Evaluation Metrics for Communication Protocols in Industrial Software Systems

Volodymyr Trykoz, Nicholas Sjöqvist Obucina, Björn Leander, Mohammad Ashjaei, Saad Mubeen

Mälardalen University, Västerås, Sweden, {firstname.lastname}@mdu.se

***Abstract*—Network convergence, driven by Industry 4.0, has introduced stringent security requirements on the distributed software in industrial control systems. Legacy plant-level protocols, originally developed with limited security considerations, are increasingly exposed to cyber threats. While security solutions such as encryption and message authentication are now being integrated into field-level protocols, they may significantly impact performance properties that are critical in these software systems, including timing predictability and resource efficiency. From the software engineering perspective, understanding these trade-offs requires a well-defined set of performance metrics that enables evaluation and comparison across the device, network, and system levels. In this paper, we present a systematic mapping study that identifies and classifies metrics used to evaluate communication protocols in industrial control systems. Based on an analysis of 20 primary studies selected from an initial pool of 198 identified studies, published between 2016 and 2026, we provide a mapping between the protocols and metrics as well as a bibliographic overview that characterizes existing research and highlights research gaps and opportunities.**

I. Introduction

Industrial control systems are undergoing a major digital transformation driven by Industry 4.0, with increased connectivity and software-based integration as key enablers. The convergence of Operational Technology (OT) and Information Technology (IT) introduces new requirements on interoperability, scalability, and maintainability, while elevating cybersecurity to a fundamental quality attribute of industrial software systems. As previously isolated networks become interconnected, the communication software stack used by controllers, sensors, and actuators is exposed to a broader and more dynamic threat landscape [13].

To address these challenges, industrial communication protocols have evolved from proprietary or fieldbus-based solutions, which primarily optimized latency and reliability, toward standardized and software-intensive protocols that incorporate security mechanisms. These newer protocols align with security requirements such as those defined in IEC 62443 [20], [15] and include features for encryption, authentication, and authorization. Protocols such as OPC UA, DDS, MQTT, and CIP EtherNet/IP exemplify this shift and are increasingly adopted in industrial software architectures [21]. While these mechanisms improve security, they also introduce additional software complexity and computational overhead that may affect performance-related quality attributes [20].

From a software engineering perspective, the introduction of such features raises important questions regarding the evaluation of extra-functional properties, like timing behavior and resource consumption. Industrial applications often operate under strict constraints, and even moderate changes in the communication software can have system-wide effects. However, current evidence on how security features influence these properties is scattered across studies that use heterogeneous metrics, experimental setups, and evaluation criteria. This lack of consistency makes it difficult for researchers and practitioners to compare and evaluate results, assess trade-offs, and reuse empirical evidence.

Despite the importance of performance evaluation in industrial software systems, there are no established guidelines that define which metrics should be used, at which system levels, and for which types of communication protocols. Consequently, protocol evaluations are commonly conducted using ad-hoc metric selections, limiting their comparability and cumulative value. A structured overview of existing metrics and their usage is therefore needed to support systematic evaluation, replication, and evidence-based decision making.

To address this gap, we conduct a systematic mapping study of research that evaluates the performance of industrial communication protocols used in these systems. The study is conducted according to the guidelines by Kitchenham and Charters [18], using snowballing as described by Wohlin [28], and includes a maturity assessment based on the Redwine–Riddle model [22]. Our results provide a structured classification of evaluation metrics, a mapping between protocols and metrics, and an overview of research trends. This contribution supports both researchers and practitioners in understanding existing evidence and identifying gaps and opportunities for future empirical studies. To enable transparency and replication, we provide a publicly available replication package[1].

II. Related Work

Hasan *et al.* [14] present a systematic literature review on communication requirements, technologies, challenges, and performance metrics for wireless mining teleoperation. Although focused on a different application domain, their work is methodologically related through its emphasis on performance metrics for communication systems. In contrast, our study targets industrial communication protocols across domains, providing a broader, protocol-centric view that supports comparative evaluation in industrial software engineering.

[1]https://github.com/nicholasob/Industrial-Protocol-Metrics-Mapping-Study

Cavalieri *et al.* [8] present a literature review based on Preferred Reporting Items for Systematic Review and Meta-Analysis (PRISMA). In it, the authors identify, categorize, and synthesize concepts from prior studies into a conceptual model. They also assess performance, but focus on a completely different performance dimension, i.e., economic, societal, and environmental performance. Nevertheless, their methodology is similar to ours and demonstrates how a comparable approach can be applied to a different performance dimension.

