This is a pre-print.
The final version will appear in the proceedings of the Applied Cryptography and Network Security Workshops - ACNS 2025 Satellite Workshops: AIHWS, AIoTS, QSHC, SCI, PrivCrypt, SPIQE, SiMLA, CIMSS by Springer Nature Switzerland AG.

IoTCat: A Multidimensional Approach to Categorize IoT Devices in order to Identify a Delegate for Cybersecurity Functions

Emiliia Geloczi[1], Nico Mexis[1], Benedikt Holler[1], Henrich C. Pöhls[1], and Stefan Katzenbeisser[1]

University of Passau, Innstr. 41, Passau, Germany
{emiliia.geloczi,nico.mexis,henrich.poehls, stefan.katzenbeisser}@uni-passau.de, benedikt.holler@gmail.com

Abstract. Despite the extensive functionality and broad diversity of Internet of Things (IoT) devices, they remain vulnerable to cybersecurity attacks. Still, many of them are unable to protect themselves due to limited resources or missing security features. One solution is to delegate security functions to a "stronger" device "encapsulating" a "weaker" one in a secure environment. In order to identify a suitable IoT device within the system to take over security responsibilities, we propose a novel flexible multidimensional approach to IoT device categorization, named IoTCat, that considers not only the technical characteristics of devices (e.g., available memory) but also user experience (e.g., trust in the vendor). In this paper, we describe our approach, compare it with existing methods, and demonstrate its applicability by presenting a proof-of-concept software solution.

Keywords: Categorization · Security · IoT · Delegation · Identification.

1 Introduction

The increasing number of Internet of Things (IoT) devices in everyday life [40] leads to more potential attack vectors, thereby increasing the risk to the personal data generated or exchanged in IoT networks [13]. At first glance, it may seem that IoT systems are powerful enough to ensure their own security, as they provide users with a wide range of functionality [29]. However, specific IoT devices are unable to ensure their own security, as most lack sufficient resources to perform complex cryptographic operations or store large amounts of data [25]. Nonetheless, these devices cannot be left unprotected, as they are also participants in the IoT system and generate, transmit, and receive potentially context-critical data [13].

It is reasonable to assume that if some devices are unable to protect themselves, they should be protected by another device. In other words, if security functions are not feasible for "weak" devices, they should be delegated to a "strong" device(s). In this case, the "strong" device is called a *delegate*, which

"encapsulates" IoT devices in a secure environment (see Figure 1). In order to use this delegation approach in an existing IoT network, the first task is to identify a device capable of acting as a delegate. Ideally, a suitable device can be identified among those already present in the network. Only if no appropriate device is found, should a "stronger" device be added to the system.

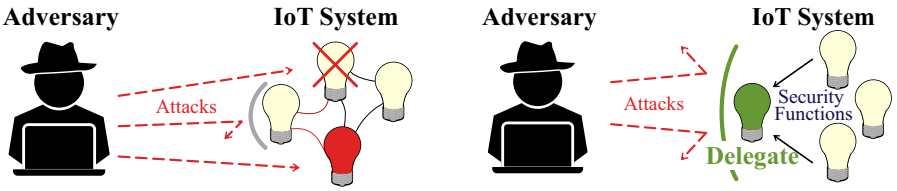


Fig. 1: IoT system (a) without and (b) with security delegation (based on [11]).

We assume that a device that outperforms others can become a delegate. Therefore, the devices connected to the network should be ranked based on a set of relevant characteristics. Depending on the purpose of selecting a delegate and its potential functions, the characteristics considered for ranking the existing devices may vary. To perform security functions (e.g., executing cryptographic protocols), the focus lies on technical characteristics organized into two groups: *Device-Related characteristics*, which we believe are most relevant to the ability to perform security tasks, and *Network-Related characteristics*, which we see as less relevant to security. Additionally, we consider *User-Related characteristics* to reflect user perception. However, since these characteristics are subjective, we assume they have the lowest weight among the groups.

Although a combination of technical and subjective characteristics reduces the likelihood of identifying multiple devices suitable for the delegate role, it does not eliminate this possibility. Having multiple devices that can act as delegates is not a disadvantage, but a benefit: it can enhance system reliability by balancing the load of delegation over several delegates or/and avoid a single point of failure by introducing redundancy. Additionally, splitting of functionality enables one to construct a defense-in-depth where devices on one level can jointly provide security functions to devices at higher levels.

1.1 Contribution

In this work, we introduce a novel flexible multidimensional approach, named *IoTCat*, which categorizes IoT devices based on their technical characteristics and the user's perspective, enabling the identification of a device(s) that can act as a delegate(s) and perform security functions on behalf of others.

1.2 Organization

Section 2 and Section 3 describe the methodology and implementation details of *IoTCat*. A review of existing related works is presented in Section 4, followed by a comparison of *IoTCat* with similar approaches and its discussion in Section 5. Finally, Section 6 concludes the paper and outlines potential directions for future research.

2 Methodology

We term our methodology *IoTCat*, as it categorizes IoT devices connected to a particular network according to nine characteristics in order to rank them for suitability in becoming a delegate for security functions. To achieve this, *IoTCat* collects device characteristics, calculates device scores, and performs ranking and classification of the devices (see Figure 2).

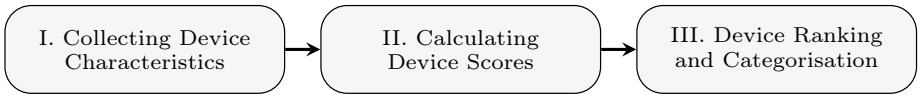


Fig. 2: Three-step workflow of *IoTCat*.

The pre-categorization task related to the identification of devices connected to a local network is not described in this section but is mentioned in Section 3, as it is performed using well-known technical tools. Since the primary novelty of our work lies in the categorization approach, we focus exclusively on the steps related to it.

Step I: Collecting Device Characteristics

Our categorization approach is based on nine device characteristics collected from devices within the network during this step. These characteristics are grouped into the following three categories:

Device-Related characteristics ($\mathcal{DR}$) include properties directly associated with a device:

1. *RAM* (Random Access Memory) is a limited resource in IoT devices. More RAM allows devices to use stronger security mechanisms, provide additional services, and better mitigate attacks.
2. *CPU* (Central Processing Unit) performance determines how fast a device can perform tasks.
3. *Availability (AV)* refers to a device's uptime. High AV indicates the absence of major faults, thereby ensuring more reliable service.

