Research online Publishing
www.ronpub.com

Open Access

Open Journal of Internet of Things (OJIOT)
Volume 10, Issue 1, 2025
http://www.ronpub.com/ojiot
ISSN 2364-7108

Automatic Identification and Classification of IoT Devices in Computer Networks – An Overview of Opportunities and Challenges

Maurizio Petrozziello, Martin Kappes, Christian Baun

Faculty of Computer Science and Engineering, Frankfurt University of Applied Sciences,
Nibelungenplatz 1, 60318 Frankfurt am Main, Germany
{petrozziello, kappes, christianbaun}@fb2.fra-uas.de

Abstract

Discovering IoT devices joining a network is essential for network management, security and optimization. Knowing what is happening on a computer network and finding those IoT devices is necessary to counter hacker attacks. To address the security challenges of IoT devices, we present identification (discovery) and classification. This gives the reader an overview of both areas, which need to be considered together; the very fact that there are many techniques and protocols for managing and communicating with IoT devices makes them both worth considering. Due to the differences in discovery and classification of IoT devices, we first present the provisioning part of the IoT device lifecycle and then discuss the different classification approaches. This thesis also describes the importance of feature extraction for classification and the difference between packet and flow features. In addition, this work discusses the difference between statistical, machine learning and artificial intelligence based classification methods, including large language models and quantum computing. In short, this thesis discusses relevant IoT device discovery and traffic classification techniques, applications, challenges and future directions.

Type of Paper and Keywords

Research review: *IoT device discovery and binding, classification of IoT network traffic, feature extraction for classification, large language models, quantum computing*

1 Introduction

The growing number of IoT devices worldwide and the resulting security risks make it necessary to identify IoT devices. The first step in improving security and management is to automatically discover IoT devices. Assigning a device type, called a fingerprint, can help identify unknown network devices. Device classification is often confused with similar approaches such as application traffic classification and intrusion detection. However, classification methods have different scopes depending on the purpose for which they are used. Traffic classification is a step towards device classification and intrusion detection. Therefore, it is important to thoroughly examine each area before evaluating the different classification approaches.

Digital transformation (Industrialization 4.0) in all industries, but especially in small and medium enterprises and smart home environments, is a challenge in terms of smart device management and security. In addition, there is skepticism about IoT networks and the protection of privacy and data security, especially in small and medium-sized businesses and smart home environments. According to studies by various institutions such as Bitcom [17], TÜV-Süd [92] and BAIN-COMPANY [141], the acceptance of IoT solutions among medium-sized companies and private users is still low, especially in the context of data security and privacy [47]. Standardization in the networking of IoT devices has not yet been established, despite existing technologies, so simple and vendor-independent solutions are lacking.

Devices and their management require easy-to-use and secure platforms. Many vendors and service providers offer solutions that are incompatible with each other and require a mandatory connection of IoT devices to the cloud, which inhibits the adoption of such systems and will lead to problems in the future due to the limited transmission capacity of the Internet. Smart gateways are one way to address the challenges of IoT networking. They are installed close to the data source in the LAN (local area network) or in an edge cloud (decentralized data processing at the edge of the network) and use intelligent data filters to minimize the amount of data to be transmitted [12, 49]. Organizations such as LF-EDGE [86], Edge-Stack [44] and others have also addressed the problem, but they are not yet fully established in IoT solutions for small and medium businesses or smart homes.

The problem with discovery and binding, also called identification in this paper, is that most frameworks use active network scanning to do this, giving the impression that the process is sufficiently secure, but active network scanning and probing is usually at odds with corporate security policies and smart home security requirements.

However, to increase the security of IoT networks, it is necessary to detect new or unknown IoT devices in a timely manner and control them through security policies before they are added to a network or smart home. Using rules to isolate them from the network if the device is not already inventoried is essential and not an easy task. The inclusion of discovery and binding techniques are, in our opinion, very important components in this paper to understand the context of IoT discovery and classification.

1.1 Motivation

Even though IoT devices and IoT networks have been established for many years, there is still no reliable and established mechanism for classifying IoT devices. This is a significant problem for the entire smart home sector and for Industry 4.0.

Existing works that do surveys on IoT device discovery and classification are in some ways outdated because they do not cover newer topics like LLMs (Large Language Models), some of the works include the use of GANs (Generative Adversarial Networks) but the scope is slightly different. LLMs are specialized AI models to generate human-like text and GANs are AI models that can generate results in different forms like images and code. Another point is the consideration of hybrid machine learning methods combined with similarity approaches on unique features like packet lang distributions [42]. So at this point, we believe that these techniques should be considered in more detail with respect to the identification and classification of IoT devices. Furthermore, they do not provide insight into statistical approaches that do not use ML-based techniques, such solutions show good results, although they have often been used in application classification, as shown in the work of [27, 28, 33, 42] and [43]. In addition, in the last few years, completely new possibilities have emerged in the field of network traffic classification, with the combination of quantum computing and artificial intelligence as a major goal. All this has encouraged us to present a survey that examines the possibilities of classifying IoT devices and compares their capabilities and limitations.

Mohd et al. [99], in their work, show a good overview about the trend and classification of the Internet of Things by analyzing surveys between 2011 and 2019. Already in this work, the growing need for further research in this field is described. However, this paper mainly analyzes the surveys.

Sánchez et al. [142] present a survey that includes a detailed consideration of the whole problem. But their work doesn't consider new AI-based approaches such as generative transformers.

Shriyal et al. [131] present an overview of the classification of IoT devices, but their focus is on securing the communication of IoT devices through blockchain technologies.

Hauda et al. [78] present a detailed survey in IoT device classification, focusing mainly on machine learning techniques.

Yongxin at al. [88] also presents a detailed overview of IoT device classification with a focus on machine learning, although they mention AI-based approaches, but do not provide specific case studies.

Nguyen and Armitage [101] present a detailed survey of machine learning based network traffic classification in general. This work provides a solid overview of the use of artificial intelligence in the application classification domain.

In recent years, some work has been published on the use of generative transformers for IoT traffic generation and classification [81, 94], but to the best of our knowledge, in the period we are doing this work, no survey has yet taken a closer look at the use of large language models (LLMs) in relation to IoT device discovery and classification.

Table 1: Compassion of the included techniques of the revisited surveys

		TECHNIQUES CONSIDERED					
Survey	Objectives	Supervised machine learning	Unsupervised machine learning	Deep learning	Statistical Similarity	Generative Transformers	LLMs
[99]	Surveys	✓	✓	✓	✗	✗	✗
[142]	Devices in general	✓	✓	✓	✗	✗	✗
[131]	IoT devices	✓	✓	✓	✗	✗	✗
[78]	IoT devices	✓	✓	✓	✗	✓	✗
[88]	IoT devices	✓	✓	✓	✗	✗	✗
[101]	Application	✓	✓	✓	✓	✗	✗
Our Work	IoT Devices	✓	✓	✓	✓	✓	✓

Our research has shown that some approaches have limited success in classifying IoT devices in real networks due to the variety of protocols they use, so we need a different perspective on these problems. For example, in the paper [35], they use the TCP window size as a feature for classification, but there are many IoT devices that only use the UDP protocol to communicate with the cloud or the gateway. The authors of [21] use TCP window size and DNS requests as features and DHCP options for a second classifier. However, this approach can only distinguish IoT from non-IoT devices, not device type.

Some researchers use the startup sequences of IoT devices and have to define the end of the startup sequence experimentally, after a number of packets [96], but this may vary in the real functionality of the IoT device. Furthermore some works report high accuracy [16, 35], but the results are produced with their own datasets and tools, so a reproduction is very difficult. Machine learning approaches usually aren't fast enow to face the challenges of real-time classification problematic that are needed for efficient working defense strategies, because to train the model or label the datasets need much time. Another example is the use of IP addresses, but the IP addresses of servers and services often do not remain constant. Due to the nature of cloud solutions, they use a few servers for the services they provide. On the other hand, some types of IoT devices use the same cloud provider for their settings and communication patterns. Analyzing DNS requests in network data can lead to privacy violations [21]. A deeper insight is reported in the chapter 4.

From our point of view, there are still challenges to overcome in this area, and the following main aspects need to be addressed for IoT network traffic classification in modern networks.

- It is difficult to objectively validate the various proposals. Most work is based on self-generated datasets tagged with techniques of unknown reliability.
- Today's networks operate at high data rates that must be analyzed in real time to isolate attacker interactions and quickly minimize the resulting damage. Such operations require expensive hardware and high throughput from monitoring and gateway devices.
- High maintenance costs: Machine learning techniques rely on a long training phase and require human intervention to label and balance the data sets.
- Published approaches based on artificial intelligence (AI) and large language models (LLMs) are not yet sufficiently researched for use in network traffic problems and require further consideration and testing. This is especially true for the quality of results and security against already compromised models.

In view of the aspects mentioned above, it is also necessary to consider the comparison of methods and solutions with standardized conditions and data sets. The following challenges should be considered:

- Scalability - Handling the massive scale of IoT deployments.
- Real-time Processing - Ensuring timely classification to meet QoS requirements.
- Privacy and Security - Protecting sensitive data while classifying traffic.
- Data Imbalance - Dealing with imbalanced datasets where some traffic types are underrepresented.
- Adaptability - Continuously adapting to evolving IoT devices and traffic patterns.

1.2 Structure of This Work

The structure of this work is as follows:

- Section 2 describes the research methodology we used.
- Section 3 provides an overview of the lifecycle, including all the steps necessary to discover, identify, bind, and manage IoT devices, and related work in this area, which is essential to understanding the security problem that arises in IoT networks and smart homes.
- Section 4 presents the classification methodology, feature extraction techniques, and a comprehensive linkage and analysis of related work. The knowledge gained from this analysis is crucial for understanding the complexities of IoT device categorization and security.
- Section 5 includes the conclusion that shows the need for further research and development work in the area of IoT device classification and intrusion detection. Finally, there is a list of future work motivated by the conclusion.

2 Our Methodology of Research

Our research study is divided into two parts because we are also looking at the provisioning part of IoT devices, which is crucial for discovering the remaining challenges in this field.

For example, to increase the security of IoT networks, it is necessary to quickly detect new or unknown IoT devices and control them through security policies before they are added to a network or smart home. In most cases, the majority of IoT solutions (see table 4) rely on encrypted data transfer between IoT devices and frameworks, but attacks can occur even before the binding is complete and go undetected by the framework. The use of rules to isolate them from the network if the device has not already been discovered is essential and not an easy task. The inclusion of discovery and binding techniques are, in our opinion, very important components in this paper to understand the context of IoT discovery and classification.

Furthermore, because network traffic is highly dynamic, there are always small variations in the behavior of IoT devices, so the task of discovering and classifying IoT devices in the context of security may take too long, making this a challenge for researchers.

Another challenge comes from the new possibilities of implementing such classification methods using generative artificial intelligence (AI), which could be a promising but as yet unproven improvement in the security of IoT device networks.

We used two research databases, namely Web of Science and IEEE Xplore, but we also used our university library resources. The following is our research methodology. We conducted a systematic search using the following keywords and filters:

- “IoT device discovery”, “IoT device binding”, “IoT device identification”, “IoT device type identification”, “IoT device classification”, “IoT device type classification”, “IoT device identification”, “IoT device type identification”.

  We also use these keywords in our research work to specify the results.

- “header features*”, “flow features*”, “statistical”, “machine learning”, “generative transformers or GANs”, “large language models or LLMs”, “quantum computing”, “survey or review”, “lifecycle”

We used a research assistant, Zotero [153], to organize our research results. The advantage of using such a tool is that we can update the references at any time, which updates the cited record. Another advantage is that we can easily find duplicate works, and the full-text search helps us filter the documents. In the following steps, we sorted the papers and surveys according to their focus and relevance. If a paper had the required focus, we included it; otherwise, we ignored it. Finally, we filtered out all documents that did not have a specific focus for our work by considering the abstract, conclusion, and future work. Then we define the related papers to include. At this point we would like to refer to a great example of a well-done literature review and research methodology [45].

3 Device Identification and Binding

This section provides background information on the variety of IoT protocols and presents the state of the art in IoT device discovery and binding, as well as the latest protocols and techniques in the transmission technologies of IoT devices (smart devices) and the different areas where IoT is implemented, such as smart homes, smart cities, smart factories, or wireless sensor networks (WSN) [6, 122].

3.1 Background

The methods for searching and binding sensors (onboarding) discussed below illustrate how frameworks and gateways work.

