

Article

Analysis of Time Drift and Real-Time Challenges in Programmable Logic Controller-Based Industrial Automation Systems: Insights from 24-Hour and 14-Day Tests

Ayah Hijazi * , Mátyás Andó  and Zoltán Pödör

Institute of Computer Science, Faculty of Informatics, ELTE Eötvös Loránd University, 1117 Budapest, Hungary; am@inf.elte.hu (M.A.); pz@inf.elte.hu (Z.P.)

* Correspondence: idws3y@inf.elte.hu

Abstract

Ensuring the reliability and temporal accuracy of real-time data transmission in industrial systems presents significant challenges. This study evaluates the performance of a Siemens Programmable Logic Controller (PLC) transmitting data to a MongoDB database via Node-RED over 24 h and 14-day intervals. Key issues observed include time drift, timestamp misalignment, and forward/backward time jumps, mainly resulting from Node-RED's internal timing adjustments. These anomalies compromised the integrity of time-sensitive data. A significant disruption on day 8 due to a power outage introduced data gaps and required manual system recovery. Additional spikes in missing data were observed after day 12. The Predictive Missing Value (PMV) model addressed these gaps. The model achieved strong accuracy at larger intervals (e.g., 5 min) but showed reduced performance at finer resolutions (1–2 min) due to the irregularity of data patterns. This research highlights the difficulty of maintaining temporal consistency in long-term, real-time systems. It also evaluates the PMV model's effectiveness in mitigating data loss while acknowledging its limitations under complex timing disruptions.



Academic Editor: Leonardo Aho Zuppa

Received: 8 September 2025

Revised: 18 October 2025

Accepted: 27 October 2025

Published: 28 October 2025

Citation: Hijazi, A.; Andó, M.; Pödör, Z. Analysis of Time Drift and Real-Time Challenges in Programmable Logic Controller-Based Industrial Automation Systems: Insights from 24-Hour and 14-Day Tests. *Actuators* **2025**, *14*, 524. <https://doi.org/10.3390/act14110524>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: PLC; MongoDB; Node-RED; real-time data transmission; time drift

1. Introduction

In recent years, the integration of Programmable Logic Controllers (PLCs) with SQL and NoSQL databases has gained significant attention due to the industrial push towards smart manufacturing and Industry 4.0 [1,2]. Node-RED's ability to integrate various devices, protocols, and data sources in an intuitive, flow-based environment has made it a popular choice for building scalable, modular IoT systems. By utilizing protocols like Modbus TCP and integrating time-series databases, Node-RED enables real-time data processing and visualization, making it an ideal platform for modern industrial applications [3]. However, as with any distributed system, one key challenge arises: ensuring proper clock synchronization across the various devices that communicate within the network.

For instance, Nițulescu and Korodi [4] demonstrated its utility in IIoT-based SCADA systems, enabling real-time monitoring and control while offering a cost-effective, scalable alternative to traditional systems. Similarly, Medina-Pérez et al. [5] highlighted its effectiveness in environmental monitoring, facilitating data processing and visualization from IoT sensors, and supporting real-time insights and future enhancements like automated alerts. These examples emphasize the adaptability and scalability of Node-RED across

diverse domains. Additionally, research in [6] examined the impact of messaging protocols on data transfer delays in IIoT applications. Using Node-RED, the study evaluated time performance across local and intercontinental connections, showing that balanced settings achieved low latency and consistent performance. The findings emphasized the need to address time drifts as Industry 4.0 progresses toward greater system interoperability.

Time drift, a phenomenon where discrepancies occur between the clocks of different devices or systems, becomes a significant concern in Node-RED-based applications [7] and in many real-time applications, including wireless sensor networks (WSNs), Time-Sensitive Networking (TSN), and 5G networks. The issue of time drift and its characterization has been a subject of continuous study since the early 1960s [8,9]. As IoT systems often rely on precise timestamps for logging, event sequencing, and data synchronization, minor discrepancies can lead to data collection, analysis, and reporting errors. In distributed systems like those built with Node-RED, where multiple devices interact with each other in real-time, time drift can distort the accuracy of time-series data stored in databases and affect the proper functioning of protocols such as MQTT and Modbus TCP [10]. Therefore, managing and mitigating time drift is essential to ensure the integrity and reliability of data, ultimately allowing for accurate monitoring and control in IoT-driven processes.

Steel et al. [11] highlighted the challenges of time drift in various devices, noting that GPS trackers maintained stable synchronization, while accelerometers like activPAL and ActiGraph devices exhibited significant drift over 9–14 days. This underscores the critical need for effective synchronization in systems with multiple devices. Moreover, Baunach [12], Hauweele et al. [13] and Elsharief et al. [14] explored time management in wireless sensor networks (WSNs), proposing a dynamic self-calibration approach integrated into the system kernel to detect and correct clock drift. This method, combined with specialized hardware like interrupt controllers, enhances the accuracy and reliability of timestamps in distributed systems, addressing synchronization challenges across diverse applications.

Moreover, different types of drift—including deployment, data, and concept Drift—impact systems across various domains, not only clock drift where encountered. In Intelligent Sensor Networks (ISNs), deployment drift due to environmental changes reduced sensor effectiveness, prompting the development of a graph-based detection algorithm for dynamic sensor redeployment [15]. Networked Control Systems (NCS) faced data drift across channels between components, addressed by a Lyapunov-based controller that improved stability [16]. Machine Learning (ML) models in 5G networks contended with data and concept drift, countered by a novel framework using contextual data and median-based detection to reduce false alarms and maintain accuracy [17,18].

To better situate the proposed research within the existing body of knowledge, the next section reviews recent work on time drift, synchronization, and real-time data management in industrial systems. With this section our aim is to highlight the specific, scientific and practical gap addressed in this paper: middleware-induced timestamp anomalies and their impact on temporal reliability.

2. Related Works

The modern industrial landscape, driven by Industry 4.0 and the Industrial Internet of Things (IIoT), demands highly reliable and temporally accurate real-time data transmission [19]. PLCs remain central to data acquisition from diverse sensors and machinery, forming the backbone of industrial control, yet their integration into distributed ecosystems faces significant challenges, including time drift, timestamp misalignment, and missing values, which can propagate errors in cyber-physical systems and compromise safety in critical applications [20–22]. Synchronization across distributed networks is essential for