4. *Firmware Version (FV) Freshness* refers to the recency of a firmware version. Firmware updates improve functionality and should address recently disclosed vulnerabilities [9,5]. Therefore, it is generally assumed that newer firmware versions are more secure than older ones. We adhere to this assumption to simplify the demonstration of the general idea of *IoTCat*'s application. However, this assumption may not always apply, especially when comparing firmware versions across different devices. A newer firmware version of one device may provide a lower level of security than an older firmware version of another device. For more accurate categorization, a more detailed comparison of firmware is needed, for which a more complex algorithm could be implemented in the future. For example, Artificial Intelligence (AI) technologies could be employed to parse changelogs and identify which known vulnerabilities have been addressed by an update.

Network-Related characteristics ($\mathcal{NR}$) refer to properties associated with device communication:

5. *Wi-Fi* captures what kind of wireless communication technology is used. Newer standards improve the security, but also provide higher link rates, broader frequency support, and lower latency.
6. *Round-Trip Time (RTT)* is the time a signal takes to travel to a destination and back, also known as a delay.
7. *MPS* refers to the MQTT [36] protocol's Maximum Packet Size a device can process. A higher MPS reduces both transmission time and bandwidth usage.

User-Related characteristics ($\mathcal{UR}$) represent subjective aspects closely related to a user:

8. *Vendor Trust (VT)* is influenced by prior positive/negative experiences with the vendor or the user's perception of a vendor.
9. *Price (PR)* is not related to security, but it remains a key factor for many users when choosing an IoT device.

Each group of characteristics is weighted based on its relevance. In our context, $\mathcal{DR}$ is considered the most critical, as it directly reflects the device's ability to perform security tasks. While $\mathcal{NR}$ is also important, it is assigned a lower weight since it influences data transmission efficiency rather than the device's performance. $\mathcal{UR}$ has the lowest weight due to its subjectivity, though it remains relevant given the user-centric nature of IoT systems. These weights can be adjusted to fit specific application scenarios.

Step II: Calculation of Device Scores

Based on the data collected during Step I, the rank values for all devices are calculated using the formulas described below.

Definition 1. *Assume that there is an IoT system SM which consists of* $n \geqslant 2$ *IoT devices* $d_i,\ i = 1, \ldots, n$, *which can be denoted as*

$$SM \coloneqq \{d_i \mid i = 1, \ldots, n\}.$$

Definition 2. *For every device* $d \in SM$, *the device can be characterized by a triple of characteristic groups as*

$$d \coloneqq \{\mathcal{DR}, \mathcal{NR}, \mathcal{UR}\},$$

where $\mathcal{DR}$ *represents Device-Related,* $\mathcal{NR}$ *represents Network-Related, and* $\mathcal{UR}$ *represents User-Related characteristics.*

Definition 3. $\mathcal{DR}$, $\mathcal{NR}$, *and* $\mathcal{UR}$ *can be described as the following collections:*

$$\begin{aligned}
\mathcal{DR} &\coloneqq \{RAM, CPU, AV, FV\},\\
\mathcal{NR} &\coloneqq \{WF, RTT, MPS\},\\
\mathcal{UR} &\coloneqq \{VT, PR\}.
\end{aligned}$$

Based on the definitions introduced, we can denote the device score S_d as follows:

$$\begin{aligned}
S_d &= 3 \times \mathcal{DR} + 2 \times \mathcal{NR} + 1 \times \mathcal{UR} =\\
&= 3 \times (S_{RAM_d} + S_{CPU_d} + S_{AV_d} + S_{FV_d}) +\\
&+ 2 \times (S_{WF_d} + S_{RTT_d} + S_{MPS_d}) + 1 \times (S_{VT_d} + S_{PR_d}),
\end{aligned}$$

where S_x is the characteristic score value of the device x. $\mathcal{DR}$, $\mathcal{NR}$, and $\mathcal{UR}$ are weighted according to their relevance, as discussed previously in Step I.

Table 1 illustrates how the individual scores for each characteristic of the device are calculated. Next, the values are normalized, with the best score being 1 and the worst score being 0. Once *IoTCat* collects and computes the scores for each device, it proceeds to the next and final step.

Step III: Ranking and Categorization of Devices

There are several methods for achieving device categorization, including linear ranking, $1/n$-dimensional k-means, or Kernel Density Estimation (KDE) [28,30]. We selected KDE because it is deterministic and provides clear categories of devices[1]. To apply KDE, we draw a graph for density estimation of device scores and identify one or more local minima. These minima serve as cut-off points for categories, as devices on opposite sides of these minima are likely significantly different from one another.

[1] In this paper, we present the most basic version of IoTCat, which means that other forms of ranking algorithms could be a possible extensions; as KDE provided good results for our prototype we did not investigate the choice further.

Table 1: Score calculation of each device characteristic.

Char.	Unit	Score S_x	Comment
RAM	kB	$\frac{RAM-RAM_{min}}{RAM_{max}-RAM_{min}}$	RAM is the available RAM value of the device.
CPU	ms	$\frac{CPU-CPU_{min}}{CPU_{max}-CPU_{min}}$	CPU denotes the time required for the device to complete a benchmark task.
AV	sec	$\frac{AV}{AV_{max}}$	AV is the uptime of d, AV_{max} is the uptime of the system.
FV	sec	$1-\frac{FV-FV_{min}}{FV_{max}-FV_{min}}$	FV represents the time elapsed between the firmware build and the $IoTCat$ request.
WF	-	$\sum_{i=1}^{9} Sup_i \times S_{WFS_i}$	S_{WFS} is the score value of Wi-Fi standard (see Appendix A), Sup_i denotes whether a device supports (1) or does not support (0) the standard with index i.
RTT	ms	$\frac{RTT-RTT_{min}}{RTT_{max}-RTT_{min}}$	RTT represents the average ping time measured over multiple requests.
MPS	B	$\frac{MPS-MPS_{min}}{MPS_{max}-MPS_{min}}$	MPS denotes the maximum packet size supported by the device.
VT	-	$\frac{VT-VT_{min}}{VT_{max}-VT_{min}}$	VT of the device is equal to $VT_{IoTCat} + VT_{user}$, where VT_{IoTCat} and VT_{user} are trust values assigned by $IoTCat$ and user, respectively (see Appendix B).
PR	$	$1-\frac{PR-PR_{min}}{PR_{max}-PR_{min}}$	PR is price of the device.