Due to the diversity of IoT devices and the protocols used, we decided to first look at the general tasks for discovering and binding an IoT device, which is the first

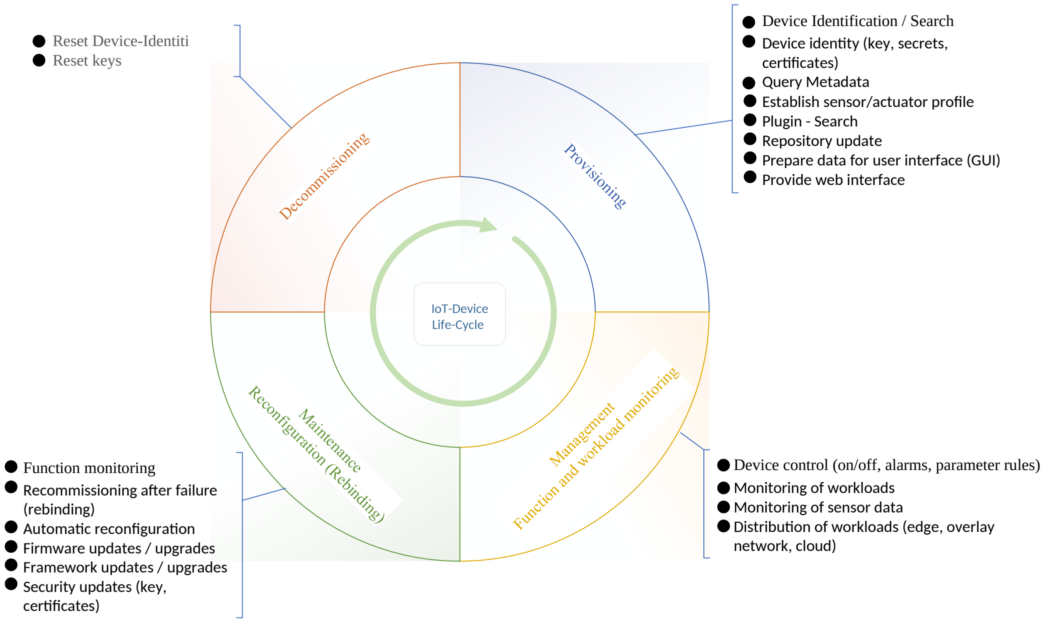
● Reset Device-Identiti
● Reset keys
Decommissioning
Provisioning
● Device Identification / Search
● Device identity (key, secrets, certificates)
● Query Metadata
● Establish sensor/actuator profile
● Plugin - Search
● Repository update
● Prepare data for user interface (GUI)
● Provide web interface
IoT-Device Life-Cycle
Maintenance
Reconfiguration (Rebinding)
Management
Function and workload monitoring
● Function monitoring
● Recommissioning after failure (rebinding)
● Automatic reconfiguration
● Firmware updates / upgrades
● Framework updates / upgrades
● Security updates (key, certificates)
● Device control (on/off, alarms, parameter rules)
● Monitoring of workloads
● Monitoring of sensor data
● Distribution of workloads (edge, overlay network, cloud)

Figure 1: Lifecycle of IoT devices

step in life cycle management, taking into account all the different protocols used by such IoT devices. This part was included to clarify that there is a lot of variation in IoT device communication methods and protocols, and in our opinion, this leads to a better understanding of the overall issues surrounding the operation of IoT devices [46, 119, 122, 137]. Figure 1 shows the general life cycle of an IoT device.

3.1.1 Ontologies and Semantic for the IoT

Ontology is definition for the interaction with digitized and formal form between processes, devices and applications. Usually ontology are a linguistic and formally organized representations of a set of concepts and the relationships between them in a particular domain. In order to systematize the communication between IoT devices and services, semantic technologies have been established with the aim of high-level abstraction of the complex information involved in IoT communication and services [24, 68, 123].

Semantic interoperability involves the ability to establish a common meaning for the data being exchanged and the ability to interpret communication interfaces in a similar way. This can be achieved using semantic models encoded in some form of formal vocabulary. The basic idea is that by providing these structured semantic models of a system, other systems can have an unambiguous understanding of the system using the same mechanisms [113]. The chapter 3.2 lists some works that deal with ontology in IoT networks.

3.1.2 The Everyday Tasks for IoT Device Administration

Below is an overview of the steps required for commissioning or onboarding from the model in Figure 1. Not all steps are necessary for every device. The exact number of steps depends on the level of automation, i.e. the automatic identification and assignment of IoT devices.

- **Commissioning:** The user turns on the IoT device and adds it to the local network.
- **Identification:** The gateway waits on all standard ports of the different protocols for a new device on the network (broadcast, advertising). Many sensors and actuators sent their basic information in these advertising messages. If a device does not send a message, the search process is automatically started

after some time $\Delta t > T$ when the gateway has detected a new network device.

- **IoT device search:** The gateway scans the local network for IoT devices. To scann the network, the gateway sends packets using protocols such as MQTT [111], CoAP [130], UPnP [18], and REST [104] to collect initial metadata about the sensor/actuator. If the search fails, the user is notified, and further details are expected. The search can also be started manually via the web interface.

- **Create sensor profile:** If the query is successful, the gateway can make a profile of the IoT device through various queries. Various device description methods are used, such as TEDS [85], SensorML [112] or IoTivity [106].

- **Plug-in search:** With the obtained data (profile), a web search for a suitable plugin is started and installed in the gateway. This indicates that everything is now available in the gateway to exchange all information with the sensor/actuator.

- **Update the repository:** The resulting metadata is stored in a repository.

- **Design the user interface:** Using the available information, the web interface in the gateway is now adapted to the sensor/actuator and the user is presented with a suitable interface for reading or operating the IoT device.for reading or operating the IoT device is presented to the user.

If the sensor/actuator mapping with the described actions is not successful, the user is redirected to the cloud platform or web interface of the IoT device provided by the manufacturer to perform manual commissioning.

3.1.3 Standards and Protocols for IoT Device Identification and Binding

With the growing number of IoT devices communicating over the Internet, these requirements are constantly changing. As a result, the requirements for quality of service (QOS) and data processing are also changing. Many Internet organizations and manufacturers offer IoT infrastructures and numerous cloud platforms, such as Works with HomeKit from Apple, Works with Nest from Google, and also Amazon, Microsoft, Siemens, and Bosch, among others. However, most require management and billing through the cloud. Some organizations and alliances are striving for standardization in the IoT, but are using different technologies to achieve this goal, making standardization difficult.

Some of the organizations and standards are:

- **Connectivity Standards Alliance (CSA)** [8] This organization and its members develop standards, tools, and platforms for global object communication.

- **The Internet Engineering Task Force (IETF)** [72] was founded in 1986 and was one of the first standards development organizations. A special feature is that all documentation is published as RFCs (Requests for Comments), which are usually of a high quality.

- **If-This-Then-That (IFTTT)** [145] is a service hub at the application layer (service gateway).

- **Open Geospatial Consortium (OGC)** [31] provides the Sensor Observation Service (SOS) and the SensorThings API defined for service interfaces in the IoT landscape; they also define a standard to describe the metadata mapping of a sensor, the Sensor Interface Descriptor (SID). According to the SID, a SID interpreter (gateway) can translate a sensor protocol into the Open Platform Communications (OPC) [109] web service interface.

- The **OpenIoT** [118] standard describes an open source web service maintained by the European Union (EU) and released jointly by developers from industry and academia in 2012.

- **Object Management Group (OMG)** [62] is an organization primarily concerned with the development of standards for vendor-independent, cross-system object-oriented programming. One of the services that has emerged from this organization is the DDS (Data Distribution Service) [61], a communication standard between distributed systems.

3.1.4 Communication Protocols for the IoT

Nowadays, wireless technologies such as WiFi [9], Bluetooth [132], cellular (LTE, 4G, 5G) [98] are widely used in addition to traditional connections via VDSL (Very High Speed Digital Subscriber Line) [1] and Ethernet. Special transmission technologies developed for IoT and building automation, such as ZigBee [10], LPWAN (Low Power Wide Area Network) [50], and Z-Wave [11], are also available. RFDI (Radio-Frequency Identification) [7] and NFC (Near Field Communication) [54] are used to read device data and transfer data between them, both technologies are widely used in modern payment systems.

The communication technologies are divided into two main categories, called short-range and long-range networks. The following lists and explains the main characteristics and technologies. In addition, to better understand the relationships between the individual technologies and protocols for the IoT landscape, we include Figure 2, which shows a representation based on the OSI (Open Systems Interconnection) reference model [19, 20, 53, 144], and Table 2, which lists the most common communication protocols for the IoT with detailed technical characteristics. An overview is provided by [36].

- **Bluetooth** [132] is a packet-based master/slave protocol with a short-range wireless technology that is most commonly used to transfer data between mobile and fixed devices over short distances. An important fact is that Bluetooth uses a different protocol stack and doesn't follow the OSI TCP/IP model [53, 144] and is suited as a wireless replacement for the serial communication interface of computers.

- **Z-Wave** [11] is a Mech-Network topology used in smart home environments. This technology connects devices to a control unit with a longer range and lower power consumption than Bluetooth. It works on top of the connection topology used by the connected devices and the control unit.

- **ZigBee** [10] is another communication protocol for connected devices that uses its own transmission standard and is used in products from Philips Hue, Xiaomi, OSRAM and others.

- **WiFi** [9] is a suite of networking protocols based mostly on the IEEE 802 standard. This communication protocol is commonly used to connect devices such as personal computers and peripherals such as printers and can also be used as a network extension and/or replacement where wired networks are not possible. In fact, WiFi is not ideal for connecting IoT devices because of its relatively high susceptibility to interference from walls and other building structures and its widespread use in home networks, resulting in overcrowded frequency bands that are often susceptible to interference. Nevertheless, WiFi is often used for IoT devices because it has a much better range.

- **RFID Radio-Frequency Identification** [7] this technology enables the contactless exchange of data using electromagnetic radio waves. An RFID system consists of an RFID reader, an RFID transponder (tag) and an antenna. The RFID transponder is the heart of the system and consists of a microchip and an antenna. Information is transmitted from an encoded memory chip (called a ”transponder” or ”tag”) to an RFID reader via the antenna.

- **NFC (Near Field Communication)** [54]. NFC is based on RFID protocols and, unlike RFID, is capable of transferring data between devices. The main difference from RFID is that an NFC device can act not only as a reader, but also as a tag (card emulation mode). In peer-to-peer mode, it is also possible to transfer information between two NFC devices.

- **4G, 5G (GSM - Global System for Mobile Communications) and LTE** [48] are cellular network technologies used for mobile communications. LTE is an advanced 4G system, the main differences between the technologies are the increased data rates and lower latency.

  NB-IoT [107] is a wireless standard designed specifically for the Internet of Things. NB-IoT uses narrow licensed radio frequency bands to efficiently transmit data, and is specifically focused on indoor deployments. The narrow technologies allow many devices to connect simultaneously without overloading the network.

- **LoRa or LoRaWAN Long Range Wide Area Network** [50] is also a wireless standard for the IoT landscape. LoRaWan has a shorter range than NB-IoT and, unlike NB-IoT, uses unlicensed frequency bands.

3.1.5 Data Protocols for the IoT

For IoT devices in the so-called consumer space, REST (Representational State Transfer) [104] based on TCP/IP protocols is typically used for data transfer. However, the HTTP (Hypertext Transfer Protocol) [103] protocol was not designed for IoT applications, but for the delivery of web content. With this protocol, the significant overhead is a disadvantage compared to MQTT [111] and CoAP [130], so communication bottlenecks can occur when bandwidth is low. The model is based on a client/server architecture and uses URI (Uniform Resource Identifier) [15] and REST methods such as GET, POST, PUT, and DELETE to access resources. The fact that HTTP [103] transfers data bidirectionally makes it unusable for push messages, but the HTTP2 extension makes this possible. On the other hand,

