Highlights

Characterizing Time-Critical Internet of Things

Sebastian Leclerc, Alessio Bucaioni, Mohammad Ashjaei

- Systematic review of 38 studies on timing in IoT selected from 1,176 publications.
- Defined key timing concepts in IoT: predictability, time-criticality.
- Extracted 113 timing metrics, categorized by architecture layers and types.
- Expert survey complements literature on timing challenges in IoT systems.

Characterizing Time-Critical Internet of Things

Sebastian Leclerc[a,*], Alessio Bucaioni[b], Mohammad Ashjaei[a]

[a] *Division of Networked and Embedded Systems, Mälardalen University, Universitetsplan 1, 722 20 Västerås, Sweden*
[b] *Division of Product Realisation, Mälardalen University, Universitetsplan 1, 722 20 Västerås, Sweden*

Abstract

The Internet of Things (IoT) is increasingly being adopted in diverse domains, many of which require strict timing constraints and predictable behavior. Despite the growing importance of timing characteristics in IoT applications, current approaches to address timing requirements are often fragmented, context-specific, and lack a unified understanding. Consequently, addressing timing aspects in IoT remains largely ad hoc and dependent on individual applications, making it challenging to generalize findings or systematically apply established solutions. The goal of this study is to provide a comprehensive understanding of how timing is defined, characterized, and measured within the IoT community. We conducted this study through a systematic and structured mix methods research approach. First, we performed a systematic review of the literature, extracting and analyzing information from 38 primary studies, selected from a rigorous process involving 1176 studies. Second, to complement the literature findings, we conducted an expert survey involving 28 respondents from academia and industry, representing a variety of roles with specialized expertise in IoT systems and timing-related issues. We identified two primary characterizations of timing within the IoT: time-criticality and predictability. Additionally, we collected and categorized 113 distinct timing metrics from literature into commonly found layers of an IoT system. The majority of the surveyed practitioners and researchers (75%) agree with our categorization and consider this research useful and relevant (71.5%). We believe that our study provides practitioners and researchers with insights into timing characteristics and metrics in IoT applications, toward the ultimate goal of standardization.

Keywords: Systematic Literature Review, Expert Survey, Internet of Things, Time-Critical Systems, Predictability, Timing Characteristics, Timing Metrics

*Corresponding author.
Email addresses: sebastian.leclerc@mdu.se (Sebastian Leclerc), alessio.bucaioni@mdu.se (Alessio Bucaioni), mohammad.ashjaei@mdu.se (Mohammad Ashjaei)

1. Introduction

The Internet of Things (IoT) is experiencing rapid adoption across diverse sectors, driven by the proliferation of connected devices. According to IoT Analytics[1], approximately 16.6 billion IoT devices were connected globally in 2023, with projections indicating continued growth. Emerging IoT applications increasingly demand strict timing constraints, especially within domains such as disaster monitoring, emergency-stop mechanisms, smart healthcare, and transportation [1, 2, 3]. These domains require systems with Real-Time (RT) capabilities, characterized by on-time response times and deterministic behavior, to effectively manage critical events. In safety-critical contexts such as industrial control and healthcare monitoring, delays or missed deadlines can have severe consequences [4, 5].

Timing challenges in IoT are compounded by its multi-layered architecture, where timing characteristics differ significantly across components such as edge devices and cloud infrastructures. Despite the recognized importance of timing, the IoT community currently lacks a standardized approach for addressing timing requirements [6, 7]. Existing practices often remain fragmented, ad hoc, and application-specific, leading to reduced reproducibility, scalability, and consistency in evaluation. Recently, a special issue of Institute of Electrical and Electronics Engineers (IEEE) IoT Magazine on Internet of Time-critical Things [8] highlighted this complexity by examining various timing considerations, including network performance, scheduling algorithms, and methodologies for timing evaluation. Additionally, within the RT community, predictability is widely considered essential in safety-critical systems, where system behavior must be analyzable to ensure timing guarantees in advance [9].

Given the above promises, this study aims to provide a comprehensive understanding of how timing is defined, characterized, and measured within the IoT community. Specifically, we address the following Research Questions (RQs):

- RQ1: How is the concept of timing characterized in IoT?
- RQ2: How are timing characteristics measured in IoT systems?

We employed a systematic mixed-methods research approach. Initially, we conducted a systematic literature review, analyzing 38 primary studies selected from an initial pool of 1176 studies using clearly defined inclusion and exclusion criteria. To complement and validate these findings, we conducted an expert survey involving 28 respondents from academia and industry. Respondents included researchers, architects, developers, and consultants with specialized expertise in IoT and timing-related topics. Our analysis identified two primary categories of timing within IoT systems: *time-criticality* and *predictability*. Additionally, we collected and classified 113 distinct timing metrics from the literature, highlighting substantial variability based on application contexts and specific domains. Most surveyed experts (75%) agreed with our categorization and found

[1] `https://iot-analytics.com/number-connected-iot-devices`

this research useful and relevant (71.5%). To enable independent replication and transparency, we provide a complete replication package[2] containing our selection process, extracted data, and survey responses. We believe our study offers valuable insights into timing characteristics and metrics for IoT applications, representing a foundational step towards standardized timing metrics — a contribution considered strongly relevant by 57.1% of the respondents.

The rest of this review is organized as follows. Section 2 presents a brief background on IoT, RT systems and analysis, and communication networks. This is followed by Section 3 which discusses the research method. Then, Section 4 presents the extracted and synthesized definitions for time-criticality and predictability within IoT. After that, Section 5 examines the key timing measurements and metrics extracted from the studies. Section 6 presents the perspective of the practitioners — examined through the survey — on this research results and their relevance. In addition, it discusses other complementary insights such as measurement techniques and challenges. Section 7 compares this review with related work. Finally, Section 8 concludes the paper.

2. Background

This section provides foundational concepts essential for understanding time-critical IoT performance. Section 2.1 offers a high-level overview of IoT, illustrated by a detailed Wireless Sensor Network (WSN) scenario, and presents our adopted IoT reference architecture. Section 2.2 introduces fundamental RT concepts, including key terminology and analysis methods related to timing behavior in wireless RT systems.

2.1. Internet of Things Fundamentals

IoT is rooted in the idea of ubiquitous computing and can be traced back to the late 1980s [6]. As a continuously evolving paradigm, IoT encompasses a wide range of distributed, heterogeneous, and interconnected systems. Typically, the systems consist of small embedded hardware platforms equipped with sensors and actuators, integrated software, and various technologies that exchange data over various networks to produce value for an individual or organizations in different domains [6, 10].

An illustrative example of an IoT system is a WSN for environmental monitoring, as seen in Fig. 1. Nodes wirelessly relay data to central sink nodes, which may process it locally or forward it to higher-level computing layers such as Edge, Fog, or Cloud infrastructure [11]. The wireless link between nodes and the sink can use technologies like Wi-Fi, Bluetooth Low Energy (BLE), Visible Light Communication (VLC), Time Slotted Channel Hopping (TSCH), or cellular networks. Messages are typically formatted using lightweight IoT protocols such as request-response-based Constrained Application Protocol (CoAP) or

[2] github.com/SebastianLeclerc/time-critical-iot

publish-subscribe-based Message Queuing Telemetry Transport (MQTT), optimized for Size, Weight, Power, and Cost-Constrained (SWaP-C) IoT devices in unreliable wireless environments. Beyond the sink node, data typically travels through wired networks, possibly through local Edge or Fog infrastructures, before reaching Cloud infrastructures via the internet.

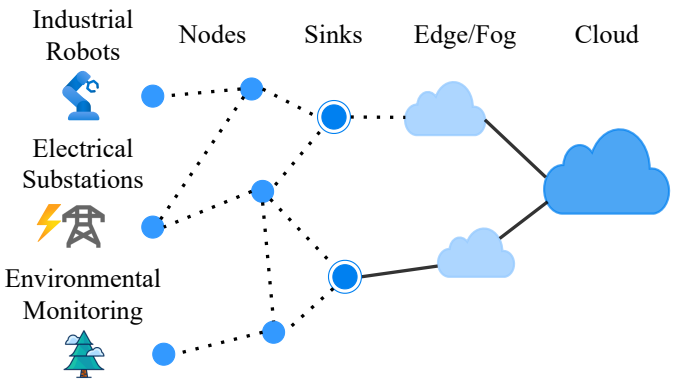

Figure 1: An overview of the components in a WSN for different application domains.

Due to the significant diversity of IoT systems, numerous reference architectures have been proposed, each targeting specific application domains. For instance, Al-Fuqaha et al. surveyed the existing IoT literature in [11] and discussed key architectural patterns and system components. Among these, they describe a basic three-layer model composed of Application, Network, and Perception layers. In a complementary approach, Yi et al. adopted a three-layer architecture comprising Device, Edge/Fog, and Cloud layers [12] — focusing on computational offloading and latency reduction. In our study, we have integrated these two widely adopted reference architectures into a unified five-layer model, consisting of Perception, Network, Edge/Fog, Cloud, and Application layers.

2.2. Real-time Fundamentals and Timeliness

Strict timing requirements are particularly critical in IoT applications within industrial, transportation, and safety-critical domains, including emergency services. Many systems in these domains can be classified as RT. In this subsection, we briefly introduce core concepts from RT theory, as well as with mechanisms and components that influence timing performance in IoT systems.

Using the WSN example described earlier, we can map timing requirements to standard RT categories [9]: *hard*, *firm*, and *soft* RT. In a *hard* RT system, missing deadlines can cause catastrophic outcomes, thus requiring strict timing guarantees. Such systems are common in safety-critical domains, such as a WSN monitoring industrial robots in automated manufacturing. *firm* RT systems

tolerate occasional missed deadlines, but any results produced after the deadline lose their usefulness. An example is a WSN in electrical substations for RT load balancing. *soft* RT systems tolerate deadline violations, which only degrade performance. A practical example is a WSN used for environmental monitoring, such as tracking temperature, humidity, and air quality in a forest for long-term analysis.

Tasks in RT systems are typically classified based on their activation patterns as *periodic*, *aperiodic*, or *sporadic* [9]. Periodic tasks are activated at regular intervals, while aperiodic tasks activate unpredictably. Sporadic tasks are a subset of aperiodic tasks with a defined minimum inter-arrival time. Time-critical events typically involve high-priority aperiodic or sporadic tasks, triggered by external stimuli such as sensor measurements exceeding thresholds. These events often originate from peripheral devices or software signals that generate Interrupt Requests (IRQs), handled by the Operating System (OS) through the Interrupt Service Routine (ISR). Real-Time Operating Systems (RTOSs) support such time-sensitive processing through task scheduling, enabling preemption of lower-priority tasks. In distributed systems such as a WSN, meeting timing constraints is more difficult due to additional complexities like tight clock synchronization and unpredictability of wireless communication.

Timing performance in wireless communication is mainly influenced by Layers 1 and 2 (L1, L2) of the Open Systems Interconnection (OSI) model [13], [14]. L1 is responsible for signal modulation, encoding, and transmission strength. L2 manages Medium Access Control (MAC) mechanisms, regulating access to a shared communication medium. For example, Wi-Fi employs Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA), whereas TSCH combines CSMA/CA, Time Division Multiple Access (TDMA), and channel hopping. While these techniques can improve determinism and reduce collisions, wireless communication remains vulnerable to unpredictable delays from multi-path fading, interference, and jamming — especially in dense environments. Consequently, ensuring communication reliability is vital for meeting stringent timing requirements in time-critical IoT systems. Wireless reliability can be defined as the Bit Error Rate (BER) and Packet Delivery Ratio (PDR) for a receiver decoding a signal with a specific Signal-to-Interference-plus-Noise-Ratio (SINR) in time [15].

Response-Time Analysis (RTA) includes methods for verifying whether a system can meet its timing constraints under worst-case conditions [9], [16]. This involves evaluating the worst-case timing behavior, which depends on the system layer and specific use case.

Various timing metrics are used to analyze RT systems from different perspectives. At the Perception layer, metrics typically address task execution in embedded and RT operating environments. Core metrics are Worst-Case Execution Time (WCET) — the maximum uninterrupted execution time — and Worst-Case Response-Time (WCRT), which accounts for interruptions, blocking, and scheduling delays. At the Network layer, analysis shifts to communication timing rather than computation. Metrics emphasize latency components such as queuing, transmission, and propagation. Common examples include

Round-Trip Time (RTT), unidirectional delay, and End-to-End (E2E) latency — the total time for data to travel from source to destination.