$X_{min/max}$ denotes the minimum/maximum value of the characteristic X among all devices.

3 Implementation Results

In order to demonstrate the application of *IoTCat*, we implemented a proof-of-concept software solution and tested it on 36 IoT devices. The test setup is illustrated in Figure 3. All IoT devices are equipped with the custom open source Tasmota firmware [37] and communicate via the MQTT protocol with a Raspberry Pi (RPi) acting as a broker. The RPi performs all necessary operations and provides the user interface in the form of a web application.

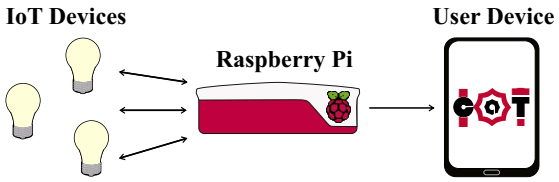


Fig. 3: Test setup of *IoTCat*.

First, the pre-categorization step is performed, during which the RPi identifies the IoT devices connected via MQTT using the Python `paho.mqtt.client` library. Next, facilitating commands offered by the Tasmota firmware, the RPi

requests data from each device regarding the characteristics specified in Section 3.I. For most characteristics, the RPi receives values in a ready-to-use format. However, the following of them still require pre-/post-processing:

- To evaluate *CPU* performance, the chip identifier can be retrieved and used to find the corresponding benchmark result in a pre-filled table. However, we decided to enhance the firmware to enable real-time performance testing. To achieve this, we extended the firmware to support the initiation of a benchmark calculation directly on the device, specifically, computing the dot product of two vectors, each containing 256 elements. This approach provides an indication of the device's performance not only on commonly used IoT chips (e.g., ESP8266EX) but also on a wide range of other platforms.
- To evaluate the RTT, the RPi sends n consecutive requests, records each individual RTT_i, and then calculates their average value. In our implementation, we set $n = 3$.
- VT cannot be directly obtained from the device and requires user input. Initially, the device's vendor is identified using the device's MAC address. The user should select the most trusted vendor. If any vendors are not automatically identified, the user is asked to enter them manually (see Appendix B).
- PR cannot be retrieved directly from a device. The user is required to either enter the device's price manually or provide the device's name, which is then used to identify the price using an AI engine. In *IoTCat*, `OpenAI` is used with the prompt: `Average price+`*`Device_Name`*`+`*`Device_Vendor`*` in $`.

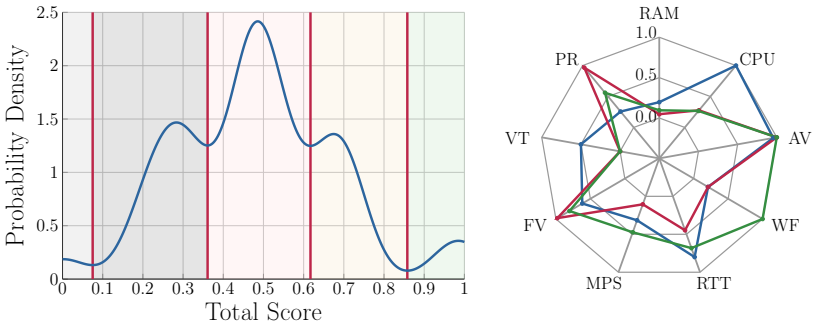
From the obtained values, a `.csv` file is generated and passed to the web application, which calculates the scores, categorizes the devices, and provides the user interface.

Device scores are calculated as described in Table 1 (see Section 2.II), and categorization is performed using Gaussian KDE (see Section 2.III). By adjusting the bandwidth of KDE, we found that a value of 0.06 resulted in five device categories, while the "ideal" bandwidth of approximately 0.092 (calculated using Silverman's rule of thumb [35]) resulted in only two categories. We believe that the five categories provide a good balance between classification granularity and clarity (see Figure 4a). The individual scores of the devices can be depicted using a radar chart, an example of which is shown in Figure 4b.

The web application is implemented in C# using the `Microsoft .NET 8.0` platform, the `MathNet.Numerics` NuGet package and `Blazor` web framework with server-side rendering. Users can access *IoTCat* via a link and open it on any device. The interface is shown in Figure 5.

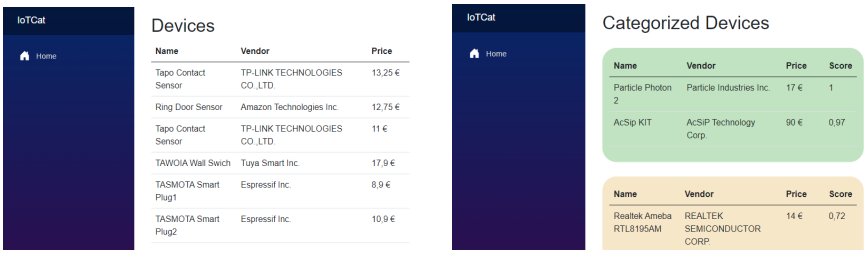
4 Related Work

In our study, two interrelated topics are addressed: the *identification* of a suitable device for the delegate role and the *categorization* of devices based on their characteristics. Accordingly, in the literature review, we discuss works related to these topics with focus on studies most closely related to ours.



(a) (b)

Fig. 4: (a) Gaussian probability density function. (b) Radar chart for: Arduino GIGA R1, Raspberry Pi Pico W RP2040, and nRF7002-DK Wi-Fi 6 DevKit.



(a) (b)

Fig. 5: *IoTCat* UI after (a) identification and (b) categorization of IoT devices.

4.1 Identification

During the study of literature on device identification, we observed that the works can be divided into two groups, based on the question each proposed approach aims to answer.