Another relevant study is the systematic review by Amjad *et al.* [2], which examines data interoperability among application-layer protocols in IIoT. The review focuses on protocol integration, interactions between IoT and data acquisition protocols, and their integration with hardware and software platforms. Although the study emphasizes interoperability rather than timing behavior or security-performance trade-offs, it provides useful context by highlighting the role of system integration architectures in the evaluation of automation systems.

As reported, there is a gap in the research area for providing metrics that are relevant for evaluating communication protocol performance in industrial software systems. None of the studies specifically conduct a literature review with the aim of identifying metrics or other aspects in the industrial automation.

III. Review methodology

The research was designed using the guidelines for conducting literature reviews, derived from works by Kitchenham and Charters [18] as well as Wohlin [28].

A. Process

The process constitutes the following steps:

1) Formulate the research topic, which guides the entire study. In this case, the topic concerns the evaluation of industrial communication protocols, with a particular focus on performance-related evaluations, especially in security-sensitive contexts.
2) Define the inclusion and exclusion criteria. Inclusion Criteria (IC) are derived from the research topic and guide the search process, while Exclusion Criteria (EC) are used to filter out irrelevant or low-value studies.
3) Define the search string based on the IC. This step was performed iteratively through pilot searches and analysis of the results.
4) Perform snowballing once the search string stabilized. Both backward and forward snowballing were applied.
5) Data extraction. The selected studies were analyzed with respect to the evaluation approaches employed.
6) Data synthesis. The extracted data were classified and presented in a way that enables a multidimensional view of the research field.

B. Research Questions

Given our focus on industrial communication protocols, we first aimed to identify which protocols are represented in the literature. Next, we examined the types of evaluation approaches used, with a particular focus on the metrics applied within those approaches. We also analyzed relevant publication trends, including publication density, venues, and the overall maturity of the studies, as well as any potential gaps in protocol coverage or evaluation approaches. Based on this, we define the following research questions:

RQ1 What industrial communication protocols are evaluated in the scientific literature?
RQ2 What metric categories are used to evaluate industrial communication protocols?
RQ3 Which publication trends exist in the evaluation of industrial communication protocols?
RQ4 What research gaps exist in the evaluation of industrial communication protocols?

C. Inclusion and Exclusion Criteria

An overview of the IC is shown in Table I, where IC1–IC4 were used to define the search string while IC5 served as a general filter. Each of IC1–IC4 represented its own group of keywords targeting a specific dimension of the search.

Several ECs, depicted in Table II, were used to filter out irrelevant studies or any studies that cannot be analyzed. Criteria EC1–EC3 ensure that we get full text studies in english that are within the scope of the work. Short papers (EC4) were excluded as such papers are typically work in progress and lack the methodological and technical detail needed for evaluation. Secondary studies (EC5) were excluded because this work reviews primary evaluation evidence; however, they were still useful for citation snowballing to optimize the search string. EC6 was introduced, so that full proceedings are excluded, while individual papers are captured by the search string. EC7 prevents scope drift by excluding studies that do not evaluate communication protocols or do not improve evaluation methods.

After the initial set of studies was identified, the EC were applied sequentially to the title, abstract, and full text. Snowballing was used to increase the likelihood of identifying relevant studies and was performed recursively until a defined cutoff point was reached. For forward snowballing, the process stopped when no new studies citing a given study were found. For backward snowballing, the cutoff point was set to studies published before 2016. Several iterations of a pilot study were conducted to identify relevant keywords for each category defined by the IC. This helped to iteratively refine the search string to minimize the number of irrelevant publications while maximizing the chances of finding relevant ones. Four databases were chosen for the search including IEEE explore, ACM, Scopus, and Web of Science.

Publications between 2016 and 2026 were included in the result. The final search string and the identified keywords belonging to each group are available in the replication package. The search string and inclusion and exclusion criteria were

TABLE I
INCLUSION CRITERIA USED FOR STUDY SELECTION.

ID	Inclusion criterion
IC1	The study has an industrial focus (e.g., industrial automation, industrial control systems, industrial automation systems).
IC2	The study describes industrial information-exchange protocols.
IC3	The study proposes a conceptual model, framework, or testbed for protocol evaluation.
IC4	The study focuses on security or performance.
IC5	The study is published in a peer-reviewed venue.

TABLE II
EXCLUSION CRITERIA USED FOR STUDY SELECTION.

ID	Exclusion criterion
EC1	The study is not available in full text.
EC2	The study is not available in English.
EC3	Unrelated domain (outside automation systems).
EC4	Short and work-in-progress papers with four pages or less.
EC5	Secondary studies (e.g., SLRs, SMSs, surveys).
EC6	Full proceedings.
EC7	The primary objective is not experimental or theoretical evaluation of communication protocols, or improvement of evaluation methods.

reviewed by another researcher to minimize bias in study selection and filtering.