3. Research Methodology

This study was conducted following Kitchenham's guidelines for secondary studies in software engineering [17]. To mitigate the limitations of a single-method approach and reduce validity threats from the lack of expert evaluation, we followed Molléri et al.'s guidelines for software engineering questionnaires and included an expert survey as an additional validation step [18]. Our method comprised three phases: *planning*, *conducting*, and *reporting*. In the planning phase, we defined the RQs and developed the research protocol, which structured all subsequent steps. The conducting phase followed this protocol and involved (i) identifying and screening relevant studies, (ii) defining an extraction form, and (iii) extracting and synthesizing data. The reporting phase addressed potential threats to validity and mitigation strategies to ensure reliability and reproducibility. All study details are documented in this paper. To facilitate independent verification and replication, we provide a replication package[2], including search and selection data, extracted and synthesized data, and survey responses.

3.1. Definition of Research Goal and Questions

Following the Goal-Question-Metric (GQM) approach [19], we defined the research goal, which is presented in Table 1, and then refined the goal in the RQs, which was already introduced in Section 1.

Table 1: Research goal expressed using the GQM perspectives.

Purpose	Identify, and classify
Issue	definitions, measurements, research methods, and application domains
Object	of time-critical and predictable IoT
Viewpoint	from the point of view of researchers and practitioners.

3.2. Screening Process

A summary of the automatic search and selection is presented in Fig. 2, following the PRISMA 2020 guidelines [20]. We performed automated searches in three academic databases: ACM Digital Library[3], SCOPUS[4], and Web of

[3] dl.acm.org
[4] scopus.com

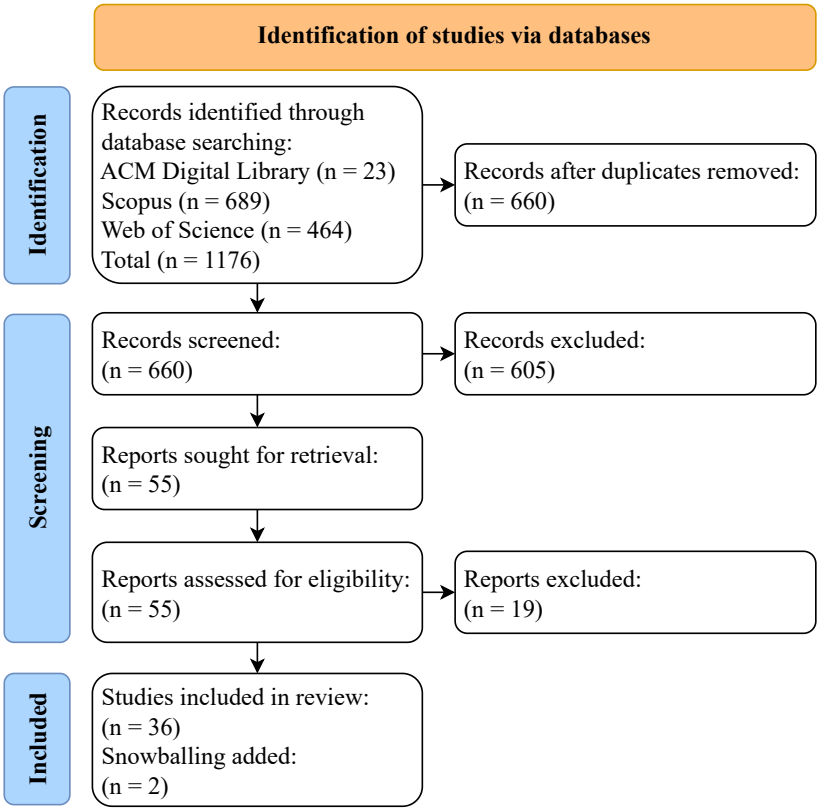

Figure 2: Overview of the search and selection process.

Science (WoS)[5], chosen for their strong support of systematic studies in computer science and software engineering [21]. To ensure broad coverage aligned with our research goals and RQs, we designed a flexible search string that captured terminology variations (e.g., treating "time critical," "time-critical," and "time?critical" equivalently). We also used "predicta*" to target relevant studies while excluding broader "predict*" results, which mainly returned unrelated Artificial Intelligence (AI)/Machine Learning (ML) papers. The final search string was:

("time-critical" OR predicta*)
AND (iot OR "Internet of Thing*")

[5] webofscience.com/wos

We kept the search string minimal to gather a broad range of studies for manual filtering, reducing construct validity threats discussed later in this section. The automated search was conducted in September 2024, covering studies from 2019-01-01 onward. It initially retrieved 1176 studies. After excluding non-research papers, summaries, and duplicates, we refined the set to 660 studies. We followed the selection process by Ali and Petersen, which emphasizes inclusion and exclusion criteria [22]. As detailed in Table 2, we applied three inclusion and ten exclusion criteria. Only studies meeting all inclusion and none of the exclusion criteria were retained for further analysis.

Table 2: Inclusion and exclusion criteria.

E/I	Criteria	Explanation
Incl.	Software or Network	Paper is published in Software or Network Engineering.
	IoT timing aspects	Paper is fully or partially focused on IoT timing aspects.
	Time bounded	Is bounded, targeting soft/hard RT system.
Excl.	Non-English	The paper is not written in English.
	Non-peer-reviewed	The paper is not peer-reviewed.
	Too few pages	The full-text paper was ≤ 4 pages.
	Not available	We could not access the full-text paper.
	Survey or review	The paper presents a survey or review.
	No results	The paper presents an idea without results.
	Non-IoT	Paper only mentioned IoT but targets another domain.
	No analysis	The paper does not provide sufficient timing analysis.
	AI or ML-based	Paper's main proposed solution was based on AI or ML techniques.
	Hardware-based	Paper's main proposed solution was based in hardware.

The selection process was conducted iteratively, refining the dataset based on title, abstract, and keywords. This resulted in 55 studies, all of which underwent full-text screening. During screening, 19 studies were excluded due to: (i) insufficient information on time-criticality or predictability (e.g., only brief keyword mentions), or (ii) exclusion criteria identified upon full-text review (e.g., AI/ML-based focus or idea papers). This yielded 36 studies. Through forward and backward snowballing [23], two additional studies were identified [24, 16], resulting in a final set of 38 primary studies. The full list of included papers is provided in Table 3, where the IDs correspond to those used in the replication package[2] for improved traceability.

3.3. Extraction Form and Data Extraction

We developed a well-defined classification framework (Table 4) to systematically extract and categorize relevant information from the primary studies. The framework consists of two facets, one for each RQ. RQ1 includes two clusters, definitions and characteristics. The definitions consist textual descriptions of time-criticality and predictability in the given study. After collecting all the definitions for each term, we synthesized a set of high-level textual characteristics of each term. RQ2 focuses on timing metrics, which are extracted as a set of key timing metrics in textual form from the respective studies. We adopted

Table 3: List of peer-reviewed primary studies.

ID	Title (truncated)	Author	Year	Ref.
89	A Comprehensive Worst Case Bounds Analysis of IEEE 802.15.7	Kurunathan et al.	2021	[25]
177	A lightweight messaging engine for decentralized data processing in the Internet of Things	Del Gaudio et al.	2020	[26]
242	A Markov chain model for IEEE 802.15.4 in time critical wireless sensor networks	Yousefi et al.	2022	[27]
572	A software architecture for the industrial internet of things - a conceptual model	Ungurean	2020	[4]
336	An Effective Communication Prototype for Time-Critical IIoT Manufacturing Factories	Kiangala et al.	2021	[5]
640	Analysis of joint scheduling and power control for predictable URLLC in industrial wireless networks	Wang et al.	2019	[15]
302	Autonomous Flow-Based TSCH Scheduling for Heterogeneous Traffic Patterns	Urke et al.	2023	[28]
504	Bounded transmission latency in real-time edge computing: a scheduling analysis	Fara et al.	2023	[29]
463	Blue Danube: A Large-Scale, End-to-End Synchronous, Distributed Data Stream Processing Architecture	Michael et al.	2022	[30]
185	Cellular network-based IIoT architecture for time-critical control tasks of building automation	Li et al.	2024	[31]
340	Coexistence Analysis of Multiple Asynchronous IEEE 802.15.4 TSCH-Based Networks	Veisi et al.	2020	[32]
132	Collaborative Task Scheduling for IoT-Assisted Edge Computing	Kim et al.	2020	[33]
28	Computation Resource Allocation for Heterogeneous Time-Critical IoT Services in MEC	Liu et al.	2020	[34]
278	Delay Analysis in IoT Sensor Networks	Althoubi et al.	2021	[35]
22	Distributed Graph Routing and Scheduling for Industrial Wireless Sensor-Actuator Networks	Shi et al.	2019	[36]
33	Dynamic Bandwidth Slicing for Time-Critical IoT Data Streams in the Edge-Cloud Continuum	Habeeb et al.	2022	[37]
156	Dynamic decision support for resource offloading in heterogeneous Internet of Things environments	Jaddoa et al.	2020	[38]
1195	End-to-end response time analysis for RT-MQTT: Trajectory approach versus holistic approach	Shahri et al.	2023	[16]
3	Ensuring End-to-End Dependability Requirements in Cloud-based Bluetooth Low Energy Applications	Spörk et al.	2021	[2]
833	Fog network task scheduling for IoT applications	Zhang et al.	2020	[39]
1199	Improving the timeliness of Bluetooth Low Energy in dynamic RF environments	Spörk et al.	2020	[24]
317	iSotEE: A Hypervisor Middleware for IoT-Enabled Resource-Constrained Reliable Systems	Li et al.	2022	[40]
324	Interrupting Real-Time IoT Tasks: How Bad Can It Be to Connect Your Critical Embedded System to the Internet?	Behnke et al.	2020	[41]
81	Jamming-Aware Simultaneous Multi-Channel Decisions for Opportunistic Access in Delay-Critical IoT	Salameh et al.	2022	[42]
274	Metascheduling Using Discrete Particle Swarm Optimization for Fault Tolerance in Time-Triggered IoT-WSN	Baniabdelghany et al.	2023	[43]
5	Non-Intrusive Distributed Tracing of Wireless IoT Devices with the FlockLab 2 Testbed	Trüb et al.	2021	[44]
870	On the Performance of Commodity Hardware for Low Latency and Low Jitter Packet Processing	Stylianopoulos et al.	2020	[45]
497	On the QNX IPC: Assessing Predictability for Local and Distributed Real-Time Systems	Becker et al.	2023	[46]
25	Optimized Scheduling for Time-Critical Industrial IoT	Brun-Laguna et al.	2019	[47]
1017	PrioMQTT: A prioritized version of the MQTT protocol	Patti et al.	2024	[48]
253	Prioritized Clock Synchronization for Event Critical Applications in Wireless IoT Networks	Bhandari et al.	2019	[14]
322	REA-6TiSCH: Reliable Emergency-Aware Communication Scheme for 6TiSCH Networks	Farag et al.	2021	[49]
1069	RT-BLE: Real-time Multi-Connection Scheduling for Bluetooth Low Energy	Li et al.	2023	[50]
295	Status Prediction and Data Aggregation for AoI-Oriented Short-Packet Transmission in Industrial IoT	Xiong et al.	2023	[51]
366	Timing Comparison of the Real-Time Operating Systems for Small Microcontrollers	Ungurean	2020	[52]
1134	Towards Multi-channel GTS Allocation in Visible Light Communication	Kurunathan et al.	2023	[53]
255	Transmission Early-Stopping Scheme for Anti-Jamming Over Delay-Sensitive IoT Applications	Halloush	2019	[54]
53	Unobtrusive, Accurate, and Live Measurements of Network Latency and Reliability for Time-Critical Internet of Things	Bhimavarapu et al.	2022	[55]

Grounded Theory as a general framework for qualitative research [56]. Specifically, we applied open coding during the data extraction phase to systematically identify and categorize key concepts.

Table 4: Form used during data extraction.

Facet	Cluster	Description
RQ1	Definitions	Definition of timing as given in the study.
	Characteristics	Synthesized characteristics based on the extracted definitions.
RQ2	Timing metrics	Metric as given in the study, categorized by the layer.

3.4. Data Analysis and Synthesis

During the synthesis phase, we utilized axial coding to establish relationships between the initial codes, forming coherent higher-level categories. We followed the recommendations described in [57] throughout the analysis and synthesis process. To ensure a structured and comprehensive approach, we combined content analysis [58] and narrative synthesis [59]. Initially, content analysis was applied to examine each study individually (vertical analysis), allowing us to identify patterns, themes, and relationships within the data. These findings were systematically classified and categorized based on the predefined structure of our data collection form. Following this, narrative synthesis was employed to synthesize and interpret the findings across all studies (horizontal analysis). This approach facilitated the development of a thematic summary.