operations like fault diagnosis, coordinated control, and event logging, but time drift, propagation delays, processing uncertainties, and network asymmetries hinder precise alignment, with requirements varying from milliseconds to nanoseconds and even minor delays degrading real-time control loops [23–26]. Emerging solutions, including Time-Sensitive Networking (TSN) for deterministic low-latency communication, Precision Time Protocol (PTP) with hardware timestamping for sub-microsecond accuracy across Wi-Fi and Ethernet, and EtherCAT enhancements (e.g., fuzzy proportional-integral schemes), mitigate these issues while analyzing propagation delay impacts in IIoT [19,27–29]. PLC data integration into databases introduces further obstacles, such as connectivity instability, high volumes, format diversity, and quality maintenance [20]. Middleware like OPC UA enables secure, metadata-rich communication for automation and digital twins [30,31], while Node-RED facilitates flexible PLC-to-database flows for visualization and storage in SCADA systems [32]. However, Node-RED's use in time-sensitive environments can induce timestamp irregularities, time jumps, and data continuity loss from internal timing adjustments or network latencies, as seen in real-time digital twin integrations with Siemens PLCs, underscoring the need for advanced synchronization, correction, and buffering mechanisms [19,33,34]. Missing data, arising from sensor malfunctions, communication failures, transmission errors, or power outages, exacerbates these issues by leading to incomplete analyses, biased models, and impaired predictive maintenance, amplified by Industry 4.0's sensor density [5,35,36]. Imputation techniques range from statistical methods for short gaps to deep learning approaches like recurrent neural networks, autoencoders, and gap-imputing algorithms for complex multivariate time series with varying durations and periodicities [5,35,37]. Distinct from general predictive imputation, the Predictive Missing Value (PMV) method prioritizes temporal continuity in industrial streams, achieving high accuracy for short-to-medium gaps but declining at finer resolutions due to irregular patterns and timing disruptions [20,38,39]. Ongoing research focuses on enhancing imputation accuracy, data fidelity, and resilience in real-time contexts [40,41].

Despite these advances, existing research largely concentrates on data fidelity or latency optimization, with limited attention to the temporal reliability of continuous data communication itself. This gap motivated the present study, which investigates how time drift, timestamp irregularities, and missing intervals evolve during long-term PLC–middleware–database operation. While prior work has primarily focused on synchronization protocols, hardware timestamping, and imputation algorithms, there is limited experimental research that formally defines middleware-induced timestamp anomalies within an end-to-end PLC–middleware–database pipeline, applies a diagnostic method to detect and correct such anomalies, and evaluates the practical outcome of producing a cleaned, time-consistent dataset for downstream analysis. This paper addresses this gap by experimentally defining and quantifying timing anomalies—including time drift, forward/backward jumps, and intermittent missing records—applying the Predictive Missing Value (PMV) approach as a diagnostic and data-cleaning tool and evaluating PMV's performance and limitations across multiple time resolutions. In doing so, we provide a practical, experiment-driven perspective on assessing synchronization performance and temporal reliability in long-running industrial data systems. Our main aim in this paper is to demonstrate the existence of this gap through experiments and to develop a practical solution for it.

3. Methodology (Data Generation and Flow)

This study was conducted in a controlled test environment that reproduces the fundamental communication structure of a typical industrial automation system. The setup consists of a Programmable Logic Controller (PLC) emulating a continuous process by incre-

menting a counter variable every second, a middleware layer (Node-RED) that collects and forwards data packets, and a MongoDB database that stores the transmitted records. MongoDB, a flexible document-based and schema-free database, efficiently handles time-series data through easy configuration, asynchronous writes, and JSON-like records, making it ideal for real-time industrial data collection and analysis [42]. Although implemented on a laboratory PC, this architecture mirrors the key components and data flow of real industrial installations, where field controllers transmit process variables through middleware gateways to supervisory or historian systems. Using this configuration enables precise control and monitoring of temporal behaviors such as timestamp preservation, time drift, ordering irregularities, and missing values—under continuous operation. The simplicity of the counter-based PLC task ensures that any observed timing anomalies originate from the communication and middleware layers rather than from the process logic itself, making the results representative of a typical industrial data-transmission pipeline.

The investigated system represents a typical Industry 4.0 architecture, where process data from a Siemens S7-1200 12/14 AC/DC/RLY PLC is continuously logged and analyzed in the PLC datalog for supervisory control. The program was developed using TIA Portal V16 software, with the logic implemented through a ladder diagram that transmitted counter values at 1 Hz, simulating the steady flow of operational data. Communication between the PLC and external systems was established using a TCP/IP connection (via ETHERNET Subnet) and PUT/GET communication with a remote partner was enabled. Additionally, the system and clock memory were activated. The PLC generated and sent counter values every second. Node-RED received each value, time-stamped it, and distributed it to MongoDB server (Version 4.4) for storage. MongoDB Compass (Version 1.37.0) was used as the graphical user interface for monitoring and managing the database.

The data transmitted during the experiment were stored sequentially in the MongoDB database as individual records. Each entry contained two key fields: the PLC counter value $C_{PLC}(s)$ and the Node-RED timestamp $T_{NR}(s)$ as in Equation (1). Figure 1 shows the dataflow of the research. This configuration allowed continuous comparison of two independent time references: the PLC time (PLC's internal clock) and the $T_{NR}(n)$ (Node-RED system clock).

$$\text{Record}_s = \{C_{PLC}(s), T_{NR}(s)\}, s = 1, 2, \dots, S \quad (1)$$

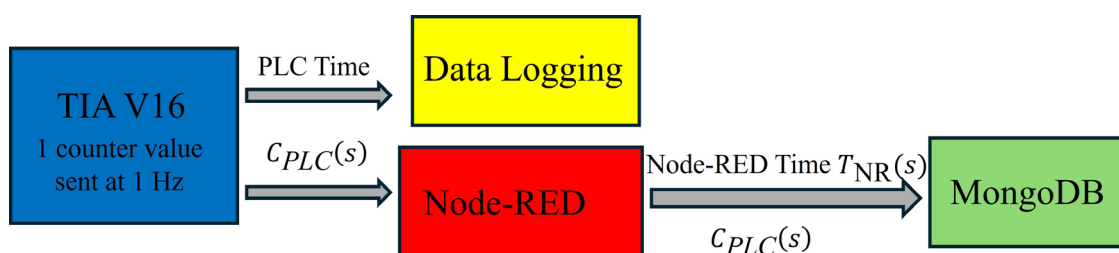


Figure 1. Experimental setup of the PLC–Node-RED–MongoDB communication chain.

The study involved continuous monitoring over a 24 h period, followed by an extended 14-Day period as a continuation of the same operation. The 24 h test revealed time series problems, like time drift, backward jumps and missing values, while the extended monitoring also showed forward jumps and the effects of a sudden electrical shutdown. Each record was examined chronologically to calculate time drift, based on the difference between Node-RED timestamps and the corresponding PLC counter values, and to identify temporal anomalies. Missing values, typically appearing regularly but never consecutively, reflected timing differences between the PLC and Node-RED. Backward jumps occurred

when buffered data were written out of sequence, causing later timestamps to precede earlier ones, and forward jumps arose from brief processing or network delays.

Figure 2 illustrates these anomaly types, combining actual observations with conceptual markers to clearly highlight missing values and backward jumps. Both periods were conducted under identical communication conditions, ensuring uninterrupted data flow and consistent network parameters. The raw data collected from both tests were systematically analyzed to identify timing irregularities and synchronization patterns, providing a comprehensive view of the temporal behavior of the PLC–Node-RED–MongoDB system.

