Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 8658278, 12 pages
https://doi.org/10.1155/2023/8658278

Review Article

AI-Enabled Learning Architecture Using Network Traffic Traces over IoT Network: A Comprehensive Review

Nagender Aneja,[1] **Sandhya Aneja**,[2] **and Bharat Bhargava**[3]

[1]*School of Digital Science, Universiti Brunei Darussalam, Brunei Darussalam*
[2]*School of Computer Science and Mathematics, Marist College, Poughkeepsie, NY, USA*
[3]*Department of Computer Science, Purdue University, West Lafayette, IN, USA*

Correspondence should be addressed to Nagender Aneja; naneja@gmail.com

Received 10 August 2022; Revised 27 December 2022; Accepted 12 January 2023; Published 6 February 2023

Academic Editor: Junaid Shuja

WiFi and private 5G networks, commonly referred to as P5G, provide Internet of Things (IoT) devices the ability to communicate at fast speeds, with low latency and with a high capacity. Will they coexist and share the burden of delivering a connection to devices at home, on the road, in the workplace, and at a park or a stadium? Or will one replace the other to manage the increase in endpoints and traffic in the enterprise, campus, and manufacturing environments? In this research, we describe IoT device testbeds to collect network traffic in a local area network and cyberspace including beyond 5G/6G network traffic traces at different layers. We also describe research problems and challenges, such as traffic classification and traffic prediction by the traffic traces of devices. An AI-enabled hierarchical learning architecture for the problems above using sources like network packets, frames, and signals from the traffic traces with machine learning models is also presented.

1. Introduction

IoT devices include a diverse range of robotic and unmanned aerial vehicles and a wide range of smartphones. These have limited computation and storage capabilities while connected over the Internet or fifth-generation wide-area network. Figure 1 shows an AI-enabled learning architecture of the B5G/6G IoT network and IoTs, where each item represents a unique Internet of Things (camera, printer, etc.) to solve device identification and traffic forecast problems. This AI-enabled architecture is a hierarchical learning architecture required for the federated learning paradigm. This is achieved through the network's hierarchy, which begins with the local network (edge server) and progresses to the vendor network (fog server) and the cloud network (cloud server).

WiFi and the current private cellular network 5G are complementary, competitive and convergent. Traditionally, private cellular is suitable in vast areas, outdoor spaces, industrial devices, and mobility. In contrast, WiFi is best suited for indoor and small outdoor area devices, especially for high density, bulk data, and ubiquitous device support. In cyberspace, edge computing offers edge servers to be operated at cellular base stations or access points to collect IoT traffic and train (or learn) a deep learning model [1]. Much like edge computing, Fog computing also brings the benefits and power of the cloud closer to the locations where data is produced and acted upon. Fog computing refers to a kind of distributed computing architecture in which the data, computation, storage, and applications are all situated between the source of the data and the cloud. Because both, fog computing and edge computing, entail moving intelligence and processing closer to where the data is produced, many people use the words fog computing and edge computing interchangeably [2].

The network traffic of an IoT device can be captured from the application, transport, and network layers for potential feature sets. According to the vision, AI-capable IoT networks are more about network traffic than anything else, which can automate the discovery of individual IoT devices among billions of IoT devices. Because of the fluid architecture and varied service offerings, predicting the

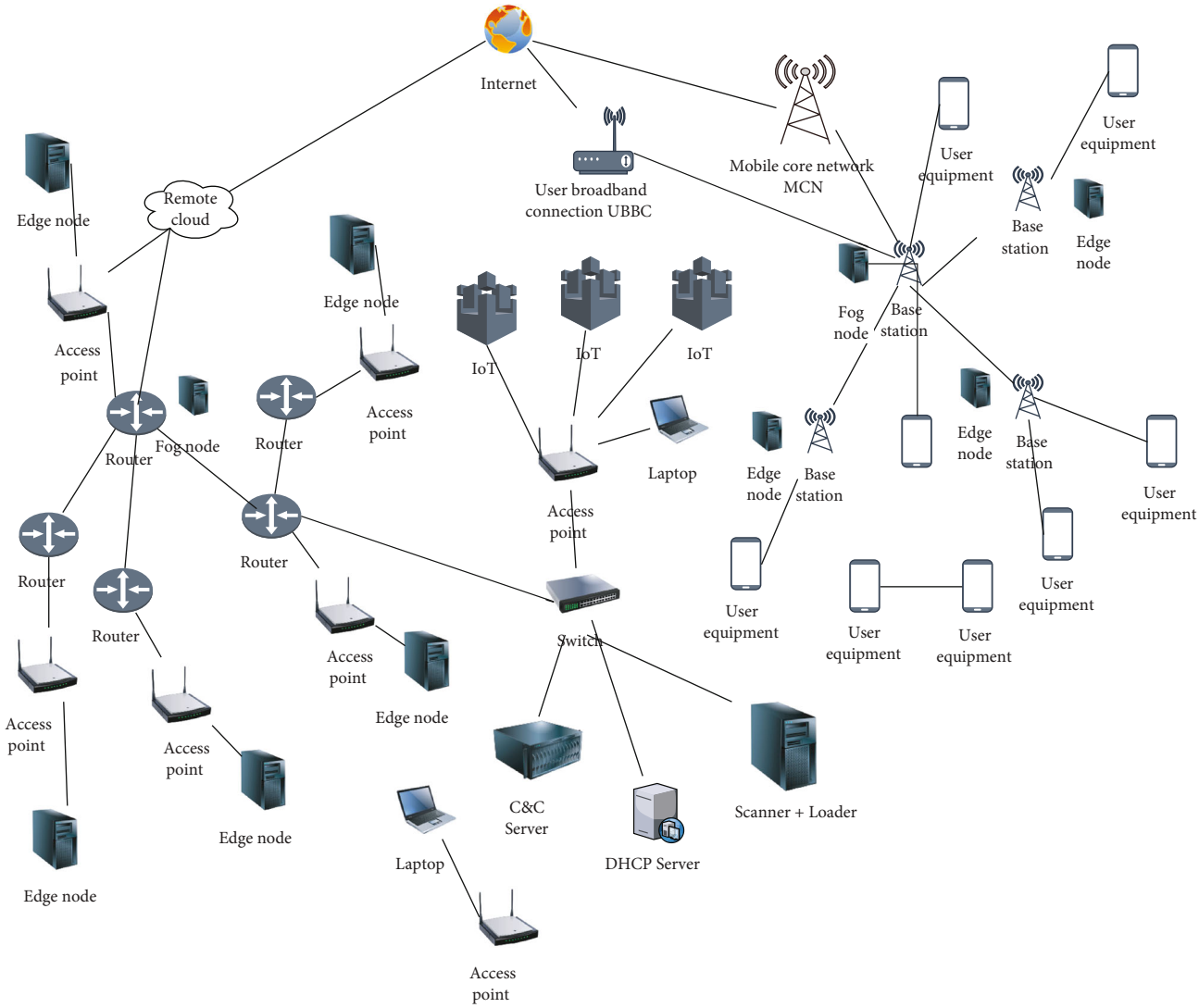

Figure 1: AI-enabled learning architecture of B5G/6G IoT network.

traffic on an intelligent IoT network is a challenging task [3]. IoT devices are vulnerable to a scenario in which devices connected to a gateway work together to deceive the IoT network's intelligent decision-making process and cause it to malfunction. By utilizing network traffic over each edge in an IoT network, physical objects can be classified as good nodes or malicious nodes [4–6].