What is/was it? The first group includes works that present approaches focused on identifying the device type, answering the question *"What is/was it?"* (see Table 2). For example, the authors analyze patterns of communication or device behavior and draw conclusions about the device type [7,22,38]. In general, all the approaches in this group rely on various AI technologies including Deep Learning, Machine Learning (ML). However, since our primary focus is not on device type identification, we do not review these approaches in detail.

What is good/bad? In the second group, we include papers that seek to answer the question *"What is good/bad?"* based on security implications. In general, the approaches are aimed at identifying trusted or suspicious devices. The authors employ various techniques, such as blacklisting, whitelisting, and ML. We

Table 2: Research works focus on IoT device type identification.

Author(s)	Title
Aksoy and Gunes [1]	Automated IoT Device Identification using Network Traffic
Ali *et al.* [2]	A Generic Machine Learning Approach for IoT Device Identification
Bao *et al.* [3]	IoT Device Type Identification Using Hybrid Deep Learning Approach...
Bezawada *et al.* [4]	Behavioral Fingerprinting of IoT Devices
Chen *et al.* [6]	IoT-ID: Robust IoT Device Identification...
Chowdhury *et al.* [7]	A Deep Learning Approach for Classifying ... IoT Devices...
Fan *et al.* [10]	An IoT Device Identification Method based on Semi-supervised Learning
Hamad *et al.* [15]	IoT Device Identification via Network-Flow ... Fingerprinting and Learning
Kotak and Elovici [18]	IoT device identification based on network communication analysis ...
Le *et al.* [19]	IoTFinder: Efficient Large-Scale Identification of IoT Devices ...
Liu *et al.* [21]	Class-Incremental Learning for Wireless Device Identification in IoT
Liu *et al.* [22]	Zero-Bias Deep Learning for Accurate Identification of ... (IoT) Devices
Meidan *et al.* [24]	ProfilIoT: A Machine Learning Approach for IoT Device Identification ...
Marchal *et al.* [23]	AuDI: Toward Autonomous IoT Device-Type Identification...
Salman *et al.* [31]	A machine learning based framework for IoT device identification ...
Scheidt and Adda [33]	Identification of IoT Devices for Forensic Investigation
Thom *et al.* [38]	FlexHash - Hybrid Locality Sensitive Hashing for IoT Device Identification
Wang *et al.* [42]	Efficient traffic-based IoT device identification ...
Yin *et al.* [43]	IoT ETEI: End-to-End IoT Device Identification Method

focus on this category in greater detail and discuss a few papers that we believe are most relevant to our approach.

Khalil *et al.* describe the trusted device identification approach based on several components: resource, trust, delegation repositories, and an ontology base [17]. In contrast to similar models, this approach uses a concept-ing technique to organize the ontology, thereby addressing the shortcomings of previous studies. According to the authors' evaluation, this approach outperforms its competitors by accounting for additional features such as completeness, prioritization, and consistency.

Miettinen *et al.* present the automated system IoT Sentinel which identifies types of devices within IoT networks and restricts external communication with vulnerable devices to mitigate potential threats [26]. The system uses the pattern of packets sent during the device setup phase as unique fingerprints. The system uses packet patterns transmitted during the device setup phase as unique fingerprints. Based on these fingerprints, devices are identified and classified into different security levels: strict, restricted, or trusted. This approach enables effective device identification while maintaining high performance.

A method of selecting a trusted element in an IoT network based on a dynamic combination of white and blacklists is proposed by Wang *et al.* in [41]. The placement of items on either list is determined by the trust relationship between pairs of end users and between the users and the provider. Furthermore, both direct and indirect trust values are considered when assigning a to each end user. The proposed method is evaluated using the Lyapunov theory. The results demonstrate its robustness against attacks targeting a service provider and its suitability for scalable systems.

Tragos *et al.* present a trust model based on a combination of five metrics: communication-based trust (communication quality), security-based trust (device behavior), data-based trust (reliability of measurements produced by IoT devices), social relationship-based trust (owner/manufacturer characteristics),

and reputation-based trust (reviews about the device). Based on the provided data analysis, the proposed model enables the detection of malfunctioning devices by verifying whether their output values are within a specified range [39].

4.2 Categorization

Next we discuss papers that focus on the categorization of IoT devices based on various parameters.

An early concept for categorizing IoT devices was presented by Gigli and Koo [12]. The authors proposed that devices be classified into four categories according to the services they provide: identity-related (a device identifies an entity using a label), information aggregation (a device gathers data and sends it to a remote application for processing), collaborative-aware (a device uses the gathered information to make decisions and executes actions accordingly), and ubiquitous (a theoretical idea where collaborative-aware services are available, always connected and interoperable – "ultimate goal of IoT" [12]).

Liu *et al.* [20] propose a method for categorizing IoT devices within a multihop mesh network based on their reputation, and hence, the threat they pose to the rest of the system. This method includes two approaches: "Hard detection" [20], which groups devices into *benign* and *malicious* categories, and "soft detection" [20], which also includes an intermediate category for *suspicious* devices.

The model for categorizing IoT services based on their inherent security risks is presented in [8]. The service assessment is based on the DREAD model developed by Microsoft [34]. The authors define seven questions, the answers to which determine the rating of each service. Based on the resulting ratings, services can be allocated into four categories: critical, high, medium, and low risk.

A five-level IoT model consisting of cloud, fog, edge, mist, and dew computing layers is presented in [14]. Devices within the described IoT network are distributed across these layers based on their processing capabilities. Depending on the layer, the devices are assigned roles (such as cloud data manager, equipment provider, service provider, etc.), in order to ensure system security under the General Data Protection Regulation (GDPR).

Sawadogo *et al.* propose an approach to clustering IoT devices using unsupervised ML [32]. Network-level traffic characteristics serve as input, collected over multiple weeks, and are then categorized using both k-means and BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) clustering methods. The evaluation results show that this approach achieves an accuracy of 70%.

5 Discussion

This section examines *IoTCat*, focusing on its distinctions from existing approaches, potential security concerns, and application domains.

5.1 Comparison with Other Approaches

As described in Section 4, existing works can be grouped into two categories: approaches that focus on *identifying* a device's type or trustworthiness, and those that concentrate on *categorizing* devices. We believe that, in general, the second category is more relevant to our work. However, *IoTCat* is designed to identify device(s) with sufficient computational power to perform security functions as delegate(s), thereby addressing the question *"What is strong enough?"*. While this question may appear similar to one addressed by approaches from *identification* category, it is fundamentally different and highlights the distinct application goal of *IoTCat*.