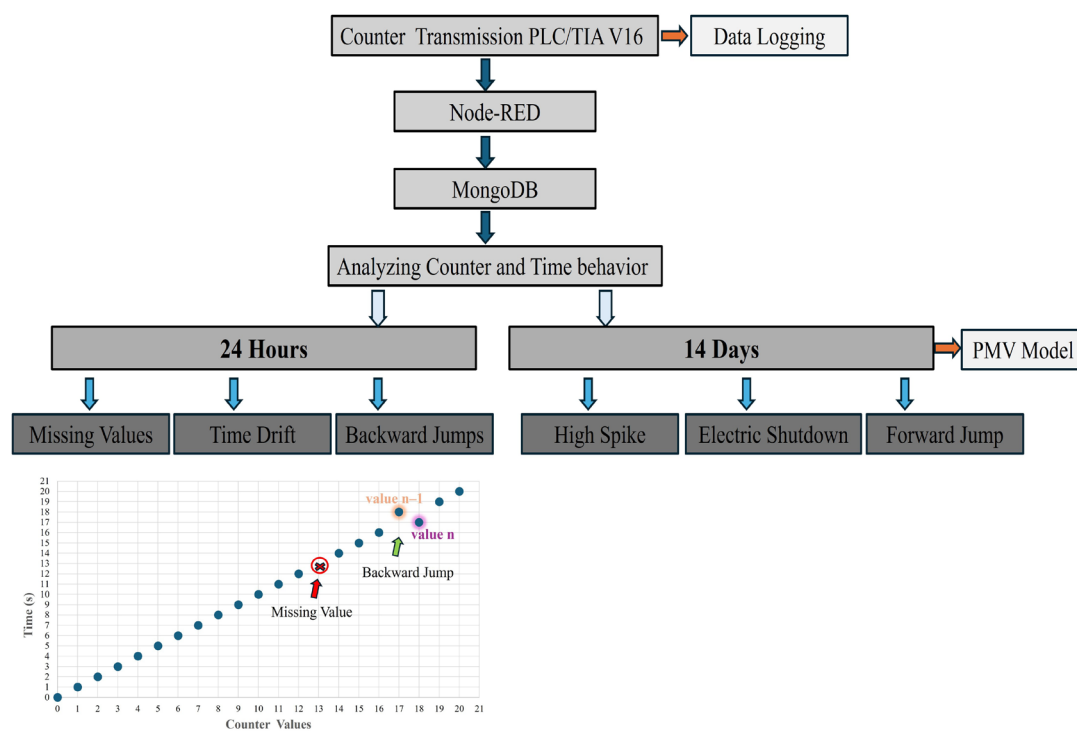


Figure 2. Observed anomalies during continuous counter transmission for the 24 h and 14-Day.

After anomalies were identified, the Predictive Missing Value (PMV) approach was applied to estimate the number of missing records arising from differences in timing between the PLC and Node-RED. The model leveraged the average timing characteristics of both systems and the duration of the monitoring period to predict expected data omissions, providing a systematic framework for understanding temporal gaps in the dataset. The accuracy and implications of this prediction were evaluated in the Results and Discussion Section 4.

The study employed a realistic industrial communication framework with continuous long-term monitoring, systematic detection of temporal anomalies, and predictive assessment of missing values using the PMV model. Integrating the PLC–Node-RED–MongoDB configuration with the PMV approach created a controlled environment to evaluate timing behavior, identify irregularities, and quantify systematic data gaps, providing a robust foundation for analyzing the temporal reliability of continuous industrial data flows.

4. Results and Discussion

This section presents and interprets the outcomes of the continuous data transmission experiments conducted using the Siemens PLC, Node-RED middleware, and MongoDB database. The goal is to analyze the temporal reliability of communication by examining how time drift, timestamp irregularities, and missing intervals evolved over both short-term 24 h and long-term 14-Day operation. The emphasis is placed not on data accuracy,

but on the stability and synchronization of timestamps that define the real-time behavior of industrial data flow.

4.1. PLC and MongoDB

The PLC counter values transmitted to MongoDB were monitored and plotted. According to the 24 h test, there were 805 missing values, which do not appear in MongoDB. Based on direct data logging (PLC time), the PLC works appropriately, and every second between 0 and 0.26 s sends the value to Node-Red. There are no signs of data loss or any time irregularity. This observation suggests that the PLC is functioning correctly, indicating that the missing values were not due to the PLC.

4.2. Time Drift for 24 H

The 24 h test results are illustrated in Figure 3. The X-axis represents the PLC real-time in hours, while the Y-axis indicates the time drift, calculated as the difference between the Node-RED transmission time and the corresponding counter value at each second.

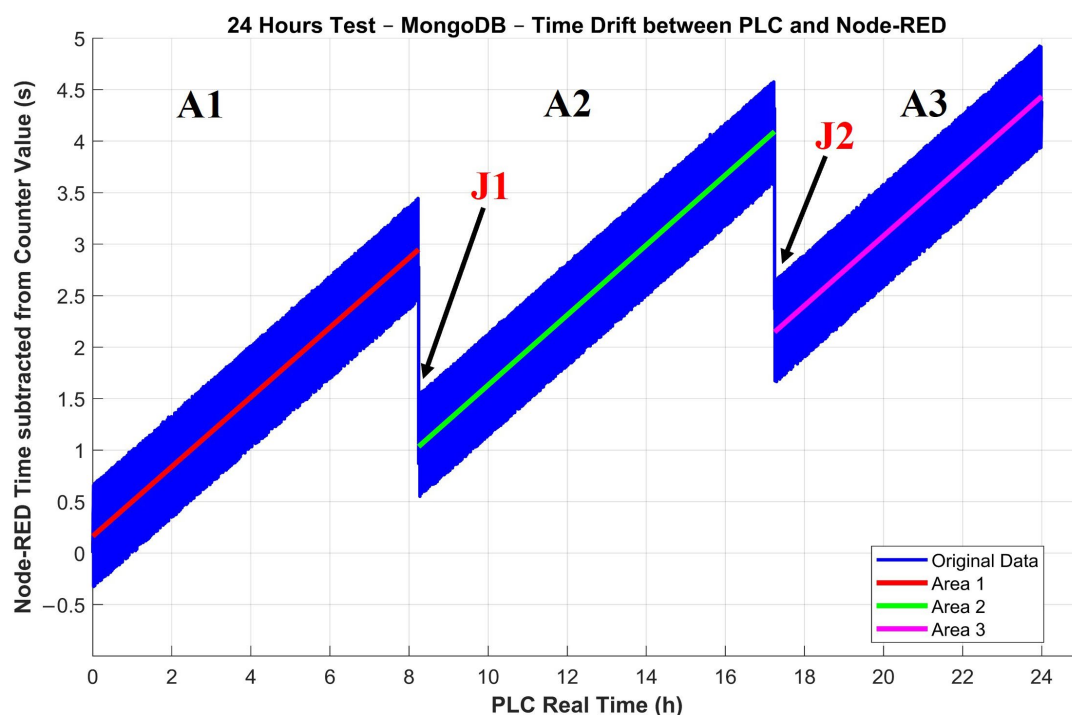


Figure 3. The nature of the time drift between the PLC and the Node-RED. A1 is Area 1, A2 is Area 2, and A3 is Area 3.