Several researchers have used neural networks to analyze network traffic. Neural networks are the universal function approximators that learn abstract, high-level, nonlinear representations of training data. It is sufficient to use a single-layer feed-forward network to estimate any continuous function to an arbitrary degree of accuracy. Another important neural network architecture is convolution neural networks (CNNs) which are specially designed to take advantage of structural information contained in an image. The graph convolution network (GCN) receives as an input a graph $G(V, E)$ which comprises a set of vertices V and a set of edges E [7]. Neural networks are used by deep learning to train and learn models. Deep learning models are as good as data as these are built on the data that they have seen. In traditional AI systems, a model is trained on the cloud and updated on edge devices. However, in federated learning (FL), a model is trained from distributed systems over the cloud [3]. Here, interesting observation for FL is that the learned model over distributed systems can be secure like other encrypted numbers communicated over the Internet [3].

In this paper, we present a survey of techniques to set up experiments for capturing network traffic traces of the IoT devices from the packet, frame, and other levels of local networks and cyberspace. We also present problems being researched to utilize network traffic traces with a fusion of AI or machine learning techniques such as deep learning, with a discussion on how the network traffic traces data pose a challenge to the privacy of IoT devices in the IoT network. Figure 1 shows a general IoT network environment where the training and learning agent(s) are located at the cloud, fog, or computing edge servers either over the Internet or private cellular network with IoT devices as end-user devices. Thereby Figure 1 presents AI/ML-enabled IoT

networks, including high-throughput, low-latency, and high-capacity 5G-and-beyond (NextG) devices which can enhance security and application experience.

2. Related Work

The primary purpose of this literature review is to scan and assess articles relating to three interrelated fields: (i) the Internet of Things; (ii) network traffic, network traces, 5G, or 6G; and (iii) machine learning or deep learning. These three fields are all interconnected. The explication of the methodology used for this review can be summarized in Table 1. The steps consist of the formulation of the study hypothesis, the location of the study, the collection of data, the critical evaluation of studies, the analysis and presentation of the data, and finally, the interpretation of the data. The keywords used are tabulated in Table 2, but the significant search terms used are IoT, "Internet of Things," network, dataset, "data set," "network traffic," "network traces," "neural network," 5G, 6G, "deep learning," "machine learning," and "artificial intelligence." The searched publications were selected if the following inclusion criteria were met:

(i) Research publications that were written in English

(ii) Research publication year from 2012 to 2022

(iii) Publications related to network traffic using machine learning or deep learning

To retrieve the relevant publications, a relevant number of queries were submitted to various databases, including Google Scholar, IEEE, Springer, Elsevier, and Scopus, to review top results. Table 3 shows queries and results submitted to the Scopus database. In the below subsections, we describe the relevant results classified into different categories.

2.1. Beyond 5G/6G-Emerging High-Throughput, Low-Latency, and High-Capacity Networks. Beyond fifth-generation (B5G) networks, or the so-called "6G," is the name given to the next-generation wireless communications systems [8]. As 5G communication networks become more widely implemented worldwide, industry and academia have begun to look at 6G communications. AI/ML-enabled 5G IoT network helps to obtain efficient resource scheduling strategies in a complex environment with heterogeneous resources and a massive number of devices [9]. The 5G systems have three technical features: enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultrareliable low-latency communications (uRLLC) meeting the reliability requirements of novel applications, such as self-driving cars, remote operations, intelligent transport systems, Industry 4.0, smart energy, e-health, and AR/VR services [10].

5G rollouts have been very fast in China; however, the coverage footprint in the United States and many other countries could be much higher. There are 17K-18K base stations in the US, while in China, it is in 2M-3M. China offered much more spectrum with mmBand from 3.3 to 3.6 GHz, while the United States offered from 3.7 to 4.0 GHz. Thus, by having much more base stations in the midband, China has been faster in deploying 5G. Today, a latency of 10-70 ms is down to 1-10 ms for applications like robotics and industrial IoT.

WiFi and 5G-and-beyond networks (NextG) are complimentary, competing, and convergent at the same time. Private cellular has traditionally been used for large regions, outdoor places, industrial equipment, and mobility. Most enterprise applications are either both wired and wireless or only wireless. Therefore, the trend here will be continued by improving the throughput, latency, user scalability, etc., of WiFi networks. On the other hand, private 5G offers solutions to specific use cases that can be deployed with high throughput and low latency. In smart manufacturing, WiFi still needs to be preferred because of latency. In applications like banking, WiFi is still not preferred because of security. WiFi and private 5G networks (P5G) support many applications, from indoor to outdoor, depending on the type of access; both coexist, and both are at par for applications like drones and self-driving cars. As a result, to compete, new open radio access network (O-RAN) designs for 5G-and-beyond networks (NextG) must smoothly execute spectrum-sharing policies and adjust their operating parameters intelligently.

Baldesi et al. [11] illustrate the performance of a controller working via a RAN intelligent controller (RIC) that monitors the spectrum and shifts cell frequency to prevent WiFi over LTE in unlicensed bands in the context of spectrum sharing over the range 5.18 GHz-5.24 GHz. An unlicensed citizens broadband radio service (CBRS) private network is preferable to a privately owned one since it is easier to set up and maintain. Since anyone may interfere, public carriers will not use the unlicensed band. In industry, this might be a danger and a criticality. The 5G deployment problems can unveil the need for 6G; for example, 6G technologies are expected to empower future IoT networks, including edge intelligence, reconfigurable intelligent surfaces, space–air–ground–underwater communications, terahertz communications, massive ultrareliable low-latency communications, and blockchain [8].

Satellite-based services are still a long way away from being commonplace. The difficulty of connecting two phones can be solved with a star link top-based antenna, but it will cost much money. 6G can pay attention to Leo, a satellite-based phone service that, if linked to space, might serve as an equalizer for rural populations. An AI-enabled network with 5G devices can automate the distribution of networking resources, storage, and other resources among the cloud and edge servers.