IV. RESULTS

In this section, we discuss the results of applying the methodology from the previous chapter to the scientific databases and present the findings.

A. Search results

A total of 198 studies were found during the search, split across four databases, which were the filtered down to just 13 relevant studies, with snowballing adding 7 more studies for a total of 20. To minimize the risk of personal bias, the results and the selected papers were reviewed by another researcher. Fig. 1 illustrates the process.

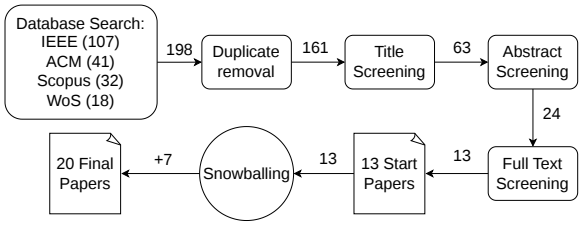

Fig. 1. Search, selection, and snowballing.

Data extraction was completed after the final set of papers had been identified. Fig. 2 shows how the selected studies are distributed across years and also shows publication venue types used by authors.

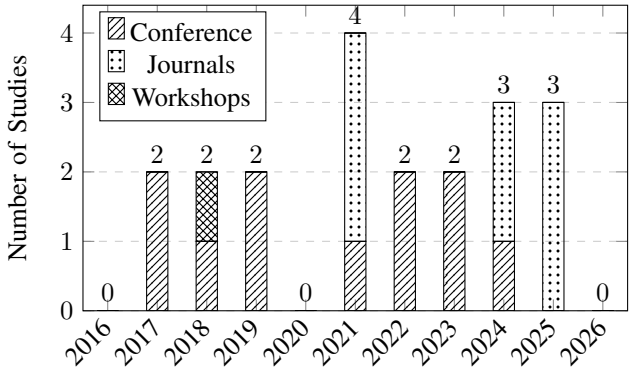

Fig. 2. Distribution of publications across years and venue types.

Overall, the publication trend has remained relatively stable, with few publications coming out each year. The zero value for 2026 is expected, since data collection took place at the beginning of that year and no publications from 2026 had been indexed yet.

In terms of venues, the majority of the works were published in distinct venues, with almost one paper per venue. The only exceptions are *Future Internet* [6], [5] and *INDUSCON* [24], [19], each accounting for two publications. In the case of *Future Internet*, both papers were authored by the same research group. Overall, the results show that there is no clearly established or dominant venue for publishing such works. This is why the breakdown is presented in terms of venue types rather than venues themselves.

B. Data extraction and classification

During the data extraction, identified metrics were extracted and classified depending on what they tell us about the system. The categories are:

- **End-to-end Timing Metrics** — The total time required for an action or message to propagate through the entire system, either to a final target node or back to the node that originated the message or initiated the action. Common labels for this category include time-to-completion (TTC) and round-trip time (RTT).
  In a system consisting of a starting node and a final target node, E2E latency is typically calculated as the difference between the timestamp recorded at the target node and the timestamp recorded at the starting node. If the starting and target nodes are the same, the metric corresponds to RTT; otherwise, it is considered an E2E delay. As the name suggests, these metrics are used to reason about the behavior of the system as a whole, particularly in scenarios where detailed intermediate timing information is of less interest and the focus is on overall system performance.