The overall drift pattern was segmented into three regions: A1, A2, and A3, separated by two noticeable jumps, labeled J1 and J2. The drift follows a consistent trend within these three regions, as detailed in Table 1. This behavior suggests that the drift remains stable within each region, while the jumps may indicate system interruptions or synchronization events.

Table 1. Drift slope and Linear Regression Coefficient (R^2).

Sections	Line Equation	Slope	Coefficient of Determination, R^2
A1	$y = 0.000094x + 0.0000$	0.000094	0.8869
A2	$y = 0.000094x - 0.0005$	0.000094	0.9048
A3	$y = 0.000094x - 0.0010$	0.000094	0.8428

At the 8th and 17th hour, the jump occurred, the relevant values are shown in Tables 2 and 3. These jumps indicate Node-RED's attempts to “catch up” after accumulating drifts. The system temporarily adjusts its internal clock or transmission process to mitigate the increasing delay. On the other hand, this leads to the phenomenon that the stored value of the database (value n) has an earlier timestamp than the earlier values (value $n-1$). According to this phenomenon, the real order is mixed when the timeline is used as shown in Figure 4.

Table 2. Presenting J1 at the 8th hour.

Counter	Node-RED Sending Time	Time Difference
29,931	08:18:53.777	00:00:01.015
29,932	08:18:54.780	00:00:01.003
29,933	08:18:53.874	−00:00:00.906
29,934	08:18:54.873	00:00:00.999
29,935	08:18:55.877	00:00:01.004

Table 3. Presenting J2 at the 17th hour.

Counter	Node-RED Sending Time	Time Difference
62,644	17:24:08.314	00:00:01.015
62,645	17:24:09.319	00:00:01.005
62,646	17:24:08.372	−00:00:00.947
62,647	17:24:09.386	00:00:01.014
62,648	17:24:10.383	00:00:00.997

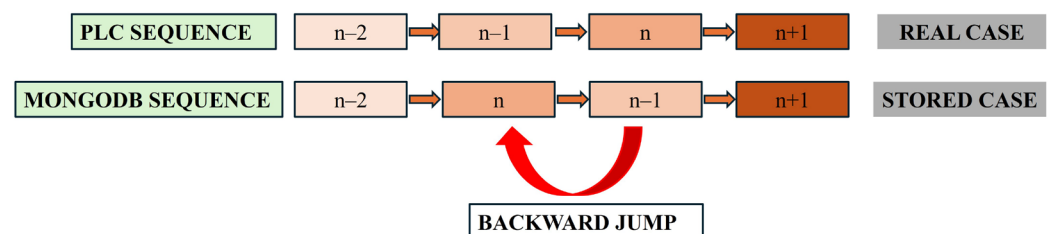


Figure 4. Illustration of the importance of backward jumps.

Even though there is some compensation (J1 and J2), there is still a significant gap between the real-time and Node-RED operations.

4.3. Time Drift for 14-Day Test

The 14-day test extended the evaluation period to assess how timing deviations evolve over the long term and to observe continuous operation built upon insights from the prior 24 h analysis, capturing a broader and more nuanced picture of the time drift behavior over an extended duration. As illustrated in the Figures presented in this section, the entire test period was systematically plotted and analyzed, following a methodology like that used for Figure 3. Data was collected and examined through database integration to ensure accuracy and continuity. The 14-day period has 3 specific cases, as shown in Figure 2, to enhance interpretability, each representing a specific operational pattern and behavior observed in the time drift.

4.3.1. Typical Operation with One Anomaly

The first case, shown in Figure 5, covers the initial 8 days of the test. The time drift exhibited a generally stable progression during this period, reflecting normal operation.

However, intermittent jumps—a total of 21 anomalies—were identified throughout this case. This equates to approximately two to three daily jumps, indicating occasional but recurrent deviations in timing synchronization.

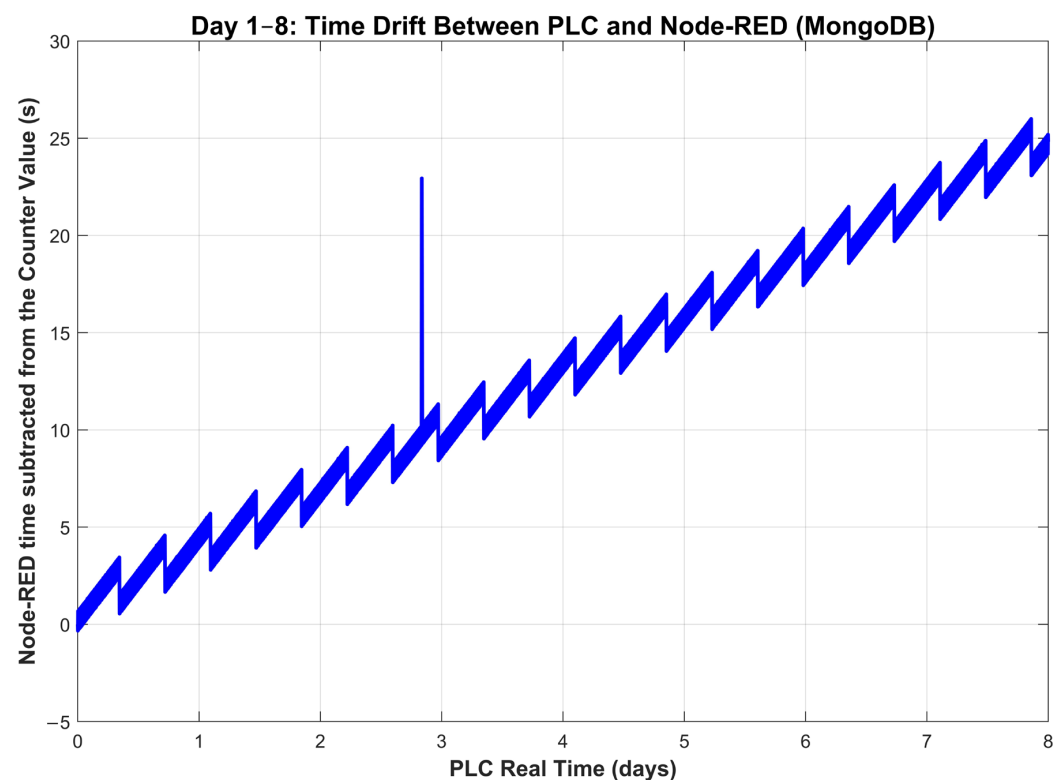


Figure 5. The nature of the time drift between the PLC and the Node-RED for the 14-day Test—(day 1 to day 8).