2.2. Network Traffic Traces Data Collect. The network traffic can be captured at the physical layer (signals) or medium access layer (frames). The traffic may also be captured at the application, transport, or network layer from the perspective of the network, which can be either local or cyberspace. This section provides an overview of the experimental configurations used by the research community to gather network traffic. In addition, the datasets generated from these testbeds are summarized in Table 4 over an example Internet of Things network depicted in Figure 1.

Table 1: Review methodology.

Steps	Explanations
Question formulation (hypothesis)	How are publications classified in IoT or network traffic datasets with machine or deep learning
Study location	Research publications
Data location	Scientific articles database: Google Scholar, IEEE, Springer, Elsevier, http://lens.org, Google and Strategies of search: keywords and citation search
Critical evaluation of studies	Tabulation of research publications according to the "keywords" search strategy and eliminating publications that do not fit the article's purpose or meet the exclusion criteria.
Data analysis and presentation and data interpretation	Elaboration of graphs and tables, discussion of results to understand the data obtained, and comparison with scenarios from other sources (scientific papers).

Table 2: Summary of database search.

Section	Keywords
Beyond 5G/6G	(("beyond 5G/6G" OR 5G OR 6G) AND ("artificial intelligence" or "machine learning" or "deep learning"))
Network traffic traces	(("network traffic" OR "network traces" OR "network log") AND ("artificial intelligence" or "machine learning" or "deep learning"))
Network traffic classification	(("network traffic" OR "network traces") AND (classification OR supervised OR prediction) AND ("artificial intelligence" or "machine learning" or "deep learning"))
Network traffic and federated learning	(("network traffic" OR "network traces") AND (classification OR supervised OR prediction) AND ("federated learning" OR "collaborative learning"))

Table 3: Scopus search queries.

Search query	Scopus results
TITLE-ABS-KEY (("beyond 5G/6G" OR 5 g OR 6 g) W/10 ("artificial intelligence" OR "machine learning" OR "deep learning") W/10 ("data set" OR dataset)) AND PUBYEAR >2012 AND PUBYEAR <2023	9
TITLE-ABS-KEY(("network traffic" OR "network trace" OR "network log") W/10 ("artificial intelligence" or "machine learning" or "deep learning") W/10 ("data set" or dataset)) AND PUBYEAR >2012 AND PUBYEAR <2023	55
TITLE-ABS-KEY((("network traffic" OR "network trace" OR "network log") W/10 ("artificial intelligence" or "machine learning" or "deep learning") W/10 ("data set" or dataset)) AND (classification OR supervised OR prediction)) AND PUBYEAR >2012 AND PUBYEAR <2023	34
TITLE-ABS-KEY((("network traffic" OR "network trace" OR "network log") W/10 ("artificial intelligence" or "machine learning" or "deep learning") W/10 ("data set" or dataset)) AND ("federated learning" OR "collaborative learning")) AND PUBYEAR >2012 AND PUBYEAR <2023	1

2.2.1. Physical Layer. The radio communication of IoT devices carries information across the space from a transmitter to a receiver by modulating the radio signals, which means encoding the signal on the radio wave by varying some aspect of the wave. Radio waves are electromagnetic waves with frequencies between 30 Hz to 300 GHz. The antenna connected to each electronic device has a transmitter to radiate the waves and a radio receiver to receive waves from neighboring devices. Traditional WiFi, Z-wave, Bluetooth, and Zigbee technologies are used in IoTs, which differentiate the transmitting wave by its carrier frequency offset in in-phase (I) and quadrature (Q) offset. The I/Q refers to two sinusoids that have the same frequency and differ in phase by 90°. The transmitted waveforms are generally different, even for the product of the same vendor and the same batch. Radio frequency fingerprinting, also known as RF fingerprinting, is a method that is based on the physical layer and makes use of hardware-based characteristics, such as electromagnetic waveforms that impose a unique signature on transmitters.

Al-Shawabka et al. [24] created an experimental testbed consisting of twenty software-defined USRP N210 radios running via GNU radio functioning as transmitters and one USRP N210 acting as a receiver. Each USRP was equipped with a CBX 1200-6000 MHz daughterboard with 40 MHz instantaneous bandwidth and one VERT2450 antenna. The authors also used two other datasets, one containing 500 WiFi with 500 ADS-B (automatic dependent surveillance-broadcast) transmissions provided by DARPA and the other 100-device dataset. The testbed by Al-Shawabka et al. [24] captured 7 TB of wireless data from devices with the same RF hardware. The testbed was first

Table 4: Summary of network traffic traces of the IoT devices testbeds.

Ref	Source	Size	No. of devices	No. of files	Instances	Features	Signature
Restuccia et al. [12]	Radio waves (signal)	7 TB, 26 GB/device	20 USRP radios	3/device	—	I/Q samples	Image
DARPA [12]	Radio waves (signal)	500 GB	500 WiFi and 500 ADS-B devices	—	18686x116+9156x76/device/ transmission	I/Q samples	Image
Qing et al. [13]	Radio waves (signal)	—	4 Zigbee devices	—	60,000/device	I/Q samples	Image
Robyns et al. [14]	Network frames	1.7 MB	748	1	122,989	Frame header fields	Text-vector
Miettinen et al. [15]	Network packets	38 MB	27	27	102,347	Packet protocol header fields	Text-vector
Sivanathan et al. [16]	Network packets	61 MB-2 GB/file/ device	28	26	6,845,378	Traffic characteristics	Text-vector
Radhakrishnan et al. [17]	Network packets	100 GB	58	3/ network	—	Inter arrival time	Text-vector
Aneja et al. [18]	Network packets	36 GB	58	—	608,864/44 device +137,300/14 devices	Inter arrival time	Image
Meidan et al. [19]	Network packets	14 MB-215 MB/file/ device	9 infected by botnets	25	7,062,606	Traffic statistical characteristics	Text-vector
Yang et al. [20]	Network packets	—	12,880	1	690,426	Packet protocol header fields	Text-vector
Pour et al. [21]	Network packets	3.6 TB	52,947	1	543,392	Packet protocol header fields	Text-vector
Zakroum et al. [22]	Network packets	2 TB	64 million	—	4.5 billion	Packet protocol header fields	Text-vector
Chowdhury et al. [23]	Network frames	171.4 MB	11	437	560,399	Frame header fields	Text-vector
Chowdhury et al. [23]	Network packets	6.0 GB	14	704	32,911,503	Packet protocol header fields	Text-vector

configured in an open room and then in an anechoic chamber, which included inserting hundreds of blue foam projecting arrowheads into the chamber to absorb undesirable radio frequency signals. Finally, each transmitter was linked to the receiver using a cable and an attenuator. The experimental testbed of Qing et al. [13] comprised four Zigbee devices and one BB50C Spectrum Analyzer of Signal Hound. The authors captured 60,000 samples of each device with 1000 sampling points.