3.5. Survey

We conducted the survey following the process proposed by Molléri et al. [18]. In the subject selection, we identified the target audience tailored terminology accordingly. The group included (i) academics with expertise in IoT and RT systems, and (ii) industry professionals working with IoT applications. We then designed the sampling plan, determining the group size and participant selection method. The questionnaire design involved defining: (i) the questions, (ii) their types, (iii) sequencing, and (iv) overall layout. Implemented via Google Forms[6], the survey primarily used close-ended (evaluation-type) questions. Following established guidelines [60], questions were organized into five categories addressing topics such as understanding of timing in IoT, timing measurements, and the relevance of our research. To minimize bias and maximize the reliability, we used Likert-scale, multiple-choice, and free-text questions, enabling both quantitative and qualitative analysis. Sensitive responses in free-text boxes were carefully reviewed and redacted if needed to ensure privacy and ethical compliance. All responses were anonymous.

To validate the questionnaire, we conducted a pilot survey with nine respondents from the target audience. This helped assess question clarity and survey relevance. Based on feedback, we refined the questionnaire — for example, by adding an explanation of timing measurements and removing a duplicated question. We then distributed the survey via e-mail to 57 potential respondents and collected 28 answers.

Experts were selected through purposive sampling from our professional network, including academic colleagues from other institutions, practitioners in IoT system development, and close collaborators in the research project. This ensured that participants had relevant domain knowledge. For transparency and to support reproducibility, we provide a full replication package[2] containing the anonymized survey responses.

3.6. Threats to validity

A common threat to construct validity in systematic studies is the potential omission of relevant work. To mitigate this, we utilized three scientific databases and applied snowballing. Although databases such as IEEE Xplore Digital Library were not directly queried, many of their papers were included via indexing in SCOPUS and WoS, helping ensure broad coverage.

Another construct validity threat is the formulation of the search string. As noted earlier, terms like "predict*" predominantly appear in AI/ML contexts, which are beyond this review's scope. The study was also limited to English-language publications, consistent with the *de-facto* norm in computer science research. To ensure data consistency, we employed descriptive statistics, cross-checked the extraction form, and performed sanity checks to validate the accuracy and reliability of the extracted data.

[6] google.com/forms/about

Other threats to internal and construct validity were mitigated through multiple strategies, including the active involvement of all authors in key methodological decisions [61, 23, 57]. This included: (i) defining clear RQs, (ii) developing a comprehensive search strategy, (iii) establishing well-defined inclusion and exclusion criteria, (iv) designing a structured data extraction form, and (v) applying Grounded Theory [56] to reduce synthesis bias.

To enhance conclusion validity, we provide a replication package[2] enabling independent analysis and reproduction of our process. Finally, conducted a representative survey using academics and professionals with experience in RT and IoT.

4. Definitions of time-criticality and predictability (RQ1)

In this section, we address the first research question regarding how timing characteristics are addressed within the IoT community. Specifically, we define two key concepts — time-criticality and predictability — by presenting existing definitions derived from an analysis of 38 studies. Readers may refer to Table 5 for an overview of the studies from which one or more characteristics were extracted. This is followed by a detailed discussion of their key characteristics, providing a comprehensive overview of how these concepts are defined and applied in the literature.

Table 5: Primary studies supporting the definitions.

Definition	Characteristic	Primary Study
Time-critical	Determinism	[44],[55],[4],[48],[53],[27],[14],[5],[51],[41],[47],[49],[33],[37],[25]
	Reliability	[2],[55],[4],[50],[54],[31],[5],[47],[49],[37]
	Low Latency	[44],[2],[55],[4],[48],[50],[31],[14],[34],[30],[5],[51],[47],[42],[26],[33],[37],[38]
	Strict Deadline	[4],[54],[27],[41],[49],[42],[33]
Predictability	Min. External	[49],[42],[24]
	Predictive	[5],[51],[24]
	Scheduling	[36],[4],[15],[40],[46],[41],[43],[28],[33],[52],[29]
	Analyzable	[36],[2],[55],[45],[39],[4],[15],[40],[46],[14],[41],[43],[28],[32],[33],[35],[52],[29],[16]

4.1. Time-critical definition

From the studies, we extracted 24 definitions of time-critical IoT applications. Due to space constraints, the full definitions are omitted here, but are available in the provided replication package[2]. Collectively, these definitions emphasize four key characteristics: Strict Deadlines, Low Latency, Reliability, and Determinism. Different studies prioritize different aspects. For example, studies in [4, 33] emphasize strict deadlines, defining time validity windows after which data becomes useless or invalid. In contrast, studies such as [48, 26] focus on low latency. Based on the collected definitions in this study, we observe an interconnection among strict deadlines, low latency, and determinism. Although different terms are used, the essential components in all three definitions are similar and focus on the guarantee of the upper bound of latency in either a process or transmission. Based on the collected definitions, we define time-critical as follows:

Time-critical. A time-critical IoT system must demonstrate robust and reliable performance to consistently meet stringent temporal requirements via means of predictability. These requirements are application-specific and typically entail low, bounded, and deterministic latency, ensuring that time-critical events are processed safely and effectively from E2E.

Hereafter, we provide a description of each of the four characteristics of time-critical based on the collected information from the studies.

Strict Deadlines. This characteristic focuses on the use of deadlines hence predetermined time windows within which data must be processed or transmitted, after which the data is considered invalid or useless. Borrowing from the RT theory, strict deadlines may be classified as hard, firm, or soft, depending on the severity of the consequences when these deadlines are missed. In Industrial Internet of Things (IIoT), strict deadlines are crucial for tasks that require RT control and monitoring, often managed by an RTOS with preemptive scheduling to ensure that time-critical tasks are prioritized [4, 41]. Edge computing is also leveraged to meet strict deadlines by reducing network congestion and latency [33]. Mechanisms like reneging, where deadline-exceeded queued packets are dropped, are employed to maintain system performance [27]. Additionally, for ensuring aperiodic time-critical traffic such as emergency alarms meets its deadline with sufficient reliability, a strategy is to sacrifice a portion of the regular traffic [49].

Low Latency. Time-critical systems often demand strict, low, and explicitly defined latency constraints (e.g., "must be less than X ms"). Centralized processing can introduce significant communication delays, making it unsuitable for time-critical applications [26]. Many IoT and stream processing platforms have latencies exceeding 10 seconds — far too high for time-critical tasks [30]. In contrast, safety systems in Internet of Vehicles (IoV) require average and maximum latencies below 15 ms, as demonstrated by benchmarks like Linear Road. Communication protocols are also critical. Transmission Control Protocol (TCP)-based MQTT is suboptimal for IIoT scenarios demanding low latency and prioritized messaging [48], whereas User Datagram Protocol (UDP) is preferred for its connectionless nature, lack of congestion control, and Application-layer retransmission management. Some applications must also balance low latency with energy efficiency [38]. In others, processing is shifted to the edge or device level to meet strict timing needs [34, 33]. Reverse task offloading — delegating tasks directly to IoT devices within defined intervals — is an emerging trend [33]. Processing closer to the data source, especially for small payloads (e.g., ¡250 B vs. Ethernet's Maximum Transmission Unit (MTU) of 1500 B), reduces both Age of Information (AoI) and E2E delays [51]. Edge computing is increasingly adopted in industrial time-critical systems, combining local responsiveness with cloud analytics while maintaining Quality of Service (QoS) under varying network and workload conditions [37].

Reliability. This characteristic focuses on reliability and safety requirements, emphasizing that any downtime or failure to meet temporal requirements can have severe safety, production, or financial consequences. In the context of IIoT, time-criticality involves systems that require hard, firm, or soft RT capabilities, low latency, and high reliability to ensure safe and efficient monitoring and control of industrial processes [4]. For example, local process control in smart buildings demands tasks with periods shorter than 1 second, low latency, and high network reliability—requirements that current IoT architectures and networks often struggle to meet [31]. Cellular networks with dedicated network slices or Wi-Fi can potentially support such IIoT applications. Additionally, time-critical applications requiring stringent E2E latency and reliability bounds increasingly utilize BLE and TSCH [2, 47].

Determinism. This characteristic focuses on deterministic scheduling and RT guarantees, which ensure communication and processing in a bounded time where sometimes the exact time that the process or communication occurs can be determined in advance. In IIoT (and Operational Technology (OT)) environments, time-critical applications demand deterministic, low-latency communication responses [5]. Enablers include Time Sensitive Networking (TSN) mechanisms, such as prioritization, resource reservation, guard bands, and preemption, to support time-critical information driving physical processes, as well as zero-loss redundancy protocols (e.g., Parallel Redundancy Protocol (PRP) and High-Availability Seamless Redundancy Protocol (HSR)) for applications where downtime is unacceptable. Non-intrusive debugging is crucial for maintaining the timing behavior of distributed systems, particularly for time-critical components like wireless radio operations, network protocols, and synchronized transmissions with high precision requirements [44]. Techniques such as code instrumentation (e.g., *printf()*) can alter execution timing, thus necessitating careful adjustments in time-critical code. Other promising approaches include the use of VLC in WSN, where the MAC supports contention-free communication via Guaranteed Time Slot (GTS) in a periodic synchronized super-frame structure, although this requires rigorous worst-case condition modeling [25]. Additionally, WSN designed for RT applications require deterministic data reception within strict deadlines, often employing strategies like reneging to ensure data validity [27].

We also performed an orthogonal analysis to identify potential correlations between the identified characteristics of time-critical applications and the layers of our IoT model, as described in Section 2. The results, illustrated by the bubble chart in Fig. 3, reveal a clear correlation between the Network layer and the characteristics of Determinism, Low Latency, and Reliability. By examining the bubble chart by IoT layer, we observe that the Network layer exhibits a significantly higher number of occurrences (46) related to the identified characteristics. In contrast, all other layers show around 20 occurrences each, with the notable exception of the Cloud element, which registers only 6 occurrences. The

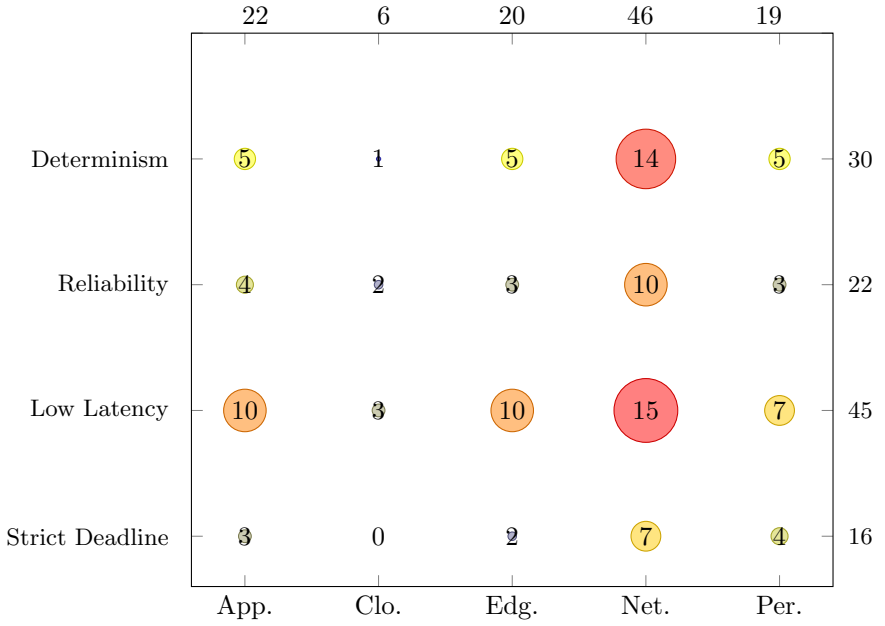

Figure 3: Orthogonal analysis of the correlation between IoT elements and time-critical definitions.

low occurrences in the Cloud layer could be explained by this layer's implied longer latency, which is not suitable for time-critical applications. Hence we see an emphasis for the edge in the extracted characteristics and in the bubble chart. Similarly, by examining the bubble chart segmented by the identified characteristics of time-critical applications, we observe that two characteristics dominate in frequency: Low Latency with 45 occurrences, and Determinism with 30 occurrences.

4.2. Predictability definition

From the studies, we extracted 24 definitions of predictable IoT applications. Due to space constraints, the full definitions are omitted here but are available in the provided replication package[2]. Collectively, these definitions emphasize four key characteristics: Analyzable Timing, Scheduling, Predictive Behavior, and Minimizing External Unpredictabilities. Different studies prioritize different aspects. For example, the study [35] focuses on interactive and latency-sensitive applications for which they propose using queuing analysis to predict the E2E latency in closed forms. Other studies, such as [49], discussed how aperiodic traffic, like an emergency alarm, is unpredictable and impractical to pre-assign resources for. According to the collected definitions, we observe that there is an interconnection between analyzable timing and scheduling definitions. In both terms, the main component is to guarantee a certain latency

in a process or transmission. However, other characteristics target different behavior of the IoT system as predictability. Based on the collected definitions, we define predictability as follows:

Predicatbility. A predictable IoT system employs rigorous scheduling, resource allocation, and temporal analysis to ensure that all timing constraints are consistently met as the application-specific tasks evolve in time from E2E. By leveraging strategies such as resource reservation and RT prediction of future system states, these systems may mitigate external sources of unpredictability, including those arising from the wireless radio environment.