Despite these anomalies, the overall data transmission remained consistent, and the drift followed a predictable trend. However, a notable outlier was observed between the end of day 2 and the beginning of day 3, where a sharp spike in the time drift momentarily disrupted the otherwise consistent pattern. This anomaly is supported by the data presented in Table 4, which reveals two consecutive counter values showing time lags of 16.188 s and 13.112 s, respectively. Together, they indicate a cumulative error of nearly 30 s in the data recording process. Such a substantial deviation strongly suggests a temporary interruption in data transmission. Notably, the PLC continued to transmit values normally and without interruption during this period, as confirmed by the continuous timestamps in Table 4. This suggests that the issue does not originate from the PLC but rather within the Node-RED environment, likely due to a network lag or a momentary processing delay. It is essential to note that this anomaly was an isolated incident, occurring only once during the entire 14-day period. The apparent reduction in MongoDB records reflects delays in data transmission from Node-RED rather than issues with the database itself. Temporary processing bottlenecks or network congestion occasionally caused Node-RED to batch or postpone sending counter values, resulting in some entries being recorded later than expected or temporarily missing during short-term analysis windows.

The time drift observed underscores the complexities of real-time data handling in distributed systems. Network or processing time inconsistencies can lead to out-of-sequence transmissions, negatively affecting the accuracy and reliability of time-sensitive applications. Over 24 h, it was noted that Node-RED experienced a drift where the recorded time stamps in the database lagged in real-time, resulting in a 4.417 s discrepancy. Specifically, the final counter value in the PLC was saved at the end of the 24 h period, whereas Mon-

goDB recorded the corresponding timestamp slightly later, at 00:00:04.417 on the following day, as illustrated in Figure 6. This indicates that while the PLC data aligns with the end of the day, the MongoDB entry occurs a few seconds into the next day. After extrapolating this drift over extended periods, the cumulative time drift on day 8 would be approximately 35.336 s (assuming a linear drift rate without other external anomalies). However, as shown in Table 4, a significant spike occurred, with the drift reaching 25.137 s. Such discrepancies emphasize the critical need for synchronization mechanisms to ensure temporal integrity over long-term operations.

Table 4. Representing the continuation of the running counter values in PLC and whether MongoDB received them.

Counter	PLC Time	Time Difference	Received by MongoDB
247,359	19:37:22.008	00:00:00.776	YES
247,360	19:37:23.224	00:00:01.216	YES
247,361	19:37:24.104	00:00:00.880	NO
247,362	19:37:25.032	00:00:00.928	NO
247,363	19:37:26.216	00:00:01.184	YES, but 16.188 s later
247,364	19:37:27.160	00:00:00.944	NO
247,365	19:37:28.120	00:00:00.960	NO
247,366	19:37:29.200	00:00:01.080	NO
247,367	19:37:30.216	00:00:01.016	NO
247,368	19:37:31.072	00:00:00.856	NO
247,369	19:37:32.008	00:00:00.936	NO
247,370	19:37:33.216	00:00:01.208	NO
247,371	19:37:34.088	00:00:00.872	NO
247,372	19:37:35.208	00:00:01.120	NO
247,373	19:37:36.056	00:00:00.848	NO
247,374	19:37:37.104	00:00:01.048	NO
247,375	19:37:38.168	00:00:01.064	NO
247,376	19:37:39.176	00:00:01.008	NO
247,377	19:37:40.136	00:00:00.960	NO
247,378	19:37:41.024	00:00:00.888	YES, but 13.112 s later
247,379	19:37:42.104	00:00:01.080	NO
247,380	19:37:43.024	00:00:00.920	NO
247,381	19:37:44.032	00:00:01.008	NO
247,382	19:37:45.152	00:00:01.120	NO
247,383	19:37:46.168	00:00:01.016	NO
247,384	19:37:47.224	00:00:01.056	NO
247,385	19:37:48.248	00:00:01.024	NO
247,386	19:37:49.120	00:00:00.872	NO
247,387	19:37:50.144	00:00:01.024	NO
247,388	19:37:51.248	00:00:01.104	NO
247,389	19:37:52.112	00:00:00.864	NO
247,390	19:37:53.032	00:00:00.920	YES
247,391	19:37:54.048	00:00:01.016	YES

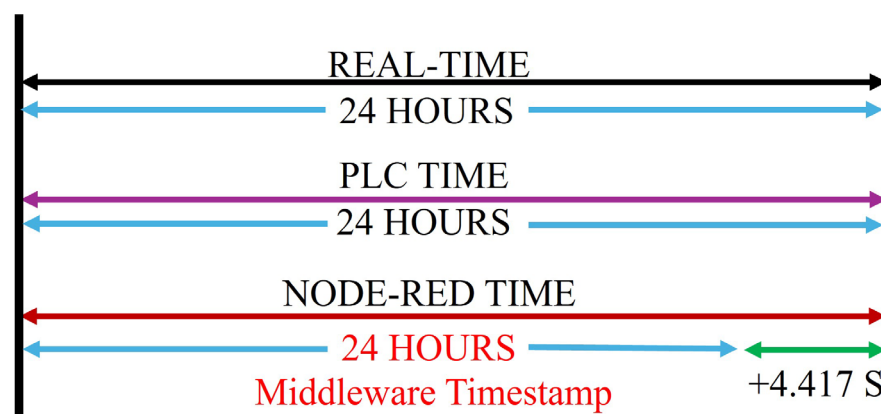


Figure 6. Time drift of Node-RED over 24 h, illustrating a 4.417 s discrepancy.

4.3.2. The Electric Shutdown

Following day 8, a sudden electric shutdown occurred, as depicted by the red line in Figure 7 and zoomed in to clearly show what happened during the electric shutdown. This shutdown and the network restoration lasted nearly 17 min, as shown in Table 5 and Figure 8. Figure 8 represents the sequence of operations and recovery following an unexpected power interruption. Initially, the system was in regular operation, with the PLC sending data consistently through Node-RED to the connected MongoDB database. Each component (PLC, Node-RED, and the MongoDB) operates synchronously, logging data as expected. However, a sudden electric shutdown disrupts this workflow, halting operations and severing the connection between the PLC, PC, Node-RED, and MongoDB. This power outage suspends data transmission and processing. When power is restored, the Network is back ON, and PLC restarts, holding onto the last recorded value, “750,068”, which remains in its memory, ready to be sent once Node-RED resumes its connection, but connectivity is still interrupted. Upon restarting the PC, Node-RED is restarted from the command prompt (restarting Node-RED establishes the connection with the MongoDB). The “750,068” counter value is sent to the Node-RED then to the MongoDB after a disconnection of 11.209 min. The TIA Portal is restarted to re-establish access to the PLC’s data log for full operational continuity. After these steps, which lasted 5.840 min, the system resumed normal operation, processing and logging data.

