×	×	√	√	√
[45]	2020	Dynamic Analysis	×	×	×	√	√	√
[37]	2020	Control Flow Graph	×	×	×	√	√	√
[39]	2020	Function Call Graph from PE files	√	√	×	×	×	×
[40]	2021	Graph-based approach using Opcode sequences	√	√	√	×	×	×
[51]	2021	2 stage hybrid analysis	×	×	×	√	√	√
[61]	2022	Control Flow Graph	×	×	×	√	√	√
[52]	2022	Hybrid Analysis using Bi-LSTM and SPP-Net	×	×	×	√	√	√

SECTION V.

Malware Analysis Using Federated Learning

A massive amount of data is generated from the billions of IoT devices connected and used today. Unfortunately, the exponential growth of IoT devices has also attracted several attackers, and the user data's security and privacy are at risk. Several research works focus on state-of-the-art techniques to detect and classify IoT malware, as discussed in the previous sections. However, all these techniques are centralized, sending the user's data to a centralized server. To protect the user's privacy and security, recent research focuses on decentralized techniques such as Federated Learning (FL), where the user's private data remains on the device, and only the model parameters are shared. In this section, we will focus on the applications of FL for IoT malware analysis and discuss them in detail.

A. Definition of Federated Learning

FL is a new branch of AI where the Machine Learning (ML) models are trained locally on the devices such as mobile phones and other smart devices [23]. The devices present in the FL setup do not exchange their local data but instead shares the parameters and gradients of the local model with a global model maintaining the privacy and security of the user. The global model resides at a server, and the topology of Federated learning is shown in Fig. 2. The global model aggregates all the model updates obtained from the local models by averaging the parameters of each individual model, and by this method, each individual model learns collaboratively from a global model. The workflow of FL is discussed below:

- *Local Model Training:* The local model training occurs in each individual device where the model is fetched with the local data. After the model is trained with the respective local data, a local model is generated.

- *Local Model Updates:* After the generation of the local model, each of the model updates contains the weights and parameters from each individual device. These updates are then sent to the aggregator.

- *Aggregating the Global model:* After receiving these updates from every local model, the global model aggregates each of the updates received by executing aggregation algorithms such as Federated Averaging (FedAvg).

- *Global Model Generation* After the global model is generated by averaging the local model updates, the global models are sent back to the local models. In this way, the individual devices learn collaboratively from a shared model.

There are various categories of FL based on the ways of splitting the data. The categories of FL include Horizontal FL, Vertical FL, and Hybrid FL.

- *Horizontal FL* - The horizontal FL contains different data samples but shares the same feature space. Some examples of Horizontal FL are Next-word prediction, recommendation systems, and wake-word detectors [23]. In [68], the authors have proposed an algorithm to achieve fairness and accuracy to reduce the uneven distribution of data across horizontal FL.

- *Vertical FL* - The Vertical FL shares different sample spaces, but the sample IDs are the same. For example, a grocery store and a bank in the same area might have similar customers, but their business structure (feature space) is different. Since a large amount of data generated from these systems are often low quality, in [69], the authors propose an explainable vertical FL to reduce the computational complexity.

- *Federated Transfer Learning (FTL)* - The FTL combines vertical and horizontal FL. In FTL, different enterprises or institutions can develop personalized models by learning from each other without sharing their own. FTL is applicable when the data samples and feature spaces differ in two clients' datasets [23]. FTL is applied to handle variance in data samples and feature space while performing on-device learning. In [70], the authors propose a semi-supervised FTL to reduce model overfitting due to insufficient overlapping training samples in FL scenarios. Here the proposed method uses non-overlapping samples from all parties to expand the training set for each party to improve local model performance. In [71], the authors propose using FTL for industrial missing data imputations where the knowledge is indirectly transferred to the target edge through helper models. In [72], the authors propose an IoTDefender, an intrusion detection framework for 5G IoT based on federated transfer learning. The IoTDefender aggregates data using federated learning and forms customized detection models using transfer learning. With it, all IoT networks can share information without compromising privacy.