- **Component- and Sub-system Timing Metrics** — A decomposition of the E2E timing metrics into distinct segments, rather than measuring only the E2E timing. Like E2E latency, component latency is calculated as the difference between timestamps, but the timestamps are taken at intermediate nodes within the system rather than only at the start and end.
  This category includes metrics such as traversal latency, send/receive delay, network transmission delay, inbound and outbound delay, PLC processing time, elaboration time, application processing delay, and server run span. Component latency enables the analysis of individual paths and interactions within the system, making it useful when intermediate behavior is of interest. From this perspective, E2E latency can be viewed as a special case of Component latency; however, we distinguish it as a separate category to highlight how frequently studies rely solely on E2E metrics versus more fine-grained segment-level measurements. In both this and the previous group, depending on how the measurements are set up, latency metrics may capture not only network delays but also encoding, application processing, and other system-level overheads.
- **Jitter** — Variation in delay that leads to deviations in the timing of otherwise periodic signals. However, the definitions of how the variation is measured and of jitter itself differ. Some studies define jitter explicitly in terms of statistics such as the standard deviation of the transmission time [3]. Other studies do not explicitly define jitter, but rather describe what it shows or measures, also using terms such as variance or variability, without explicitly defining what jitter is [1], [19]. Variance in this case is not used as a statistical measure, but rather as a synonym for difference or variability, and therefore it is not measured as a statistical metric. In other cases, jitter is not defined but standard deviation is implicitly used as a measure of jitter [12]. Some studies measure jitter as the difference between a set cycle time and the measured cycle time, without a direct statistical interpretation [24].
  Overall, there is a large discrepancy in how jitter is perceived. In much of the literature, jitter is used as a synonym for measurement spread, whereas a more accurate definition refers to deviation from an expected period.
- **Cycle Time** — The elapsed time between two consecutive executions of a periodic function in the system. This metric can help identify drifts from the expected cycle period. For example, if a system is configured to perform an action every 100 ms but in practice performs it every 110 ms, the measured cycle time reflects this behavior. It can also be used as a measure of best-effort cyclical performance. It is important to distinguish this metric from jitter: while cycle time represents the measured interval itself, jitter is classically defined as the deviation of the measured cycle time from the expected periodicity.
- **Timing Reliability** — Given a specific deadline, this metric captures how often the system meets it. It is typically evaluated over a series of execution cycles and expressed as the percentage of successful deadline hits relative to the total number of runs. This metric is not limited to hard real-time systems; it can also be applied to systems with soft timing requirements.
- **Throughput** — A cross-sectional metric measuring the volume of data transferred per unit of time. Some example units include gigabit per second, megabit per second, and messages per second. It is often also referred to as bandwidth, data rate, or outgoing data rate. It is worth noting that although many papers include throughput as a metric, they do not explicitly state how it is measured, for example which tools, frameworks, or applications were used [1], [19], [3].
- **Data Loss** — A measure of delivery reliability describing the proportion of transmission units that fail to reach their destination under adverse conditions, such as high load or deliberate interference. It is typically measured as a ratio or percentage of lost units relative to the total number of transmitted units. Depending on the system and protocol layer, these transmission units may correspond to packets [27], frames [19], or similar entities.
- **CPU Utilization** — The processor load experienced by a node while handling messages. This metric is typically expressed as the percentage of CPU resources utilized by the application or protocol. Similar to throughput, this metric is often reported without a detailed description of how it is measured, including which tools or sampling methods are used.
- **Memory Utilization** — The memory load experienced by a node while processing messages. This metric is usually expressed as a percentage of the total available memory or as an absolute value, such as kilobytes or megabytes. Even here it is not always clear how the memory usage was measured [1], [30].
- **Protocol Overhead** — The additional data and processing required by a protocol beyond the payload itself, including headers, control messages, and connection setup procedures such as handshakes. It is typically calculated by accounting for all additional bytes introduced by a given protocol. Compared to most other metrics, protocol overhead is largely static and therefore relatively straightforward to evaluate.
- **Clock Synchronization** — Measurement uncertainty caused by clock de-synchronization between nodes or by local clock drift, which can influence the accuracy of latency or timing measurements. This issue is addressed in works by Gaffurini *et al.* [11], [10], where synchronization uncertainty and clock deviations are quantified as part of the experimental setup.

Moreover, the individual metric categories can be grouped into higher-level metric groups representing the broader system aspects targeted by the metrics.

- **Timing** — metrics concerned with communication timing

and protocol performance with respect to timing constraints. This group includes E2E and component latency, jitter, cycle time, and timing reliability.

- **Network** — metrics describe the performance of the underlying communication medium. This group includes throughput- and packet-loss-related metrics. While these metrics do not directly evaluate protocol behavior, they provide essential network context to interpret protocol performance.
- **Efficiency** — metrics capturing the resource impact of the communication protocol on the host system. This includes CPU and memory utilization as well as protocol overhead.
- **Meta-metrics** — metrics used to assess other metrics. Currently, this group consists of clock synchronization metrics, which are used to characterize clock drift and deviation both across nodes and within individual nodes. These metrics provide insight into timestamp accuracy and can be used to adjust or correct timing measurements.

The summary of which studies used which metric category is presented in Table III. Note that the same study may use multiple metric categories.

TABLE III
METRIC CATEGORIES EXTRACTED FROM THE SELECTED PAPERS, GROUPED BY HIGHER-LEVEL METRIC GROUPS.