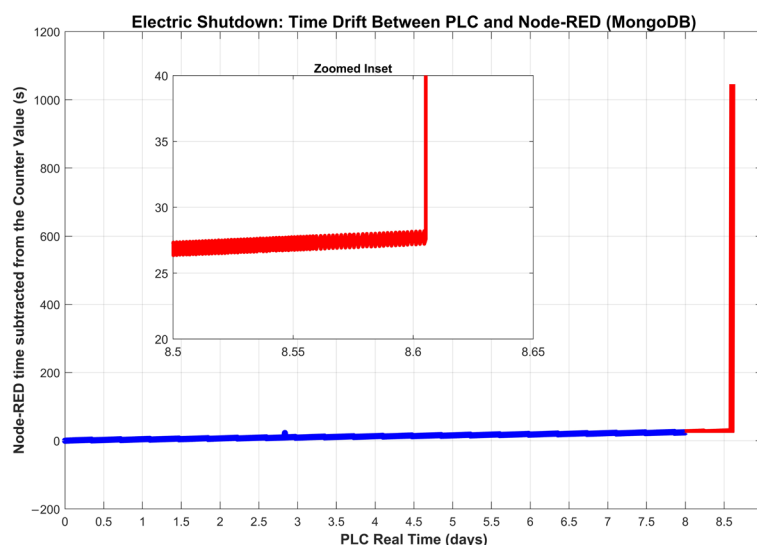
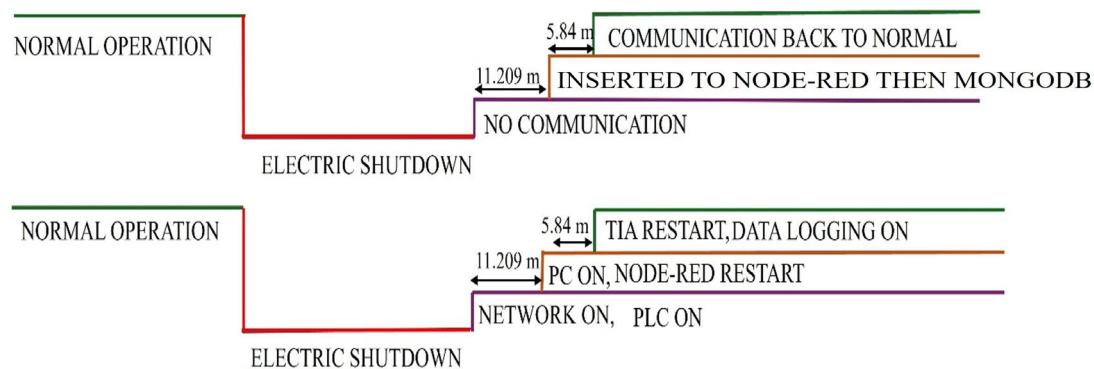


Figure 7. The nature of the time drift between the PLC and the Node-RED for the 14-day Test (the electric shutdown after day 8).

Table 5. Representing the counter values in MongoDB during the electric shutdown.

Counter	PLC Time	Time Difference	Received by MongoDB
750,067	15:16:02.112	00:00:01.088	YES
750,068	15:16:03.115	00:00:01.003	NO
750,068	15:16:03.115	00:00:00.000	NO
750,068	15:16:03.115	00:00:00.000	NO
750,068	15:16:03.115	00:00:00.000	NO
750,068	15:16:03.115	00:00:00.000	NO
750,068	15:16:03.115	00:00:00.000	NO
750,068	15:16:03.115	00:00:00.000	NO
750,068	15:16:03.115	00:00:00.000	NO
750,068	15:16:03.115	00:00:00.000	NO
750,068	15:16:03.115	00:00:00.000	NO
750,068	15:16:03.115	00:00:00.000	YES, but 11.209 min later
750,069	15:33:13.048	00:17:08.933	NO
750,070	15:33:14.176	00:00:01.128	YES, but 5.840 min later

**Figure 8.** The PLC, PC, Node-RED, and TIA software operation due to the electric shutdown.

4.3.3. Time Drift of Day 12 to Day 14

There was a new phenomenon in the last 2 days compared with the first 8 days. In the highlighted FJ area, it is evident that a forward jump occurred, as shown in Figure 9, allowing the data to continue without any time drift, as illustrated in Table 6. There are no irregularities based on the number of counter values and the Sending Time by the PLC (the relative transmission time for each counter value). On the other hand, in the case of the database, there are no missing values, only a high time difference (around 2.011 s). This upward slope was presented only once during the 14-day. This effect also makes data processing harder because next to the backward jumps, the forward jump also can change the timeline.

The other new phenomenon is that distinct spikes appear in the data, after day 12.4. Table 7 shows the relevant values in the case of one spike. Table 7 shows the relevant values in the case of one spike. For instance, at the missing counter values “1,205,716” and “1,205,717”, the time difference is 1.950 s, which is notably higher than the surrounding values. Two missing values next to each other only appear after 12 days. This kind of problem can be connected to the network problem and Node-RED. Unfortunately, this 2 s anomaly is too small to obtain enough information, making it more challenging to do the proper timeline analysis.

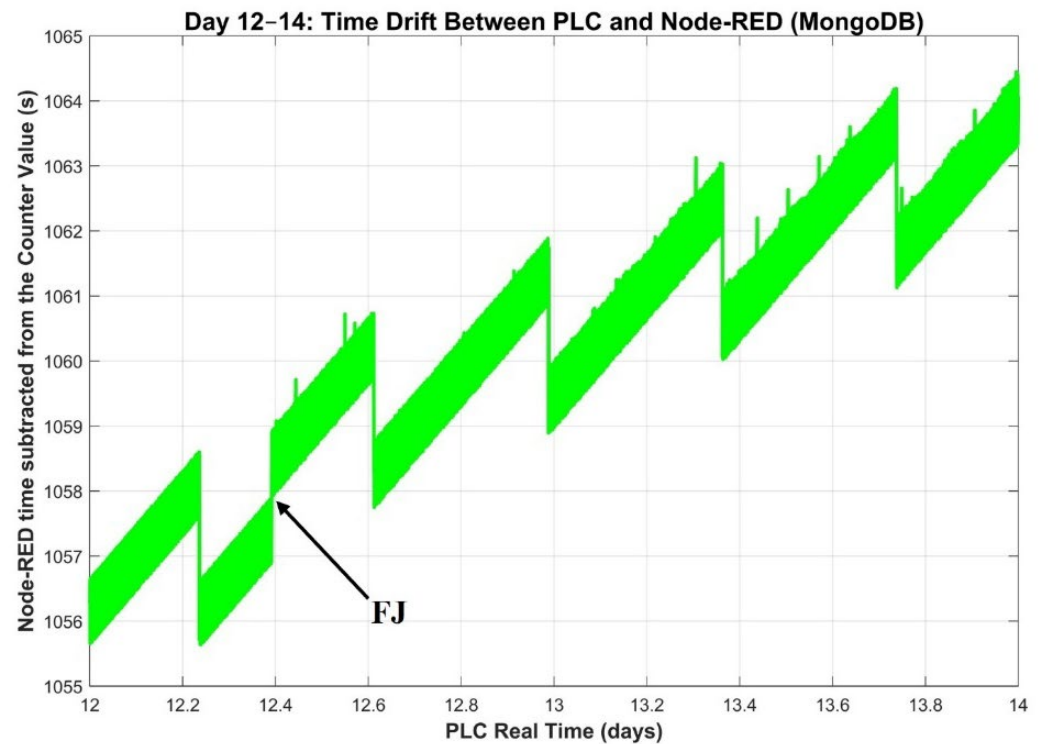


Figure 9. The nature of the time drift between the PLC and the Node-RED from day 12 to day 14.