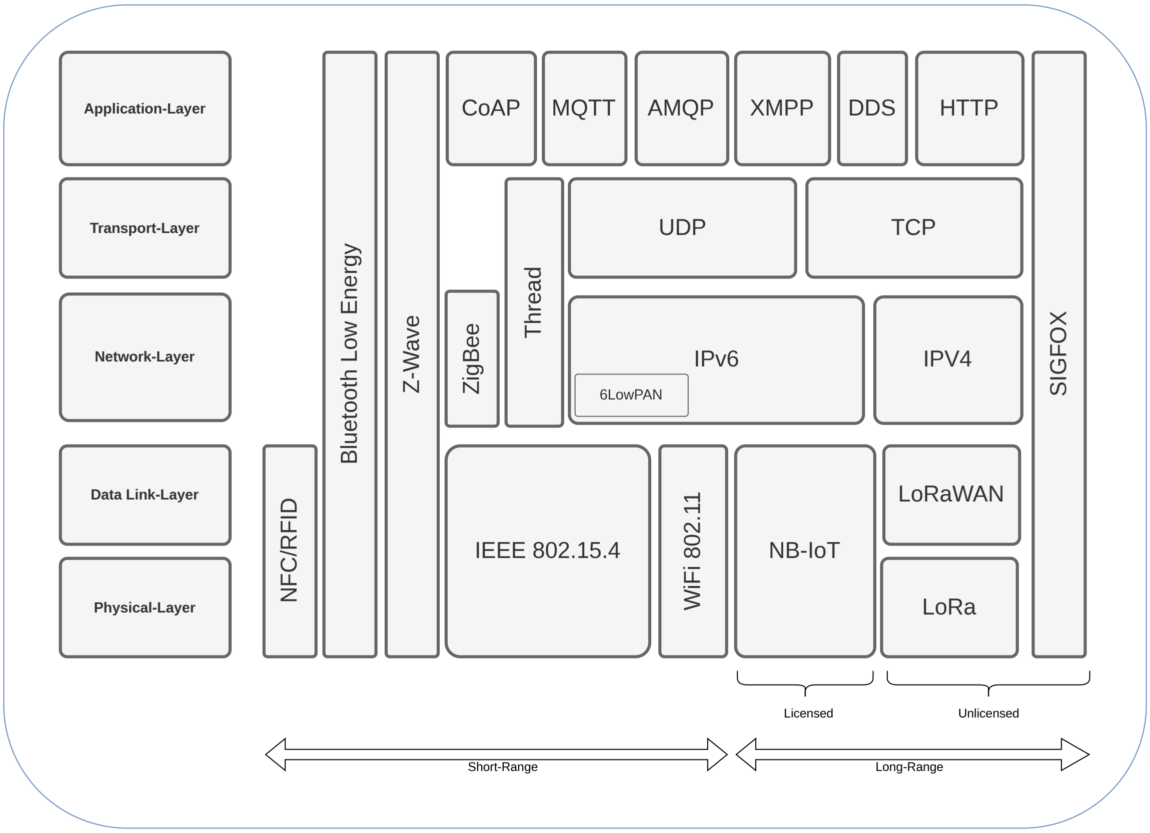

Figure 2: Protocols in the IoT landscape

Table 2: Communication technologies for IoT

		SHORT-RANGE-NETWORKS			
Name	**Frequency Band**	**Data Rate**[a]	**Topology**	**Range**	**Power**
Bluetooth LE	2.4 GHz	125 Kbps - 2 Mbps	Mesh	30 m	Low
Z-Wave	800 MHz - 900 MHz	100 Kbps	Mesh	65 m	Low
ZigBee	2.4 GHz	250 Kbps	Star, Mesh	300 m	Low
WiFi	2.4/5 GHz	5 Mbps - 5 Gbps	Star	100 m	High
RFID	125 kHz - 960 MHz	10 - 640 Kbps	Point to Point	300 m	Low
NFC	13.5 MHz	10 - 420 Kbps	Point to Point	5 cm	Low
Thread	2.4 GHz	250 Kbps	Mesh	70 m	Low
		WIDE-RANGE-NETWORKS			
Name	**Frequency Band**	**Data Rate**[a]	**Topology**	**Range**	**Power**
4G	2 - 8 GHz	3 - 100 Mbps	Star	15 km	High
5G	3 - 300 GHz	3 - 10 Gbps	Star	25 km	High
LTE	700 MHz - 2.6 GHz	3 - 100 Mbps	Star	15 km	High
NB-IoT	1 GHz	100 Kbps	Star	32 km	Low
LoRa	1 GHz	100 Kbps	Star	16 km	Low

[a] The data rate always depends on many factors of a given scenario like frequency, topology, obstacles, interference, and number of participants.

HTTP is integrated into many programming languages via SDKs (Software Development Kits) and can be implemented quickly.

Industries and developers can choose from various technologies and protocols to build their solutions, mostly referring to the Open System Interconnection Model OSI [53, 144] (see figure 2). The most common protocols and standards in the IoT landscape are:

- **AMQP** [110] (Advanced Message Queuing Protocol) is an M2M protocol developed by John O'Hara in the United Kingdom in 2003. It is a messaging protocol. AMQP supports both request/response and publish/subscribe methods.
- **CoAP** [130] (Constrained Application Protocol, RFC 7252) is based on UDP and is designed for sensors and actuators with low processing power. The overhead of the protocol is low, which has a positive impact on the bandwidth of data packet transmission. CoAP uses a client/server architecture and is designed for machine-to-machine (M2M) applications such as building automation and smart energy projects.
- **DDS** [61] (Data Distribution Service) is a standard developed by the Object Management Group (OMG) for real-time systems. This protocol and API enables device data exchange with high performance.
- **MQTT** [111], (Message Queuing Telemetry Transport). The MQTT protocol is one of the most widely used IoT communication paths because it is very resource efficient, using a publish/subscribe architecture based on TCP/IP. The overhead is low, but a MQTT broker must be developed to translate the information from client to server and vice versa [129].
- **XMMP** [125] (Extensible Messaging and Presence Protocol) is a set of open technologies for instant messaging, presence, multi-party chat, voice and video calling, collaboration, content syndication, and general routing of XML data. XMPP is an XML-based protocol that uses extensible, real-time instant messaging and presence information. For more information, see RFC 3920 [125] and RFC 3921 [124].
- **UPnP** (Universal Plug and Play) [18] (RFC-6970) is a technology designed for smart homes and small offices based on local area networks. It was originally developed by Microsoft Corporation in 1999. The technical inspiration for UPnP was to provide a distributed computing framework based on Web technologies for small networks, especially home environments. UPnP simplifies device installation and connectivity. However, its security implications remain a risk and potential for exploitation by malicious actors.

Table 3 shows the common application layer protocols for IoT. However, application protocols are often listed together with transmission technologies such as WiFi [9], Z-Wave [11], or ZigBee [10]. Strictly speaking, they do not describe the same working environment. For this reason, we present the protocols and technologies in separate tables.

Many Internet organizations and manufacturers offer IoT infrastructures and numerous cloud platforms, such as Works with HomeKit from Apple, Works with Nest from Google, and also Amazon, Microsoft, Siemens, and Bosch, among others. In recent years, a lot of work has been done on machine-to-machine (M2M) gateways, such as FIWARE [55], OpenMTC [52], OpenIoT [118] or GSN [2, 3, 67, 4]. These gateways act as a layer between physical sensors and virtual sensor data.

The papers [2] and [136], show how long research has been done on optimal solutions in the IoT. A middleware such as Global Sensor Network (GSN) [4] is repeatedly described, which virtually replicates the IoT device through an abstraction. More recent implementations of such methods are platforms such as OpenHab [57].

3.1.6 Frameworks for the IoT

There are several frameworks and APIs (Application Programmable Interface) for building IoT infrastructures. Most are based on programming languages such as C++, Java, and Python. Some primarily use one IoT protocol, such as MQTT [111]. These frameworks can often interact with other IoT protocols through extensions. Some of these frameworks include

- **Node-RED** [58] is a graphical framework originally developed by IBM for the development of event-based applications. The framework allows flow-based programming of the behavior of IoT devices. This project has been open source since 2013, and a large number of users in the community have developed extensions.
- **IoTivity** [106] is a framework that implements the Open Connectivity Foundation (OCF) standard. Modules can be created and modified using any editor in JSON format. This software framework enables device-to-device connectivity and is an open source project released under the Apache 2.0 license.

Table 3: Common data protocols for IoT

Name	Architecture	Transport Protocol	QoS	Security	Areas of application
AMQP	Publish/Subscribe and Request/Response	TCP	✓	SSL/TLS	client/server messaging, IoT device management
CoAP	Request/Response	UDP	✓	DLTS	Smart energy, healthcare, monitoring, automation
DDS	Publish/Subscribe	TCP and UDP	✓	TLS and DTLS	Industrial automation, healthcare, transportation
MQTT	Publish/Subscribe	TCP	✓	SSL/TLS	Indrustie and home automation, remote sensing, agriculture
XMPP	Publish/Subscribe and Request/Response	UDP	✗	SASL and TLS	Social networking, gaming, collaboration, healthcare
UPnP	Plug and Play	UDP	✗	✗	Discover devices in a Local area Network

- **Thinger.IO** [135] is an open source platform for connecting and monitoring IoT devices. It supports Arduino, Linux, Sigfox [148], and MQTT IoT devices and can be combined with Node-RED. This platform is focused on monitoring IoT devices and does not have low-code capabilities.

- **Zetta** [60] is a programming environment built on Node.js that allows devices to be connected through an API representation. The platform combines REST APIs and WebSockets, exposing each IoT device as an API. A Zetta server can communicate with microcontrollers such as Arduino and Spark Core and continuously stream large amounts of data.

- **ThingsSpeak** [147] is a cloud platform for analyzing, aggregating, and visualizing data from IoT devices. The framework is tightly integrated with Matlab to visualize and analyze the sensor data. The services are freely available for non-commercial projects.

- The **OpenHAB** [57] Foundation provides a cloud-free framework that can be installed on the most modern operating systems, including Linux, Windows, macOS, and Raspberry Pi, but can also be combined with cloud solutions such as Google Assistant, Amazon Alexa, and Apple HomeKit, among others. This open source solution provides a high level of privacy for user data.

- **ThingsBoard** inc. [146] evolved from a startup that provides an IoT framework, namely ThingsBoard, for device management and monitoring. This solution enables connectivity over various standard IoT protocols such as MQTT, CoAP and HTTP. This framework can also be installed on-premises, including on Linux, Windows, macOS, Raspberry Pi, and Docker, but can also be integrated with cloud solutions from AWS (Amazon), Azure (Microsoft), Google Cloud, and DigitalOcean. They offer a community edition that is open source and free, as well as professional and cloud editions.

- **DeviceHive** [39] is an open source IoT framework solution that is offered as a Docker image, manually installed on Linux or in a cloud platform called DeviceHive Playground. To present IoT data and metrics, the framework can be combined with applications such as Grafana [84], an open source data visualization framework.

- **ioBroker** [73] is another open source IoT framework that is offered on-premises (own hardware) or in the cloud with several licensing models and a free community version. However, the installation is a bit more demanding than the other solutions and sometimes requires more in-depth know-how.

- **Matter** [8] is an IPv6-based application layer framework that supports TCP and UDP in the transport layer and sits virtually on top of WiFi, ZigBee, and Thread technologies. This standard is supported by the Connectivity Standards Alliance and comes with an open source software development kit (SDK). This standard is already supported by many companies, including Amazon, Apple, Google, and Samsung, and can be integrated into various operating systems, including Android, Wear OS, iOS, iPadOS, watchOS, and Windows.

The table 4 shows frameworks that are commonly used in practice. However, most of these frameworks are offered with many licensing rules, but most offer a free community version or a free license for a limited number of IoT devices.

Table 4: IoT frameworks

		PRORRIETARY		
Name	**Protocols**	**Architecture**	**Device Mgmt.**	**License**
(AWS)IoT Platform	HTTP, MQTT, WebSockets	Cloud and EDGE	✓	As per customer requirement
Bosch IoT Suite	HTTP, MQTT	Cloud-Service	✗	As per customer requirement
Cisco IoT Platform	MQTT among other	Cloud and EDGE	✓	As per customer requirement
Microsoft Azure IoT Suite	MQTT, AMQP, WebSockets, HTTPS	Cloud-Service	✓	As per customer requirement
IBM Watson IoT Platform		Cloud-Service	✗	As per customer requirement
Oracle IoT	HTTP, MQTT	Cloud-Service	✗	As per customer requirement
Siemens Cloud Connect	OPC Unified Architecture, MQTT	Cloud-Service	✗	As per customer requirement
Thinger.IO	MQTT	Cloud and On-premise	✗	As per customer requirement
		NON PROPRIETARY		
Name	**Protocols**	**Architecture**	**Device Mgmt.**	**License**
Node-RED	MQTT, HTTP, Raw TCP/IP	Run on Host	✗	Open Source, Free
IoTivity	CoAP, MQTT, AMQP	Cloud and Machine-to-Machine	✗	
Zetta	API, all device protocols	Run everywhere	✗	Open Source, Free
Things-Speak	Resr API, MQTT API	Cloud	✓	Free for non-commercial use
ioBroker	MQTT and adapters	On premise and Cloud (with Alexa or GoogelHome)	✓	Free for non-commercial use
Home-Assistent	MQTT and adapters	On premise and Cloud (with integration platforms)	✓	Open Source, Free
OpenHab	MQTT and adapters	On premise and Cloud (with integration platforms)	✓	Open Source, Free
ThingsBoard	MQTT, CoAP, HTTP	Cloud-Service and On-premise	✓	Free for non-commercial use
DeviceHive	REST API, WebSocket, MQTT and adapters	On premise and Cloud	✗	Open Source, Free
Matter	SDK, all device protocols with adapters	On top of application	✗	Open Source, Free

3.2 Related Works in Device Discovery and Binding

Several works exist that provide literature reviews in the field of device onboarding and binding. One example is the work of Bellendorf et al. [14]. This section focuses on some selected works and solutions that provide compatibility with the IoT protocols that are most used in practice.

Nugent et al. [105] present HomeML, an XML-based format for data exchange in intelligent environments. The goal is to solve problems caused by heterogeneity. The proposed format can be used to describe sensors within a room. Such models can help to create new frameworks, but do not provide a ready-made solution for the user.

Ishaq et al. [74] present a REST-based interface for accessing sensors and retrieving data. However, the authors also assume the existence of a sensor network.

Li et al. [87] and Da Silva et al. [59] propose to use the TOSCA standard to classify the basic structure of IoT applications. However, this approach is based on a pure cloud solution, which does not meet our requirement to of process data as locally as possible.

Mayer et al. [93] present a method that uses semantic metadata reasoning with a visual modeling tool to overcome the challenges of configuring smart devices. In this approach, the user specifies the characteristics of his smart environment. The system then determines whether the goals can be achieved and what actions are required based on the available services. However, this approach assumes that devices are already integrated into the network environment.

Vogler et al. [150] present LEONORE, a scalable deployment framework for deploying and running custom application logic directly on the IoT gateway. However, in order to know which IoT gateways are available for deployment, the IoT gateways must have a pre-installed local agent. This agent binds the gateway to the framework by providing its unique identifier and profile data such as ID, MAC address, and command set.

Martínez et al. [91] describe the Sensor Deployment Files (SDF), which represent the metadata of sensors. These can be retrieved and registered via a framework (Sensor-Thing-API). Such techniques offer the possibility to describing sensors in an automated way.

Broering et al. [23] presented a survey of search (discovery) technologies for IoT devices. This work provides a foundation for understanding the existing techniques and the issues related to the heterogeneity of the IoT landscape.