Group	Category	Appears in
Timing	E2E Latency	[10], [11], [1], [19], [3], [27], [6], [9], [25], [4], [30], [26], [21], [5], [29], [17]
	Component Latency	[10], [11], [12], [7], [29], [30]
	Jitter	[1], [19], [3], [12], [29], [24]
	Cycle Time	[7], [24], [11], [12]
	Timing Reliability	[16]
Network	Throughput	[1], [19], [3], [4], [24], [7]
	Data Loss	[19], [27], [25], [4], [30], [16], [17]
Efficiency	CPU Utilisation	[1], [27], [30], [7], [21]
	Memory Utilisation	[1], [30], [7], [21]
	Protocol Overhead	[3], [27], [26], [21]
Meta	Clock Synchronization	[10], [11]

From a purely technological perspective, we also collected information on the protocols evaluated in the selected studies. This analysis helps identify current research interests and trends, as well as protocols that may be niche or underexplored in the existing literature. The results are presented in Table IV. As in the previous category, a single study may evaluate multiple protocols. Protocols mentioned only once were grouped under the *Other* category to improve the readability of the chart.

In order to better visualize the overall result, understand how the metrics relate to the protocols, and what are the clear gaps we combine them into a categorical bubble chart, which is shown in Fig. 3.

TABLE IV
PROTOCOLS EXPLORED IN THE SELECTED STUDIES.

Protocol	Investigated in
OPC UA	[27], [12], [9], [25], [26], [7], [21], [16]
MQTT	[1], [19], [3], [27], [9], [25], [26], [21]
CoAP	[1], [19], [3], [25]
HTTP	[1], [3]
WebSocket	[1], [3]
S7comm	[10]
PROFINET	[11]
EtherCAT	[29]
GOOSE	[4], [5]
Modbus	[1], [19], [6], [30], [5]
DNP3	[6]
IEC 870-5-101	[30]
SERCOS III	[24]
DDS	[21], [16]
ROS	[21]
ISA100.11a	[17]
WirelessHART	[17]

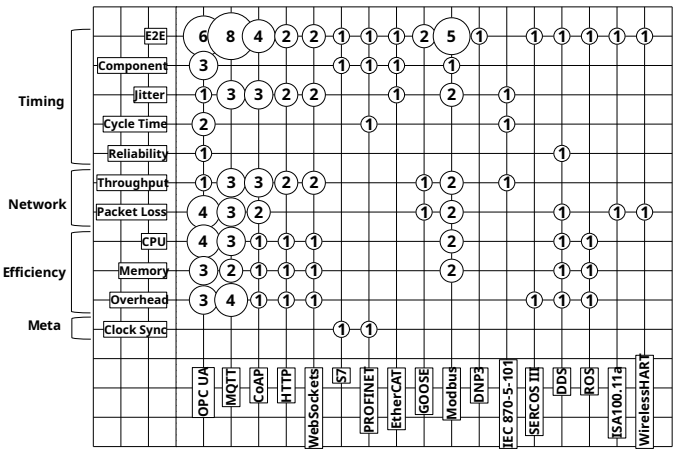

Fig. 3. Classification across metric group and protocol dimensions.

The data shows that most research has concentrated on a small set of protocols. While the metrics themselves are relatively evenly represented, there are still noticeable gaps, with some metrics appearing far more frequently than others. OPC UA and MQTT are the most commonly investigated protocols, which may reflect the growing interest towards them in modern industrial systems. CoAP and Modbus also appear in a fair number of studies, though less frequently than OPC UA and MQTT. Other protocols receive considerably less attention, indicating either niche usage or limited research interest. Regarding metrics, E2E measurements such as end-to-end latency and round-trip time are the most widely used. Jitter, throughput, packet loss, CPU usage, memory

consumption, and protocol overhead are also evaluated, but less frequently than the E2E latency metrics.

C. Maturity assessment

A scoring system based on Redwine-Riddle maturity model [22] and used in other works such as Satka *et al.* [23] was applied to better understand the maturity of the solutions proposed in the papers. The score breakdown is as follows:

- **Score 0 - Inconclusive**. The paper does not make a clear contribution or is too vague to classify.
- **Score 1 - Not mature at all**. The paper presents an idea, concept, architecture, taxonomy, or method, but does not provide working proof.
- **Score 2 - Somewhat mature**. The paper provides a proof of concept (PoC) and demonstrates it in one or more use cases, but there is no evidence that the contribution is used in the research community.
- **Score 3 - Mature**. The paper includes a PoC, use case(s), and credible evidence of uptake from the research community, adapted in industry, or their proposed artifact builds on an already established and reused tool chain.

Table V shows the classification of studies according to their maturity score.

TABLE V
MATURITY EVALUATION.