Hereafter, we provide a description of each of the four characteristics of predictability.

Analyzable Timing. This characteristic encompasses requirements for bounded latencies, consideration of WCET, predictable jitter, and overall performance that can be analyzed deterministically (e.g., via some form of WCRT). This is crucial for interactive and latency-sensitive applications, where predictable performance in terms of tail latency, jitter, or response time is essential [35]. Analyzability can sometimes be achieved through queuing analysis using closed-form formulas. In industrial event-based applications, ultra-low and predictable latency (e.g., less than 10 ms) along with minimal jitter is often required, although this is challenging due to the influence of multiple components, such as queue behavior and how the OS and hardware manage thread isolation [45]. Moreover, the integration of Information Technology (IT) into OT systems frequently does not fully consider RT requirements, leading to a pessimistic view of bounded network latency [16]. For instance, the TSCH channel hopping mechanism mitigates the impact of multi-path fading and external interference, and its use of TDMA avoids intra-TSCH collisions, thereby providing efficient, reliable, and predictable communication that is well-suited for IIoT environments [32]. Additionally, performance predictability with minimal RT overhead is often essential for critical (I)IoT applications, and the use of an RTOS can enable the predictable performance needed for hard RT applications. Finally, SWaP-C constrained devices, being relatively simple, can offer predictable on-chip memory access latencies, further supporting the goal of analyzable timing.

Scheduling. This characteristics focuses on ensuring predictability by coordinating system resources, such as processors, memory, and network links, to meet RT requirements. The behavior of the system's scheduler is crucial and must be analyzable to determine timing bounds [46]. Techniques such as temporal isolation, which guarantees dedicated resource access, and priority inheritance, designed to mitigate the priority-inversion phenomenon, are commonly employed to enhance predictability. In industrial contexts, scheduling unpredictable heterogeneous traffic, often affected by the radio environment, remains a significant challenge [28]. More robust L2 scheduling mechanisms, when combined with effective power control, are essential for achieving predictable per-packet

communication reliability, a cornerstone of Ultra Reliable Low Latency Communication (URLLC) in IIoT applications [15]. Moreover, leveraging fog/edge computing and decoupling software components with and without hard RT requirements across different processors enables more predictable latencies, which is vital for monitoring and controlling time-critical operations in IIoT [4].

Predictive Behavior. This characteristic refers to the run-time prediction of system states, events, or updates to enhance overall efficiency and reliability. For instance, IIoT data, such as location and velocity, often exhibit time correlations that can be leveraged to reduce the AoI [51]. In this context, authors have proposed strategies such as immediately transmitting a predicted status update when it matches the source data or aggregating multiple predicted status updates into a single packet, with careful optimization to balance prediction accuracy against transmission error probabilities. Additionally, downtime and delays caused by unpredictable faults can be mitigated through redundancy mechanisms, robust protocols, and cloud-based analytics, all of which support predictive maintenance [5]. Another approach involves forecasting the number of future connection events necessary for successful transmissions by filtering data through an observation window, thereby further enhancing system performance [24].

Minimizing External Unpredictabilities. This characteristic focuses on mitigating environmental factors that can cause unpredictable behavior, such as radio interference, jamming, or contention. For example, in WSNs, reliability can be compromised by reactive jammers that intelligently predict and interfere with communication [42]. Aperiodic traffic, such as emergency alarms, is inherently unpredictable and challenging to allocate resources for in advance [49]. Radio interference, particularly in BLE applications, may lead to prolonged and unpredictable transmission delays [24]. Moreover, accurately forecasting future noise in the radio environment is often not feasible, as many existing models assume ideal conditions or rely on atypical data. Although BLE employs mechanisms such as Adaptive Frequency Hopping (AFH) and autonomous retransmissions, these do not guarantee an upper bound on latency. Additionally, variations in AFH implementations may result in inconsistent behavior and inaccurate predictions of the Bluetooth frequencies in use.

We also performed an orthogonal analysis to identify potential correlations between the identified characteristics of predictability and the layers of our IoT model, as described in Section 2. The results, illustrated by the bubble charts in Fig. 4, reveal a clear correlation between the Network layer and the characteristics of Analyzable Timing and Scheduling. By examining the bubble chart by IoT layer, we observe that the Network layer exhibits a significantly higher number of occurrences (32) related to these characteristics. In contrast, all other layers show around 10 occurrences each, with the notable exception of the Cloud element, which registers only 2 occurrences. This pattern is very similar to the one observed for time-critical applications, suggesting that the Cloud may be

the least explored or significant layer, while the Network remains the primary focus of current research. This similarity is not surprising, as predictability is inherently more stringent than time-criticality. Moreover, when segmenting the bubble chart by the identified characteristics of predictability, we observe that two characteristics dominate in frequency: Analyzable Timing, with 36 occurrences, and Scheduling, with 23 occurrences.

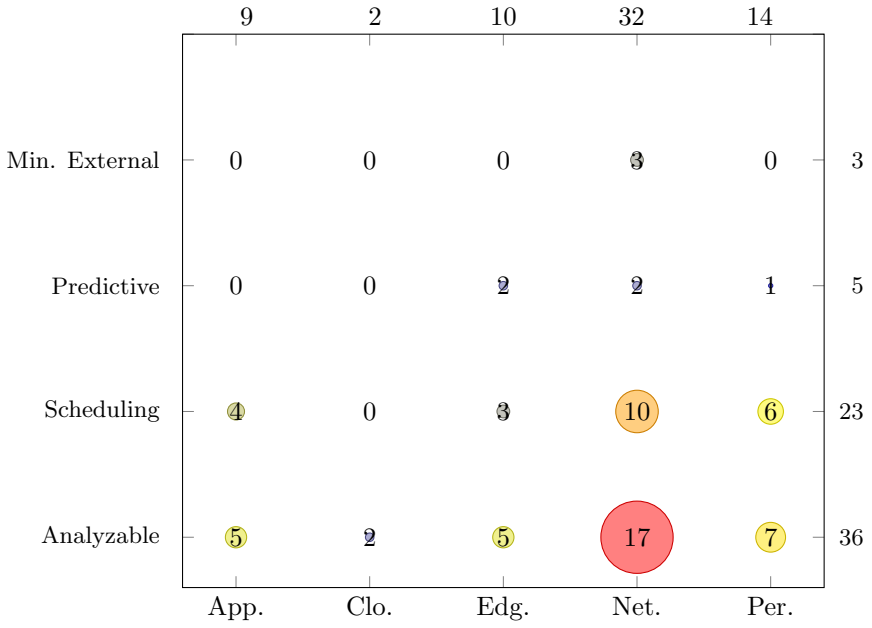

Figure 4: Orthogonal analysis of the correlation between IoT elements and predictable definitions.

Highlights - RQ1 Timing definitions and characteristics

- Time-Criticality is defined by four characteristics: Strict Deadlines, Low Latency, Reliability, Determinism. Hard, firm, or soft deadlines are supported by edge computing and task offloading. Applications often require latency < 15 ms. UDP is preferred over TCP. Techniques like BLE, TSCH, and network slicing are explored to increase reliability. Determinism is enabled through TSN, VLC, preemption, guard bands, and redundancy protocols. Network layer and Low Latency are most closely associated with time-criticality in IoT systems.
- Predictability is defined by 4 key characteristics: Analyzable Timing, Scheduling, Predictive Behavior, Minimizing External Unpredictabilities. WCRT, WCET, and queuing models are used for latency/jitter analysis. Predictable resource access via temporal isolation, priority inheritance, and L2 scheduling. Forecasting system states/updates (e.g., AoI, status prediction, fault tolerance). Tackling interference, jamming, and aperiodic traffic is explored. Network layer and Analyzable timing are most closely associated with predictability in IoT systems.

5. Timing metrics (RQ2)

This section addresses the second research question by examining the key metrics used to capture timing aspects in IoT systems. In total, we identified and collected 113 timing metrics from the literature. Most of these are highly specialized and situational, tailored to specific systems, environments, conditions, and assumptions. For example, some studies propose unique metrics, such as [24], which introduces the metric n_{CE} to estimate the future number of connection events required for successful transmission; [34], which defines "sojourn time" as the duration an object remains in a node — a metric closely related to deadline adherence. Other studies integrate established metrics into novel composite Key Performance Indicators (KPIs). For instance, [31] merges network transmission latency with IoT controller processing time into a "time cost" metric. Additionally, many proposals explore a wide range of configurations, including variations in task sets, priorities, network sizes, data rates, and radio environments, to encompass diverse operational scenarios. These factors complicated the development of a unified timing metrics framework capable of serving as both a broad descriptive and prescriptive instrument. Our attempt to provide such a framework is presented in Table 6.

For the sake of brevity and clarity, Table 6 displays some of the collected timing metrics, categorizing them first according to the IoT layer and then by category type, which includes Latency, Data rate, Reliability, Scheduling, and Resource. It is worth remarking that the category types were identified by applying open coding on the selected studies and further verified with our survey respondents (of whom 75% agreed with our categorization). The interested reader is encouraged to consult the cited works and our replication package[2] for additional metrics and detailed measurement information. In addition, Table 6 also includes a brief description of each measurement along with references to the corresponding studies. When a study presented a timing metric addressing multiple IoT layers, we included the reference for each layer individually. For instance, [41] examines the impact of network interrupts on tasks across the Application, Network, and Perception layers.

We group all measurements and metrics into the above mentioned five timing categories. Latency covers time-based performance measures in delivery or processing such as E2E delay, WCRT of messages, execution time, and jitter. Data rate focuses on how much data the system can handle or transfer over time such as network or task throughput, and network utilization. Reliability addresses success or failure metrics in network packet delivery such as PDR, Packet Reception Ratio (PRR), and various probabilities of successful transmission. Scheduling captures the scheduler's behavior in terms of metrics such as Deadline Satisfaction Ratio (DSR), more complex measurements of adaptability or scalability, but also temporal metrics derived from some RTA method, etc. Finally, Resource encompasses overhead such as Central Processing Unit (CPU), memory usage, queue length, capacity, and other system-level factors.

The observant reader may notice that there are timing metrics that might not seem to fit a certain IoT layer, e.g., typical network measurements under the

Table 6: Summary of key measurements taken in each IoT layer.

IoT Layer	Type	Description	Paper ID
Application	Latency	Overhead	[26]
		End-to-End	[30], [50]
		Round-Trip Time	[48], [31], [46]
		Queue	[48]
		Execution Time	[50]
		Network Interrupt Effect on Tasks	[41]
	Data rate	Network: Throughput, Capability, Utilization	[30], [37]
		Task Throughput	[33]
	Reliability	Message Loss Ratio, Packet Loss Rate	[48], [31]
	Scheduling	Task Distribution	[33]
		Adaptability to Requirement Changes, Connection Re-Schedule Delay	[50]
		Response-Time Analysis	[46], [33], [16]
	Resource	Connection Capacity	[50]
Cloud	Latency	End-to-End	[35]
		Jitter	[35]
		End-to-End Unidirectional	[2]
	Data rate	Network Utilization	[37]
	Reliability	End-to-End Necessary Transmissions	[2]
	Scheduling	Task Distribution	[38]
		Offloading Response-Time	[38]
Edge/Fog	Latency	End-to-End	[35], [5]
		Jitter	[35]
	Data rate	Network Utilization	[37]
		Task Throughput	[33]
	Reliability	Failure Recovery	[5]
	Scheduling	Task Distribution	[33], [38]
		Task Timeout Probability	[34]
		Offloading Response-Time	[38]
		End-to-End delay	[39]
		Response-Time Analysis	[29], [33]
Network	Latency	End-to-End	[35], [50], [5]
		Jitter	[35]
		End-to-End Unidirectional	[2], [55]
		Round-Trip Time	[31], [45], [46]
		Execution Time	[50]
		Worst-Case Bounds	[25], [53], [24]
		n-th Packet Delay	[27]
		Reneging Effect	[27]
		Possible Deadline Limits	[36]
		Latency Exceeded Data	[24]
		Network Interrupt Effect on Tasks	[41]
		Measurement Accuracy (Tightness)	[44]
	Data rate	Network: Throughput, Utilization	[45], [25], [49], [37]
		Task Throughput	[33]
		Maximal Measurement Event Rate	[44]
	Reliability	End-to-End Necessary Transmissions	[2]
		Failure Recovery	[5]
		Packet Delivery Ratio, Packet Loss Rate, Packet Reception Ratio	[27], [36], [49], [54], [31], [32]
		Collision-Free Transmissions	[32]
		Latency Bounds at Expected Reliability Levels	[55]
		Mean-Time Attempting to Transmit	[54]
		Channel Idle Probability	[42]
	Scheduling	Task Distribution	[33]
		Response-Time Analysis	[29], [46], [33]
		Adaptability to Requirement Changes, Connection Re-Schedule Delay	[50]
		Superframe Scalability	[25]
		Transmission Probability (with Required Signal-to-Interference-plus-Noise-Ratio)	[15]
		Pessimistic Bounds (via Key Performance Indicator)	[28]
		Convergence	[28]
		Node Join Time	[36]
		Generated Schedules Validity	[43]
		Medium Access Control Delay	[14]
		End-to-End in Ideal and Non-Ideal Radio Environment	[47], [49]
	Resource	Connection Capacity	[50]
Perception	Latency	Round-Trip Time	[31], [46]
		Network Interrupt Effect on Tasks	[41]
		Measurement Accuracy (Tightness)	[44]
		Context Switch, Execution time	[52]
	Data rate	Task Throughput	[33]
		Real-time Operating System's function throughput	[40]
		Maximal Measurement Event Rate	[44]
	Reliability	Packet Loss Rate, Packet Delivery Ratio	[31], [54]
		Mean-Time Attempting to Transmit	[54]
	Scheduling	Task Distribution	[33]
		Response-Time Analysis	[46], [33]

Application layer. The reason is that we have categorized the study belonging (in part, or fully) to the Application layer, and the authors have tested their application under various network scenarios.