Table 3 presents a comparison of *IoTCat* with the most closely related approaches from both categories.

The *Characteristics* column represents features of devices considered when determining their category.[2] From the table, we can see that most approaches evaluate the behavior-related characteristics, e.g., traffic patterns [20,26,32], or reputation [17,39,41]. Khalil *et al.* [17] consider also device resources, which we can interpret as analogous to our *Device-Related characteristics*. Furthermore, Tragos *et al.* [39] evaluate communication quality-related features that can be seen as similar to our *Network-Related characteristics*.

The *Evaluation Method* column describes the strategies used by the authors to evaluate devices. Some approaches adopt a concept-ing method [17,39], where characteristics are assigned s based on their significance, followed by the computation of rank values for the identification of a trusted device. *IoTCat* follows a similar method but also incorporates KDE for categorization. Other approaches classify devices based on their types [26], roles [12], capabilities [14], potential threats [20] and risks [8].

Regarding *Device Categories*, we can see that only *IoTCat* and the approach presented by Sawadogo *et al.* [32] offer the possibility of flexible categorization. Other approaches strictly define fixed categories. However, in contrast to our approach, Sawadogo *et al.* focuses on categorizing individual IoT devices but does not include any ranking mechanism.

Drawing conclusions from the comparison, we observe that *IoTCat* has a unique goal that is not commonly found in other approaches. Existing works primarily focus on either identifying the type or trustworthiness of devices or on categorizing them, typically without integrating these goals. In contrast, *IoTCat* combines both: it categorizes devices and identifies the "strongest" one(s) capable of acting as the delegate(s) and performing the required security functions. In addition, unlike other approaches, *IoTCat* provides users with a comprehensive overview of network device resources while taking user preferences into account.

Moreover, *IoTCat* offers flexible configuration to address specific needs. First, it provides customizable categorization, allowing devices to be distributed across a variable number of categories. This can be achieved by adjusting the bandwidth

[2] For consistency, we assume that approaches focusing on *identification* classify devices into trusted/untrusted groups.

Table 3: Comparison of *IoTCat* with the existing approaches.

Paper	Characteristics	Evaluation Method	Device Categories
Identification			
Khalil *et al.* [17]	resources, trust, ontology	concept-ing	trusted, untrusted
Miettinen *et al.* [26]	traffic patterns	type-based	strict, restricted, trusted
Wang *et al.* [41]	relationships between nodes	black-and-white lists	trusted, untrusted
Tragos *et al.* [39]	communication quality, behavior, measurements reliability, social and reputational trust	concept-ing	trusted, untrusted
Categorization			
Gigli and Koo [12]	services provided	role-based	identity, data aggregation, collaborative-aware, ubiquitous
Liu *et al.* [20]	traffic patterns	threat-based	benign, malicious, suspicious
Dominik *et al.* [8]	DREAD risks	numerical ranks	critical, high, medium, low
Hadzoovic *et al.* [14]	functions and capabilities	capability analysis following GDPR	cloud, fog, edge, mist, dew computing
Sawadogo *et al.* [32]	traffic patterns	k-means, BIRCH	flexible
IoTCat	RAM, CPU, AV, FV, WF, RTT, MPS, VT, PR	concept-ing, KDE	flexible

of the KDE or by using alternative methods (see Section 2.III). Second, using an evaluation method to calculate devices' scores allows for flexible customization of the ranking and categorization results, as the weights can be adjusted to emphasize the characteristics most relevant to the specific task. For our current research prototype presented here, we choose to give the subjective user-related characteristics a weight of 1, which is the lowest weight of the three groups; exploring the added flexibility by setting other weights to better match given real-life scenarios is subject to further research.

In this paper, we present the most basic version of *IoTCat*. However, we see significant potential for extending it by incorporating additional characteristics to enable more comprehensive device evaluation and more accurate categorization. We believe that *IoTCat* can seamlessly integrate any of the approaches presented in Table 3. For example, traffic pattern or behavior analysis could be implemented to identify devices that not only have sufficient resources for the

required tasks but also exhibit trustworthy behavior. Such devices could then also be assigned tasks involving sensitive or critical data. Moreover, other aspects can be added to adapt the categorization to the real-world situation. For example, the following characteristics of the devices can be taken into account: the updateability [5], the availability of dedicated security hardware, or specific encryption abilities, or power limitations [27].

Also, it is important to note that a direct comparison of *IoTCat* with other approaches in terms of categorization results is not feasible. As discussed earlier, each approach is designed with a specific purpose, leading to inherently non-comparable outcomes.

5.2 Security Concerns

Applying the security delegation method to a system inevitably redirects an adversary's attack vector to a device acting as a delegate. On the one hand, delegation enhances the system's security, since weak devices are no longer the primary target and the delegate is able to perform more sophisticated protection mechanisms. However, on the other hand, the delegate becomes the main target of an adversary, resulting in a single point of failure vulnerability, which certainly introduces a new challenge to the security strategy. In order to address this vulnerability, *IoTCat* provides the device categorization functionality, enabling the selection of multiple devices for the delegate role. It ensures fault tolerance and distributes attack vectors across several devices, rather than concentrating them on a single target.

5.3 Possible Applications

Categorization and ranking of IoT devices can be applied not only within smart home systems but also across other domains. Regardless of the application goals, categorization solutions enhance the overall security of systems by enabling network load balancing, resource optimization, device monitoring, access control management, etc.

For example, in the healthcare and medical IoT device domain, the security and reliability of patient-critical devices must be strictly ensured. Conversely, non-medical devices such as smart beds and room thermostats can be designated as low priority. In the case of an emergency power outage, the load should be correctly distributed to prioritize critical devices. Furthermore, medical devices access data with varying levels of sensitivity. The more critical the data, the more security mechanisms must be applied; hence, more reliable and powerful devices should be employed.