Radio fingerprinting focuses on extracting signal characteristics such as modulation, bandwidth, center frequency, protocols, and emitter identity to identify a device, among others. Moreover, RF leverages waveform-level faults such as I/Q imbalance, phase noise, frequency offset, and sampling offset imposed by the RF circuitry to distinguish wireless devices. Modulation domain metrics such as frequency error, SYNC correlation, IQ offset, magnitude error, phase error, I/Q imbalance due to hardware imperfections, and modulation shape are used as the features.

A wireless device's waveforms may be linked to small hardware-level faults detected in off-the-shelf radio circuitry. Restuccia et al. [12] demonstrated a system DeepRadioID for improving accuracy of deep learning-based radio fingerprinting algorithms using the dataset [24] by utilizing a filter for each device waveform-level faults. It is envisaged that adversarial devices will not be able to imitate the device's filter. An adversary's accuracy is greatly lowered by DeepRadioID when it attempts to produce the fingerprints of other devices by utilizing their filters. Among the others, the low, slow, small unmanned aerial vehicles (LSSUAVs) use signals in the 2.4 GHz band. The proposed method [25] recognized LSSUAV signals without any mistakes and falsely recognized IEEE 802.11b and IEEE802.11n signals as LSSUAV.

Jagannath et al. [25] presented a few datasets for deep learning in wireless communications. The datasets are as follows:

(1) Bluetooth waveform dataset from 86 smartphones [26]. The database contains subsets of Bluetooth signals sampled at 5, 10, 20, and 250 GSps. Each dataset collected 150 Bluetooth signals from 86 different COTs smartphones. A COT antenna operating in ISM2400 band connected to a oscilloscope is used to record the samples in the duration of 10 μs. Bandpass filtering is used to remove oscilloscope spurs. Filtered samples are normalized from -1 to +1

(2) Three wireless standards signals from USRP X310 radio [27]. In this dataset, four USRP X 310 radios are set up to transmit in a standard compliant IEEE 802.11a, long-term evolution (LTE) or 5G new radio frames generated from WLAN, LTE, and 5G toolboxes of MATLAB. The USRP B210 receiver is used at a rate of 5 MSps for WiFi and 7.69 MSps for LTE and 5G. Five sets of two secs of IQ samples are captured from each emitter for two days

(3) Automatic dependent surveillance-broadcast (ADS-B) signal dataset from over 140 commercial aircrafts [28, 29]. An aircraft broadcasts its location and identifier to air traffic control centers in ADS-B format. The ADS-B signals at Daytona Beach International Airport are captured for 24 hours using a USRP B210 receiver set at 1090 MHz with eight mega samples per second (MSps) sampling rate in this dataset

(4) Waveforms from seven hovering unmanned aerial vehicles (UAVs) [30]. The dataset comprises of waveforms from all seven identical DJI M100 UAVs. The signals are captured by flying UAVs at the distances of 6, 9, 12, and 15 feet from the receiver in an RF anechoic chamber. The receiver is USRP X310 with UBX 160 USRP daughterboard. The receiver is tuned at 10 MHz downlink channel frequency centered at 2.4065 GHz. There are 140 examples from each UAV at 4 distances and with 4 bursts. The dataset provides total 13k examples of 92k IQ samples per example. The dataset is in SigMF format (binary format) with metadata in JSON file

2.2.2. Medium Access Layer. WiFi-enabled devices employ the MAC layer (or data link layer) management frame-probe request frame repeated over and over again sporadically in a wireless local area network (WLAN) for network scanning and to connect with nearby accessible WiFi access points (specific or broadcast). In frame analysis, probe request frames are a preferable choice. There are some reasons for this: (i) WiFi-enabled devices transmit probe request frames, (ii) the information carried in a probe request frame is in the form of plain text [31], and (iii) WiFi-enabled devices exchange probe request frames regularly with access points for the connection [32]. Depending on the individual device configuration and capability, a probe request frame contains a variety of information element tags, including service set identifier (SSID) parameter set, supported rates, extended supported rates, high-throughput (HT) capabilities, and vendor-specific information [33].

To establish a connection with the neighboring access point, Chowdhury et al. [33] developed an experimental testbed. In this testbed, a WiFi interface on a system in master mode was set up using a WiFi adapter connected to the system (guest OS). The OS was configured to use a WiFi interface in monitor mode to collect IEEE 802.11 MAC frame traces using tcpdump. The devices with wireless connectivity exchanged probe request frames periodically inside a wireless local area network (WLAN) for network scanning. The network frame dataset (network frames) includes probe request frames of the IEEE 802.11 MAC protocol. The dataset includes information on both associated and unassociated device states.

The frame-level (probe request frame) Glimps [14] dataset was captured with 738 devices in a music festival. It is observed that there are 26,648 unique MAC addresses with multiple frames. The Glimps dataset includes probe request frames from devices with a maximum of 12 tag parameters. The dataset is available at https://github.com/rpp0/wifi-mac-tracking.

2.2.3. Local Network. At the local network level, packets from the application, transport, and network layers are captured. These packets include different protocols, packet fields, and a bag of words (BoW) from the protocols, such as TCP, UDP, IP, HTTP, DNS, mDNS, SSDP, and DHCP.

Chowdhury et al. [33] created an experimental testbed of IoT devices connected to a destination PC on LAN through an access point to capture passively observed network traffic traces. The MAC address is used to identify a device on the LAN and separate its traffic from other devices on the network. There are several publicly accessible datasets for device fingerprinting research and practice. IoT sentinel dataset consists of 31 IoT devices from 27 different manufacturers [15]. This dataset is relatively challenging since it comprises devices setup network traffic only. Each set was repeated 20 times per device type during the setup of devices. The dataset can be downloaded from https://github.com/andypitcher/IoT_Sentinel. The UNSW dataset includes 28 IoT devices of different kinds [16]. In this dataset, all the traffic on the LAN side was collected using the tcpdump tool running on OpenWrt. The traffic on the network was tracked for 26 weeks. The raw trace data had information about packet headers and payloads. The dataset can be downloaded from https://iotanalytics.unsw.edu.au/iottraces. The GTID dataset [17] was captured from 58 devices of 16 distinct device types on campus networks.