5.1. Application layer metrics

Latency in the Application layer involves two primary performance concerns for time-critical IoT systems: application overhead-induced latency [26] and the execution time of the proposed application, such as the centralized RT BLE scheduler in [50]. The study in [26] evaluates a prototype of a conveyor belt system transporting items between production machines, focusing on Machine-to-Machine (M2M) communication with RT requirements. This layer also includes network-centric measurements evaluating application performance. For example, PrioMQTT, introduced in [48], assesses the impact of priority levels on MQTT message transmission by measuring average queue times under different priority conditions. In the Data rate metrics, task throughput measurements are considered from multiple perspectives, such as the throughput of IoT nodes and the Edge/Fog layer in [33]. Additionally, network-centric evaluations include the large-scale data stream processing performance in [30], which simulates a large intelligent transportation system, and network throughput (or utilization) assessments in both [30] and [37]. Reliability metrics play a crucial role in IoT applications. The Message Loss Ratio (MLR) has been analyzed in [48], measuring the ratio of lost messages to transmitted messages. Similarly, Packet Loss Rate (PLR) has been evaluated using ping measurements within a (4G) cellular IoT application in [31], applied to a real-world building automation scenario involving Heating, Ventilation, and Air Conditioning (HVAC) systems. For Scheduling metrics related to distributed Edge/Fog applications, task distribution has been measured in [33] under varying task arrival rates and different offloading strategies. Time-critical applications require robust RTA, which has been assessed through the DSR metric in [33], particularly concerning local and time-critical tasks. Additionally, traditional RTA approaches have been employed in [46] to evaluate the QNX RTOS distributed scheduling behavior, focusing on execution traces and WCRT. Study [16] compares various RTA approaches for RT-MQTT, an Application-layer protocol responsible for transmitting MQTT messages across a network. Moreover, specialized performance evaluations of BLE scheduling adaptability have been conducted in [50], analyzing worst-case latency under deadline requirement changes and the delay incurred when modifying the wireless schedule. Finally, in the Resource category, connection capacity in BLE applications has been assessed for its ability to meet deadline requirements [50].

5.2. Cloud layer metrics

In Latency metrics, study [35] focuses on time-critical operations across the Cloud, Edge/Fog, and Network layers using queuing analysis. This study evaluates E2E latency through synthetic network traces and modeling, while also conducting jitter analysis under varying data rates. Similarly, study [2] examines

cloud-based BLE applications by measuring unidirectional E2E latency. This assessment involves evaluating transmission latency (Tx) from the BLE device within the Local Area Network (LAN) to the cloud server in the Wide Area Network (WAN) via different wireless communication technologies (Wi-Fi, cellular), as well as the reverse direction (Rx). For Data rate metrics, study [37] investigates network utilization over time in both the Edge/Fog and Cloud layers. This research explores the role of Software-Defined Networking (SDN)'s bandwidth slicing and (5G) cellular technology in supporting time-critical applications. The study is based on a simulated self-driving car scenario, emphasizing high-bandwidth, low-latency communication requirements. Regarding Reliability metrics, study [2] further examines the reliability of BLE communication between the cloud and BLE devices, analyzing the number of required transmissions based on the network path's transmission reliability. Finally, in Scheduling metrics, study [38] evaluates Cloud and Edge/Fog offloading by measuring an application's average response time, accounting for delays in uploading, processing, and downloading. Additionally, this study assesses the average task distribution across different layers under various offloading strategies.

5.3. Edge/Fog layer metrics

In the Edge/Fog layer, several performance measurements overlap with those discussed in the Application and Cloud layers. Therefore, this section focuses only on unique measurements. For Latency metrics, study [5] develops an IoT prototype incorporating industrial wired zero-loss redundancy protocols (PRP, High-Availability Seamless Redundancy Protocol (HSRP)), TSN, and edge computing. This study derives a formal E2E frame communication delay, accounting for transmission, propagation, and processing delays. Regarding Reliability metrics, the same study [5] evaluates network link failure recovery time using the aforementioned zero-loss redundancy protocols. For Scheduling metrics, study [34] investigates time-critical computational resource allocation in mobile Edge/Fog scenarios. This study assesses the probability of task timeouts across different configurations, including varying task sets and data rates, using different queuing models. Additionally, a distinct RTA approach is identified in the Edge/Fog layer. Study [29] develops an RT Edge/Fog system model that schedules transmission operations using a compatible RTA. The proposed RTA is experimentally evaluated based on system schedulability ratio, using randomly generated WCET tasks across different network scenarios varying in size and utilization. The evaluation is applied to modelled crowdsensing applications and networks of smart sensors, representing time-critical distributed data collection scenarios. Similarly, study [39] examines the scheduling of bursty and unpredictable tasks in IoT systems utilizing an Edge/Fog layer, measuring E2E task scheduling-induced delays for different task sets. Finally, another Scheduling metric already discussed in the Application layer, study [33], evaluates the scheduling DSR within this context.

5.4. Network layer metrics

In the Latency metrics, a variety of typical network performance metrics are considered. Study [50] provides a Cumulative Distribution Function (CDF) of E2E latency over BLE, while [31], [45], and [46] report various RTT distributions. Study [55] uses a precise industrial testbed to evaluate unidirectional latency over Wi-Fi 6 and 5G, aligning with [2, 44]. Latency bounds for VLC are analyzed in [25] and modeled further in [53]; worst-case BLE latency under dynamic conditions is examined in [24]. Specific behaviors are also studied: [27] measures n-th packet delay and analyzes the reneging effect in IEEE 802.15.4, while [36] compares a decentralized TSCH scheduler against Orchestra using latency CDF. Study [24] assesses latency-exceeded data in BLE under interference. In the Data rate metrics, studies [25] and [49] evaluate throughput for IEEE 802.15.7 and emergency TSCH applications, respectively. Study [45] analyzes the impact of a Virtual Network Function (VNF) on throughput in a software switch. In the Reliability metrics, we mainly find PDR assessments. Study [27] uses periodic transmissions, while [36] provides a CDF across network sizes. Studies [49] and [32] evaluate alarm message delivery and PRR in coexisting networks. Study [32] also estimates collision-free transmissions in concurrent TSCH networks. Jamming resilience is analyzed in [42], measuring idle channel probability across protocols. Study [55] shows 5G achieves lower latency than Wi-Fi 6 at $\geq$ 99.9% reliability. This evaluation was conducted on a real-world industrial control testbed, where a Programmable Logic Controller (PLC) communicated with a motor controller over wireless PROFINET. In the Scheduling metrics, various RTA approaches are used to determine performance bounds, measured through schedulability ratio in [29], WCRT and execution traces [46], and DSR in [33]. The evaluation in [46] targets a simulated and hardware-based autonomous driving scenario on a Raspberry Pi platform, focusing on detection and localization tasks. Study [28] computes KPIs from experimental data to estimate pessimistic performance bounds. Study [14] examines MAC delay scalability with prioritized nodes. TSCH is a recurring focus. Studies [47, 49] evaluate E2E upstream latency in ideal and non-ideal radio environments for emergency traffic. Study [25] assesses superframe scalability in VLC-based networks. Study [15] shows that integrating power control with scheduling improves successful transmission rates based on SINR thresholds. Scheduling convergence is addressed in [28], which analyzes PDR, duty cycles, and latency during network formation and topology changes. Study [36] compares join times across scheduling methods and network sizes. Study [43] introduces an offline metascheduler that maintains feasible schedules during single- and two-failure events, mitigating state space explosion.

5.5. Perception layer metrics

In the Perception layer, there is a significant overlap with previously discussed performance metrics; therefore, this section highlights only a few unique ones. For Latency metrics, study [44] proposes a non-intrusive method for tracing a distributed wireless IoT system. The method was evaluated on a real

testbed running a generic event-based application. To evaluate this approach, the accuracy (tightness) of its measurements is compared to ground-truth logic tracing, quantifying the magnitude of measurement errors. Another study investigates the timing behavior of RTOSs suitable for IoT through experiments on real controllers, assessing the delay and jitter associated with context switching triggered by various sources, such as events, semaphores, and mailboxes [52]. Additionally, the same study measures the execution time of key RTOS primitives, including semaphores. For Data rate metrics, study [44] further examines the maximal measurement event frequency of its proposed tracing system, determining how frequently a variable can be accessed or read without causing delays or overflow. Another approach investigated in the Perception layer is found in study [40], which proposes an RTOS hypervisor middleware for IoT, providing resource allocation to enhance system predictability. Various RTOS function throughputs are then measured using a benchmarking framework. For Reliability metrics, study [54] explores strategies to mitigate wireless jamming in time-critical IoT applications. The authors evaluate the packet success rate (or PDR) of their proposed wireless communication schemes under various hostile radio environments via MATLAB simulations. Additionally, they analyze the mean time required for the system to attempt a transmission, offering insights into the resilience of their approach.

Highlights - RQ2 Timing metrics

- 113 timing metrics were identified and categorized into Latency, Data rate, Reliability, Scheduling, and Resource, then grouped by IoT architecture layers.
- Latency and Scheduling metrics (e.g., E2E delay, RTT, DSR) are most associated with the Network and Edge/Fog layers.
- Resource and Reliability metrics (e.g., CPU usage, PDR, redundancy success) are found across all layers, with emphasis on system robustness.
- The Cloud layer shows fewer metrics overall, while the Application, Edge/Fog, and Perception layers present more context-specific measures.

6. Discussion

This section discusses our findings in conjunction with the survey results, providing the practitioners' perspective of our findings and complementary insights. Section 6.1 explores broader research trends in time-critical IoT, Section 6.2 reports on the respondents' opinion on our synthesized definitions and timing metrics, while Section 6.3 addresses general considerations and challenges in measuring time-critical IoT systems.

6.1. Relevance of the research topics

The relevance of this research is evident in the rising trend observed in our collected studies, as shown in the publication trends over the last five years[7] in Fig. 5. This observation is further supported by our survey respondents, 71.5% of whom strongly indicated that unifying time-critical IoT applications would be useful (See Q23 in Table 9). Moreover, Fig. 5 shows that most of the selected studies were published in 2020. Notably, only one study appeared at a workshop, while the rest were featured in venues such as the IEEE Internet of Things Journal and IEEE Access.

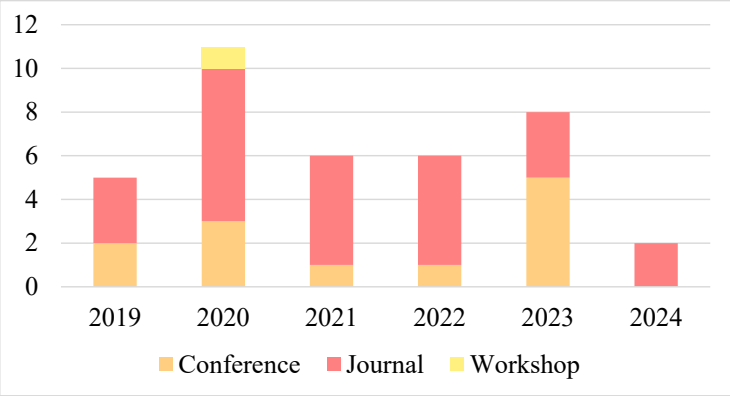

Figure 5: Distribution of primary studies by year and by type of publication.