Another example of the application of device categorization is in the Small or Medium-sized Enterprise (SME) domain. In SMEs that rely on IoT networks for their operations, such as manufacturing plants using smart sensors and microcontrollers to monitor production lines, categorization software similar to *IoTCat* can play a crucial role in optimizing network performance. For instance, in a manufacturing plant with multiple production lines, each line may have its

own set of IoT devices monitoring various parameters. The software can identify which microcontroller consistently performs best under load and assign it the responsibility of managing inter-device communication and data aggregation. This delegated microcontroller can then handle more demanding tasks, such as coordinating real-time alerts when machines show signs of malfunction, thereby improving the overall efficiency and reliability of the production process. By using this software regularly, the company can adapt to changes in the IoT network. This ensures that the most capable microcontroller is always in charge, resulting in more reliable operations, reduced downtime, and better resource allocation, ultimately contributing to more streamlined and cost-effective production processes.

6 Conclusions and Future Works

In this paper, we introduce a novel multidimensional approach *IoTCat* designed to rank and categorize IoT devices. We use it to assess their suitability to act as a delegate for cybersecurity functions. By incorporating nine characteristics, covering both technical aspects of IoT devices and user preferences, our approach enables a comprehensive evaluation of devices within the system and identification of those "stronger" device(s) that are capable of performing critical security functions on behalf of "weaker" devices. Furthermore, we designed the *IoTCat* methodology to be highly flexible and customizable, allowing it to be adapted to a wide range of systems and use cases. Its full potential can be further explored through future research:

- Incorporation of continuous system monitoring and behavior/reputation-related metrics to determine whether the current delegate remains optimal.
- Exploration of *IoTCat*'s applicability beyond prototype settings, including its integration into real-world and large-scale IoT systems.
- Applying different clustering algorithms and ML techniques to offer even more flexible categorization.
- Design and implementation of a delegation protocol, to be applied following
- Explore backup and fail-over strategies to avoid the delegate becoming a single-point of failure. *IoTCat*, which accounts for diverse network topologies and security functions to be delegated.

Acknowledgments. This work has been partially funded by the Bavarian State Ministry of Science and Arts (BayStMWK), under Project "ForDaySec: Security in Everyday Use of Digital Technologies (fordaysec.de)" of the Bavarian Research Association and by the Interreg VI-A Programme Germany/Bavaria–Austria 2021–2027, as part of Project BA0100016: "CySeReS-KMU: Cyber Security and Resilience in Supply Chains with focus on SMEs" of the European Union.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

A Appendix: Scoring of Wi-Fi Standards

Different Wi-Fi standards are maintained by the IEEE 802.11 working group. While each standard is assigned a unique letter code, major standard "generations" are also numbered chronologically [16]. Newer standards offer several advantages over older ones, e.g., higher maximum link rates, broader frequency band support, and reduced latency. In IoT environments, consistent and fast data transmission is essential for maintaining smooth interconnectivity. Furthermore, the ability to operate across a wider range of frequency bands enhances network flexibility.

The supported Wi-Fi standards of a device represent a qualitative rather than a quantitative characteristic. Therefore, to calculate a score for this characteristic, we first compute individual scores for each Wi-Fi standard in use (S_{WFS}) using the following formula (see Table 4):

$$S_{WFS} = CW_{WFS} + FREQ_{WFS} + DARA_{WFS},$$

where S_{WFS} is a score value of Wi-Fi standard WFS, CW_{WFS} and $FREQ_{WFS}$ are numbers of supported frequencies and channel widths by Wi-Fi standard WFS, respectively, and $DARA_{WFS}$ is a data rate of Wi-Fi standard WFS. All values are min-max normalized.

Table 4: Wi-Fi Standards scores.

№	Standard	Denotation*	Channel Width (MHz)	Frequency (GHz)	Data Rate (mbps)	Score
1	802.11	Wi-Fi 0	20	2.4	2	0.00
2	802.11b	Wi-Fi 1	20	2.4	11	0.00
3	802.11a	Wi-Fi 2	20	54	6	0.00
4	802.11g	Wi-Fi 3	20	2.4	6	0.00
5	802.11n	Wi-Fi 4	20/40	2.4/5	72	0.26
6	802.11ac	Wi-Fi 5	20/40/80	5	433	0.27
7	802.11ah	HaLow**	1/2/4/8/16	0.9	347	0.67
8	802.11ax	Wi-Fi 6	20/40/80/160	2.4/5/6	574	0.72
9	802.11be	Wi-Fi 7	20/40/80/160/320	2.4/5/6	1376	1.00

* Wi-Fi generations only received numerical nomenclature starting with Wi-Fi 4, others generations were numbered retroactively.
** During calculation, HaLow gets 1 additional score point to emphasize its low power consumption provided specifically for resource-constrained IoT devices.

B Appendix: Vendor Trust Calculation

Trust is a subjective characteristic of a device and can be influenced by various factors, such as prior positive or negative experiences. Therefore, we combine user preferences with a quantitative assessment of the vendors.

We calculate a vendor trust value for the device (VT) using the following formula:

$$VT = VT_{IoTCat} + VT_{user},$$

where VT_{IoTCat} is the weight value assigned to the vendor by *IoTCat*, and VT_{user} is the weight value that reflects the user's preference.

To determine VT_{IoTCat}, we analyzed multiple reports from the Statista.com portal on the most widely owned device brands across various countries (e.g., the USA, UK, India, China, Germany, and Italy). Based on this analysis, we identified 14 vendors with the highest overall popularity. These vendors were then clustered into three groups using the k-means algorithm, with each cluster assigned a specific weight value (see Table 5). Accordingly, if a device's vendor appears in this list, its VT_{IoTCat} is set to the corresponding weight value. Devices from vendors not included in the list receive $VT_{IoTCat} = 0$ by default.

It is important to note that for *IoTCat* we have compiled a generalized list of vendors and their corresponding weights. However, this list can be adjusted on demand to reflect specific market conditions and contextual factors such as country, application domain, or device type, as trust in vendors may vary in different environments.

In determining the value of VT_{user}, all vendors are initially assigned $VT_{user} = 0$. However, if the IoT system includes devices from multiple vendors, the user is required to indicate their preferred vendor, which is then assigned $VT_{user} = 1$.

Table 5: IoT device vendors and assigned weights.