In IAT dataset [18], the 100 GB, pcap files of the GTID dataset were converted to 2 GB MATLAB files of inter arrival time (IAT) provided at [17]. The MATLAB files were restored to compatible HDF5 data format using Python by converting every 1000 packets IAT to a jpg image using the Python matplotlib library resulting in 36 GB jpg images. The IAT dataset (as image) consists of 137,300 images from the isolated network of 14 devices and a total of 608,864 images from the campus network of 44 devices, including active DFP (34,898 images) and passive DFP (573,966 images) signatures. This IAT dataset is publicly available on the DOI link: 10.21227/d9e4-wm90. The size of each protocol traffic signature trained over the dataset ranges from 2 GB to 6 GB.

In N-BaIoT [19] network traffic capturing setup, IoT devices were connected to an access point via WiFi, and that access point was connected to a central switch. Port mirroring was carried out on the switch to sniff the network traffic on a server. The components of two botnets—BASHLITE and Mirai—were deployed on a computer and communication server connected to the switch as a part of the experiment setup, which infects IoT devices by brute-forcing the default credentials of devices with open telnet ports. The compromised/infected devices as bots were then controlled by the server to launch the attacks. This dataset with and without attacks is publicly available on the link [34].

2.2.4. Cyberspace. The Internet is becoming home to many IoT devices, such as webcams, routers, and net printers. Gathering network traffic in the Internet process requires sending query packets to a remote IoT device and collecting the response packets at the server. The authors [20] utilized the Zmap tool to detect online IoT devices in the network. They extracted vendor and product information from web pages using a web crawler. The device labels and product labels are extracted from vendor websites and product web pages—the vendor stores product-related information in application services. Using 20 application protocols, the regex of vendor strings was utilized to match the data. The web crawler collected 67 device types, 2228 vendors, and 64,532 product labels.

Another platform for collecting IoT data is a network telescope (i.e., darknet). The darknet is a set of routable, allocated unused IP addresses through which Internet-scale traffic may be routed. Researchers [21] categorized and cleaned the darknet network traffic dataset gathered over twenty-four hours using an a/8 network telescope provided by the Center for Applied Internet Data Analysis. This dataset comprises 3.6 TB of captured network traffic from 440,000 compromised IoT devices and evidence-based artifacts associated with 350 IoT botnets. Using IoT botnet-specific properties, the authors constructed a probabilistic model to clean unrelated traffic and to discriminate between misconfiguration and malicious traffic. The probabilistic model formulates, computes, and compares the joint probabilities of a source causing a misconfiguration flow and a source being malicious. Suppose the likelihood of a source creating a misconfiguration flow for the number of destinations is greater than the likelihood of the source being malicious. In that case, the source is considered generating misconfiguration traffic. The dataset is available at https://github.com/COYD-IoT/COYD-IoT.

The authors in [22] looked at several characteristics of traffic recorded by an a/20 network telescope. The collected traffic is 2 TB and contains over 4.5 billion packets. Exploration patterns of network probers with geolocation at the port level and top services are parts of the reported research.

2.3. Network Traffic Classification Problem. As part of the network traffic classification, network traffic (or packets from the application, transport, and network layers) is studied for potential feature sets that might be used to identify the device from the cohort of devices [16, 17, 33, 35–39]. Several research projects have tried their hand at IoT device classification called fingerprinting by monitoring network traffic.

One way of taking the feature set for classification is through statistical preprocessing on network traffic, e.g., packet length statistics, traffic flow volume, traffic flow duration, device sleep time, a bag of words (BoW) from network traffic-port numbers, and DNS server name string [16, 37, 39]. Pinheiro et al. [39] used statistical preprocessing for features on encrypted network traffic. Another way to take the feature set is to use packet fields and a bag of words (BoW) from the protocols, e.g., TCP, UDP, IP, HTTP, DNS, mDNS, SSDP, and DHCP [33, 35, 36, 38]. This feature set fingerprints the device for device identification.

In the MAC layer (or data link layer) fingerprint approaches, features are extracted from MAC frames [14, 31, 32, 40]. MAC layer fingerprinting does not require analyzing raw radio signals in contrast to RF fingerprinting approaches. In frame analysis, probe request frames are a preferable choice to be used for device identification. Robyns

et al. [14] performed per-bit entropy analysis of probe request frames to compute suitable bitmask for device fingerprinting. Gu et al. [31] utilized multilayer perceptron (MLP) to extract a feature set from probe request frames to classify devices.

Radio frequency (RF) is the physical layer property of a device; it can be considered as DNA of a wireless-enabled communication device in the network [13, 41]. RF fingerprinting leverages hardware-based characteristics, e.g., electromagnetic waveforms, that impose a unique signature emitted by the transmitters [42]. Thus, these hardware-level imperfections (such as solder variations, monolithic microwave integrated circuit (MMIC) fabrication, digital-to-analog converters, bandpass filters, frequency mixers, and power amplifier [43–45]) occur unintentionally during the manufacturing process [41, 46]. RF fingerprinting allows identifying wireless-enabled devices separately even when the same batch of devices come from the same manufacturer [45, 47].

Jian et al. [42] utilized raw and processed in-phase and quadrature (I/Q) samples as features set for RF fingerprinting. Huang et al. [48] investigated the I/Q imbalance of quadrature modulation signal for feature extraction to identify specific emitter. Bassey et al. [43] used software-defined radio devices in received mode to capture RF (I/Q) traces, and they filtered those traces to extract just the burst transmission regions for analysis. Sankhe et al. [45] presented a system called ORACLE to identify unique radio emitters based on the analysis of I/Q samples (I/Q imbalance and DC offset) using CNN.

A shortcoming of the network traffic (or packets from the application, transport, network layers, frames from the MAC layer, and signals from the physical layer) is that their scope is limited to local IoT networks. Thus, Pour et al. [21] presented an Internet-wide perspective to leverage a large-scale network. The authors in [20–22] utilized numerous tools such as Zmap to detect online IoT devices and to collect Internet-wide network traffic (or packets from the application, transport, and network layers) from those devices. Based on UPnP and DNS replies, HTTP data banners, and network-layer metadata, the feature set from the collected traffic was used to fingerprint the device and create an ensemble of four supervised classifiers.

The well-known Mirai and BASHLITE botnets commanded vulnerable IoT devices to launch attacks. Mirai botnet included a computer and communication server and a scanner and loader server. The scanner and loader components identify vulnerable IoT devices and load malware onto the devices. The BASHLITE botnet server also commands the infected device to launch attacks [15] such as TCP or UDP packets flooding servers on the cloud. When using federated learning, the idea is to gather models from various edge devices, retrained into a cloud-based system, and update back on the fog or edge devices [49–51]. Federated learning on Internet-wide network traffic (known as network telescopes) has the purpose of learning the network traffic from a variety of edge IoT devices and updating back on the fog or edge devices to combat botnets such as Mirai and BASHLITE. The discovery and analysis of IoT-centric botnets reveal crucial cyber threat intelligence relating to the discovery of malware attack vectors and disclose possible vulnerabilities or intrusion points within globally deployed IoT devices [21].