Table 6. The PLC Counter values, the PLC sending time, and the time difference at the occurrence of FJ (the forward Jump).

Counter	PLC Sending Time	PLC Time Difference	Received by MongoDB
1,057,925	06:09:34.872	00:00:00.968	YES
1,057,926	06:09:35.824	00:00:00.952	YES
1,057,927	06:09:36.944	00:00:01.120	YES
1,057,928	06:09:37.856	00:00:00.912	YES, but 2.011 s later
1,057,929	06:09:38.984	00:00:01.128	YES
1,057,930	06:09:40.016	00:00:01.032	YES
1,057,931	06:09:40.928	00:00:00.912	YES
1,057,932	06:09:41.816	00:00:00.888	YES

Table 7. The MongoDB Counter values presenting the occurrence of high spikes.

Counter	Node-RED Sending Time	Time Difference
1,205,714	23:12:58.984	00:00:01.005
1,205,715	23:12:59.998	00:00:01.014
1,205,718	23:13:03.010	00:00:01.950
1,205,719	23:13:04.032	00:00:01.022
1,205,721	23:13:05.049	00:00:01.017
1,205,722	23:13:06.073	00:00:01.024
1,205,723	23:13:07.120	00:00:01.047

4.4. PMV Calculation for the 24 H

During the 24 h period, the MongoDB recorded the number of missing values, supporting the hypothesis that the issue is independent of the PLC and the MongoDB themselves, yet influenced by Node-RED. Investigation into missing values revealed a distinct temporal

trend, as illustrated in Figure 10. The outliers (indicated by red circles) were removed from the dataset using the interquartile range (IQR) threshold for defining outliers at a multiplier of 4.

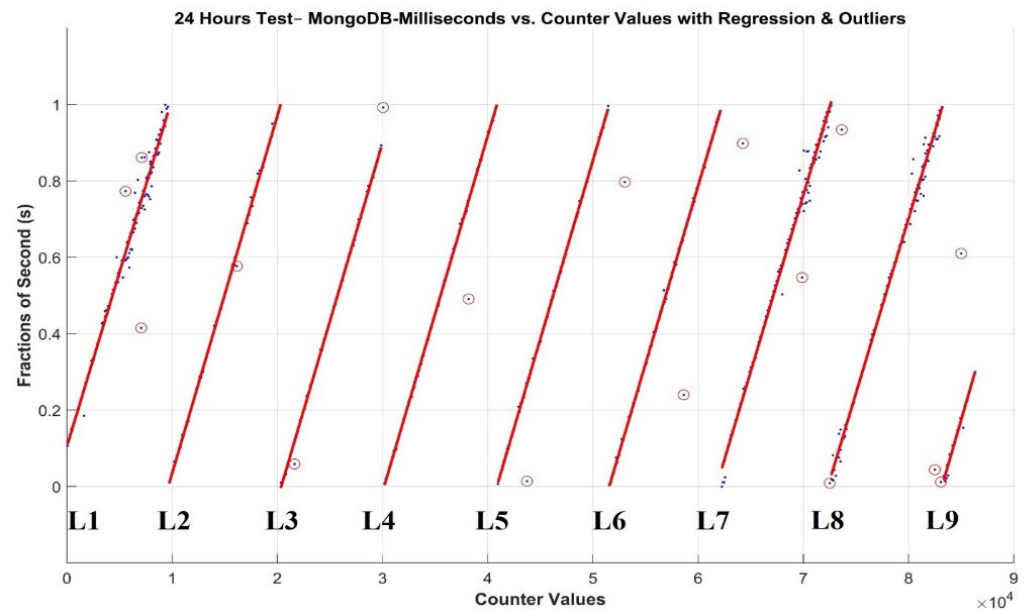


Figure 10. The occurrence of missing values according to the original Node-RED time and highlighting the outliers.

It is important to note that these regular missing values have continuously appeared throughout the test, and there are no two next to each other. The main reason of it is that different devices have different average time differences. So, suppose the data collection is slow (Node-RED) compared to the data generation (PLC). In that case, it is natural that the data collection does not record a particular value after a specific time, as seen in Figure 11. This theory was introduced in our earlier publication [20].

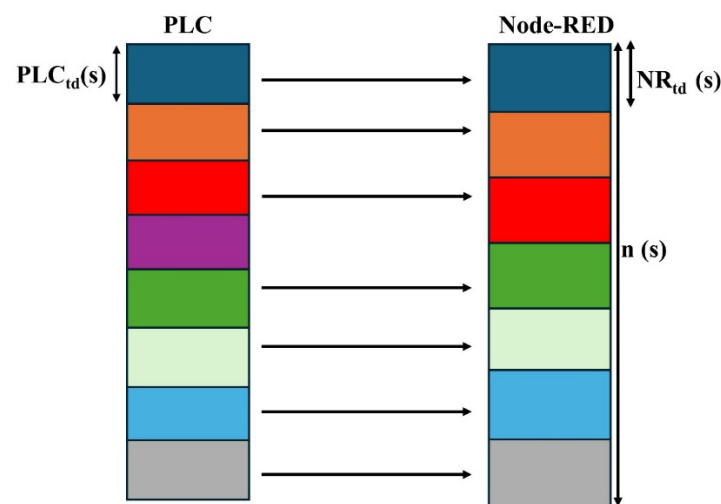


Figure 11. Systematic Illustration based on the Predicted Missing Values (Equation (2)). “Each color block represents a counter value”.

PMV Equation (2) was applied to calculate the possible number of missing values [20]

$$\text{PMV} = (\text{NR}_{td} - \text{PLC}_{td}) \cdot n \quad (2)$$

where

PMV is the number of predicted missing value,

NR_{td} is the average time difference of Node-RED in seconds,

PLC_{td} is the average time difference of PLC in seconds,

and n is the length of the test in seconds.

As the PMV model has already been proven to work, the primary focus of the current investigation is its accuracy. Different time intervals are used to determine how the accuracy of the model is affected. Table 8 shows the absolute and relative differences related to the model.

Table 8. The results of the difference between PMV and the real number of missing values for different time intervals for the first 9 days.