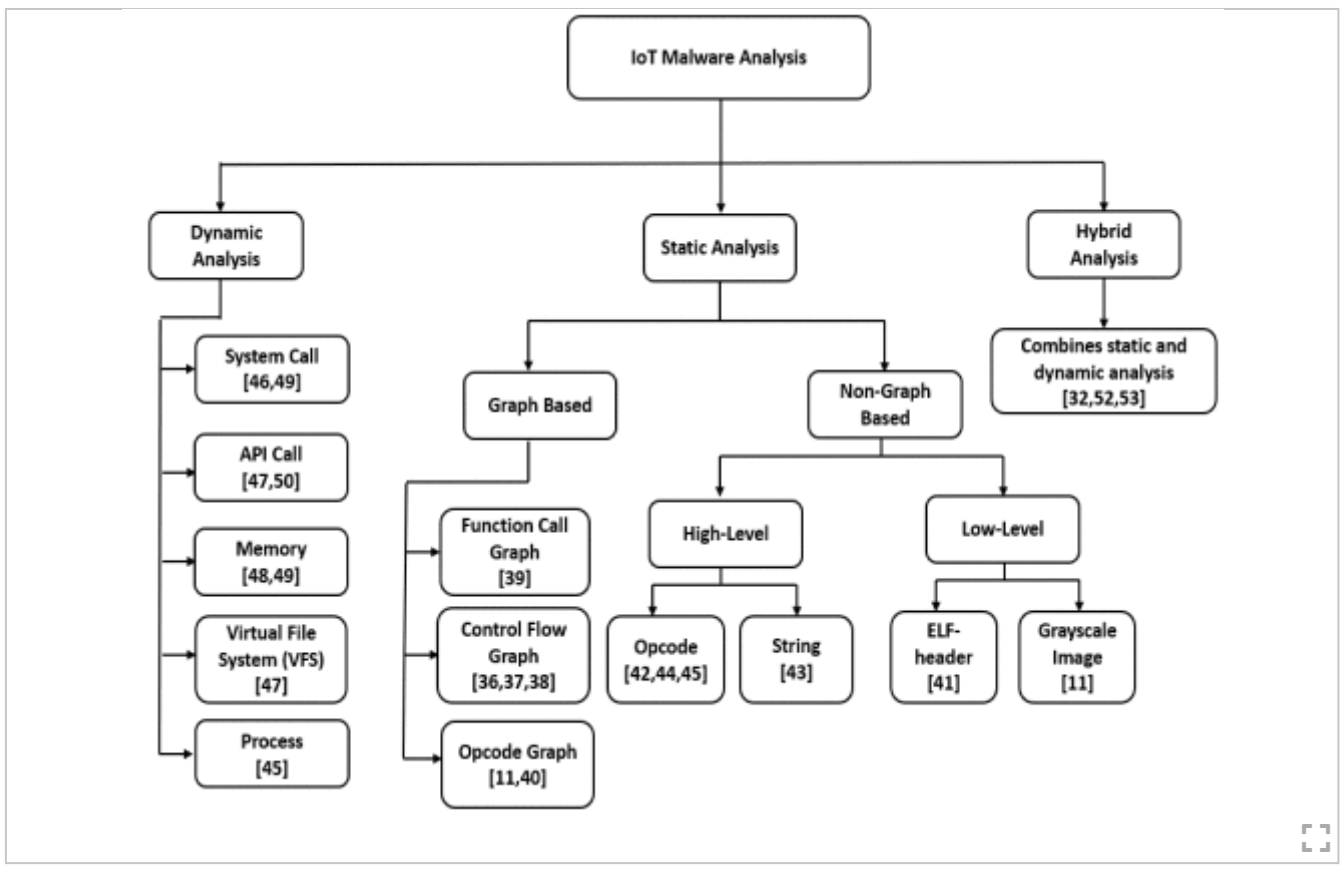


FIGURE 1.
Taxonomy of IoT Malware Analysis.