To provide a broader perspective on research trends within time-critical IoT, we identified nine distinct solution and target domains, as illustrated in Fig. 6. The solution domains represent a given study's potential domains of utilization, whereas the target domains refer to the specific domains that were actually targeted in a given study. From both of these categories combined, the industrial sector accounts for the largest share (28.21%), followed by environmental monitoring (14.74%) and transportation (14.10%). A small portion of studies (4.49%) did not specify a domain. This distribution aligns with the expectation that time-critical behavior holds greater importance in sectors such as the industrial compared to typical consumer products.

6.2. Practitioners Perspective

The survey results provide practitioner perspectives that both validate and complement the literature findings. To support transparency and provide context for the survey analysis, Table 7 lists all survey questions along with their formats. While we do not elaborate on each response in the paper, we focus on the most meaningful results relevant to our contributions. We did not collect

[7]The exception being 2024, for which data was not collected for the entire year.

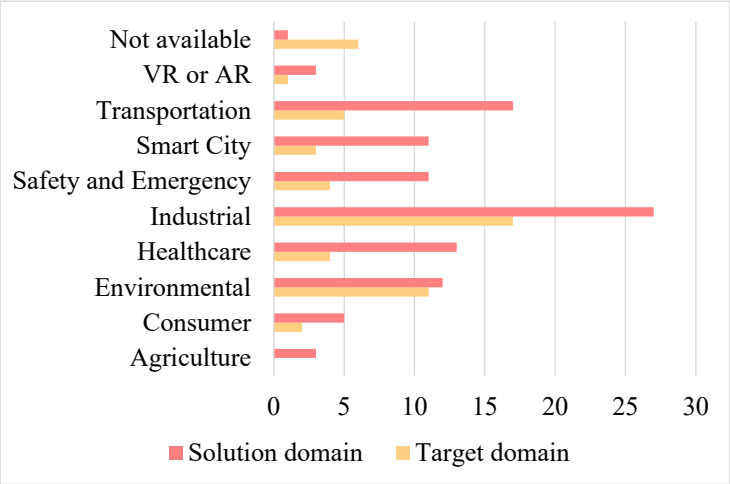

Figure 6: Identified solution domains and target domains from primary studies.

broader demographic data (e.g., age, gender, location), as it was not central to the research focus, we did capture respondents' roles and levels of experience. Most respondents identified as researchers (82.1%), with others identifying as developers (7.1%), developer–researchers (3.6%), a software architect (3.6%), or a researcher/consultant (3.6%). Notably, 78.6% reported having worked with time-critical IoT applications. Respondents also reported moderate-to-high familiarity with performance and timing evaluations as seen from Q3 in Table 9. These results suggest that the survey responses reflect informed perspectives grounded in practical experience with time-critical IoT systems. Tables 8 and 9 summarize the yes/no/partially and Likert-scale survey responses, respectively, while other survey results are available in the replication package[2].

From our synthesized definitions of time-criticality and predictability, we identified four key characteristics for each. Of the survey participants, 53.6% agreed with these, while 35.7% suggested minor changes as seen from Q5 in Table 8.

Feedback indicated that time-criticality and predictability sometimes overlap and may be merged in certain contexts. Respondents emphasized that definitions should be grounded in application-specific requirements. For example, one described time-criticality as a general term that "*could be refined into requirements*," while others linked it to QoS, Quality of Experience (QoE), or RT categories. The perspective varied by system layer — some considered the view from individual IoT nodes (Perception layer), others from an E2E standpoint.

Additional comments suggested that predictability could be seen as a means to achieve time-critical behavior. One noted, "*you cannot have complete predictability of the communication*," particularly in dynamic wireless environments. Another aligned predictability with Buttazzo's view [9] that "*the system*

Table 7: Survey questions and their types.

Q#	Question (truncated)	Type
Q1	What is your primary role in IoT?	Free-text
Q2	Have you worked with time-critical IoT applications?	Multiple choice
Q3	How familiar are you with performance/timing evaluations in IoT?	Likert (1–5)
Q4	How often are timing requirements defined in IoT applications?	Likert (1–5)
Q5	Our research identified two ways timing is addressed in IoT: Time-Critical and Predictability. Do you agree with this categorization?	Multiple choice
Q6	Q5 Follow-up: Specify which modifications or alternatives to the definitions above.	Free-text
Q7	How frequently do you encounter timing measurements in IoT applications?	Likert (1–5)
Q8	We categorized timing measurements based on these IoT layers: ... Are there additional layers where timing measurements are relevant?	Multiple choice
Q9	Q8 Follow-up: If yes, please specify additional layers.	Free-text
Q10	Which IoT layer do you believe requires the most stringent timing guarantees?	Multiple choice
Q11	We categorized timing measurements into: ... Are there any measurement categories that were not mentioned?	Multiple choice
Q12	Q11 Follow-up: If yes, please specify additional categories.	Free-text
Q13	Our findings indicate that the Network layer contains the most timing measurements, whereas the Cloud layer has the least. Do you agree?	Multiple choice
Q14	Q13 Follow-up: If not, please explain.	Free-text
Q15	What are the greatest challenges you face when measuring timing performance in IoT applications?	Multiple choice
Q16	Q15 Follow-up: If other, please specify the challenge(s).	Free-text
Q17	Do you believe timing aspects in IoT are underexplored?	Multiple choice
Q18	Q17 Follow-up: Please explain why or why not?	Free-text
Q19	Our findings suggest that timing measurements in IoT are often ad-hoc and case-specific ... do you agree with this observation?	Likert (1–5)
Q20	Would standardized timing metrics be beneficial for IoT applications?	Likert (1–5)
Q21	What are the biggest challenges in adopting standardized timing metrics for IoT applications?	Free-text
Q22	Beyond standardization, what other timing-related challenges or research gaps do you think should be addressed in IoT?	Free-text
Q23	How useful do you find this research on defining unified time-critical IoT applications?	Likert (1–5)
Q24	Would you be willing to engage with the team in an interview?	Multiple choice

Table 8: Summary of Yes, No, and Partially survey responses.

Question (truncated)	Yes	No	Partially
Q5: Agree: time-critical and predictive categorization?	15 (53.6%)	3 (10.7%)	10 (35.7%)
Q8: Are additional IoT layers relevant for timing?	9 (32.1%)	19 (67.9%)	-
Q11: Are there additional measurement categories?	7 (25%)	21 (75%)	-
Q13: Agree: Net. most, Clo. least timing measurements?	25 (89.3%)	3 (10.7%)	-
Q17: Are timing aspects underexplored in IoT?	16 (57.1%)	12 (42.9%)	-

Table 9: Summary of Likert-scale survey responses.

Question (truncated)	Mean ± SD	95% CI	% Agree (4–5)
Q3: Familiarity with performance/timing evaluations	3.64 ± 1.22	[3.17, 4.12]	71.4%
Q4: Frequency of defined timing requirements	2.89 ± 0.83	[2.57, 3.22]	21.5%
Q7: Frequency of encountering timing measurements	3.54 ± 0.79	[3.23, 3.84]	57.1%
Q19: Agreement on ad-hoc timing standards	3.96 ± 0.64	[3.72, 4.21]	78.6%
Q20: Usefulness of standardized metrics	3.82 ± 0.82	[3.50, 4.14]	57.1%
Q23: Usefulness of research findings	3.93 ± 0.90	[3.58, 4.28]	71.5%

should be able to predict the evolution of the tasks and guarantee in advance that all critical timing constraints will be met."

These responses reflect diverse interpretations and reinforce the need for unified definitions. Based on the feedback, we refined our definitions, presented in Sections 4.1 and 4.2.

To categorize timing metrics, we adopted a five-layered IoT model based on [11, 12] and classified all 38 reviewed studies accordingly. Survey responses show 67.9% agreed with the layer structure, while 32.1% proposed more granularity (e.g., device, OS, middleware, or OSI layers) as seen from Q8 in Table 8. Some noted confusion around the Perception layer, suggesting separation of sensing and actuating or adding a dedicated RT layer. Others questioned conceptual clarity, asking "*how does Application, Perception, and Cloud live on a single plane?*". This highlights that strict layering may not fully capture the interconnected nature of IoT systems.

Survey respondents were also asked to identify layers requiring the most stringent timing guarantees. Results: Application (46.4%), Cloud (10.7%), Edge/Fog (42.9%), Network (64.3%), and Perception (42.9%). These responses suggest a broad distribution of timing-critical points, with a lower emphasis on the Cloud. This aligns with findings from Table 6, where the Network layer has the most timing metrics, and the Cloud the fewest. This observation was validated by 89.3% of respondents (Q13 in Table 8), with comments such as "*the Cloud is a complex ecosystem*" and "*can only to a certain degree rely on network timing information*". Several noted that the Cloud must often reconstruct timing indirectly from other layers.

Our analysis revealed variability in how explicitly timing requirements are defined in IoT systems. Study [16] noted that integrating IT into OT often ignores RT requirements. Similarly, [31] argued that current smart building IIoT architectures are inadequate for time-critical tasks. One trade-off strategy is to relax energy efficiency constraints [38]. Survey results showed only 21.5% strongly agreed that timing requirements are explicitly defined during design or operation (Q4 in Table 9), indicating a need for stronger emphasis in future systems.

We also mapped reviewed studies to the lower OSI layers: L1 (16.98%), L2 (54.72%), L3 (22.64%), and L4 (5.66%). As expected, L2 had the strongest focus due to its influence on timing. However, as [28] notes, robust L2 scheduling can be undermined by upper-layer behavior. Environmental context was also raised,

with one respondent noting higher latency risks in signal-dense environments (e.g., offices) compared to open fields.

From the survey, 57.1% of respondents strongly indicated frequent use of timing measurements, as seen from Q7 in Table 9. Our review identified 113 distinct timing metrics, a subset of which is listed in Table 6. The variety and frequency of these metrics highlight the need for standardization to enable consistency across studies.

We grouped timing metrics into five categories: Latency, Data rate, Reliability, Scheduling, and Resource. This categorization was supported by 75% of respondents (Q11 in Table 8). The remainder suggested adding metrics for error handling, resilience, and precision. We argue that such metrics may fit under Reliability. One respondent noted the interdependence between Latency and Reliability, especially for hard deadlines. Another suggested a scalability category, which is indeed important in dynamic IoT systems.

On the use of RTT, one respondent recommended the TWAMP protocol[8] for accurate two-way measurements without clock synchronization.

Our findings indicate that timing metrics in IoT are often ad hoc and context-specific, with 78.6% agreeing on the lack of standardization (Q19 in Table 9). The diversity of metrics makes comparison across studies difficult. Still, 57.1% strongly agreed (Q20 in Table 9) that standardizing metrics — similar to those used in RT systems — would benefit IoT. While new metrics may be needed for niche cases, a shared baseline would support consistent evaluation.

Several challenges were raised by respondents. The most common was ecosystem heterogeneity — diverse technologies and architectures make unified timing guarantees difficult. Respondents also called for standardization of practices, especially around WCET, and better methodologies for evaluating timing under resource constraints. Some suggested applying hard RT methods to time-critical systems and using QoS techniques for soft RT.

Additional gaps include the lack of tools for timing analysis, integration challenges, and coordination issues between Edge/Fog and Cloud. Security and usability were also noted.

Although most respondents supported standardization, only 57.1% felt that timing is underexplored in IoT as seen in Q17 in Table 8. This suggests a research-industry gap: academia may emphasize theoretical timing models (e.g., TSN), while industry faces deployment challenges. Collaboration may be needed to bridge this divide.

Some respondents noted that not all IoT systems need strict timing; in many, reliability or availability is more important. However, as adoption grows, more applications will demand strict timing. The challenge remains to provide guarantees in large-scale, heterogeneous, and unpredictable systems.

Several respondents emphasized that E2E timing remains underexplored. Others pointed out that while timing may be well-studied in RT contexts, it remains insufficiently addressed in the specific landscape of IoT.