Maturity Score	Publication	Total
Inconclusive	-	0
Not mature at all	[29]	1
Somewhat Mature	[10], [11], [19], [3], [17], [27], [12], [6], [9], [25], [4], [30], [24], [26], [7], [5], [16]	17
Mature	[1], [21]	2

Most studies are considered to be somewhat mature, with only one study classified as not mature and two studies categorized as fully mature. While it would be beneficial for more studies to reach a higher level of maturity, the timeframe of only ten years makes this understandable, especially for the more recent publications, which simply have not had enough time to mature.

V. DISCUSSION

In terms of **RQ1**, OPC UA and MQTT stand out as the most frequently explored protocols. CoAP and Modbus also appear in several studies, but not as often as the first two. This makes sense since these protocols were designed for industrial environments and for use with constrained, embedded devices. What is interesting, however, is that other industrially relevant protocols are far less represented. For example, GOOSE, Profinet, and WirelessHART receive noticeably less attention than the four protocols mentioned above. One explanation could be that some of these protocols are quite niche—they have narrow, specialized use cases and may be less appealing to the broader industry. In contrast, something like OPC UA has wide applicability and includes built-in mechanisms for modeling a variety of systems, which likely contributes to its popularity in research.

Additionally, protocols such as OPC UA and MQTT may benefit from larger communities, stronger tooling support, and more extensive documentation. MQTT in particular is also widely used beyond the industrial automation domain, which may further increase its visibility and attractiveness for research. These factors could contribute to the observed focus on these protocols in the literature.

However, this does not fully explain the distribution, as Profinet is also widely adopted in industrial settings yet appears to be comparatively underrepresented in the selected studies. One possible explanation is related to temporal trends in research focus. Since this study considers publications from the past decade, it may reflect a shift in attention toward protocols such as OPC UA, which have gained traction alongside the rise of Industry 4.0. As a result, research activity may have increasingly concentrated on protocols perceived as better aligned with emerging industrial paradigms, rather than those with longer-standing deployment.

When it comes to **RQ2**, researchers tend to focus on E2E measurements, while paying less attention to metrics that capture latency per component. This may indicate that they are more concerned with the overall behavior of the system rather than with the performance of individual components. If a system meets its deadlines, it is seen as functioning correctly; if it does not, then something in the system is not working well. However, practitioners may still find traversal-latency measurements valuable because they can reveal choke points and bottlenecks that affect overall performance. This is especially notable given that other common network-performance indicators (such as jitter, packet loss, and throughput), have received more attention in the literature than traversal latency, though still not as much as end-to-end latency. Using these metrics together can provide a more complete picture and help complement the insights gained from end-to-end measurements. Another explanation may be that the systems are very simple, having only two nodes and measuring only the overarching latency between them, without any measurements in between. This, on the other hand, suggests that the measurement setup itself was simple and not particularly complicated. In the future, it may be interesting to implement more complicated setups and thus be able to do more complex measurements.

Jitter is a metric that stands out in the sense that its definition varies between publications. As mentioned previously, some studies use the classical definition, describing jitter as deviations from an expected deadline or period (in the latter case, the deadline is implicitly equal to the period). However, other works use the term more loosely and apply it, for example, to denote the standard deviation of timing measurements. It would be preferable if jitter were used more consistently and not conflated with real-time terminology. Authors should more clearly distinguish between jitter and statistical measures such

as standard deviation and variance in their work.

It is worth pointing out that some metrics, e.g, throughput, CPU utilization, and Memory utilization have a reproducibility issue. Several studies report them and include in their results. But, they do not specify exactly how it was measured, like which tools were used and such. This is especially important for something like CPU usage and memory usage where the monitoring tool itself can influence the measurements. It would be nice if authors reported and are more explicit with how do they measure it, even if it is as simple as using htop or process explorer. It helps reproducibility.

Beyond this, communication in these works is not limited to network-level and inter-node performance (such as latency), but also includes intra-node behavior. This is reflected in the interest in CPU usage, memory consumption, and protocol overhead. These metrics describe what happens on the device itself rather than on the network, and they help reveal how different protocols affect the computational load and resource usage of the nodes. Considering both network-level and device-level metrics provides a more complete picture of system performance, so it would be interesting to see them used together more.

Clock synchronization is an interesting but often overlooked category of metrics. It can be seen as a kind of meta-metric, since it reflects how accurate time-based measurements actually are. Good clock-sync visibility can help identify clock drift or other timing issues between devices. In the studies that did examine it, the differences were usually quite small, which may explain why most researchers do not place much emphasis on it. Still, even small improvements in measurement accuracy can be valuable. Clock-sync information allows you to normalize timestamps reported by different nodes, which is useful because nodes are not always perfectly synchronized. For this reason, it would be beneficial for future protocol-focused work to include clock-synchronization metrics as part of their evaluation.