The classification algorithms used in traditional AI systems have been successfully adapted. The federated learning approach is valued in various applications, including the health industry, where data collected from various health organizations is used to generalize the learning model. Therefore, the dynamic edge, fog, or cloud computing architecture with Internet-wide network traffic collected from millions of devices and an overview of the federated learning approach-based classification algorithms appears promising for securing IoT devices. The AI-enabled framework, for IoT device identification or to classify between misconfiguration and other malicious traffic, utilizing IoT botnet-specific properties is also beneficial for IoT security. IoT network is modeled as a GCN to classify physical objects as good or malicious nodes by utilizing network traffic over each edge [4–6]. On the other hand, given that the legacy networks are already over-provisioned, we anticipate that AI and ML will be able to improve the efficiency of the network systems, mainly to assist security professionals in securing the network.

2.4. The Network Traffic Prediction Problem. The IoT device log includes the packets' protocol-specific events in chronological order. The packets include port numbers, IP addresses, sequence numbers, flags, checksum, window size, and domain names. [52–54]. Analyzing and processing logs of communication protocols in network traffic would gradually become a necessary component of IoT cyberspace. Log analysis can be significant (i) due to big size of IoT cyberspace, (ii) for predicting possible attacks, (iii) avoiding vulnerabilities, and (iv) node plan for IoT system [53]. For example, predicting the busy hours at a smart shop can help determine the number of cameras and sensors.

The problem of a requester making a specific resource request, i.e., provisioning resources dependent on the traffic prediction, can be addressed by automating how the supplier works. Overprovisioning, tighter coupling and interdependence, sophisticated resource flexing, and locked-step procedures are some of the challenges encountered by application-specific network traffic prediction. Deep reinforcement learning (DRL) and deep learning are used in many contexts. They are becoming popular to learn policies that optimize the performance of complex wireless networks by optimal resource allocation.

In AI-enabled next-generation 6G communication networks, network traffic prediction may be beneficial. High-speed networking at one trillion bits per second extended battery life with low energy consumption, enhanced security and privacy, and enhanced intelligence via application, service, and operational intelligence via deep learning which are some of the 6G visions advanced by [55]. With a perspective of 6G visions, Zhao et al. [1] proposed a learning-based mobile fog scheme to fully use edge computing and storage resources to learn the offloading decision to optimize the system performance automatically. The authors first optimized the bandwidth allocation and then optimized

the offloading strategy based on the allocated bandwidth. A new reward function was developed by [56–58] which jointly takes the transmission packet rate, the system throughput, the power consumption, communication requirements, the content request hit rate, average content access delay, system backhaul traffic, and the transmission delay into account to allocate resources among different types of IoT devices in a network.

One of AI's visions is a data-based prediction in which a requester can request desired resources with decoupled provision over the network. At the same time, a provider can offer feasible options using a declarative interface that captures the requester's intent (desired resources). In this scenario, the provider can capture their intent. For instance, a network intent strategy that uses traffic prediction can induce weak coupling and lesser interdependence which can further induce resource flexibility without locked-step operation and thereby make effective use of resources.

Aneja et al. [52] analyzed network traffic logs of IoT devices distributed in a network behind the application gateways. These logs are used to identify compromised devices as well as collaborative adversaries.

Our goal is to eventually arrive at an infrastructure that is not just deliberate but also clever and unnoticeable. There are several networks, various devices, numerous application protocols, and even more applications. AI and ML can do a root-cause analysis on data collected from a variety of sources across IoT networks. Network traffic traces make one network communicate with another network. Thereby, the investigation might result in the base-lining of behavior and the prediction of any failure in the application.

2.5. Network Traffic Traces Data Pose a Challenge to Federated Learning. The federated averaging algorithm (FedAvg) is recommended by taking a weighted average of the training models. An interesting observation for federated learning (FL) is that the learned model over distributed systems can be secured in the same way as other encrypted numbers communicated over the Internet [49, 50, 59].

Tang et al. [60] proposed an approach to intrusion detection that involved enabling multiple Internet service providers (ISPs) or other institutions to collaborate on deep learning training under the premise that local data would be retained. It not only improves the accuracy of the model's detection but also protects users' privacy when using the network. Experiments were run on the authors' CICIDS2017 network intrusion detection data. The findings of this experiment indicate that workers who participate in federated learning have a higher detection accuracy. The accuracy and performance of other aspects of federated learning are almost on par with that of centralized deep learning models.

The hierarchical FL algorithm iterates for subglobal aggregations at various small cell base stations (SBSs) and global aggregation at macro cell base stations (BSs). Collaborative FL (CFL) is an FL framework that allows edge devices to implement FL without relying on a central controller. CFL allows devices to connect so that the transmission time between neighboring devices is less than connecting to the BS [51]. The optimal use of limited wireless resources can be enabled via sparsification, allowing only a small set of devices to send their local learning model parameters to the edge or cloud server. In dispersed FL, first, subglobal models are computed within different groups of closely located end devices. The subglobal models are then aggregated to yield a global model [61].

Research is being conducted on the hierarchical FL, CFL, and DFL frameworks, which provide distributed AI controllers to gather data from network edges or devices to enable IoT applications to learn and interact with one another. Even though collecting network traffic is one of the measurement-based solutions to numerous essential security issues of IoT devices, it is possible that the proposed FL architectures—HFL, CFL, and DFL—may cause additional issues for IoT devices, distributed controllers, and servers due to the collection of network traffic traces. For example, analyzing network traffic over federated learning architectures may unveil straggler or weak clients. A straggler client fails to share its model with the server at a proper time convenience. A weak client precludes participating in FL due to heterogeneity in hardware [62].

All clients interact with the server simultaneously in synchronous FL, while the training period can differ in asynchronous FL. Hence, it is essential to determine the training period for all local participants, which we call scheduling. Considering resource-constrained clients, there are better solutions to carry out a schedule frequently. Analyzing network traffic over federated learning architectures may unveil the scheduling among the clients using network features such as IAT and activity patterns.

In FL, security and privacy are ensured by the distributed artificial intelligence, in addition to being trustworthy and communication-efficient [63]. To build secure neural networks, it is necessary to use robust aggregation, detection, and reputation mechanisms over untrusted devices. However, Pinheiro et al. [39] used statistical preprocessing of features on encrypted network traffic for device information. Privacy-preserving algorithms can protect gradient information. The integration of 6G and federated learning, with efficient training and inference, as well as potential federated learning applications for 6G, was proposed by Liu et al. [55] to achieve ubiquitous artificial intelligence in 6G by 2025.