№	Vendor Name	Weight
1	Samsung Electronics Co.	3
2	Philips	3
3	Google LLC	3
4	Robert Bosch GmbH	3
5	TP-LINK Technologies CO.	2
6	Amazon Technologies Inc.	2
7	Xiaomi Tech	2
8	Ring	2
9	Hive	2
10	LG Group	1
11	Siemens AG	1
12	Huawei Technologies Co.	1
13	D-Link	1
14	Espressif Inc.	1

References

1. Aksoy, A., Gunes, M.H.: Automated IoT Device Identification using Network Traffic. In: ICC 2019 - 2019 IEEE International Conference on Communications (ICC). p. 1–7. IEEE, Shanghai, China (2019). https://doi.org/10.1109/ICC.2019.8761559
2. Ali, Z., Hussain, F., Ghazanfar, S., Husnain, M., Zahid, S., Shah, G.A.: A Generic Machine Learning Approach for IoT Device Identification. In: 2021 International Conference on Cyber Warfare and Security (ICCWS). p. 118–123. IEEE, Islamabad, Pakistan (Nov 2021). https://doi.org/10.1109/ICCWS53234.2021.9702983
3. Bao, J., Hamdaoui, B., Wong, W.K.: IoT Device Type Identification Using Hybrid Deep Learning Approach for Increased IoT Security. In: 2020 International

Wireless Communications and Mobile Computing (IWCMC). p. 565–570. IEEE, Limassol, Cyprus (2020). https://doi.org/10.1109/IWCMC48107.2020.9148110
4. Bezawada, B., Bachani, M., Peterson, J., Shirazi, H., Ray, I., Ray, I.: Behavioral Fingerprinting of IoT Devices. In: Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security. p. 41–50. ACM, Toronto Canada (Jan 2018). https://doi.org/10.1145/3266444.3266452
5. Brenner, R., Leithäuser, M., Jänich, S., Pöhls, H.C.: Updatefähigkeit als konstruktionsanforderung. Recht Digital (RDi) pp. 252 –264 (2024), https://beck-online.beck.de/Bcid/Y-300-Z-RDI-B-2024-S-252-N-1
6. Chen, Q., Song, Y., Jennings, B., Zhang, F., Xiao, B., Gao, S.: IoT-ID: Robust IoT Device Identification Based on Feature Drift Adaptation. In: 2021 IEEE Global Communications Conference (GLOBECOM). p. 1–6. IEEE, Madrid, Spain (2021). https://doi.org/10.1109/GLOBECOM46510.2021.9685693
7. Chowdhury, R.R., Idris, A.C., Abas, P.E.: A Deep Learning Approach for Classifying Network Connected IoT Devices Using Communication Traffic Characteristics. Journal of Network and Systems Management **31**(1), 26 (2023). https://doi.org/10.1007/s10922-022-09716-x
8. Dominik, O., Miljenko, M., Marin, V.: Categorizing IoT services according to security risks, vol. 382, p. 154–166. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-78459-1_11
9. European Commission: What constitutes the 'latest version of the firmware' under the regulation (eu) 2019/424? (2025), https://energy-efficient-products.ec.europa.eu/faqs-0/what-constitutes-latest-version-firmware-under-regulation-eu-2019424_en
10. Fan, L., Zhang, S., Wu, Y., Wang, Z., Duan, C., Li, J., Yang, J.: An IoT Device Identification Method based on Semi-supervised Learning. In: 2020 16th International Conference on Network and Service Management (CNSM). p. 1–7. IEEE, Izmir, Turkey (Nov 2020). https://doi.org/10.23919/CNSM50824.2020.9269044
11. Geloczi, E., Klement, F., Struck, P., Katzenbeisser, S.: SoK: Delegated Security in the Internet of Things. Future Internet **17**(5), 202 (2025). https://doi.org/10.3390/fi17050202
12. Gigli, M., Koo, S.: Internet of Things: Services and Applications Categorization. Advances in Internet of Things **1**(2), 27–31 (Jul 2011). https://doi.org/10.4236/ait.2011.12004
13. Gupta, B., Quamara, M.: An overview of internet of things (iot): Architectural aspects, challenges, and protocols. Concurrency and Computation: Practice and Experience **32**(21), e4946 (2020). https://doi.org/10.1002/cpe.4946
14. Hadzovic, S., Mrdovic, S., Radonjic, M.: Identification of IoT Actors. Sensors **21**(6), 2093 (Mar 2021). https://doi.org/10.3390/s21062093
15. Hamad, S.A., Zhang, W.E., Sheng, Q.Z., Nepal, S.: 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). p. 103–111. IEEE, Rotorua, New Zealand (2019). https://doi.org/10.1109/TrustCom/BigDataSE.2019.00023
16. Institute of Electrical and Electronics Engineers: IEEE 802.11, The Working Group Setting the Standards for Wireless LANs (2023), https://www.ieee802.org/11/
17. Khalil, U., Ahmad, A., Abdel-Aty, A.H., Elhoseny, M., El-Soud, M.W.A., Zeshan, F.: Identification of trusted IoT devices for secure delegation. Computers & Electrical Engineering **90**, 106988 (2021). https://doi.org/10.1016/j.compeleceng.2021.106988