It is also worth noting the lack of research that uses timing-reliability metrics. Since most of the literature we examined does not focus on hard real-time systems, where meeting deadlines exactly is critical, this gap is somewhat understandable. Still, more work in this area would be valuable. Even when a system is not strictly hard real-time, understanding how reliably it meets its deadlines over time and under different conditions can provide important insights into its behavior and robustness. For example, soft-real time systems with quality-of-service–driven deadlines may tolerate occasional misses, making metrics such as deadline hit rate a useful indicator of overall timing reliability.

For **RQ3** the publication trend analysis shows that the number of publications on this topic has not grown over the past decade. More practical research on industrial protocols and technologies would be useful, especially studies carried out in environments that resemble real industrial deployments. This would give a better understanding of how these protocols actually behave in real-world or near-real-world conditions.

Another observation is the lack of a centralized venue for publishing such works, as there is practically one venue per publication. This suggests that there is no clear research community in this area, since authors publish in very different venues. Finally, it can be observed that conferences and journals are the primary means of sharing such works, with only one workshop identified among the selected studies. In practice, this may indicate that there is no well-established venue or research community specifically focused on the holistic evaluation of communication protocol performance. Instead, most studies concentrate on a limited number of performance aspects and are published in venues that align with those specific aspects. This fragmentation may further contribute to the lack of comprehensive evaluation approaches observed in the literature.

In terms of **RQ4**, several potential research gaps were already highlighted when addressing the previous research questions. At a higher level of analysis, it can be observed that most studies focus on only a subset of the identified metric groups, with only a few works covering a broader range of metrics. This suggests that many authors prioritize the evaluation of specific aspects of protocol performance rather than providing comprehensive assessments.

As a result, holistic evaluations of industrial communication protocols appear to be limited in the reviewed literature. Consequently, researchers and practitioners seeking a comprehensive understanding of protocol performance must rely on multiple studies conducted under different experimental setups. This fragmentation makes it difficult to compare results and to assess protocol behavior in a consistent and systematic manner.

VI. Conclusion

This paper presented a systematic mapping study of research evaluating the performance of industrial communication protocols. As a primary research contribution, the study provides a structured classification of performance metrics, a mapping between metrics and protocols, and an analysis that characterizes publication trends and research maturity. The results show clear imbalances in current evaluation practices, where several metric categories and protocols are extensively studied while others remain largely unexplored. The maturity assessment further indicates that most studies are at a moderate level of maturity, reflecting the relatively recent and evolving nature of this research field.

From a research perspective, the mapping highlights gaps and underrepresented areas that deserve further investigation, particularly with respect to metric diversity and system-level evaluation. The catalog of metrics and the protocol-to-metric mapping support reuse of empirical evidence and can facilitate more comparable and cumulative research, which is a central goal of systematic studies in software engineering. Additionally, dispersed publication trends and venues suggest a lack of centralized venues or communities dedicated to holistic performance evaluation.

From a practitioner perspective, the results provide guidance for selecting evaluation metrics when assessing communica-

tion protocols in industrial systems. While not all metrics are applicable in every context, the study shows that focusing on narrow subsets of metrics limits the ability to reason about trade-offs and performance implications. The overview provided in this paper supports more informed and transparent evaluation decisions in practice.

As future work, we plan to integrate the identified metrics into a conceptual model that supports reusable and systematically comparable protocol evaluations. Using this model, we intend to conduct empirical studies that assess multiple protocols across a broader range of metric categories. This work aims to advance evidence-based performance evaluation of industrial communication software and to support more mature and holistic empirical studies in this area.

REFERENCES

[1] Al-Hawawreh, M., Sitnikova, E.: Developing a Security Testbed for Industrial Internet of Things. IEEE IoT-J **8**(7), 5558–5573 (2021)

[2] Amjad, A., et al.: A Systematic Review on the Data Interoperability of Application Layer Protocols in Industrial IoT. IEEE Access **9**, 96528–96545 (2021)

[3] Aquilina, J., et al.: A Comparative Analysis of Application Layer Protocols Within an Industrial Internet of Things Monitoring System. In: IEEE M&N. pp. 1–6 (2024)

[4] Boeding, M., et al.: A testbed for evaluating performance and cybersecurity implications of IEC-61850 GOOSE hardware implementations. In: IEEE CCNC (2023)

[5] Boeding, M., et al.: End-to-End Framework for Identifying Vulnerabilities of Operational Technology Protocols and Their Implementations in Industrial IoT. Future Internet **17**(1), 34 (2025)