[8] `https://datatracker.ietf.org/doc/html/rfc5357`

6.3. Identifying, Capturing, and Presenting Timing Data

Depending on the application, we must first identify what exactly to measure. Some studies clearly specify the precise start and end points of their measurements ([45, 30, 47, 49]) or define exactly what is included in the measurement such as the response time in [38]. For instance, studies [47, 49] measure WSNs schedulers' E2E upstream latency from the moment sensor data (or an alarm) is generated until it arrives at the sink in different radio environments. Additional examples of such start and end point variations can be observed in Table 6, where the measured latency might be captured from an E2E perspective (which itself varies), as RTT, as one-way Tx/Rx latency, or upstream/downstream. While this level of granularity aids in comparing different proposals, it is not always clearly reported in every study. Moreover, depending on system constraints — such as the lack of access to low-level timing information — some level of measuring abstraction may be necessary, and certain in-depth measurements might be impractical.

Beyond addressing exactly what to measure, we must also consider appropriate methods of capturing the timing data in validating that the timing requirements are met. We have identified five high-level validation methods from the studies: Formal Analytical Approaches (21.82%), Comparative Performance Analysis (26.36%), Experimental Approaches (18.18%), System Design and Construction (15.45%), and Simulation or Emulation (18.18%). Note that the studies often included some form of comparative performance analysis to justify their contributions, thereby explaining their high frequency. Several studies also compare various validation methods to assess the tightness or accuracy of results. For instance, study [46] examines experimental and formal methods for evaluating WCRT, while study [16] contrasts an experimental approach with two formal ones. The latter shows that a holistic formal WCRT analysis (classic RTA of WCET task sets) is more pessimistic than a more complex trajectory-based WCRT (which follows packets backward in trajectory). When validating the timing requirements via an experimental testbed or deployment, the act of measuring a system can itself alter the system's timing behavior as discussed in [44, 55]. Examples of this include logging overhead consuming CPU (*printf()*) or network probes that consume bandwidth. Both studies offer extensive guidance on achieving tight, unobtrusive, and accurate measurements, discussing methods such as serial interface logging, buffered transfers, hardware-based logic tracing (e.g., via General-Purpose Input/Output (GPIO) pins), and dedicated debugging hardware. The two latter methods here represent the better options in general.

We also surveyed our respondents on the challenges of measuring timing performance through a multiple-choice question supplemented by an optional free-text response (Q15 and Q16 in Table 7). The majority (60.7%) identified high variability in wireless conditions and difficulties in synchronization across distributed nodes as the most significant challenges. Following this, 25% considered measurement overhead and limited access to low-level timing information to be the primary obstacles. Beyond these predefined categories, respondents also highlighted additional challenges such as the heterogeneity of IoT devices

crossing different (software, network, IoT) conceptual layers, which increases measurement complexity, and the application-dependent nature of timing evaluations, such as the possible wireless signal strengths in a particular environment. These observations highlight the complexity of ensuring that timing requirements are met within the broad and dynamic IoT ecosystem. Through this research, we aim to provide guidance on key considerations and resources, including where to measure, how to measure, and which metrics to use — ultimately contributing to the broader goal of standardization.

Once we know what and how to measure, we must now decide on an appropriate presentation of time. As previously discussed, in many time-critical IoT systems, we are primarily concerned with worst-case timing behavior or bounded latencies [27], which aligns with our definitions. Several studies report these bounds using tail Latency metrics — for instance, first or last percentiles (or more stringent thresholds, depending on reliability requirements) of tasks missing their deadlines [28, 35]. Consequently, the CDF is frequently used to depict this distribution, such as in [55, 48, 35], allowing an assessment of how reliably a given latency constraint is met. In this sense, latency and reliability function as two sides of the same coin. To ensure reliable findings, a statistically significant amount of empirical data should be collected, thereby revealing any temporal fluctuations [2].

7. Related work

To the best of our knowledge, no prior peer-reviewed study provides a holistic view of time-critical IoT. Existing research primarily focuses on specific components or isolated aspects.

Behnke and Austad offer a comprehensive review of RT performance in IIoT communication, identifying common use cases and corresponding RT requirements [62]. The use cases include autonomous vehicles, worker safety and hazard protection, and augmented reality. Their five-year study highlights challenges such as lack of standardization, wireless limitations, scalability-predictability trade-offs, and complexity in adopting technologies like 5G and TSN. Our work differs by taking a foundational step toward standardizing timing metrics — an identified gap in the field.

As discussed earlier, timing is closely linked to wireless reliability. Vlavianos et al. [63] assessed link quality metrics in IEEE 802.11a/g networks, showing that no single metric is sufficient. The evaluation was conducted through a measurement-based study on a physical testbed in an office environment, using multiple nodes under varying parameters and conditions. Their study found: Received Signal Strength Indicator (RSSI) is useful only at low data rates; SINR is accurate but hard to measure; PDR depends on transmission rate and packet size; and BER requires large samples and careful outlier handling. They concluded that combining metrics offers better link quality estimation.

Few studies examine time-criticality in relation to connectivity and cloud technologies. Perez-Ramirez et al. [1] explored new Wi-Fi features for Industry 4.0, proposing multi-frame scheduled MAC, packet redundancy via Multi-Link

Operation (MLO) in Wi-Fi 7, and time synchronization using Fine Timing Measurement (FTM), targeting use cases such as safety-critical wireless control (e.g., emergency stops) and autonomous mobile robots on the factory floor.

Shukla et al. [64] reviewed latency reduction techniques in IoT and cloud computing, using a Systematic Literature Review (SLR) process to identify time-critical applications and emergency response scenarios requiring low latency. Their work emphasizes fog computing techniques, including use cases such as secure social networks demanding ultra-low latency. Gowri et al. [65] proposed a Resource Allocation and Service Placement (RASP) strategy leveraging Reinforcement Learning (RL) and Energy-Efficient Computing (EEC) in Fog/Cloud settings. These works focus on improving timing predictability in specific system layers, complementing our broader metric-centered perspective.

Other studies address scheduling. Kharb and Singhrova [66] reviewed TSCH, focusing on scheduling algorithms, advertisement policies, and challenges in IIoT. Their work follows a SLR process and identifies real-world scenarios such as safety, control, and monitoring in industrial process automation, as well as in smart metering, body area networks, and home automation. Khajeh et al. [67] reviewed RT scheduling across diverse IoT domains such as healthcare, smart cities, and industrial systems. Using a SLR process, they discuss a range of application scenarios, including traffic monitoring with RT traffic light scheduling and smart building systems for energy consumption tracking. These works emphasize resource management in time-critical applications and complement our focus on timing analysis.

Mitra et al. [68] surveyed the design of time-critical systems, focusing on timing interference in single-core, multi-core, and distributed environments. They discussed how interference affects execution time bounds and presented modeling and mitigation strategies to ensure timing guarantees under variable workloads.

Finally, Soularidis et al. [69] reviewed time-critical IoT systems in mission-critical contexts, such as decision support for Search and Rescue (SAR) operations. Their work proposes a conceptual framework, leveraging heterogeneous collaborative IoT entities with diverse data sources and edge devices. Their work further underscores the growing need for time-critical capabilities in modern IoT applications.

8. Conclusion and Future Work

The IoT market has seen extensive growth over the last few years. Some of the targeted domains and applications have requirements on timing performance. We have identified two key timing characteristics in the literature, time-criticality and predictability, which are represented by a broad spectrum of features affecting the timing. Moreover, the methods of measuring the IoT system's performance and the metrics to use, in ensuring these features are spread. We have therefore conducted a systematic literature review consisting of 38 primary studies, combined with surveying 28 IoT experts, in order to provide clear definitions of our characteristics of interest along with showcasing the

current state of the practice and theory in measuring timing in an IoT system across our five-layered system model.

Our hope is that this research will contribute to the future standardization of time measurement in IoT. Our findings offer actionable insights that could directly inform standardization efforts in organizations such as IEEE or Internet Engineering Task Force (IETF) and ongoing working groups such as DetNet and 6TiSCH. A fundamental next step is the establishment of a unified IoT reference model — existing models are often fragmented or outdated, lacking consensus across academia and industry. This study contributes to such efforts by identifying a structured set of timing-related measurement categories (e.g., Latency, Data rate, Reliability, Scheduling, Resource) that emerged both from literature analysis and were validated by over 75% of expert survey respondents. These categories, aligned with a layered IoT architecture, can serve as a foundation for categorizing and standardizing performance metrics across application domains. Furthermore, the observed imbalance in timing measurements across layers — particularly the relative scarcity of metrics in the Cloud layer compared to the Network layer — suggests targeted areas where standardization efforts could focus. This work may also support the harmonization of timing requirements across IoT applications, much like ongoing initiatives in adjacent fields such as digital twins, where research outputs have informed real-world standardization processes [70, 71]. We see potential in engaging with relevant working groups to translate these findings into formal specification drafts.

Future work should tackle the key timing challenges in IoT systems, such as device heterogeneity, multi-layered solutions, unpredictable wireless conditions, and the absence of a unified standard. Developing tools and methodologies for unobtrusive E2E timing analysis across IoT layers is essential. It is also vital to bridge the gap between theoretical models and practical deployment challenges, ensuring timing guarantees in large-scale, distributed, and resource-constrained environments. Collaboration between academia and industry will be crucial to aligning timing requirements with real-world use cases. Future studies could investigate the adaptation of hard RT techniques for time-critical IoT applications and the refinement of soft RT methods for diverse scenarios, ultimately supporting more reliable and predictable IoT systems.

Acknowledgements

This work is supported by the Swedish Agency for Innovation Systems via the iSecure project. We sincerely thank all survey respondents for their valuable time and insights. The authors also thank industrial partners, especially Canarybit, for their input.

References

[1] J. Perez-Ramirez, O. Seijo, I. Val, Time-Critical IoT Applications Enabled by Wi-Fi 6 and Beyond, IEEE Internet Things Mag. 5 (3) (2022) 44–49.

[2] M. Spörk, M. Schuß, C. A. Boano, K. Römer, Ensuring End-to-End Dependability Requirements in Cloud-based Bluetooth Low Energy Applications, in: Proc. Int. Conf. Embedded Wireless Syst. and Netw., 2021, pp. 55—66.

[3] G. Ali, M. M. Mijwil, I. Adamopoulos, J. Ayad, Leveraging the Internet of Things, Remote Sensing, and Artificial Intelligence for Sustainable Forest Management, Babylon. J. Internet Things 2025 (2025) 1–65.

[4] I. Ungurean, N. C. Gaitan, A Software Architecture for the Industrial Internet of Things—A Conceptual Model, Sensors 20 (19) (2020) 5603.

[5] K. S. Kiangala, Z. Wang, An Effective Communication Prototype for Time-Critical IIoT Manufacturing Factories Using Zero-Loss Redundancy Protocols, Time-Sensitive Networking, and Edge-Computing in an Industry 4.0 Environment, Processes 9 (11) (2021) 2084.

[6] M. Weyrich, C. Ebert, Reference Architectures for the Internet of Things, IEEE Softw. 33 (1) (2016) 112–116.

[7] M. Fahmideh, A. Ahmad, A. Behnaz, J. Grundy, W. Susilo, Software Engineering for Internet of Things: The Practitioners' Perspective, IEEE Trans. Softw. Eng. 48 (8) (2022) 2857–2878.

[8] IEEE Internet of Things Magazine, The Internet of Time-Critical Things: Advances and Challenges in Computing and Communications, *ieeexplore.ieee.org*, 2022. [Online]. Available: `https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9945844`. (accessed 2025-03-11).

[9] G. C. Buttazzo, Hard Real-Time Computing Systems - Predictable Scheduling Algorithms and Applications, Third Edition, Springer, New York, NY, USA, 2011.

[10] ISO/IEC, Internet of Things (IoT), Preliminary Report 2014, *iso.org*, 2014. [Online]. Available: `https://www.iso.org/files/live/sites/isoorg/files/developing_standards/docs/en/internet_of_things_report-jtc1.pdf`. (accessed 2025-02-01).

[11] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, IEEE Commun. Surv. and Tut. 17 (4) (2015) 2347–2376.

[12] S. Yi, C. Li, Q. Li, A Survey of Fog Computing: Concepts, Applications and Issues, in: Proc. 2015 Workshop Mobile Big Data, 2015, pp. 37—-42.

[13] W. Stallings, C. Beard, Wireless Communication Networks and Systems, First Edition, Pearson, Hoboken, NJ, USA, 2016.

[14] S. Bhandari, X. Wang, Prioritized Clock Synchronization for Event Critical Applications in Wireless IoT Networks, IEEE Sensors J. 19 (16) (2019) 7120–7128.

[15] L. Wang, H. Zhang, Analysis of Joint Scheduling and Power Control for Predictable URLLC in Industrial Wireless Networks, in: IEEE Int. Conf. Ind. Internet, 2019, pp. 160–169.

[16] E. Shahri, P. Pedreiras, L. Almeida, End-to-End Response Time Analysis for RT-MQTT: Trajectory Approach versus Holistic Approach, in: 19th Int. Conf. Factory Commun. Syst., 2023, pp. 1–8.