As a result, we conclude that the data contained in network traffic traces present a challenge to federated learning, even though these data offer benefits to IoT bots, network traffic prediction, and IoT device identification over Internet-wide automation intelligence.

3. Conclusion

The problems of device identification based on traffic classification and intruder detection based on traffic prediction utilizing traffic traces of Internet of Things devices have been addressed in many published research publications. However, there needs to be more in-depth research on topics like network traffic traces, traffic categorization, traffic prediction, and federated learning applications beyond 5G and 6G. To give classification, research directions, and problems in machine learning applications for network traffic, the

survey's objective is to investigate those areas. An AI-enabled hierarchical learning architecture for the challenges listed above that uses sources such as network packets, frames, and signals from the traffic traces with machine learning models might be effective for improving the model's performance by employing federated learning models. However, an intruder may also understand the architecture and put any edge servers, fog servers, or cloud servers in danger. As a result, future research might investigate ways to secure the network or devices if an attacker can anticipate the device by analyzing network data.

Data Availability

No data to share.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] R. Zhao, X. Wang, J. Xia, and L. Fan, "Deep reinforcement learning based mobile edge computing for intelligent Internet of Things," *Physical Communication*, vol. 43, 2020.

[2] M. Chen, T. Wang, S. Zhang, and A. Liu, "Deep reinforcement learning for computation offloading in mobile edge computing environment," *Computer Communications*, vol. 175, 2021.

[3] L. Nie, Z. Ning, M. S. Obaidat et al., "A reinforcement learning-based network traffic prediction mechanism in intelligent internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, 2021.

[4] B. Yan, G. Wang, J. Yu, X. Jin, and H. Zhang, "Spatial-temporal Chebyshev graph neural network for traffic flow prediction in IoT-based ITS," *IEEE Internet of Things Journal*, vol. 9, 2022.

[5] C. Li, G. Shen, and W. Sun, "Cross-architecture Intemet-of-things malware detection based on graph neural network," in *In 2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, Shenzhen, China, 2021.

[6] H.-N. D. YuleiWu and H. Tang, "Graph neural networks for anomaly detection in industrial internet of things," *IEEE Internet of Things Journal*, vol. 9, 2022.

[7] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, 2020.

[8] C. Dinh, "6g internet of things: a comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, 2022.

[9] Y. Wu, "Deep learning for privacy preservation in autonomous moving platforms enhanced 5G heterogeneous networks," *Computer Networks*, vol. 185, 2021.

[10] L. Chettri and R. Bera, "A comprehensive survey on internet of things (IOT) toward 5g wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2020.

[11] L. Baldesi, F. Restuccia, and T. Melodia, "Charm: Nextg spectrum sharing through data-driven real-time o-ran dynamic control," 2022, https://arxiv.org/abs/2201.06326.

[12] F. Restuccia, S. D'Oro, A. Al-Shawabka et al., "Deepradioid: real-time channel-resilient optimization of deep learning-based radio fingerprinting algorithms," in *In Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 51–60, Catania, Italy, 2019.

[13] G. Qing, H. Wang, and T. Zhang, "Radio frequency fingerprinting identification for zigbee via lightweight cnn," *Physical Communication*, vol. 44, article 101250, 2021.

[14] P. Robyns, B. Bonne, P. Quax, and W. Lamotte, "Noncooperative 802.11 mac layer fingerprinting and tracking of mobile devices," *Security and Communication Networks*, vol. 2017, Article ID 6235484, 21 pages, 2017.

[15] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "IoT Sentinel: Automated Device-Type Identification for Security Enforcement in IoTLocation," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 2177–2184, Atlanta, GA, USA, 2017.

[16] A. Sivanathan, H. H. Gharakheili, F. Loi et al., "Classifying iot devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, 2018.

[17] S. V. Radhakrishnan, A. Selcuk Uluagac, and R. Beyah, "Gtid: a technique for physical device and device type fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 519–532, 2014.

[18] S. Aneja, N. Aneja, B. Bhargava, and R. R. Chowdhury, "Device fingerprinting using deep convolutional neural networks," *International Journal of Communication Networks and Distributed Systems*, vol. 28, no. 2, pp. 171–198, 2022.

[19] Y. Meidan, M. Bohadana, Y. Mathov et al., "N-BaIoT—network-based detection of IOT botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[20] K. Yang, Q. Li, and L. Sun, "Towards automatic fingerprinting of IOT devices in the cyberspace," *Computer Networks*, vol. 148, pp. 318–327, 2019.

[21] M. S. Pour, A. Mangino, K. Friday et al., "On data-driven curation, learning, and analysis for inferring evolving internet-of-things (iot) botnets in the wild," *Computers & Security*, vol. 91, article 101707, 2020.

[22] M. Zakroum, A. Houmz, M. Ghogho et al., "Exploratory data analysis of a network telescope traffic and prediction of port probing rates," in *2018 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 175–180, Miami, FL, USA, 2018.

[23] R. R. Chowdhury, S. Aneja, N. Aneja, and P. E. Abas, "Packet-level and IEEE 802.11 mac frame-level network traffic traces data of the d-link IOT devices," *Data in Brief*, vol. 37, p. 107208, 2021.

[24] A. Al-Shawabka, F. Restuccia, S. D'Oro et al., "Exposing the fingerprint: dissecting the impact of the wireless channel on radio fingerprinting," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 646–655, Toronto, ON, Canada, 2020.

[25] A. Jagannath, J. Jagannath, and P. S. P. V. Kumar, "A comprehensive survey on radio frequency (rf) fingerprinting: Traditional approaches, deep learning, and open challenges," 2022, https://arxiv.org/abs/2201.00680.

[26] E. Uzundurukan, Y. Dalveren, and A. Kara, "A database for the radio frequency fingerprinting of bluetooth devices," *Data*, vol. 5, no. 2, p. 55, 2020.

[27] G. Reus-Muns, D. Jaisinghani, K. Sankhe, and K. Chowdhury, "Trust in 5g open rans through machine learning: Rf

fingerprinting on the powder pawr platform," in *IEEE Globecom 2020-IEEE Global Communications Conference*, pp. 1–6, Taipei, Taiwan, 2020.

[28] Y. Liu, J. Wang, H. Song, S. Niu, and Y. Thomas, *A 24-hour signal recording dataset with labels for cybersecurity and iot*, IEEE, Piscataway, NJ, USA, 2020.

[29] Y. Liu, J. Wang, J. Li et al., "Zero-bias deep learning for accurate identification of internet-of-things (iot) devices," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2627–2634, 2021.

[30] N. Soltani, G. Reus-Muns, B. Salehi, J. Dy, S. Ioannidis, and K. Chowdhury, "Rf fingerprinting unmanned aerial vehicles with non-standard transmitter waveforms," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 15518–15531, 2020.