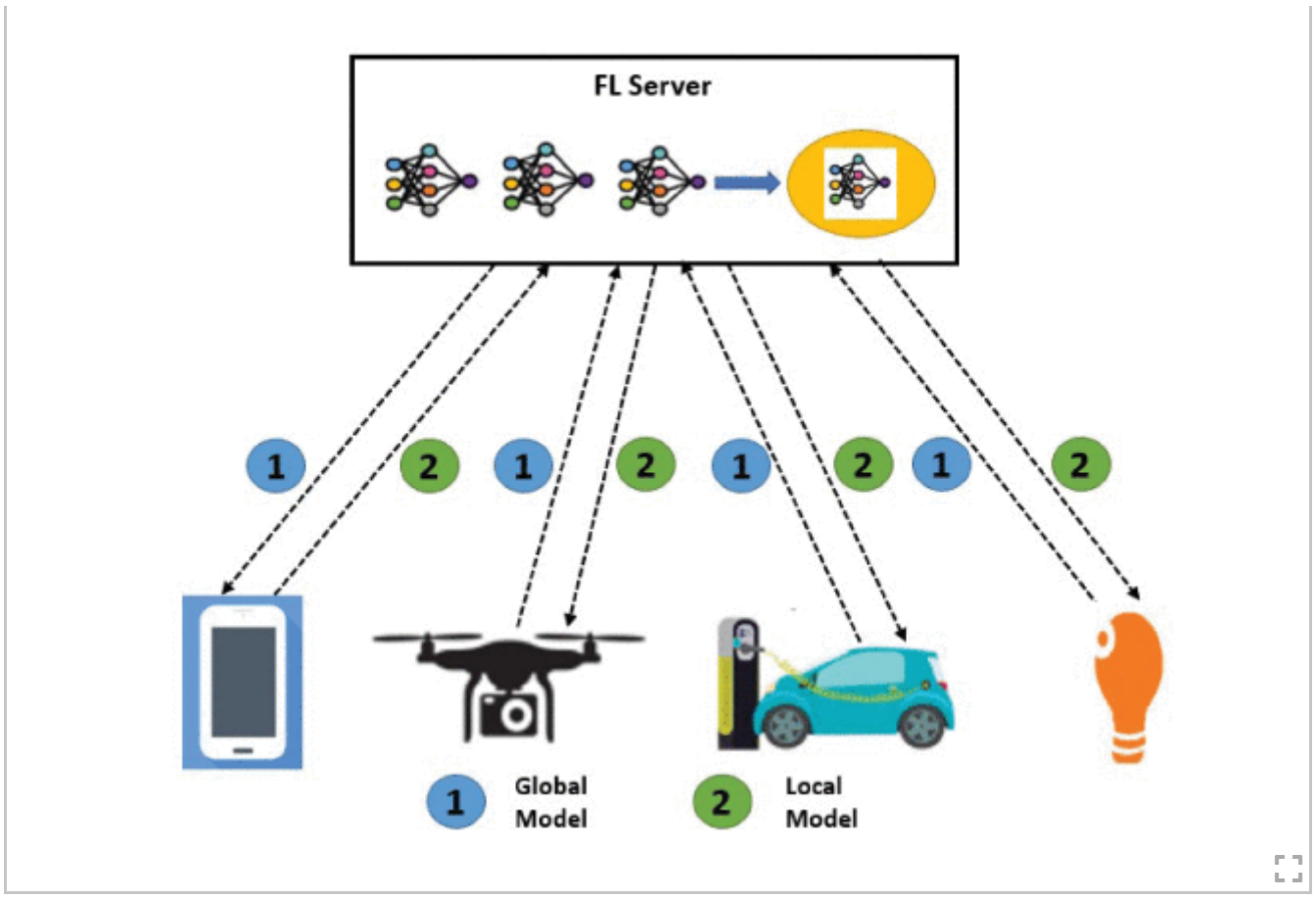


FIGURE 2.
Topology of Federated Learning.

B. Applications of Federated Learning for Malware Analysis

As discussed in the previous sections, several state-of-the-art Machine Learning (ML) and Deep Learning (DL) techniques detect and classify IoT malware. Recent research studies focus on Federated Learning as it has significant advantages over traditional ML and DL models. FL ensures data privacy, security, reduced latency, lower power consumption, and on-device training. In addition, FL also delivers personalized ML models to the users, where the models learn collaboratively and ensure enhanced user experience.

In [73], the authors used the autoencoder model to use FL for botnet detection. Here, the IoT network traffic is collected on an edge device that contains the local model and a virtual worker. The global model aggregates the local model updates and sends them back to the virtual worker to train the new local model with the local data. The method achieved 99% accuracy in classifying the IoT network traffic as benign or malicious. The authors of the paper [74] used a Convolutional Neural Network (CNN) for the asynchronous FL model to select the heterogenous nodes to participate in the global model aggregation. In [75], the authors have used attention-based federated incremental learning for network traffic classification. The proposed method achieved an accuracy of 96%, reducing network failure risk due to long transmission distances between the nodes. The application of FL also extends to IoT healthcare due to the dynamic generation of a large amount of data.