Ali H. et al. [37] provides a secure search for IoT services, which also analyzes the different protocols, such as multicast DNS (mDNS) and DNS service discovery (DNS-SD). In their paper, they propose a broker-based solution for IoT device discovery.

Khudoyberdiev et al. [82] presents the registration and discovery of embedded systems using a DNS service. This method assumes that users want to access their IoT devices over the Internet. This solution uses the protocol CoAP to communicate with the IoT device.

OntoSensor [68, 123, 139] is a sensor repository for modeling and managing sensors. It combines SensorML [112], IEEE SUMO [115, 116], ISO 19115 [75], OWL [140], and GML standards [108]. This repository aims to provide a methodology for describing sensors in different application domains. By combining many sensor definition languages, the ontology, in turn, becomes very complex. The description and configuration of sensors are standardized in IEEE1451.2 to define an interface. These Transducer Electronic Data Sheet (TEDS) [85] allow sensors to describe themselves. Other description languages exist, such as Siren (hypermedia specification for representing entities) from Google.

In recent years, some work on machine-to-machine (M2M) gateways has also been presented, e.g., FIWARE [55], OpenMTC [120], OpenIoT [32] or GSN [3, 4, 136]. These gateways are a layer between physical sensors and *virtual* sensor data.

However, recent research in this area also shows solutions that take into account criteria such as semantic and syntactic parameters for searching IoT devices, as presented in the work of Cimmino et al. [29].

The listed and following works show how long research has been carried out on this topic to develop optimal solutions for the control and management of IoT devices [3, 67, 136]. In this context, middleware such as Global Sensor Network (GSN) [3, 4] is described, which practically replicates the IoT device through an abstraction. The authors of the paper [2] presented a middleware solution in 2006. Recent implementations of such methods include platforms such as OpenHab [57].

3.3 Summary

Today, various organizations are still working on standardization and specialization to address the challenges of IoT, such as Bosch IoT Suite, Web of Things, and IoTivity, among others, and some have been revived, as in the case of CoRE. However, they are all evolving individually and standardization is still a long way off. Most vendors today offer good IoT platforms to manage and control IoT devices for individual customer needs. For an overview of modern platforms from an academic research perspective, see the work of De Nardis et al. [34]. However, there is still a lack of solutions that can be easily deployed in small and medium enterprises or smart home environments without cloud registration. As presented in the introduction, there is a need for a new perspective on processing and monitoring data from sensor networks and deploying applications for IoT networks, especially in the context of privacy.

4 Classification

As mentioned at the beginning, from our point of view, the classification relevant tasks need to be separated from the administrative tasks in the lifecycle of IoT devices. In the previous chapter, we presented standards and methods for discovery (searching and binding); now we will consider classification at the network level. Identifying and classifying IoT devices at a very early stage when they are connected to the network is essential for the security of IoT networks. Although IoT devices and IoT networks have been around for many years, there is still no reliable and established mechanism for classifying IoT devices. This is a significant problem for the entire smart home sector and for Industry 4.0. For this reason, we explore the possibilities of classifying IoT devices and compare their capabilities and limitations.

4.1 Background

Identifying IoT devices in networks is a classification problem. Each IoT device, with its unique functions and characteristics, presents a challenge to different analysis techniques. Matching these devices from the transmitting network packets is a particularly difficult aspect of the classification process. Classifying IoT devices involves implementing fingerprint mapping at different layers of the OSI protocol stack. The method and features used can vary significantly depending on the research focus.

- The first level of classification is to distinguish whether the network features belong to an IoT device or not, as shown in the work of Bremler et al. [21]. In their work, they use different feature sets and machine learning to compare the accuracy of the classifier on an unseen dataset.
- The second level of classification is to predict the device name (type), e.g., TP-Link Tap, Amazon plug. Most of the published work in this area uses machine learning and packet-level features, for example, the work of Miettinen at al. [96].
- The third level of classification examines the functions of devices to predict whether the device is a sensor or an actor. For example, the work of Hadzovic et al. [65] presents the identification of IoT actors.

Classification, the process of assigning predefined classes to a set of unseen data based on observed attributes or features, is the cornerstone of understanding and interpreting data. The class, a label that describes an object in a particular context, is what allows us to make sense of the data. For example, animals can be classified as dogs, cats, mice, etc. A case refers to an object (an animal in the previous example) whose class is known, while an ”unseen case” is an object whose class is currently unknown (such as a new species). In the case of an IoT device, transmission characteristics are used to determine its class, demonstrating the practicality of classification in different domains. The most common classification approaches are:

- Statistical Methods
- Statistical Methods with machine-learning
- Deep-learning Methods
- Neural-Networks Methods

4.2 Methods for IoT Device Classification

This section provides background information on the classification methods of IoT devices.

4.2.1 Disambiguation

The term device classification is often confused with similar approaches such as traffic classification, intrusion detection, or device fingerprinting. Traffic classification is the field of research that deals with classifying network traffic for various purposes, but it is most commonly used for malicious traffic detection, also known as intrusion detection. Device classification generally categorizes devices with similar functions, such as hubs, cameras, or light bulbs. Device identification can distinguish between the model and manufacturer of the device, such as Google Home Assistant, Alexa, D-Link camera, or TP-LinkPlug. Device fingerprinting gives each device a unique label that can describe the manufacturer and model, such as D-Link-camera1 and D-Link-camera2, two instances of the same device.

Most classifier approaches rely on machine learning to tackle the complex task of identifying devices in a network. However, classification remains a challenge due to the technological and functional diversity of IoT devices.

4.2.2 Port-Based Classification

This classification technique was first used to identify applications. The port number and associated transport layer protocols, TCP and UDP, registered by the Internet Assigned Numbers Authority (IANA) [71], can be used to identify an application generating network traffic. Today, applications use unregistered port numbers or random ports, such as peer-to-peer (P2P) applications. In addition, this approach runs into problems when Network Address and Port Translation (NAT) is used between communication points or when traffic is encrypted at the IP layer.

4.2.3 Payload-Based Classification

This method is advantageous because it produces low false-negative rates; most payload analysis methods involve inspecting the packets and comparing them to a stored signature (pattern). Payload inspection is not compliant with privacy policies and has some significant drawbacks when dealing with encrypted traffic and protocol obfuscation or encapsulation, as much of the network traffic is left unclassified. In addition, payload-based classification requires significantly more processing power to inspect many flows at the high-speed rates of today's network traffic. Regardless of the depth of inspection, the payload-based classifier must be kept up-to-date because application semantics can change with updates.

4.2.4 Statistical-Based Classification

Statistical methods require knowledge of the problem and prior structuring of the parameter selection. In addition, these approaches typically have an explicit underlying probability model. Statistical classifiers use packet or flow-level features to specify patterns that can distinguish different applications or device types. The most commonly used features in this area are packet or flow length, duration, inter-arrival time, or flow idle time. These statistical measurements can be used to analyze the correlation between classes of network traffic. Statistical methods quickly reach their limits due to the large amount of data required to achieve a given level of accuracy, so the researcher combines statistical methods with machine learning methods to overcome the challenges.

4.2.5 Behavioral-Based Classification

This classification method moves the observation point further up the network stack. It includes all traffic transferred from a device to a destination, such as a gateway or cloud service, and attempts to identify the type of devices using heuristic information from the selected features. These can be the IP addresses contacted with the associated port numbers, the transport layer protocol involved, and other features such as the type of service. From this perspective, behavioral classification has the same advantage as statistical classification because it avoids payload inspection and is lightweight.

4.2.6 Machine Learning-Based Classification

In recent years, machine learning approaches have become increasingly important. Such models use a learning process with training data. Since this is similar to statistical classification models, many researchers have combined statistical methods with machine learning methods to achieve high accuracy in network traffic classification and device identification. However, these methods require large amounts of data and usually take some time to collect. An overview of traffic classification techniques that include machine learning techniques is presented in the 2008 paper by Nguyen and Armitage [101], which shows that the topic has been an important area of research for decades. In the field of machine learning classification research, three methods can be distinguished: supervised, semi-supervised and unsupervised.

- Supervised Learning Classification solves a prediction problem. In other words, this method analyzes new network traffic and assigns it to a predefined class according to its feature characteristics. This method requires a labeled data set, which means a significant amount of time.
- Semi-supervised is a method in which some of the samples in the training data are not labeled. Therefore, the algorithm is trained on both labeled and unlabeled data. Such methods are able to take advantage of this additional unlabeled data. Unlike supervised learning, semi-supervised learning is able to classify data faster and more effectively than unsupervised learning.
- Unsupervised learning does not involve predefined classes and labeled data sets. The classifier tries to group similar patterns into the same class. Due to the complicated validation of the results, such methods often require large amounts of data over long periods of time, which is always associated with high costs and effort.

The goals of these methods are different. In unsupervised learning, the goal is to get results from a large amount of new data, while in supervised learning, the nature of the output is already known and must be predicted for new unknown input data. If we look at it closely, the supervised method is classification, and the unsupervised is clustering. Table 5 shows a summary of the different methods with their characteristics.

4.2.7 Deep Learning and Neural Networks-Based Classification

Deep learning (DL), which uses algorithms that mimic human thinking, is a subset of machine learning. The key feature is the use of neural networks that enable computers to make decisions. DL relies on multi-layer artificial neural networks that do not require structured data input, whereas machine learning primarily uses structured data sets. These approaches require large amounts of labeled data and have high power consumption, but achieve higher recognition accuracy than other methods. Figure 3 shows the main difference between machine learning and deep learning approaches from a classification point of view.

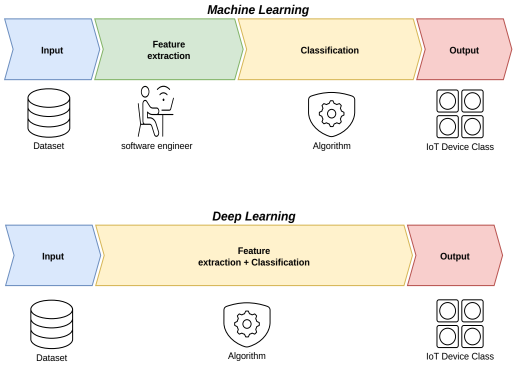

Figure 3: Machine- and Deep-Learning procedures

Several publications and literature are presented by researchers in this area of computer science (see [89, 126, 127, 128]). Researchers use the above techniques for network traffic classification and anomaly detection in data networks. In addition, publications on this topic have been presented over a decade ago; for example, Donald Michie et al. [95] presented an overview of the various classification methods as early as 2009.

4.2.8 Large Language Models

Large language models (LLMs) are actually deep learning methods for human language; they work on the basis of neural networks that have been trained with large amounts of data. Nowadays, especially since the introduction of ChatGPT, there are more and more new approaches in this field of research that deal with large language models. Researchers are trying to adapt generative language models for network traffic classification and intrusion detection systems. Researchers use such models, which are usually intended for language-specific tasks, but are increasingly being used to analyze IoT network traffic or to generate synthetic datasets. These approaches typically use generative transformers that process raw data at the token level, which is an important task in natural language processing. Traditional methods using traffic generators often require manual tuning and are usually very time-consuming to implement. Therefore, approaches based on artificial intelligence represent a new trend today.

In Figure 4 an overview is presented that shows the scopes in Artificial Intelligence with the range of action in the individual approaches, and Figure 5 shows the working procedure from a large language model (LLM).

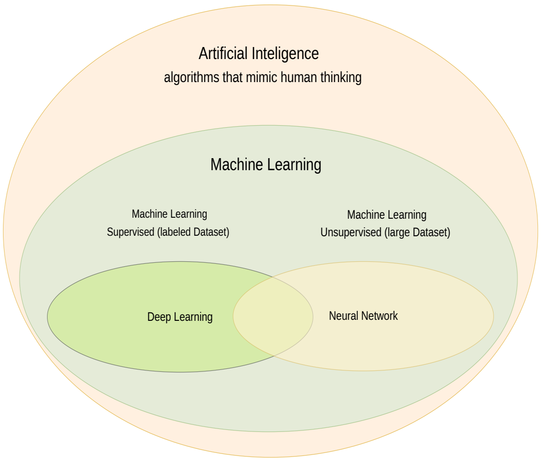

Figure 4: Scopes of ML-Approaches

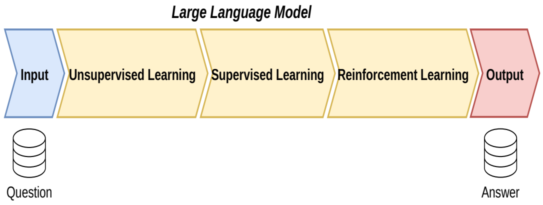

Figure 5: Large Language Models procedure

Table 5: Common Classification Methods

Method	Advantages	Challenges	Detection Time	areas of application
Port-Based	Easy to implement, lightweight, Real-Time	Random ports, NAT Low	First packet	Application classification
Payload -Based	Low false negative rate	High computing power, privacy issues, encrypted payloads	First payload	Application and IoT device classification, anomaly detection
Statistical	Flow or packet based, lightweight	Limits with large datasets, prior knowledge of the problem	Few packets or flow	Application and IoT device classification
Behavioral	Lightweight	Limits with large datasets	After exploiting heuristic information	IoT device classification, anomaly detection
Supervised Machine-learning	Good accuracy	Moderate computing power labeled dataset	Depends on the model	IoT device classification, anomaly detection
Unsupervised Machine-learning	Good accuracy	High computing power, unlabled dataset	Depends on the model	IoT device classification, anomaly detection
Deep-learning	Better accuracy	High computing power, labeled dataset	Depends on the model	IoT device classification, anomaly detection
Neural-Networks	Better accuracy	High computing power, unlabeled dataset, feature engineering	Depends on the model	IoT device classification, anomaly detection
AI based with Generative Transformers	promise better accuracy	High computing power	Depends on the model	IoT device traffic generation, IoT device classification, anomaly detection
AI based with Large Language Models	promise better accuracy	High computing power	Depends on the model	IoT device traffic generation, IoT device classification, anomaly detection
Quantum Computing	Very fast and high performance with large datasets	Cost, Hybrid approaches with AI	Fast	IoT device classification, anomaly detection

4.2.9 Quantum Computing

Quantum computing also plays an important role in pattern recognition, especially on large data sets. Combining machine learning approaches with quantum computing technologies QML (quantum machine learning) [97] has promising advantages in detecting malicious network traffic. In addition, quantum computing can help solve complex problems involving many variables that interact in complicated ways. Such solutions can also be used in network technology, with the promise of faster anomaly detection in large amounts of data, such as those generated by the Internet and IoT networks [138, 149].