[6] Boeding, M., et al.: Exponential Backoff and Its Security Implications for Safety-Critical OT Protocols over TCP/IP Networks. Future Internet **17**, 286 (2025)

[7] Burger, A., et al.: Bottleneck Identification and Performance Modeling of OPC UA Communication Models. In: 2019 ACM/SPEC ICPE. pp. 231–242 (2019)

[8] Cavalieri, A., et al.: A Conceptual Model Proposal to Assess the Effectiveness of IoT in Sustainability Orientation in Manufacturing Industry: An Environmental and Social Focus. Applied Sciences **12**(11), 5661 (2022)

[9] Freitas, L., et al.: OPC-UA in interoperability – a performance comparative testing. IFAC-PapersOnLine **58**(8), 240–245 (2024)

[10] Gaffurini, M., et al.: Virtual PLC in Industrial Edge Platform: Performance Evaluation of Supervision and Control Communication. IEEE TIM **73**, 1–10 (2024)

[11] Gaffurini, M., et al.: Characterizing the Real-Time Communication Performance of Virtual PLC in Industrial Edge Platform. IEEE OJIM **4**, 1–11 (2025)

[12] Gruner, S., et al.: Towards Performance Benchmarking of Cyclic OPC UA PubSub over TSN. In: 2022 IEEE 27th ETFA). pp. 1–8. IEEE (2022)

[13] Göppert, J., et al.: A Survey on Life-Cycle-Oriented Certificate Management in Industrial Networking Environments. JSAN MDPI **13**(2), 26 (2024)

[14] Hasan, S., et al.: Wireless Communication in Underground Mining Teleoperation: A Systematic Review. IEEE Access **13**, 72577–72602 (2025)

[15] IEC 62443 part 4-1: Security for industrial automation and control systems - secure product development lifecycle requirements. Standard, IEC (2009-2018)

[16] Ioana, A., Korodi, A.: DDS and OPC UA Protocol Coexistence Solution in Real-Time and Industry 4.0 Context Using Non-Ideal Infrastructure. Sensors **21** (2021)

[17] Jecan, E., et al.: A dual-standard solution for industrial Wireless Sensor Network deployment: Experimental testbed and performance evaluation. In: 14th WFCS. pp. 1–9. IEEE (2018)

[18] Kitchenham, B., Charters, S., et al.: Guidelines for performing systematic literature reviews in software engineering (2007)

[19] Monges, Y., et al.: Performance Evaluation of MQTT and CoAP Communication Protocols in IIoT for Electrical Substations Based on the IEC 61850 Standard. In: 15th INDUSCON. pp. 774–779. IEEE (2023)

[20] Ponomar, S., Sarkans, M.: Overview of the development of cybersecurity in data transmission protocols used in industry. Proceedings of the Estonian Academy of Sciences **74**(2), 143–148 (2025)

[21] Profanter, S., et al.: OPC UA versus ROS, DDS, and MQTT: Performance Evaluation of Industry 4.0 Protocols. In: 2019 IEEE ICIT. pp. 955–962. IEEE (2019)

[22] Redwine, S., Riddle, W.: Software technology maturation. Proceedings - International Conference on Software Engineering pp. 189–200 (1985)

[23] Satka, Z., et al.: A comprehensive systematic review of integration of time sensitive networking and 5G communication. JSA **138**, 102852 (2023)

[24] Sestito, G.S., et al.: Evaluating Real-Time Ethernet performance indicators for SERCOS III networks. In: 2021 14th INDUSCON. pp. 1191–1197. IEEE (2021)

[25] Silva, D., et al.: A Performance Analysis of Internet of Things Networking Protocols: Evaluating MQTT, CoAP, OPC UA. Applied Sciences **11**(11), 4879 (2021)

[26] Silveira Rocha, M., et al.: Performance Comparison Between OPC UA and MQTT for Data Exchange. In: 2018 MetroInd4.0&IoT. pp. 175–179. IEEE (2018)

[27] Wang, X., et al.: A Comparative Study to Evaluate the Performance of Communication Protocols for Process Industry. In: 32nd ITNAC. pp. 170–177. IEEE (2022)

[28] Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: 18th EASE. ACM (2014)

[29] Wu, X., Xie, L.: End-to-End Delay Evaluation of Industrial Automation Systems Based on EtherCAT. In: 2017 IEEE 42nd LCN. pp. 70–77. IEEE (2017)

[30] Yi, M., et al.: Benchmarking Cloud-Based SCADA System. In: 2017 IEEE CloudCom. pp. 122–129. IEEE (2017)