In [76], the authors have used FL for IoT healthcare data to secure data collaboration for the IoT environment and reduce overheads. Furthermore, they have combined blockchain technology and FL to enable a secure architecture for privacy-preserving in smart healthcare. In [77], the authors have used Artificial Neural Network (ANN) as the global model for federated intrusion detection for IoT healthcare applications. The proposed method improved the performance in heterogeneous IoT data and tackled poisoning attacks. The FL application also extends to agricultural IoT where in [78], the authors have used three different global models, including Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Deep Neural Network (DNN) and evaluated three different datasets for intrusion detection in IoT using Hierarchical Federated Learning.

FL has also been used in the field of Unmanned Aerial Vehicles (UAVs). In paper [79], the authors propose to use FL for UAVs where the UAV coordinates are distributed to ground devices for shared model training. Using the UAV's high altitude and mobility, it can proactively establish short-distance line-of-sight links with devices and prevent any device from being a communication straggler. In [80], the authors use FL for UAVs to form a swarm for distributed model training. They also explore the impact of the distance change between the training node of the UAVs and the parameter server UAV on the training accuracy [80]. In [81], the authors use decentralized FL for UAV Networks known as DFL-UN, enabling FL within UAV networks without a central entity. Finally, in [82], the authors use hierarchical FL for edge-enabled UAV networks. Here, the edge-aided UAV network exploits the edge servers located in base stations as intermediate aggregators by employing commonly shared data.

Federated Machine Learning has gained much attention due to how it handles privacy by decentralizing the data generated at the IoT devices and aggregating the global model at the

centralized server [23]. In addition, Federated Machine Learning for searching malware [83] has been used to speed up learning without compromising the data of users. In [84], proposed a method for malware classification using Federated Machine Learning. The authors of [85] have reviewed different research works on Federated Machine Learning regarding multi-level classification, reliable client selection, and resource management. The discussion of research work, along with their contribution, is discussed in detail in Table 4.

TABLE 4 Applications of Federated Learning in IoT

Reference	Year	FL clients	Local Learning model	Dataset Used	Description	Contribution	Accuracy
[73]	2022	IoT Devices	Multi-Layer Perceptron and Auto Encoder	N-BaIoT Dataset	Uses supervised and unsupervised federated learning model to detect malware affecting seen and unseen IoT devices.	The proposed method proves that the use of more diverse and large data increases the model performance in both supervised and unsupervised learning techniques	98.5%
[86]	2022	IoT Devices	Auto Encoder	IoT Botnet	The paper uses a deep autoencoder to detect botnet attacks using on-device decentralized traffic data.	Significant improvement in attack detection and added data security by using an edge layer for each IoT device.	98%
[87]	2020	IoT Devices	CNN	ISCXVPN2016 packet dataset	The paper uses CNN in Federated learning for traffic classification. It performs model training in a non- IID environment as well as dynamic Internet traffic application distribution.	Resolves the problem of a fault-tolerance in client-server communication protocol and operates in an environment where the application to be classified is dynamically added	92%
[77]	2022	IoT Healthcare devices	ANN	Bot IoT Dataset	Proposes a lightweight Artificial Neural Network for federated intrusion detection for IoT Healthcare application.	Improved performance in heterogeneous IoT data and tackles poisoning attacks.	99%
[88]	2022	IoT Devices	CNN	MNIST dataset	Proposes a differentially private self-normalizing neural network combining differential-privacy, self-optimisation and algorithm for client selection.	Improves adversarial robustness without reducing privacy.	96%
[78]	2022	Agricultural IoT	CNN, RNN, DNN	CSE-CIC-IDS2018 dataset, MQTT dataset, InSDN Dataset	Proposes a federated learning-based intrusion detection to prevent agricultural IoT attacks.	The proposed method has an equal or higher detection rate compared to centralized techniques.	98%
[89]	2021	IoT Devices	Neural Network	NSL-KDD	Hierarchical Federated learning with respect to detection accuracy and speed of convergence.	Intrusion detection for IoT systems using Hierarchical Federated Learning proved higher efficiency for two iid and one non-iid client compared to FL.	85%
[24]	2020	IoT Devices	CNN, 3NN	MobiAct Dataset	Personalized Federated learning in cloud-based architecture to tackle device, model, and statistical heterogeneity.	Uses Federated Transfer Learning and Federated Distillation to tackle heterogeneities and outperform centralized learning techniques.	95%
[74]	2021	IoT Devices	CNN	MINIST Dataset	Asynchronous parameter updates for federated learning to improve training efficiency in heterogeneous IoT devices.	Proposed method outperforms in both iid and non-iid data distribution settings and proposes a node selection technique.	80%
[76]	2021	IoT healthcare data	Machine Learning	-	Blockchain and Federated learning enabled secure architecture for privacy-preserving in smart healthcare.	Secured data collaboration for IoT environment with low overheads.	-
[75]	2022	IoT Devices	Unsupervised and Semi-supervised Learning	Moore and SCX-VPN dataset	Propose attention-based Federated Incremental learning to improve the weight of the parameters.	Faster convergence and higher accuracy rate.	98%