4.3 Feature Extraction for IoT Device Classification

This section describes the main techniques for deciding which features to use to classify IoT devices. One of the main challenges in classification is determining which network traffic features are best suited for the process.

4.3.1 Background

Feature extraction is a procedure that defines a set of features that will be most efficient for our classification approach. Network traffic is the volume of data flowing through a viewpoint from source to destination. This traffic is carried in packets, which take care of the physical topology and routing information of the endpoints. In general, we can choose between packet-level or flow-level feature extraction. Packets can describe the traffic individually (packet by packet) or as a stream, also called a flow. The network traffic can be observed and analyzed in real time or captured as files that we trace at the viewpoint (gateway) in the network.

4.3.2 Packets

A network packet with a header and a payload contains all the information about the communication. Depending on the protocol we observe, the extraction can be done on several layers of the OSI reference model [53, 144].

From a network traffic perspective, a network packet is a construct of header and data (payload). Both may be encrypted at different layers of the protocol stack. The most commonly used header information for feature extraction for IoT device classification includes packet length, transport layer protocol used (TCP or UDP), time-to-live (TTL) information, TCP window size, and application layer protocol, such as HTTP, HTTPS, DHCP, or DNS (Domain Name System) [69].

With packet-level feature selection, we face some problems caused by the dynamics of network traffic; we get different fingerprints (classes) for an IoT device. In such cases, we need to explain the main differences for classifying IoT devices in general. Section 4 already showed some classification methods, but what does this mean for the classifier? Well, if we look at the taxonomy of traffic classification techniques, we find some important points to consider. For example, port-based techniques only need to look at the first packet in the transmission to determine what application it is, but with the dynamic use of ports today, this method is not as present. On the other hand, flow-based approaches typically require the entire data stream to determine the appropriate behavioral patterns.

4.3.3 Streams and Flows

Streams can be described as packets sent between endpoints (devices or hosts) with the same characteristic. Examples of relevant information in the context of this work are the source IP address and port number, destination IP address and port number, the layer-4 protocol used (UDP or TCP), and the application layer protocol used (e.g., HTTP, HTTPS, DNS, DHCP, and NTP). Streams of packets are so-called flows with particular characteristics, such as the same destination addresses and port numbers in the case of TCP or UDP network traffic.

Flows can be defined in several ways. According to the Internet Engineering Task Force (IETF) working group [72], flows are *"IP packets that pass through a point in the network during a time interval that belongs to the common characteristics"*. Relevant header information for feature extraction for IoT device classification includes the source IP address and port number, destination IP address and port number, and the transport layer protocol used. Definitions of flows in the Request for Comments (RFC-2063) [22] date back to 1997 and correspond to the definition of a stream. The IPFIX standard (RFC-7011) [5] defines network flows, generally based on the NetFlow v9 format described in (RFC-3954) [30].

NetFlow v9 (RFC-3954) [30]) was the basis for IP Flow Information Export (IPFIX (RFC-7011) [5]). Unlike NetFlow v9, IPFIX is an open standard supported by many network vendors in addition to Cisco. The formats are nearly identical except for a few additional fields added in IPFIX. Other definitions of flows exist for specific needs, such as OpenFlow, which is used in software-defined networking (SDN) and is managed by the Open Networking Foundation [56].

Relevant in the context of this work is that flows can exist without a TCP connection in the case of a UDP packet that is transmitted between a known source (host or device) to a certain destination without the need for an endpoint connection between them; the flow must be defined in a time window. Moreover, flows do not have any size restriction.

4.3.4 Communication Behavior of IoT Devices

It is not unlikely that IoT devices will establish multiple connections to servers and services with multiple destination IP addresses and ports during operation. An important feature is the number of connections the devices establish within a given startup sequence or time window and the protocols involved. Each IoT device shows an individual connection scheme; this could be a good starting point for fingerprinting, as shown, for example, in the work of Sivanathan et al. [133]. The authors use Sankey plots to visualize the behavior of the devices; Sankey plots are typically used to visualize statistical flows in financial or marketing analysis.

4.3.5 Feature Ranking Methods for Classification

The problem of determining the most appropriate features for classification has long been a concern of researchers. A number of papers have been published on this topic that analyze the best selection of features and then incorporate them into the classification. As examples, see the works of [38] and [76]. Other researchers have addressed the problem of feature ranking according to cost, which is a non-negligible factor [25].

4.4 Related Work in IoT Device Classification

IoT-Sentinel (see Miettinen et al. [96]) focuses on classifying device types at the device boot sequence by analyzing 23 header features using machine learning methods. The authors build a vector of 276-dimensional feature vector (12 packets × 23 features). This study represents one of the first methods to generate classification features from traffic headers. The main purpose of this work is to detect compromised devices and isolate them from the network to prevent damage. However, the accuracy of this behavior-based approach is limited.

Based on the IoT Sentinel dataset and the UNSW dataset [100], some other work has been presented; for example, the authors [26] present a solution for device identification. In their work, the authors extract 218 features from a single packet and archive an accuracy of 83.35% based on the IoT Sentinel dataset.

Bremler et al. [21] use TCP window size and DNS request as features and DHCP options for a second classifier. The authors use machine learning techniques and two classifiers with different feature sets and compare the F1 scores of each with the classification latency. This provides a solid overview of possible approaches. However, the approach is to distinguish IoT from non-IoT devices, not the device type.

Other researchers use behavioral classification to fingerprint IoT devices. Bezawada et al. [16] present a dynamic behavioral model based on the command-response activities of IoT devices. This approach does not use features such as IP addresses and port numbers for classification. However, it is very difficult to automatically observe all the interactions of the devices, so the authors have to manually interact with the devices under test, which is a time-consuming task.

Deng et al. [35] present IoTSpot, a framework for identifying IoT devices using anonymous network data with a short training phase of 40 minutes. In this approach, the authors used unsupervised machine learning methods and archived high accuracy; the limitation is that classifying devices with the same hardware and firmware significantly reduces the accuracy. Another drawback of this method is the fact that the TCP window size cannot be used for devices that only use UDP.

Sivanathan et al. [134] use an active TCP port scan technique to classify devices in their work. In a subsequent work, the authors use the number of DNS queries and cipher suite information, among other things, to build a multi-stage classifier and report a high accuracy of over 99%. In this work, the authors show approaches to visualize the behavior of IoT devices using Sankey diagrams [133]. The challenges of this approach are certainly the dynamics in the communication behavior of IoT devices and the fact that the analysis of DNS requests in network data can lead to a violation of privacy.

Khandait et al. [80] present in their work *IoTHunter* a framework that uses the occurrence of keywords belonging to flows with deep packet inspection methods for the classifier. Disadvantages of the method used are the fact that most IoT devices today use encryption, so for some devices it may result that they do not have unencrypted keywords or have none at all.

Entropy-based features are chosen to perform traffic classification. For example, if a packet carries plain text, then the entropy of the payload is lowest. The entropy will increase proportionately if the packet carries audio/XML/JSON-encoded, compressed, or encrypted data. Such methods were mainly used to detect anomalies in network traffic and are not so common for device-type fingerprinting (see [102, 114]).

Several works on flow-based classification have been published in the last decade. As mentioned before, a flow is a set of packets passing through an observation point in the network, and all packets belonging to a particular flow have a set of common properties. Some of these papers are presented in the works [41, 64, 66, 70, 114].

Some authors analyze the radio frequency transmitted over WiFi signals to fingerprint devices [151, 152] or in the 5G and 6G radio wave bands, as in the work of Takasaki et al. [143]. Other authors use the modulation information of the transmitted signal [83]. This work provides an interesting insight into the related challenges in IoT device classification, but it is not directly comparable to the other studies in this work because they analyze not only packet or flow characteristics, but also the frequency spectrum and waveforms transmitted by devices.

Other works implement statistical approaches using the Euclidean rules or cosine similarity, among others, for comparison with and without machine learning, as shown, for example, in the papers [27, 28, 33, 43]. However, most of this work deals with the classification of network traffic for application detection.

Duan et al. [42] in their work present ByteIoT, an identification approach based on the frequency distribution of packet length with the k-nearest neighbors algorithm, using a distance metric to build the classifier. This work shows that such methods are well suited for classification approaches in IoT networks. For new devices joining the network at this point, a consideration of hybrid approaches with generative transformers would be conceivable.

However, new approaches have been recognized to improve the security of IoT networks. Rieger et al. [121] present a framework that can detect infiltration attacks by defining normal and suspicious behavior situations in an everyday household with context-based detection methods. Göbel et al. [64] present in their work ”Find My IoT Device” a method based on the Approximate Matching Algorithm to identify IoT traffic flows.

Today, researchers are trying solutions using generative pre-trained transformers (GPT), a type of large language model (LLM), to build frameworks for improving network security or to generate large synthetic datasets useful for training machine learning algorithms for better accuracy and performance. However, the use of artificial intelligence (AI) promises improvements in this area of research that have yet to be proven.

Meng et al. [94], the authors present an approach to use a generative pre-trained language model (NetGPT) to generate an educational task. Kholgh et al. [81] present an LLM approach based on ChatGPT-3 Davinci and Babbage for fine-tuning the model that generates synthetic network flows that can be used to train machine learning and especially deep learning models.

However, the number of papers and surveys published in this research area demonstrates its high relevance. The works [40, 78, 101] show that classification approaches have been a hot topic for more than a decade.

4.5 Summary

Several works in this research area have analyzed the ranking of features that are best suited for classification. The results of these studies show that packet size is the most promising factor for classifying IoT devices. The TCP window size, which depends on the memory and processing speed of the device, is widely used to distinguish IoT from non-IoT devices, but cannot be used for devices that only work with UDP or other protocols. For example, the Dynamic Host Control Protocol (DHCP) service provides IP addresses, gateway information, and other relevant parameters such as device names that can be used for further classification. When a new IoT device joins the network, most will send a DHCP Discovery Broadcast message at startup to obtain an IP address and other communication parameters. For example, the device name could be used to initialize a further search on the Internet and extract more information about the devices from the content of websites; this approach was taken by Feng al al. [51]. They present the ARE framework to identify the IoT devices from the banner information. This method was re implemented in the work of Javed et al. [77] with different results. If the device does not have DHCP options enabled by default, or if an administrator (Network-Policies) disables them, and if the device does not have a client name set by the manufacturer, this method fails.

Most of the work shown takes the approach of monitoring traffic at the TCP/IP layer. However, IoT devices use protocols such as CoAP and MQTT, so traffic diagnosis has open challenges before the packets reach the gateway. In addition, most of the work is related to WiFi devices and does not consider other transmission techniques such as Bluetooth Low Energy, Z-Wave, LoRaWAN, and ZigBee; this confirms the assumption that this area of research is very diverse due to the many variations in the transmission techniques used. The authors of the article [13] present a framework that identifies devices working with ZigBee and Z-Wave protocols. Other authors try to reduce the number of features with individual methods. For example, Santos et al. [41] used four statistical features combined with the user agent text information extracted from the packet payload and the random forest algorithm to classify the devices. Some works show that the packet length and window size are good features for discriminating the device types with tractable accuracy, as shown in the work of Pinheiro et al. [117].

Many researchers are trying to adapt deep learning and neural networks to solve the problems of IoT device traffic identification. However, it remains a challenge because both methods face privacy (payload inspection) and computational issues. This problem exists even if we use cloud solutions where computing power is not the big problem because most of the measurement data is needed close to the IoT device, so latency and data transmission afford back to the local networks is still a challenge [89, 90, 127].

In recent years, researchers have worked on new approaches to traffic classification and anomaly detection, such as approximate matching [63, 64] and context-based detection [121]. Recently, other solutions have emerged that use generative AI and large language models (LLMs) to overcome the challenges, as shown in the work of [81, 94], among others.