[31] G. Xiaolin, W. Wenjia, G. Xiaodan, Z. Ling, M. Yang, and A. Song, "Probe request based device identification attack and defense," *Sensors*, vol. 20, no. 16, p. 4620, 2020.

[32] Y. Li, J. Barthelemy, S. Sun, P. Perez, and B. Moran, "A case study of WIFI sniffing performance evaluation," *IEEE Access*, vol. 8, pp. 129224–129235, 2020.

[33] R. R. Chowdhury, S. Aneja, N. Aneja, and E. Abas, "Network traffic analysis based iot device identification," in *2020 the 4th International Conference on Big Data and Internet of Things*, pp. 79–89, Singapore, 2020.

[34] Y. Meidan, M. Bohadana, Y. Mathov et al., *N-Baiot—Network-based Detection of Iot Botnet Attacks Using Deep Autoencoders*, UCI Machine Learning Repository, 2018.

[35] A. Aksoy and M. H. Gunes, "Automated iot device identification using network traffic," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–7, Shanghai, China, 2019.

[36] N. Ammar, L. Noirie, and S. Tixeuil, "Network-protocol-based iot device identification," in *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 204–209, Rome, Italy, 2019.

[37] B. Bezawada, M. Bachani, J. Peterson, H. Shirazi, I. Ray, and I. Ray, "Behavioral fingerprinting of iot devices," in *Proceedings of the 2018 Workshop on attacks and solutions in hardware security*, pp. 41–50, Toronto, Canada, 2018.

[38] M. A. Muhammad, A. Ayesh, and I. Wagner, "Behavior-based outlier detection for network access control systems," in *Proceedings of the 3rd International Conference on Future Networks and Distributed Systems*, pp. 1–6, Paris, France, 2019.

[39] A. J. Pinheiro, B. JdM, C. A. Burgardt, and D. R. Campelo, "Identifying iot devices and events based on packet length from encrypted traffic," *Computer Communications*, vol. 144, pp. 8–17, 2019.

[40] M. Uras, R. Cossu, E. Ferrara, A. Liotta, and L. Atzori, "Pma: a real-world system for people mobility monitoring and analysis based on wi-fi probes," *Journal of Cleaner Production*, vol. 270, article 122084, 2020.

[41] N. Soltanieh, Y. Norouzi, Y. Yang, and N. C. Karmakar, "A review of radio frequency fingerprinting techniques," *IEEE Journal of Radio Frequency Identification*, vol. 4, no. 3, pp. 222–233, 2020.

[42] T. Jian, B. C. Rendon, E. Ojuba et al., "Deep learning for rf fingerprinting: a massive experimental study," *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 50–57, 2020.

[43] J. Bassey, D. Adesina, X. Li, L. Qian, A. Aved, and T. Kroecker, "Intrusion detection for iot devices based on rf fingerprinting using deep learning," in *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 98–104, Rome, Italy, 2019.

[44] G. Li, Y. Jiabao, Y. Xing, and H. Aiqun, "Location-invariant physical layer identification approach for wifi devices," *IEEE Access*, vol. 7, pp. 106974–106986, 2019.

[45] K. Sankhe, M. Belgiovine, F. Zhou et al., "No radio left behind: radio fingerprinting through deep learning of physical-layer hardware impairments," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 1, pp. 165–178, 2019.

[46] V. Brik, S. Banerjee, M. Gruteser, and O. Sangho, "Wireless device identification with radiometric signatures," in *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pp. 116–127, San Francisco, CA, USA, 2008.

[47] S. Japertas, A. Budnikas, and G. Činčikas, "Identification technology of mobile phone devices using rff," in *Proceedings of the 11th International Conference on Wireless Information Networks and Systems*, pp. 1–6, Vienna, Austria, 2014.

[48] F. Zhuo, Y. Huang, and J. Chen, "Radio frequency fingerprint extraction of radio emitter based on i/q imbalance," *Procedia Computer Science*, vol. 107, pp. 472–477, 2017.

[49] S. Wang, T. Tuor, T. Salonidis et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, 2019.

[50] M. Asad, A. Moustafa, T. Ito, and M. Aslam, "Evaluating the communication efficiency in federated learning algorithms," in *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 552–557, Dalian, China, 2021.

[51] H. Mingzhe Chen, "Wireless communications for collaborative federated learning," *IEEE Communications Magazine*, vol. 58, 2020.

[52] S. Aneja, M. A. X. En, and N. Aneja, "Collaborative adversary nodes learning on the logs of iot devices in an iot network," in *2022 14th International Conference on COMmunication Systems NETworkS (COMSNETS)*, pp. 231–235, Bangalore, India, 2022.

[53] W. Pin, L. Zhihui, Q. Zhou et al., "Bigdata logs analysis based on seq2seq networks for cognitive internet of things," *Future Generation Computer Systems*, vol. 90, pp. 477–488, 2019.

[54] Y. Shen, E. Mariconti, P. A. Vervier, and G. Stringhini, "Tiresias predicting security events through deep learning," in *Proceedings of 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 592–605, Toronto, Canada, 2018.

[55] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6g communications: challenges, methods, and future directions," *China Communications*, vol. 17, 2020.

[56] H. Yang and X. Xie, "An actor-critic deep reinforcement learning approach for transmission scheduling in cognitive internet of things systems," *IEEE Systems Journal*, vol. 14, 2020.

[57] Z. Shi, X. Xie, H. Lu, H. Yang, M. Kadoch, and M. Cheriet, "Deep-reinforcement-learning-based spectrum resource management for Industrial Internet of things," *IEEE Internet of Things Journal*, vol. 8, 2021.

[58] X. Wang, C. Wang, and X. Li, "Federated deep reinforcement learning for internet of things with decentralized cooperative edge Caching," *IEEE Internet of Things Journal*, vol. 7, 2020.

[59] S. Savazzi, M. Nicoli, and V. Rampa, "Federated learning with cooperating devices: a consensus approach for massive IoT networks," *IEEE Internet of Things Journal*, vol. 7, 2020.

[60] Z. Tang, H. Haiyang, and X. Chonghuan, "A federated learning method for network intrusion detection," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 10, article e6812, 2022.

[61] U. Latif, "Federated learning for internet of things: recent advances, taxonomy, and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, 2021.

[62] A. Imteaj, U. Thakker, S. Wang, J. Li, and M. H. Amini, "Federated learning for resource-constrained iot devices: panoramas and state-of-the-art," 2020, https://arxiv.org/abs/2002.10610.

[63] Y. Liu, "Privacy-preserving traffic flow prediction: a federated learning approach," *IEEE Internet of Things Journal*, vol. 7, 2020.