To preserve the privacy of the ML models, several techniques are used in FL, and they include differential privacy (DP), homomorphic encryption and secure multi-party computation [23].

1) Homomorphic Encryption

The computation and analysis use several encryption techniques, making it difficult for the attacker to decrypt the user's original information. In [90], the authors use homomorphic encryption in FL for IoT healthcare data to prevent the adversary from inferring private medical data by various attacks, such as model reconstruction attacks or model inversion attacks. Furthermore, in [91], the authors combine homomorphic encryption and Verifiable computing to secure against confidentiality and integrity threats from the aggregation server. Finally, in [92], the authors develop a method for multi-party homomorphic aggregation where the central node only receives an encrypted version of the individual gradients from the local model.

2) Differential Privacy (DP)

Differential privacy determines the amount of data available for third-party analysis. The differential privacy contributes to the adversarial robustness of a machine learning model. In the paper [88], the authors have added a differential privacy noise layer to maintain the privacy characteristics of Federated Learning. In [93], the authors use differential privacy for hierarchical FL based on Local Differential Privacy (LDP). The method involves adding the noise to the shared model parameters before uploading them to edge and cloud servers. In [94], the authors track the privacy loss by accounting for the log moments. Finally, in [95], the authors combine FL and differential privacy approaches based on update optimization of relative-staleness and a semi-synchronous approach for fast convergence in heterogeneous networks. Some of the differential privacy framework's properties are protecting sensitive personal information, privacy protection and group privacy [23].

3) Secure Multi-Party Computation

A model where multiple parties compute and prevent data leakage to third parties. In [96], the authors propose using partially encrypted multi-party computation to reduce the communication and computation cost compared to conventional multi-party computation, and it achieves as high accuracy as traditional distributed learning.

C. Advantages of Integrating Federated Learning and IoT

There are several benefits to integrating FL for IoT malware analysis and in the section, we will discuss them in detail.

1) Data Privacy and Security
To understand the pattern of data, train the data and get insights, centralized learning techniques such as ML and DL algorithms are used. In these techniques, the data of different businesses present in different locations are sent to a central server where all the data are stored and trained. As the IoT application's user data can be sensitive and contain sensitive user information, the centralized techniques can potentially expose data to potential attackers and intruders. In Federated learning, the sensitive user data is not transferred to any central location for training the algorithm but stays on the IoT device, and only the parameters of the model are shared with a central server for collaborative learning.

2) Improved Network Performance
IoT devices require a huge network infrastructure to communicate and handle the data generated from these devices. This potentially affects the performance of the network. In FL, since user data is present in the IoT device and not transferred to a central server the traffic in the network is reduced. This increases the overall performance of the network.

3) Scalability
The conventional ML algorithms fail to scale to the massive amount of data being generated from IoT devices. The integration of FL with IoT enables it to scale the learning as it is not required to train large volumes of data but focuses on the aggregation of model parameters. This improves the scalability of the FL over the other centralized techniques available.

SECTION VI.
Research Challenges and Future Research Directions

The IoT devices are heterogeneous and complex in their design and nature. Although there are several advantages to combining FL and IoT, there are many technical difficulties in implementing and deploying them in real time. In this section, we will discuss the challenges and future research directions in detail and provide a summary of challenges and future directions in Table 5.