5 Conclusion and Future Works

Many papers have been published on the classification of network traffic and IoT devices. All approaches face specific challenges when using different classification methods. For example, the IP addresses of servers and services are often not constant; the nature of cloud solutions is that they use a few servers for the services they provide. Another challenge is that some types of IoT devices use the same cloud provider for their settings and communication patterns, such as Google devices. Such approaches therefore run the risk of low detection rates. Using artificial intelligence (AI) on constrained devices at the edge of the network, with federated or split learning methods, could help overcome some of the latency and security issues of the native cloud solution, but introduces new challenges in terms of the limited processing power of constrained IoT devices.

In addition, because network traffic is highly dynamic, there are always small variations in the behavior of IoT devices. Testing the same device multiple times will result in multiple different fingerprints, creating a multi-class problem. In the case of two or more corresponding classes, a similarity coefficient must be defined to decide which class the unseen data most likely belongs to. These approaches could be implemented using the euclidean rules or cosine similarity, among others, as shown in the works of Cunha et al. [33] and Duan et al. [42].

The Challenges are not solved yet at all, so researchers keep working on this research field, from our point of view, with the following key aspects:

- Standardization - Creating standard protocols and benchmarks for IoT traffic classification.
- Cross-layer Approaches - Integrating information from multiple network stack layers for more accurate classification.
- Edge Computing - Processing data closer to the source to reduce latency and bandwidth usage.
- Federated Learning - Leveraging distributed models to enhance privacy and scalability.

Statistical methods have advantages, such as providing a structured approach to collecting, organizing, analyzing, and interpreting data. A downside could be the potential for erroneous conclusions with inappropriate statistical methods. On the other hand, new approaches are often prioritized in research, creating a ”publish or perish” culture that can bias studies in favor of new methods over existing ones. This can hinder meaningful comparisons between different approaches, making it difficult for end users to determine the most appropriate method for their research questions. Machine learning methods can provide better results in terms of accuracy and F1 score, but require large datasets and, in the case of supervised learning, a significant amount of technical work to label and balance the datasets. Most existing work uses machine learning techniques to solve network behavior classification problems. Still, most of them use data-driven models on features of network packets and/or flows without looking at gains versus costs [41]. Nevertheless, there are still many unsolved classification challenges in this area.

New and re-established ways in traffic classification, as shown by [63, 64, 121], but also techniques without machine learning, as demonstrated in the work [33], offer possible new research directions that can be further explored in the future. Furthermore, moving classification processes closer to the edge of networks, commonly referred to as ”edge computing”, is an important trend in the field of IoT traffic classification. This approach involves processing data near the source of data generation (i.e., the IoT devices themselves), rather than relying on centralized cloud servers. Such solutions are often used in medical research, where latency is critical. Using such methods for traffic classification is another possible area of research.

In recent years, but increasingly since the advent of ChatGPT, new methods have been researched that can be classified in the class of AI-based learning. Today, there are many promising new methods that use large language models to analyze IoT network traffic to prevent attacks or to generate synthetic datasets for classification. However, it remains to be proven that these approaches can meet the challenges. In particular, the results of such generative models need to be verified, as the model may already be compromised when it is used to detect anomalies and cyber-attacks.

The research areas that we will consider for our future work are:

- A deeper look into similarity approaches with the possibilities of machine learning to overcome the challenges that exist in real-time network traffic classification. One of these points is to minimize the training phase for unknown IoT devices.
- The use of generative AI such as ChatGPT to generate synthetic IoT device traffic datasets, as shown in the work of Meng et al. [94] and Kholgh et al. [81], to train machine learning models. The idea is to train a LLM (Large Language Model) with the network traffic behavior of several IoT devices, which can then be used for classification to improve accuracy and false positive rate. Another research direction we can imagine is to make the behavior of IoT devices understandable to large language models, for example, to represent network traffic features in tokens so that they can be used to describe the behavior of IoT devices in a large language model.

In addition, from our point of view, edge computing can lead to the following research directions for future work:

- Using artificial intelligence (AI) on constrained devices at the edge of the network, with federated or split learning methods. Do such solutions need to be implemented on edge gateways because the processing power required would be too much for constrained IoT devices?
- Using statistical approaches with reduced complexity methods that can be processed on constrained IoT devices, but will they overcome the immense increase in IoT device implementations in the future?

While traditional methods provide a solid foundation, machine learning techniques offer promising solutions to address the unique challenges of IoT traffic classification. Continued innovation and collaboration are essential to developing robust, scalable, and capable solutions to support the growing IoT ecosystem. For example, new approaches have emerged in recent years using quantum computing. These solutions combine classical AI-based approaches such as machine learning, generative transformers, and LLMs with the power of quantum computing to overcome the challenges of analyzing very large data sets and detecting malicious network traffic in real time. Quantum computing is not the focus of our work, but to give an idea of the possibilities of such approaches, we include the work of Kalinin et al. [79] and Spadari et al. [138].

IoT traffic classification is an important area of research with significant implications for network management, security and optimization, and remains challenging due to the technological and functional diversity of IoT devices.

References

[1] R. Abbi and B. Ray, "Definitions of Managed Objects for Very High Speed Digital Subscriber Lines (VDSL)," RFC 3728, Feb. 2004. [Online]. Available: https://www.rfc-editor.org/info/rfc3728

[2] K. Aberer, M. Hauswirth, and A. Salehi, "A middleware for fast and flexible sensor network deployment," in *Proceedings of the 32nd International Conference on Very Large Data Bases*, ser. VLDB '06. VLDB Endowment, 2006, p. 1199–1202.

[3] K. Aberer, M. Hauswirth, and A. Salehi, "Infrastructure for data processing in large-scale interconnected sensor networks," in *2007 International Conference on Mobile Data Management*. Conference and Custom Publishing, 2007, pp. 198–205. [Online]. Available: http://ieeexplore.ieee.org/document/4417143/

[4] K. Aberer, M. Hauswirth, and A. Salehi, "The global sensor networks," 07 2010.

[5] P. Aitken, B. Claise, and B. Trammell, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information," RFC 7011, Sep. 2013, accessed on: Dec 17, 2024:. [Online]. Available: https://www.rfc-editor.org/info/rfc7011

[6] I. Akyildiz, S. WY, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks*, vol. 38, pp. 393–422, 03 2002.

[7] J.-M. P. Alexandre Dolgui, "Radio-frequency Identification (RFID): Technology and Applications," in *Supply Chain Engineering: Useful Methods and Techniques*. London: Springer London, 2010, pp. 163–194. [Online]. Available: https://doi.org/10.1007/978-1-84996-017-5_5

[8] C. S. Alliance. (2024) Connectivity standards alliance. Accessed on: Dec 9, 2024:. [Online]. Available: https://www.csa-iot.org/

[9] W.-F. Alliance. (2025) Wifi. Accessed on: Mar 4, 2025:. [Online]. Available: https://www.wi-fi.org/discover-wi-fi/internet-things/

[10] Z. Alliance. (2024) ZigBee. Accessed on: Dec 10, 2024. [Online]. Available: https://zigbeealliance.org/

[11] Z. Alliance. (2024) Z-wave. Accessed on: Dec 9, 2024:. [Online]. Available: https://z-wavealliance.org/

[12] 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)*, 2019, pp. 204–209.

[13] L. Babun, H. Aksu, L. Ryan, K. Akkaya, E. S. Bentley, and A. S. Uluagac, "Z-IoT: Passive device-class fingerprinting of ZigBee and z-wave IoT devices," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/9149285/

[14] J. Bellendorf, "Cloud topology and orchestration using TOSCA: A systematic literature review," in *Service-Oriented and Cloud Computing*, K. Kritikos, P. Plebani, and F. De Paoli, Eds. Springer, 2018, vol. 11116, pp. 207–215. [Online]. Available: https://link.springer.com/10.1007/978-3-319-99819-0_16

[15] T. Berners-Lee, R. T. Fielding, and L. M. Masinter, "Uniform Resource Identifier (URI): Generic Syntax," RFC 3986, Jan. 2005. [Online]. Available: https://www.rfc-editor.org/info/rfc3986

[16] 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*. ACM, 2018, pp. 41–50. [Online]. Available: https://dl.acm.org/doi/10.1145/3266444.3266452

[17] Bitcom. (2024) Iot-plattformen – aktuelle trends und herausforderungen. Accessed on: Dec 10, 2024:. [Online]. Available: https://www.bitkom.org/sites/default/files/file/import/180424-LF-IoT-Plattformen-online.pdf

[18] M. Boucadair, R. Penno, and D. Wing, "Universal Plug and Play (UPnP) Internet Gateway Device - Port Control Protocol Interworking Function (IGD-PCP IWF)," RFC 6970, Jul. 2013, accessed on: Dec 17, 2024:. [Online]. Available: https://www.rfc-editor.org/info/rfc6970

[19] R. T. Braden, "Requirements for Internet Hosts - Application and Support," RFC 1123, Oct. 1989. [Online]. Available: https://www.rfc-editor.org/info/rfc1123

[20] R. T. Braden, "Requirements for Internet Hosts - Communication Layers," RFC 1122, Oct. 1989. [Online]. Available: https://www.rfc-editor.org/info/rfc1122

[21] A. Bremler-Barr, H. Levy, and Z. Yakhini, "IoT or NoT: Identifying IoT devices in a ShortTime scale," 2019. [Online]. Available: http://arxiv.org/abs/1910.05647

[22] N. Brownlee, C. G. Mills, and D. G. R. Ruth, "Traffic Flow Measurement: Architecture," RFC 2063, Jan. 1997, accessed on: Dec 10, 2024:. [Online]. Available: https://www.rfc-editor.org/info/rfc2063

[23] A. Bröring, S. Below, and T. Foerster, "Declarative sensor interface descriptors for the sensor web," 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:7893003

[24] V. Caballero, S. Valbuena, D. Vernet, and A. Zaballos, "Ontology-defined middleware for internet of things architectures," *Sensors (Basel, Switzerland)*, vol. 19, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:76663689

[25] B. Chakraborty, D. M. Divakaran, I. Nevat, G. W. Peters, and M. Gurusamy, "Cost-aware feature selection for iot device classification," *IEEE Internet of Things Journal*, vol. 8, no. 14, pp. 11 052–11 064, 2021.

[26] R. R. Chowdhury, S. Aneja, N. Aneja, and E. Abas, "Network traffic analysis based IoT device identification," in *Proceedings of the 2020 4th International Conference on Big Data and Internet of Things*. ACM, 2020, pp. 79–89. [Online]. Available: https://dl.acm.org/doi/10.1145/3421537.3421545

[27] J. Y. Chung, B. Park, Y. J. Won, J. Strassner, and J. W. Hong, "Traffic classification based on flow similarity," in *IP Operations and Management*, G. Nunzi, C. Scoglio, and X. Li, Eds. Springer, 2009, vol. 5843, pp. 65–77, series Title: Lecture Notes in Computer Science. [Online]. Available: http://link.springer.com/10.1007/978-3-642-04968-2_6

[28] J. Y. Chung, B. Park, Y. J. Won, J. Strassner, and J. W. Hong, "An effective similarity metric for application traffic classification," in *2010 IEEE Network Operations and Management Symposium - NOMS 2010*, 2010, pp. 286–292.

[29] A. Cimmino and R. García-Castro, "Wothive: Enabling syntactic and semantic discovery in the web of things," *Open Journal of Internet Of Things (OJIOT)*, vol. 8, no. 1, pp. 54–65, 2022. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:101:1-2022090515503251402854

[30] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954, Oct. 2004, accessed on: Dec 17, 2024:. [Online]. Available: https://www.rfc-editor.org/info/rfc3954

[31] O. G. Consortium. (2024) Open geospatial consortium. Accessed on: Dec 9, 2024:. [Online]. Available: https://www.ogc.org/

[32] M. Corici, H. Coskun, A. Elmangoush, A. Kurniawan, T. Mao, T. Magedanz, and S. Wahle, "OpenMTC: Prototyping machine type communication in carrier grade operator networks," in *2012 IEEE Globecom Workshops*. IEEE, 2012, pp. 1735–1740. [Online]. Available: https://ieeexplore.ieee.org/document/6477847/

[33] V. C. Cunha, A. A. Z. Zavala, P. R. M. Inácio, D. Magoni, and M. M. Freire, "Classification of encrypted internet traffic using kullback-leibler divergence and euclidean distance," in *Advanced Information Networking and Applications*, L. Barolli, F. Amato, F. Moscato, T. Enokido, and M. Takizawa, Eds. Springer, 2020, vol. 1151, pp. 883–897. [Online]. Available: http://link.springer.com/10.1007/978-3-030-44041-1_77

[34] L. De Nardis, A. Mohammadpour, G. Caso, U. Ali, and M.-G. Di Benedetto, "Internet of things platforms for academic research and development: A critical review," *Applied Sciences*, vol. 12, p. 2172, 02 2022.