[17] B. Kitchenham, P. Brereton, A systematic review of systematic review process research in software engineering, Inf. and Softw. Technol. 55 (12) (2013) 2049–2075.

[18] J. S. Molléri, K. Petersen, E. Mendes, Survey Guidelines in Software Engineering: An Annotated Review, in: Proc. 10th ACM/IEEE Int. Symp. Empirical Softw. Eng. and Meas., 2016, pp. 1–6.

[19] R. V. Basili, G. Caldiera, D. H. Rombach, Goal Question Metric Approach, in: Encyclopedia Softw. Eng., Wiley, Hoboken, NJ, USA, 1994.

[20] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, L. Shamseer, J. M. Tetzlaff, E. A. Akl, M. Brennan, et al., The PRISMA 2020 statement: an updated guideline for reporting systematic reviews, BMJ 372 (2021) n71.

[21] P. Brereton, A. B. Kitchenham, D. Budgen, M. Turner, M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, J. Syst. and Softw. 80 (4) (2007) 571–583.

[22] N. B. Ali, K. Petersen, Evaluating strategies for study selection in systematic literature studies, in: Proc. 8th ACM/IEEE Int. Symp. Empirical Softw. Eng. and Meas., 2014, pp. 1–4.

[23] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: Proc. 18th Int. Conf. Eval. and Assessment Softw. Eng., 2014, pp. 1–10.

[24] M. Spörk, C. A. Boano, K. Römer, Improving the Timeliness of Bluetooth Low Energy in Dynamic RF Environments, ACM Trans. Internet Things 1 (2) (2020) 1–32.

[25] H. Kurunathan, R. Severino, E. Tovar, A Comprehensive Worst Case Bounds Analysis of IEEE 802.15.7, J. Sensor and Actuator Netw. 10 (2) (2021) 23.

[26] D. Del Gaudio, P. Hirmer, A lightweight messaging engine for decentralized data processing in the internet of things, Softw.-Intensive Cyber-Physical Syst. 35 (1) (2020) 39–48.

[27] H. Hadadian Nejad Yousefi, Y. Kavian, A. Mahmoudi, A Markov chain model for IEEE 802.15.4 in time critical wireless sensor networks under periodic traffic with reneging packets, J. Ambient Intell. and Humanized Comput. 13 (4) (2022) 2253–2268.

[28] A. R. Urke, Ø. Kure, K. Øvsthus, Autonomous Flow-Based TSCH Scheduling for Heterogeneous Traffic Patterns: Challenges, Design, Simulation, and Testbed Evaluation, IEEE Open J. Commun. Soc. 4 (2023) 2357–2372.

[29] P. Fara, G. Serra, F. Aromolo, Bounded transmission latency in real-time edge computing: a scheduling analysis, in: 26th Euromicro Conf. Digit. System Des., 2023, pp. 618–625.

[30] P. A. Michael, P. D. Tsanakas, D. S. Parker, Blue Danube: A Large-Scale, End-to-End Synchronous, Distributed Data Stream Processing Architecture for Time-Sensitive Applications, in: IEEE/ACM 26th Int. Symp. Distrib. Simul. and Real Time Appl., 2022, pp. 39–48.

[31] X. Li, S. Wang, J. Cao, Cellular network-based IIoT architecture for time-critical control tasks of building automation, Automat. Construction 162 (2024) 105387.

[32] F. Veisi, M. Nabi, H. Saidi, Coexistence Analysis of Multiple Asynchronous IEEE 802.15.4 TSCH-Based Networks, IEEE Access 8 (2020) 150573–150585.

[33] Y. Kim, C. Song, H. Han, H. Jung, S. Kang, Collaborative Task Scheduling for IoT-Assisted Edge Computing, IEEE Access 8 (2020) 216593–216606.

[34] J. Liu, Q. Zhang, Computation Resource Allocation for Heterogeneous Time-Critical IoT Services in MEC, in: IEEE Wireless Commun. and Netw. Conf., 2020, pp. 1–6.

[35] A. Althoubi, R. Alshahrani, H. Peyravi, Delay Analysis in IoT Sensor Networks, Sensors 21 (11) (2021) 3876.

[36] J. Shi, M. Sha, Z. Yang, Distributed Graph Routing and Scheduling for Industrial Wireless Sensor-Actuator Networks, IEEE/ACM Trans. Netw. 27 (4) (2019) 1669–1682.

[37] F. Habeeb, K. Alwasel, A. Noor, D. N. Jha, D. AlQattan, Y. Li, G. S. Aujla, T. Szydlo, R. Ranjan, Dynamic Bandwidth Slicing for Time-Critical IoT Data Streams in the Edge-Cloud Continuum, IEEE Trans. Ind. Inform. 18 (11) (2022) 8017–8026.

[38] A. Jaddoa, G. Sakellari, E. Panaousis, G. Loukas, P. G. Sarigiannidis, Dynamic decision support for resource offloading in heterogeneous Internet of Things environments, Simul. Model. Pract. and Theory 101 (2020) 102019.

[39] C. Zhang, F. Shen, J. Jin, Y. Yang, Fog Network Task Scheduling for IoT Applications, in: 2nd Workshop Fog Comput. and IoT, 2020, pp. 10:1–10:9.

[40] Y. Li, H. Takada, iSotEE: A Hypervisor Middleware for IoT-Enabled Resource-Constrained Reliable Systems, IEEE Access 10 (2022) 8566–8576.

[41] I. Behnke, L. Pirl, L. Thamsen, R. Danicki, A. Polze, O. Kao, Interrupting Real-Time IoT Tasks: How Bad Can It Be to Connect Your Critical Embedded System to the Internet?, in: IEEE 39th Int. Perform. Comput. and Commun. Conf., 2020, pp. 1–6.

[42] H. A. Bany Salameh, M. H. Khadr, M. Al-Quraan, M. Ayyash, H. Elgala, S. Almajali, Jamming-Aware Simultaneous Multi-Channel Decisions for Opportunistic Access in Delay-Critical IoT-Based Sensor Networks, IEEE Sensors J. 22 (3) (2022) 2889–2898.

[43] H. Baniabdelghany, R. Obermaisser, A. Khalifeh, P. Muoka, Metascheduling Using Discrete Particle Swarm Optimization for Fault Tolerance in Time-Triggered IoT-WSN, IEEE Internet Things J. 10 (14) (2023) 12666–12675.

[44] R. Trüb, R. Da Forno, L. Daschinger, A. Biri, J. Beutel, L. Thiele, Non-Intrusive Distributed Tracing of Wireless IoT Devices with the FlockLab 2 Testbed, ACM Trans. Internet Things 3 (1) (2021) 1–31.

[45] C. Stylianopoulos, M. Almgren, O. Landsiedel, M. Papatriantafilou, T. Neish, L. Gillander, B. Johansson, S. Bonnier, On the performance of commodity hardware for low latency and low jitter packet processing, in: Proc. 14th ACM Int. Conf. Distrib. and Event-Based Syst., 2020, pp. 177—182.

[46] M. Becker, D. Dasari, D. Casini, On the QNX IPC: Assessing Predictability for Local and Distributed Real-Time Systems, in: IEEE 29th Real-Time and Embedded Technol. and Appl. Symp., 2023, pp. 289–302.

[47] K. Brun-Laguna, P. Minet, Y. Tanaka, Optimized Scheduling for Time-Critical Industrial IoT, in: IEEE Global Commun. Conf., 2019, pp. 1–6.

[48] G. Patti, L. Leonardi, G. Testa, L. Lo Bello, PrioMQTT: A prioritized version of the MQTT protocol, Comput. Commun. 220 (2024) 43–51.

[49] H. Farag, S. Grimaldi, M. Gidlund, P. Österberg, REA-6TiSCH: Reliable Emergency-Aware Communication Scheme for 6TiSCH Networks, IEEE Internet Things J. 8 (3) (2021) 1871–1882.

[50] Y. Li, J. Lv, B. Li, W. Dong, RT-BLE: Real-time Multi-Connection Scheduling for Bluetooth Low Energy, in: IEEE Conf. Comput. Commun., 2023, pp. 1–10.

[51] Q. Xiong, X. Zhu, Y. Jiang, J. Cao, X. Xiong, H. Wang, Status Prediction and Data Aggregation for AoI-Oriented Short-Packet Transmission in Industrial IoT, IEEE Trans. Commun. 71 (1) (2023) 611–625.

[52] I. Ungurean, Timing Comparison of the Real-Time Operating Systems for Small Microcontrollers, Symmetry 12 (4) (2020) 592.

[53] H. Kurunathan, R. Sámano-Robles, M. G. Gaitán, R. Indhumathi, E. Tovar, Towards Multi-channel GTS Allocation in Visible Light Communication, in: South Amer. Conf. Visible Light Commun., 2023, pp. 130–134.

[54] R. D. Halloush, Transmission Early-Stopping Scheme for Anti-Jamming Over Delay-Sensitive IoT Applications, IEEE Internet Things J. 6 (5) (2019) 7891–7906.

[55] K. Bhimavarapu, Z. Pang, O. Dobrijevic, P. Wiatr, Unobtrusive, Accurate, and Live Measurements of Network Latency and Reliability for Time-Critical Internet of Things, IEEE Internet Things Mag. 5 (3) (2022) 38–43.

[56] K. Charmaz, L. L. Belgrave, Grounded Theory, in: The Blackwell Encyclopedia of Sociology, Wiley, Hoboken, NJ, USA, 2007.

[57] D. S. Cruzes, T. Dyba, Recommended Steps for Thematic Synthesis in Software Engineering, in: Int. Symp. Empirical Softw. Eng. and Meas., 2011, pp. 275–284.

[58] R. Franzosi, Quantitative Narrative Analysis, Sage, Newbury Park, CA, USA, 2010.

[59] M. Rodgers, A. Sowden, M. Petticrew, L. Arai, H. Roberts, N. Britten, J. Popay, Testing Methodological Guidance on the Conduct of Narrative Synthesis in Systematic Reviews: Effectiveness of Interventions to Promote Smoke Alarm Ownership and Function, Evaluation 15 (1) (2009) 49–73.

[60] Carnegie Mellon Softw. Eng. Inst., Pittsburgh, PA, USA, Designing an effective survey, 1st Edition (2005).

[61] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, in: Proc. 12th Int. Conf. Eval. and Assessment Softw. Eng., 2008, pp. 68—-77.

[62] I. Behnke, H. Austad, Real-Time Performance of Industrial IoT Communication Technologies: A Review, IEEE Internet Things J. 11 (5) (2024) 7399–7410.

[63] A. Vlavianos, L. K. Law, I. Broustis, S. V. Krishnamurthy, M. Faloutsos, Assessing link quality in IEEE 802.11 Wireless Networks: Which is the right metric?, in: 19th Int. Symp. Pers., Indoor Mobile Radio Commun., 2008, pp. 1–6.

[64] S. Shukla, M. F. Hassan, D. C. Tran, R. Akbar, I. V. Paputungan, M. K. Khan, Improving Latency in Internet-of-Things and Cloud Computing for Real-Time Data Transmission: A Systematic Literature Review (SLR), Cluster Comput. 26 (2023) 1–24.

[65] A. Gowri, P. S. Bala, I. Z. Ramdinthara, Comprehensive Analysis of Resource Allocation and Service Placement in Fog and Cloud Computing, Int. J. Adv. Comput. Sci. and Appl. 12 (3) (2021) 62–79.

[66] S. Kharb, A. Singhrova, A Survey on Network Formation and Scheduling Algorithms for Time Slotted Channel Hopping in Industrial Networks, J. Netw. and Comput. Appl. 126 (2019) 59–87.

[67] S. Abolhassani Khajeh, M. Saberikamarposhti, A. M. Rahmani, Real-Time Scheduling in IoT Applications: A Systematic Review, Sensors 23 (1) (2022) 232.

[68] T. Mitra, J. Teich, L. Thiele, Time-Critical Systems Design: A Survey, IEEE Des. and Test 35 (2) (2018) 8–26.

[69] A. Soularidis, K. I. Kotis, G. A. Vouros, Real-Time Semantic Data Integration and Reasoning in Life-and Time-Critical Decision Support Systems, Electronics 13 (3) (2024) 526.

[70] E. Ferko, A. Bucaioni, M. Behnam, Architecting Digital Twins, IEEE Access 10 (2022) 50335–50350.

[71] E. Ferko, A. Bucaioni, P. Pelliccione, M. Behnam, Standardisation in Digital Twin Architectures in Manufacturing, in: Proc. IEEE 20th Int. Conf. Softw. Archit., 2023, pp. 70–81.