TABLE 5 Summary of Research Challenges and Future Directions

Challenges	Causes	Future Direction	References
Device Heterogeneity	Massive number of devices generating different types of data such as CCTV cameras, smart phones, medical sensors etc., are connected to the network.	Future research directions to address the device heterogeneity should include fault tolerance, active device sampling and asynchronous communication.	[74], [97], [98], [99]
Statistical Heterogeneity	With massive number of devices connected the data is non-independently and identically distributed across the network (non-iid) and algorithms such as machine and deep learning are built on assumption of identical and distributed (iid) dataset.	Utilization of clustering based on statistical heterogeneity and adaptive selection of clients for aggregation within the cluster.	[100]
Efficient Data Management	Efficient data management is necessary for transmitting data to the required destination in real-time.	Efficient data management policies is crucial. Research on utilizing blockchain technology is important for efficient data management and privacy of data.	[101], [102]
Server-side attacks	Model updates can be tampered by the attackers, and the attacker may try to steal the model updates from the cloud resulting in inconsistency and noise.	This can be tackled by using privacy-preserving techniques such as homomorphic encryption and differential privacy techniques.	[88], [93], [103]
Client-side attacks	Client-side servers are prone to model-poisoning attacks and data-poisoning attacks. In a model poisoning attack, the attacker might upload poisoned updates, leading to performance degradation and classification errors. In data poisoning attacks, the attackers infiltrate and enter misleading information, tampering with the training of the models	The usage of blockchain technology to verify each of the model updates using miners and ledgers.	[104]
Communication Overhead	FL methods rely on recursively communicating and exchanging model updates throughout the process.	Generating algorithms to reduce communication overheads.	[105]
Resource-Constrained	IoT devices have low bandwidth, power capabilities and limited storage. IoT devices are resource-constrained and may not contain CPU or GPU capabilities to utilize complex ML or DL models	Utilizing resource aware training, virtual workers and edge devices.	[17], [106]

1) Device Heterogeneity
Millions of IoT devices are connected, and integrating these multiple heterogeneous devices is a huge challenge. Their storage, computational capabilities and communication capacities vary significantly. In some cases, it is possible that only some devices are active due to power constraints or due to connectivity issues. So, FL must handle heterogeneous hardware and be robust to dropped devices. In [100], the authors propose an adaptive client sampling algorithm to tackle heterogeneity, and the

proposed system significantly reduces the convergence time compared to several baseline sampling schemes. In [99], the authors leverage Federated Reinforcement Learning to accelerate and stabilize the process with heterogeneous data. There is active research work on the linear convergence in FL for heterogeneous data where the authors in [97] have proposed a method for linear convergence rates under aggressive gradient sparsification and quantified the effect of the compression level on the convergence rate.

In [98], the authors used self-attention-based transformers by replacing CNN to improve FL over heterogeneous data. Some future research directions to address the device heterogeneity should include fault tolerance, active device sampling and asynchronous communication. First, for fault tolerance, a potential approach to solve it could consider dropping the inactive devices and ignoring device failure, which may also lead to biased device sampling. Hence it is essential to consider all aspects while solving fault tolerance. Next is active device sampling; this could be solved by setting a threshold and certain conditions to the number of devices depending on their activity status. This approach could select the participating devices at each FL round. Finally, for asynchronous communication, the paper [74] has proposed a lightweight node selection algorithm to select the nodes to carry out the task efficiently.

2) Statistical Heterogeneity

Since there are different types of devices connected to the network in IoT, the ability of a device to participate in training more than the other is inevitable. This leads to statistical heterogeneity, where the devices collect data in a non-identically distributed (non-IID) manner across the network. Moreover, the number of data points across devices may vary significantly, and there may be an underlying statistical structure present that captures the relationship among devices and their associated distributions [107].

This data-generation paradigm violates frequently used independent and identically distributed (i.i.d.) assumptions in distributed optimization and may add complexity in terms of problem modelling, theoretical analysis, and the empirical evaluation of solutions. Some of the future challenges to addressing the statistical heterogeneity are to identify and include the clients with valuable data and poor communication capabilities. This type of difference in the data can lead to complexity in the modelling, analysis and evaluation of the Federated Learning model. This could be potentially tackled by using Adaptive Client Sampling [100].