[35] L. Deng, Y. Feng, D. Chen, and N. Rishe, "IoTSpot: Identifying the IoT devices using their anonymous network traffic data," in *MILCOM 2019 - 2019 IEEE Military Communications Conference (MILCOM)*. IEEE, 2019, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9020977/

[36] Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Kosice, Slovak Republic, J. Mocnej, A. Pekar, School of Engineering and Computer Science, Victoria University of Wellington, New Zealand, W. K.G. Seah, School of Engineering and Computer Science, Victoria University of Wellington, New Zealand, E. Kajati, Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Kosice, Slovak Republic, I. Zolotova, and Department of Cybernetics and Artificial Intelligence, Faculty of Electrical Engineering and Informatics, Technical University of Kosice, Slovak Republic, "INTERNET OF THINGS UNIFIED PROTOCOL STACK," vol. 19, no. 2, pp. 24–32, 2019. [Online]. Available: http://www.aei.tuke.sk/papers/2019/2/04_Mocnej.pdf

[37] Deptartment of Information Technology, Assiut University, Egypt, A. H. Ahmed, N. M. Omar, and H. M. Ibrahim, "Secured service discovery technique in IoT," pp. 40–46, 2019. [Online]. Available: http://www.jocm.us/index.php?m=content&c=index&a=show&catid=216&id=1307

[38] B. Desai, D. M. Divakaran, I. Nevat, G. Peters, and M. Gurusamy, "A feature-ranking framework for iot device classification," 01 2019.

[39] DeviceHive. (2024) Devicehive. Accessed on: Dec 17, 2024:. [Online]. Available: https://www.devicehive.com/

[40] A. Dhakad, S. Singh, Mohana, M. Moharir, and A. K. A. R, "Real time network traffic analysis using artificial intelligence, machine learning and deep learning: A review of methods, tools and applications," in *2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)*. IEEE, 2023, pp. 372–378. [Online]. Available: https://ieeexplore.ieee.org/document/10331855/

[41] R. Du, J. Wang, and S. Li, "A lightweight flow feature-based IoT device identification scheme," vol. 2022, pp. 1–10, 2022. [Online]. Available: https://www.hindawi.com/journals/scn/2022/8486080/

[42] C. Duan, H. Gao, G. Song, J. Yang, and Z. Wang, "Byteiot: A practical iot device identification system based on packet length distribution," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1717–1728, 2022.

[43] G. Dupont, C. Leite, D. R. Dos Santos, E. Costante, J. Den Hartog, and S. Etalle, "Similarity-based clustering for IoT device classification," in *2021 IEEE International Conference on Omni-Layer Intelligent Systems*

(COINS). IEEE, 2021, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/9524201/

[44] Edge-Stack. (2024) Edge-stack. Accessed on: Dec 10, 2024:. [Online]. Available: https://www.edgestack.io/

[45] F. J. Ekaputra, M. Sabou, E. Serral, E. Kiesling, and S. Biffl, "Ontology-based data integration in multi-disciplinary engineering environments: A review," *Open Journal of Information Systems (OJIS)*, vol. 4, no. 1, pp. 1–26, 2017. [Online]. Available: http://nbn-resolving.de/urn:nbn:de:101:1-201711266863

[46] U. Elangovan, *Product lifecycle management (plm): a digital journey using industrial internet of things (iiot)*. CRC Press, 2020.

[47] P. Emami-Naeini, H. Dixon, Y. Agarwal, and L. F. Cranor, "Exploring how privacy and security factor into iot device purchase behavior," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1–12. [Online]. Available: https://doi.org/10.1145/3290605.3300764

[48] V. Fajardo, J. Arkko, J. A. Loughney, and G. Zorn, "Diameter Base Protocol," RFC 6733, Oct. 2012. [Online]. Available: https://www.rfc-editor.org/info/rfc6733

[49] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing-based iot," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2146–2153, 2018.

[50] S. Farrell, "Low-Power Wide Area Network (LPWAN) Overview," RFC 8376, May 2018. [Online]. Available: https://www.rfc-editor.org/info/rfc8376

[51] X. Feng, Q. Li, H. Wang, and L. Sun, "Acquisitional rule-based engine for discovering internet-of-thing devices," 2018.

[52] fiware. (2025) Openmtc-agent. Accessed on: Feb 27, 2025:. [Online]. Available: https://fiware-openmtc.readthedocs.io/en/latest/

[53] I. O. for Standardization. (2024) International organization for standardization. Accessed on: Dec 9, 2024:. [Online]. Available: https://www.iso.org/home.html

[54] N. Forum. (2024) NFC (near field communication). Accessed on: Dec 10, 2024:. [Online]. Available: https://nfc-forum.org/

[55] F. Foundation. (2024) Fiware. Accessed on: Dec 17, 2024:. [Online]. Available: https://www.fiware.org/

[56] O. N. Foundation. (2024) Open networking foundatio. Accessed on: Dec 17, 2024:. [Online]. Available: https://opennetworking.org

[57] O. Foundation. (2024) OpenHab. Accessed on: Dec 10, 2024:. [Online]. Available: https://www.openhab.org/

[58] O. Foundation. (2024) Node-RED. Accessed on: Dec 10, 2024:. [Online]. Available: https://nodered.org/

[59] A. C. Franco da Silva, U. Breitenbücher, P. Hirmer, K. Képes, O. Kopp, F. Leymann, B. Mitschang, and R. Steinke, "Internet of things out of the box: Using TOSCA for automating the deployment of IoT environments," in *Internet of Things Out of the Box: Using TOSCA for Automating the Deployment of IoT Environments*, 2017, pp. 358–367.

[60] GitHub. (2024) Zetta api first iot plattform. Accessed on: Dec 17, 2024:. [Online]. Available: https://github.com/zettajs/zetta

[61] O. M. Group. (2024) Data distribution service. Accessed on: Dec 17, 2024:. [Online]. Available: https://www.omg.org/omg-dds-portal/

[62] O. M. Group. (2024) Object management group. Accessed on: Dec 17, 2024:. [Online]. Available: https://www.omg.org/

[63] T. Göbel, F. Uhlig, and H. Baier, "EVALUATION OF NETWORK TRAFFIC ANALYSIS USING APPROXIMATE MATCHING ALGORITHMS," in *Advances in Digital Forensics XVII*, G. Peterson and S. Shenoi, Eds. Springer, 2021, vol. 612, pp. 89–108. [Online]. Available: https://link.springer.com/10.1007/978-3-030-88381-2_5

[64] T. Göbel, F. Uhlig, and H. Baier, "Find my IoT device – an efficient and effective approximate matching algorithm to identify IoT traffic flows," in *Digital Forensics and Cyber Crime*, P. Gladyshev, S. Goel, J. James, G. Markowsky, and D. Johnson, Eds. Springer, 2022, pp. 72–92.

[65] S. Hadzovic, S. Mrdovic, and M. Radonjic, "Identification of iot actors," *Sensors*, vol. 21, no. 6, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/6/2093

[66] S. A. Hamad, W. E. Zhang, Q. Z. Sheng, and S. Nepal, "IoT device identification via network-flow based fingerprinting and

learning," in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE, 2019, pp. 103–111. [Online]. Available: https://ieeexplore.ieee.org/document/8887356/

[67] M. Hauswirth, K. Aberer, and A. Salehi, " Invited Talk: Zero-Programming Sensor Network Deployment ," in *International Symposium on Applications and the Internet Workshops (SAINTW'06)*. Los Alamitos, CA, USA: IEEE Computer Society, Jan. 2007, p. 1. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/SAINT-W.2007.57

[68] P. Hirmer, M. Wieland, U. Breitenbücher, and B. Mitschang, "Dynamic ontology-based sensor binding," in *Dynamic Ontology-Based Sensor Binding*, 2016, pp. 323–337.

[69] P. E. Hoffman and K. Fujiwara, "DNS Terminology," RFC 9499, Mar. 2024. [Online]. Available: https://www.rfc-editor.org/info/rfc9499

[70] F. Hu, S. Zhang, X. Lin, L. Wu, N. Liao, and Y. Song, "Network traffic classification model based on attention mechanism and spatiotemporal features," vol. 2023, no. 1, p. 6, 2023. [Online]. Available: https://jis-eurasipjournals.springeropen.com/articles/10.1186/s13635-023-00141-4

[71] IANA. (2025) Service name and transport protocol port number registry. Accessed on: Feb 27, 2025:. [Online]. Available: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml

[72] I. E. T. F. (IETF). (2024) Internet engineering task force. Accessed on: Dec 9, 2024:. [Online]. Available: https://www.ietf.org

[73] ioBroker GmbH. (2024) iobroker. Accessed on: Dec 10, 2024:. [Online]. Available: https://www.iobroker.net/

[74] I. Ishaq, J. Hoebeke, J. Rossey, E. De Poorter, I. Moerman, and P. Demeester, "Facilitating sensor deployment, discovery and resource access using embedded web services," in *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, 2012, pp. 717–724. [Online]. Available: http://ieeexplore.ieee.org/document/6296943/

[75] ISO. (2025) Suggested upper merged ontology. Accessed on: Feb 26, 2025:. [Online]. Available: https://www.iso.org/standard/53798.html

[76] S. Izadi, M. Ahmadi, and R. Nikbazm, "Analysis of feature selection methods for network traffic classification," in *Proceedings of the 8th International Conference on Advanced Intelligent Systems and Informatics 2022*, A. E. Hassanien, V. Snášel, M. Tang, T.-W. Sung, and K.-C. Chang, Eds. Springer, pp. 65–77.

[77] T. Javed, M. Haseeb, M. Abdullah, and M. Javed, "Using application layer banner data to automatically identify IoT devices," vol. 50, no. 3, pp. 23–29, 2020. [Online]. Available: https://dl.acm.org/doi/10.1145/3411740.3411744

[78] H. Jmila, G. Blanc, M. R. Shahid, and M. Lazrag, "A survey of smart home IoT device classification using machine learning-based network traffic analysis," vol. 10, 2022.

[79] M. Kalinin and V. Krundyshev, "Security intrusion detection using quantum machine learning techniques," *Journal of Computer Virology and Hacking Techniques*, vol. 19, no. 1, pp. 125–136, jun 2022. [Online]. Available: https://link.springer.com/10.1007/s11416-022-00435-0

[80] P. Khandait, N. Hubballi, and B. Mazumdar, "Iothunter: Iot network traffic classification using device specific keywords," *IET Networks*, vol. 10, no. 2, pp. 59–75, 2021. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/ntw2.12007

[81] D. K. Kholgh and P. Kostakos, "Pac-gpt: A novel approach to generating synthetic network traffic with gpt-3," *IEEE Access*, vol. 11, pp. 114 936–114 951, 2023.

[82] A. Khudoyberdiev, W. Jin, and D. Kim, "A novel approach towards resource auto-registration and discovery of embedded systems based on DNS," vol. 8, no. 4, p. 442, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/4/442

[83] M. Kose, S. Tascioglu, and Z. Telatar, "RF fingerprinting of IoT devices based on transient energy spectrum," vol. 7, pp. 18 715–18 726, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8631016/

[84] G. Labs. (2025) Grafana. Accessed on: Mar 4, 2025:. [Online]. Available: https://grafana.com/

[85] K. Lee, *IEEE 1451: A standard in support of smart transducer networking*, ser. IMTC-00. IEEE, 2000, vol. 2.

[86] LF-EDGE. (2024) Lf-edge. Accessed on: Dec 10, 2024:. [Online]. Available: https://lfedge.org/

[87] F. Li, M. Vogler, M. Claessens, and S. Dustdar, "Towards automated IoT application deployment by a cloud-based approach," in *2013 IEEE 6th International Conference on Service-Oriented Computing and Applications*. IEEE, 2013, pp. 61–68. [Online]. Available: http://ieeexplore.ieee.org/document/6717286/

[88] Y. Liu, J. Wang, J. Li, S. Niu, and H. Song, "Machine learning for the detection and identification of internet of things devices: A survey," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 298–320, 2022.

[89] Y. Liu, J. Wang, J. Li, H. Song, T. Yang, S. Niu, and Z. Ming, "Zero-bias deep learning for accurate identification of internet-of-things (IoT) devices," vol. 8, no. 4, pp. 2627–2634, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9173537/

[90] E. Lo and J. Kohl, "Internet of things (IoT) discovery using deep neural networks," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2020, pp. 795–803. [Online]. Available: https://ieeexplore.ieee.org/document/9093371/

[91] E. Martínez, D. Toma, S. Jirka, and J. Del Río, "Middleware for plug and play integration of heterogeneous sensor resources into the sensor web," vol. 17, no. 12, p. 2923, 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/12/2923

[92] J. Mauere. (2024) Iot-plattformen – aktuelle trends und herausforderungen. Accessed on: Dec 10, 2024:. [Online]. Available: https://www.tuvsud.com/de-de/-/media/de/cyber-security/pdf/allgemein/marketing/studie_internet-ofthings_2019_2020.pdf

[93] S. Mayer, N. Inhelder, R. Verborgh, R. Van De Walle, and F. Mattern, "Configuration of smart environments made simple: Combining visual modeling with semantic metadata and reasoning," in *2014 International Conference on the Internet of Things (IOT)*. IEEE, 2014, pp. 61–66. [Online]. Available: https://ieeexplore.ieee.org/document/7030116

[94] X. Meng, C. Lin, Y. Wang, and Y. Zhang, "Netgpt: Generative pretrained transformer for network traffic," 2023. [Online]. Available: https://arxiv.org/abs/2304.09513

[95] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, "Machine learning, neural and statistical classification," in *Machine Learning, Neural and Statistical Classification*, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:15773445

[96] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "IoT SENTINEL: Automated device-type identification for security enforcement in IoT," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2017, pp. 2177–2184. [Online]. Available: http://ieeexplore.ieee.org/document/7980167/

[97] N. Mishra, M. Kapil, H. Rakesh, A. Anand, N. Mishra, A. Warke, S. Sarkar, S. Dutta, S. Gupta, A. Dash, R. Gharat, Y. Chatterjee, S. Roy, S. Raj, V. Jain, S. Bagaria, S. Chaudhary, V. Singh, R. Maji, and P. Panigrahi, *Quantum Machine Learning: A Review and Current Status*, 01 2021, pp. 101–145.