18. Kotak, J., Elovici, Y.: IoT device identification based on network communication analysis using deep learning. Journal of Ambient Intelligence and Humanized Computing **14**(7), 9113–9129 (2023). https://doi.org/10.1007/s12652-022-04415-6
19. Le, F., Ortiz, J., Verma, D., Kandlur, D.: Policy-Based Identification of IoT Devices' Vendor and Type by DNS Traffic Analysis, pp. 180–201. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-17277-0_10
20. Liu, X., Abdelhakim, M., Krishnamurthy, P., Tipper, D.: Identifying Malicious Nodes in Multihop IoT Networks Using Diversity and Unsupervised Learning. In: 2018 IEEE International Conference on Communications (ICC). p. 1–6. IEEE, Kansas City, MO (2018). https://doi.org/10.1109/ICC.2018.8422484
21. Liu, Y., Wang, J., Li, J., Niu, S., Song, H.: Class-Incremental Learning for Wireless Device Identification in IoT. IEEE Internet of Things Journal **8**(23), 17227–17235 (2021). https://doi.org/10.1109/JIOT.2021.3078407
22. Liu, Y., Wang, J., Li, J., Song, H., Yang, T., Niu, S., Ming, Z.: Zero-Bias Deep Learning for Accurate Identification of Internet of Things (IoT) Devices. IEEE Internet of Things Journal **8**(4), 2627–2634 (Feb 2021). https://doi.org/10.1109/JIOT.2020.3018677
23. Marchal, S., Miettinen, M., Nguyen, T.D., Sadeghi, A.R., Asokan, N.: AuDI: Toward Autonomous IoT Device-Type Identification Using Periodic Communication. IEEE Journal on Selected Areas in Communications **37**(6), 1402–1412 (2019). https://doi.org/10.1109/JSAC.2019.2904364
24. Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J.D., Ochoa, M., Tippenhauer, N.O., Elovici, Y.: ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis. In: Proceedings of the Symposium on Applied Computing. p. 506–509. SAC '17, Association for Computing Machinery, New York, NY, USA (Apr 2017). https://doi.org/10.1145/3019612.3019878
25. Mexis, N., Anagnostopoulos, N.A., Chen, S., Bambach, J., Arul, T., Katzenbeisser, S.: A Lightweight Architecture for Hardware-Based Security in the Emerging Era of Systems of Systems. ACM JETC **17**(3), 1–25 (Jun 2021). https://doi.org/10.1145/3458824
26. Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A.R., Tarkoma, S.: IoT sentinel: Automated device-type identification for security enforcement in IoT. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). pp. 2177–2184. IEEE (2017)
27. Mössinger, M., Petschkuhn, B., Bauer, J., Staudemeyer, R.C., Wójcik, M., Pöhls, H.C.: Towards quantifying the cost of a secure IoT: Overhead and energy consumption of ECC signatures on an ARM-based device. In: Proc. of The 5th workshop on IoT-SoS: Internet of Things Smart Objects and Services (WOWMOM SOS-IOT 2016). IEEE (July 2016). https://doi.org/http://dx.doi.org/10.1109/WoWMoM.2016.7523559, http://henrich.poehls.com/papers/2016_Moessinger_et_al-Towards_quantifying_the_cost_of_a_secure_IoT.pdf
28. Parzen, E.: On estimation of a probability density function and mode. The Annals of Mathematical Statistics **33**(3), 1065–1076 (Sep 1962)
29. Pereira, F., Correia, R., Pinho, P., Lopes, S.I., Carvalho, N.B.: Challenges in resource-constrained iot devices: Energy and communication as critical success factors for future iot deployment. Sensors **20**(22) (2020). https://doi.org/10.3390/s20226420
30. Rosenblatt, M.: Remarks on Some Nonparametric Estimates of a Density Function. The Annals of Mathematical Statistics **27**(3), 832–837 (1956). https://doi.org/10.1214/aoms/1177728190

31. Salman, O., Elhajj, I.H., Chehab, A., Kayssi, A.: A machine learning based framework for IoT device identification and abnormal traffic detection. Transactions on Emerging Telecommunications Technologies **33**(3), e3743 (2022). https://doi.org/https://doi.org/10.1002/ett.3743, e3743 ETT-19-0273.R1
32. Sawadogo, F., Violos, J., Hameed, A., Leivadeas, A.: An Unsupervised Machine Learning Approach for IoT Device Categorization. In: 2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom). p. 25–30. IEEE, Athens, Greece (Sep 2022). https://doi.org/10.1109/MeditCom55741.2022.9928766
33. Scheidt, N., Adda, M.: Identification of IoT Devices for Forensic Investigation. In: 2020 IEEE 10th International Conference on Intelligent Systems (IS). p. 165–170. IEEE, Varna, Bulgaria (2020). https://doi.org/10.1109/IS48319.2020.9200150
34. Shostack, A.: Experiences Threat Modeling at Microsoft. In: MODSECMoDELS (2008), https://api.semanticscholar.org/CorpusID:2508643
35. Silverman, B.W.: Density Estimation for Statistics and Data Analysis. CRC Press, London (1986)
36. Spielvogel, K., Pöhls, H.C., Posegga, J.: TLS beyond the broker: Enforcing fine-grained security and trust in publish/subscribe environments for IoT. In: 17th Int. Workshop on Security and Trust Management (STM). LNCS, vol. 13075, pp. 145–162. Springer (2021)
37. Tasmota Community: Tasmota Documentation (2025), https://tasmota.github.io/docs/, accessed: 2025-04-03
38. Thom, N., Thom, J., Charyyev, B., Hand, E., Sengupta, S.: FlexHash - Hybrid Locality Sensitive Hashing for IoT Device Identification. In: 2024 IEEE 21st Consumer Communications & Networking Conference (CCNC). p. 368–371. IEEE, Las Vegas, NV, USA (Jan 2024). https://doi.org/10.1109/CCNC51664.2024.10454657
39. Tragos, E., Bernal Bernabe, J., Staudemeyer, R., Hernández-Ramos, J., Fragkiadakis, A., Skarmeta, A., Nati, M., Gluhak, A.: Trusted IoT in the complex landscape of governance, security, privacy, availability and safety, p. 30. River Publishers (2016). https://doi.org/10.1201/9781003337966-6
40. Vailshery, L.S.: Number of IoT connected devices worldwide 2019-2023, with forecasts to 2030 (2023), https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/
41. Wang, B., Li, M., Jin, X., Guo, C.: A Reliable IoT Edge Computing Trust Management Mechanism for Smart Cities. IEEE Access **8**, 46373–46399 (2020). https://doi.org/10.1109/ACCESS.2020.2979022
42. Wang, B., Kang, H., Sun, G., Li, J.: Efficient traffic-based IoT device identification using a feature selection approach with Lévy flight-based sine chaotic sub-swarm binary honey badger algorithm. Applied Soft Computing **155**, 111455 (2024). https://doi.org/10.1016/j.asoc.2024.111455
43. Yin, F., Yang, L., Wang, Y., Dai, J.: IoT ETEI: End-to-End IoT Device Identification Method. In: 2021 IEEE Conference on Dependable and Secure Computing (DSC). p. 1–8. IEEE, Aizuwakamatsu, Fukushima, Japan (Jan 2021). https://doi.org/10.1109/DSC49826.2021.9346251