3) Efficient Data Management

Since there is a massive amount of data being generated from IoT devices every day, it is crucial to have efficient data management techniques in place. The massive amount of gathered data is raw and needs to be processed before analyzing and making decisions in real time. After processing, transmitting it to the required destination in real-time is also necessary. Hence, it is crucial to have efficient data management policies to handle and store massive data. In the paper [101], the authors have proposed techniques to tackle data management using blockchain technology. In [102], the authors use Deep Reinforcement Learning (DRL) to analyze the data characteristics of IoT devices. This increased the model aggregation rate and reduced communication costs.

4) Server-Side Attacks

Preserving the privacy of the data and model is a significant issue in Federated Learning. There is a high chance that the model updates can be tampered by the attackers, and the attacker may try to steal the model updates from the cloud resulting in inconsistency and noise. Therefore, it is essential to use homomorphic encryption techniques, where the computation and analysis use several encryption techniques, making it difficult for the attacker to decrypt. This can be tackled by using differential privacy techniques [103], which improve convergence and protects from attacks.

5) Client-Side Attacks

Privacy security mechanisms are computationally expensive, and the various security issues on the client and server sides remain challenging to address when data communication is restricted. Similar to the server side, it is essential to protect the client side using encryption techniques. In addition, the client-side servers are prone to model-poisoning attacks and data-poisoning attacks. In a model poisoning attack, the attacker might upload poisoned updates, leading to performance degradation and classification errors. In data poisoning attacks, the attackers infiltrate and enter misleading information, tampering with the training of the models. In the paper [104], the authors use blockchain technology, miners, and ledgers to regularly verify the local model updates.

6) Communication Overhead

Federated Learning involves several rounds of communication, considering the massive number of IoT devices connected. Therefore, FL methods rely on recursively communicating and exchanging model updates throughout the process. In the paper [105], the authors proposed gradient-descent FL

that involves local updates and global convergence measures using a control algorithm to reduce the loss function for reduced resource consumption.

7) Resource-Constrained

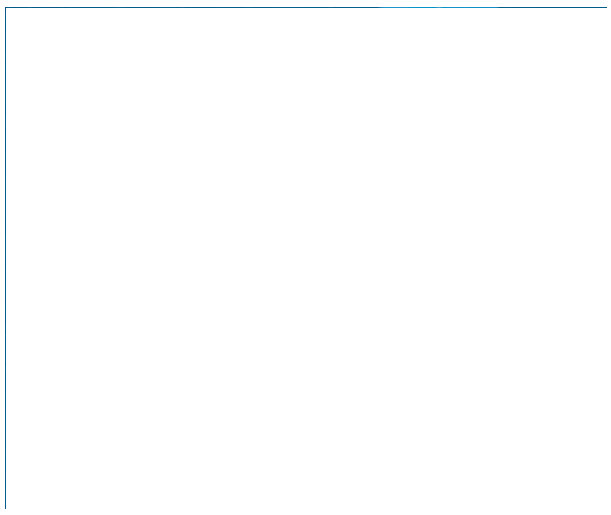
Most IoT devices are resource-constrained and may not contain CPU or GPU capabilities to utilize complex ML or DL models. The IoT device does not have the processing ability, low bandwidth and power, or limited storage capacity. In [17], the authors review the latest research work and explore the research directions for FL in resource-constrained IoT devices. The resource requirements of FL are not met in certain IoT devices due to weak computation [16]. This could be tackled using resource-aware training for the neural network [106].

SECTION VII.
Conclusion

With the increasing number of IoT devices, it is essential to protect the privacy and security of the user data. Hence, it is crucial to preserve confidential user data effectively. This paper presented a comprehensive survey of integrating Federated Learning for IoT malware analysis and discussed several associated approaches and techniques in detail. Specifically, we have discussed the IoT malwares highlighting the different types and natures of IoT malwares. We also discussed the different types of malware analysis and their taxonomy in depth. Subsequently, the paper addressed the motivation behind integrating Federated learning and IoT malware analysis and reviewed and compared the differences between Federated Learning and centralized learning techniques such as Machine Learning and Deep Learning. Finally, at the end of the paper, we analyzed the research challenges in integrating Federated Learning with IoT and discussed future research directions in detail.

ACKNOWLEDGMENT

The authors acknowledge the grant from the Natural Sciences and Engineering Research Council of Canada—NSERC (#RGPIN-2020-04701) and Mitacs Global Research Internship (GRI)—to Arash Habibi Lashkari.



PDF

Help