[98] M. Mohammed, A. Ghazi, A. Awad, S. Hassan, H. Jawad, and K. Jasim, "A comparison of 4g lte and 5g network cybersecurity performance," 04 2024.

[99] A. H. Mohd Aman, E. Yadegaridehkordi, Z. S. Attarbashi, R. Hassan, and Y.-J. Park, "A survey on trend and classification of internet of things reviews," *IEEE Access*, vol. 8, pp. 111 763–111 782, 2020.

[100] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, 2015, pp. 1–6.

[101] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," vol. 10, no. 4, pp. 56–76, 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4738466/

[102] H. Nguyen-An, T. Silverston, T. Yamazaki, and T. Miyoshi, "Entropy-based IoT devices identification," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2020, pp. 73–78. [Online]. Available: https://ieeexplore.ieee.org/document/9236963/

[103] H. Nielsen, J. Mogul, L. M. Masinter, R. T. Fielding, J. Gettys, P. J. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1,"

RFC 2616, Jun. 1999. [Online]. Available: https://www.rfc-editor.org/info/rfc2616

[104] M. Nottingham, "Building Protocols with HTTP," RFC 9205, Jun. 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc9205

[105] C. D. Nugent, D. D. Finlay, R. J. Davies, H. Y. Wang, H. Zheng, J. Hallberg, K. Synnes, and M. D. Mulvenna, "homeML – an open standard for the exchange of data within smart environments," in *Pervasive Computing for Quality of Life Enhancement*, T. Okadome, T. Yamazaki, and M. Makhtari, Eds. Springer, 2007, vol. 4541, pp. 121–129. [Online]. Available: http://link.springer.com/10.1007/978-3-540-73035-4_13

[106] O. C. F. (OCF). (2024) Iotivity. Accessed on: Dec 10, 2024:. [Online]. Available: http://iotivity.org/

[107] G. of Mobile Standards 3GGP. (2025) Nb-iot. Accessed on: Mar 4, 2025:. [Online]. Available: https://www.3gpp.org/news-events/3gpp-news/nb-iot-complete/

[108] OGC-Org. (2025) Geography markup language. Accessed on: Feb 27, 2025:. [Online]. Available: https://www.ogc.org/publications/standard/gml/

[109] OPC-Org. (2025) Open platform communications. Accessed on: Feb 27, 2025:. [Online]. Available: https://opcfoundation.org/

[110] O. Open. (2024) Advanced message queuing protocol. Accessed on: Dec 10, 2024:. [Online]. Available: https://www.amqp.org/

[111] M. Org. (2024) Message queuing telemetry transport. Accessed on: Dec 10, 2024:. [Online]. Available: https://mqtt.org/

[112] O. Org. (2024) Suggested upper merged ontology. Accessed on: Dec 10, 2024:. [Online]. Available: https://www.ogc.org/publications/standard/sensorml/

[113] A. Palavalli, D. Karri, and S. Pasupuleti, "Semantic internet of things," in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, 2016, pp. 91–95.

[114] A.-S. K. Pathan, M. M. Monowar, and Z. M. Fadlullah, Eds., *Characterizing Flow-Level Traffic Behavior with Entropy Spaces for Anomaly Detection*, 0th ed. CRC Press, 2013. [Online]. Available: https://www.taylorfrancis.com/books/9781466507647/chapters/10.1201/b14574-18

[115] A. Pease, "Sumo: a sharable knowledge resource with linguistic inter-operability," in *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003*, Oct 2003, pp. 827–.

[116] A. Pease. (2025) Sensor model language (sensorml). Accessed on: Feb 27, 2025:. [Online]. Available: https://www.ontologyportal.org/index.html

[117] A. J. Pinheiro, J. d. M. Bezerra, C. A. P. Burgardt, and D. R. Campelo, "Identifying IoT devices and events based on packet length from encrypted traffic," vol. 144, pp. 8–17, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366419300052

[118] O. Platforms and E. T. for the Internet of Things. (2024) Open platforms and enabling technologies for the internet of things. Accessed on: Dec 17, 2024:. [Online]. Available: https://openiot.fbk.eu/

[119] L. Rahman, T. Ozcelebi, and J. Lukkien, "Understanding iot systems: A life cycle approach," *Procedia Computer Science*, vol. 130, pp. 1057–1062, 01 2018.

[120] F. Ramparany, F. G. Marquez, J. Soriano, and T. Elsaleh, "Handling smart environment devices, data and services at the semantic level with the FI-WARE core platform," in *2014 IEEE International Conference on Big Data*. IEEE, 2014, pp. 14–20. [Online]. Available: http://ieeexplore.ieee.org/document/7004417/

[121] P. Rieger, M. Chilese, R. Mohamed, M. Miettinen, H. Fereidooni, and A.-R. Sadeghi, "ARGUS: Context-based detection of stealthy IoT infiltration attacks," in *32nd USENIX Security Symposium (USENIX Security 23)*. USENIX Association, 2023, pp. 4301–4318. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/rieger

[122] B. Russell and D. Van Duren, *Practical Internet of Things Security*, ser. Community experience distilled. Packt Publishing, 2016. [Online]. Available: https://books.google.de/books?id=IaExDQEACAAJ

[123] D. J. Russomanno, C. R. Kothari, and O. A. Thomas, "Building a sensor ontology: A practical approach leveraging iso and ogc models," in *International Conference on Artificial Intelligence*, 2005. [Online]. Available: https://api.semanticscholar.org/CorpusID:12884921

[124] P. Saint-Andre. Extensible messaging and presence protocol (XMPP): Instant messaging and presence. Accessed on: Dec 17, 2024:. [Online]. Available: https://www.rfc-editor.org/rfc/rfc3921.html

[125] P. Saint-Andre, “Extensible Messaging and Presence Protocol (XMPP): Core,” RFC 3920, Oct. 2004, accessed on: Dec 10, 2024:. [Online]. Available: https://www.rfc-editor.org/info/rfc3920

[126] I. H. Sarker, “Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions,” vol. 2, no. 6, p. 420, 2021. [Online]. Available: https://link.springer.com/10.1007/s42979-021-00815-1

[127] F. Scheidegger, L. Benini, C. Bekas, and A. C. I. Malossi, “Constrained deep neural network architecture search for iot devices accounting hardware calibration,” 09 2019.

[128] S. Schmidgall, J. Achterberg, T. Miconi, L. Kirsch, R. Ziaei, S. P. Hajiseyedrazi, and J. Eshraghian, “Brain-inspired learning in artificial neural networks: a review,” 2023. [Online]. Available: http://arxiv.org/abs/2305.11252

[129] C. Sengul and A. Kirby, “Message Queuing Telemetry Transport (MQTT) and Transport Layer Security (TLS) Profile of Authentication and Authorization for Constrained Environments (ACE) Framework,” RFC 9431, Jul. 2023. [Online]. Available: https://www.rfc-editor.org/info/rfc9431

[130] Z. Shelby, K. Hartke, and C. Bormann, “The Constrained Application Protocol (CoAP),” RFC 7252, Jun. 2014. [Online]. Available: https://www.rfc-editor.org/info/rfc7252

[131] S. S. Shriyal and B. S. Ainapure, “Iot device classification techniques and traffic analysis - a review,” in *2021 International Conference on Technological Advancements and Innovations (ICTAI)*, 2021, pp. 244–250.

[132] B. SIG. (2024) Bluetooth. Accessed on: Dec 9, 2024:. [Online]. Available: https://www.bluetooth.com/

[133] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, “Classifying IoT devices in smart environments using network traffic characteristics,” vol. 18, no. 8, pp. 1745–1759, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8440758/

[134] A. Sivanathan, H. H. Gharakheili, and V. Sivaraman, “Can we classify an IoT device using TCP port scan?” in *2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)*. IEEE, 2018, pp. 1–4. [Online]. Available: https://ieeexplore.ieee.org/document/8913346/

[135] I. O. T. SL. (2024) Thinger.IO. Accessed on: Aug. 1, 2024:. [Online]. Available: https://docs.thinger.io/

[136] J. Soldatos, N. Kefalakis, M. Hauswirth, M. Serrano, J.-P. Calbimonte, M. Riahi, K. Aberer, P. P. Jayaraman, A. Zaslavsky, I. P. Žarko, L. Skorin-Kapov, and R. Herzog, “OpenIoT: Open source internet-of-things in the cloud,” in *Interoperability and Open-Source Solutions for the Internet of Things*. Springer, 2015, vol. 9001, pp. 13–25. [Online]. Available: https://link.springer.com/10.1007/978-3-319-16546-2_3

[137] G. Soós, D. Kozma, F. N. Janky, and P. Varga, “Iot device lifecycle – a generic model and a use case for cellular mobile networks,” in *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2018, pp. 176–183.

[138] V. Spadari, I. Guarino, D. Ciuonzo, and A. Pescapé, “A comparison between classical and quantum machine learning for mobile app traffic classification,” in *2024 IEEE/ACM Symposium on Edge Computing (SEC)*, 2024, pp. 461–467.

[139] W. S. W. Standards. (2025) Review of sensor and observations ontologies. Accessed on: Feb 27, 2025:. [Online]. Available: https://www.w3.org/2005/Incubator/ssn/wiki/Review_of_Sensor_and_Observations_Ontologies.html

[140] W. S. W. Standards. (2025) Web ontology language (owl). Accessed on: Feb 27, 2025:. [Online]. Available: https://www.ontologyportal.org/index.html

[141] A. B. Syed Ali and F. Ford. (2024) Cybersecurity is the key to unlocking demand in the internet of things. Accessed on: Dec 10, 2024:. [Online]. Available: https://www.bain.com/de/

[142] P. M. S. Sánchez, J. M. J. Valero, A. H. Celdrán, G. Bovet, M. G. Pérez, and G. M. Pérez, “A survey on device behavior fingerprinting: Data sources, techniques, application scenarios, and datasets,” *IEEE Communications Surveys, Tutorials*, vol. 23, no. 2, pp. 1048–1077, 2021.

[143] C. Takasaki, T. Korikawa, K. Hattori, and H. Ohwada, “Traffic behavior-based device type classification,” in *2023 International Conference on Computing, Networking and Communications (ICNC)*, 2023, pp. 353–357.

[144] A. S. Tanenbaum and D. J. Wetherall, *Computer Networks*, 5th ed. USA: Prentice Hall Press, 2010.

[145] I. T. T. That. (2024) If-this-then-that. Accessed on: Dec 9, 2024:. [Online]. Available: https://ifttt.com/

[146] ThingsBoard. (2024) Thingsboard open-source iot platform. Accessed on: Dec 17, 2024:. [Online]. Available: https://thingsboard.io/

[147] ThingSpeak. (2024) Thingspeak for iot projects. Accessed on: Dec 17, 2024:. [Online]. Available: https://thingspeak.com/

[148] unabiz Technology. (2024) Sigfox. Accessed on: Aug. 1, 2024:. [Online]. Available: https://www.sigfox.com/en,https://www.thethingsnetwork.org/

[149] P. B. Upama, M. J. H. Faruk, M. Nazim, M. Masum, H. Shahriar, G. Uddin, S. Barzanjeh, S. I. Ahamed, and A. Rahman, "Evolution of quantum computing: A systematic survey on the use of quantum computing tools," in *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, 2022, pp. 520–529.

[150] M. Vögler, J. M. Schleicher, C. Inzinger, and S. Dustdar, "A scalable framework for provisioning large-scale IoT deployments," vol. 16, no. 2, pp. 1–20, 2016. [Online]. Available: https://dl.acm.org/doi/10.1145/2850416

[151] L. Xie, L. Peng, J. Zhang, and A. Hu, "Radio frequency fingerprint identification for internet of things: A survey," vol. 3, 2023.

[152] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," 2015. [Online]. Available: http://arxiv.org/abs/1501.01367

[153] Zotero-Org. (2025) Zotero. Accessed on: Mar 4, 2025:. [Online]. Available: https://www.zotero.org/

Author Biographies

Maurizio Petrozziello is a Researcher and lab engineer at the Faculty of Computer Science and Engineering of the Frankfurt University of Applied Sciences in Frankfurt am Main, Germany. He earned his diploma in communications engineering in 1992 at the University of Applied Sciences in Darmstadt, Germany. His research interests include IoT networks, IT security, and network protocols. Currently, he works on IoT device virtualization and generating Datasets for IoT device Classification.

Dr. Christian Baun is a professor of computer science, particularly computer networks and operating systems, at the Faculty of Computer Science and Engineering of the Frankfurt University of Applied Sciences in Frankfurt am Main, Germany. He is the author of several books, articles, and research papers, e.g., in public and private cloud infrastructure and platform services, single-board computers, distributed computing, and distributed storage. His research interests include operating systems, cloud services, distributed systems, and computer networks.

Dr. Martin Kappes is a professor of computer science, particularly computer networks and operating systems, at the Faculty of Computer Science and Engineering of the Frankfurt University of Applied Sciences in Frankfurt am Main, Germany. His work focuses on IT security, particularly network and system security, security organization, evaluation, and management, and the reliability and availability of complex systems.