Hours	Sum of the Predicted Missing Values	Absolute Differences [pc]	Relative Differences [%]
24 h	7048.4	174.4	2.54%
12 h	7050.6	176.6	2.57%
8 h	7052.4	178.4	2.59%
6 h	7050.9	176.9	2.58%
3 h	7053.6	179.6	2.61%
1 h	7069.8	195.8	2.85%
30 min	7057.4	183.4	2.67%
25 min	7099.9	225.9	3.29%
10 min	7256.5	382.5	5.57%
5 min	7049.8	175.8	2.57%
2 min	7059.3	185.3	2.70%
1 min	6984.3	199.3	2.94%

It is important to note that the model cannot predict more significant anomalies like those introduced in Section 4.3. In the first nine days, there are several forward jumps and one peak. The model performed well at larger intervals (e.g., 24, 12, 8, and 3 h) with relative differences under 3%. However, it also works well with smaller intervals (1 to 5 min), with around 3% accuracy. This means that after 1 min of operation, the PMV model can predict approximately 95% of the missing values.

The rise in PMV error for the 10 min interval (5.57%) results from accumulated time drift irregularities and non-linear timestamp behavior over longer prediction gaps. The PMV approach is therefore most effective for short-interval reconstruction, where temporal behavior remains nearly linear and stable.

A clean dataset (remove the anomalies listed in Section 4.3) was used to assess the PMV model's performance better. Table 9 presents the predicted missing values for the 2.5 days period with smaller time intervals across the first seven distinct areas in Figure 5.

The results indicate that the PMV model performed reasonably well at predicting missing values for 15 min, 10 min, and 5 min intervals, with low relative differences. However, the performance significantly declined at 1 min and 2 min intervals, where the model's predictions diverged from the actual values, showing higher discrepancies (1.90% for 1 min and 1.36% for 2 min intervals). Based on the results, considering the 5 min operation, the PVM model has already reached maximum accuracy, which does not become significantly better.

Table 9. The results of the difference between PMV and the real number of missing values for small time intervals for the 2.5 days.

Hours	Sum of the Predicted Missing Value	Absolute Differences [pc]	Relative Differences [%]
15 min	2052.9	23.9	1.18%
10 min	2052.0	23.0	1.13%
5 min	2051.6	22.6	1.11%
2 min	2055.5	27.5	1.36%
1 min	2067.6	38.6	1.90%

It is worth emphasizing that the PMV model cannot account for data collection errors caused by external factors such as Node-RED jumps or power cuts. These external disruptions, which are not related to actual data gaps, represent a limitation of the PMV model.

The results demonstrate that temporal deviations, even when small, can accumulate over time and distort the true sequence of industrial events. The identification of drift, jumps, and missing intervals confirms that timing irregularities stem mainly from middleware-level scheduling rather than from physical network instability. The use of the PMV model for temporal gap analysis provides a novel method to quantify and diagnose these irregularities systematically.

These results demonstrate how middleware behavior directly affects the temporal reliability of industrial data systems. Even under stable network and processing conditions, asynchronous middleware scheduling can accumulate time drift and cause timestamp misalignment that disrupts data continuity. This phenomenon can cause many practical problems in an industrial environment. The application of the Predictive PMV model shows how timing anomalies can be detected and corrected to produce a clean, time-consistent dataset. This process provides a practical means to assess synchronization performance by focusing on communication-layer timing rather than solely on sensor accuracy. Overall, the findings generalize that ensuring reliable industrial data transmission requires continuous supervision of middleware timing behavior alongside traditional synchronization and fault-tolerance mechanisms.

5. Conclusions

This study examines the challenges of long-term, real-time data transmission in industrial environments, explicitly focusing on Siemens PLC transmitting data to MongoDB using Node-RED. Over both 24 h and 14-day observation periods, various temporal anomalies were identified, including time drift, forward and backward timestamp jumps, and cumulative time deviations, highlighting the inherent complexity of maintaining consistent timing in distributed systems. These anomalies reveal the sensitivity of real-time applications to processing delays and network instability, directly affecting timestamp accuracy, data sequencing, and the overall reliability of time-sensitive records. A key observation was the progressive time drift in Node-RED, where the recorded timestamps gradually deviated from real-time, especially over extended durations. This misalignment sometimes caused Node-RED to “catch up” via abrupt timestamp jumps, resulting in disordered sequences where newer data appeared to precede older entries. Forward jumps also introduced instances where timestamps effectively “lost” a second, compromising data continuity. A significant disruption occurred on day 8 due to a power outage, halting data transmission. During this event, the PLC retained its last recorded value (750,068) and resumed transmission after an 11.209 min gap—recovery required restarting both Node-RED and the TIA Portal, taking 5.840 min to restore regular operation. Although the system resumed without

a PLC reset, the event introduced unresolved gaps in the timeline, emphasizing the critical need for the robust handling of power failures and emergency stops in industrial systems. To address missing data, a Predictive Missing Value (PMV) model was applied to a clean dataset at 2.5 days to address missing data. The model showed high accuracy at shorter intervals (10 min and 5 min segments), with relative prediction errors as low as 1.11%. The novelty of this research lies in its explicit focus on temporal reliability—an often-overlooked dimension of industrial data quality. Unlike previous works that primarily evaluate the accuracy or trustworthiness of sensor values, this study isolates and analyzes the timing behavior of data transmission itself, demonstrating how asynchronous middleware processes such as Node-RED can distort timestamp order even when values remain correct. Furthermore, by employing the Predictive Missing Value (PMV) model as a diagnostic tool rather than a reconstruction mechanism, the study introduces a new methodological framework for identifying, quantifying, and interpreting missing intervals in long-term real-time data streams. This perspective bridges the gap between communication reliability and temporal precision, establishing a foundation for future synchronization-aware industrial monitoring systems. In conclusion, this research underscores the importance of designing data transmission systems that can tolerate timing inconsistencies, particularly in industrial IoT and real-time monitoring contexts. The observed time drift and compensatory timestamp jumps illustrate a fundamental challenge in distributed systems: preserving temporal integrity. For applications requiring high-frequency, time-ordered data, improvements are needed in platforms like Node-RED—potentially through enhanced synchronization protocols, timestamp correction mechanisms, or buffering strategies to accommodate network-induced delays without compromising data fidelity.

Author Contributions: A.H.: Conceptualization, Data Curation, Formal Analysis, Methodology, Visualization, Writing—Original Draft, Writing—review and editing. M.A.: Conceptualization, Project administration, Methodology, Supervision, Writing—review and editing. Z.P.: Conceptualization, Formal Analysis, Project administration, Supervision, Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available on request.

Acknowledgments: This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences. During the preparation of this work, the authors used ChatGPT 4o-mini in order to improve only readability and language grammar. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liao, Y.; Deschamps, F.; de Freitas Rocha Loures, E.; Ramos, L.F.P. Past, present and future of Industry 4.0—A systematic literature review and research agenda proposal. *Int. J. Prod. Res.* **2017**, *55*, 3609–3629. [[CrossRef](#)]
2. Xu, L.D.; Xu, E.L.; Li, L. Industry 4.0: State of the art and future trends. *Int. J. Prod. Res.* **2018**, *56*, 2941–2962. [[CrossRef](#)]
3. Blackstock, M.; Lea, R. Toward a Distributed Data Flow Platform for the Web of Things (Distributed Node-RED). In Proceedings of the WoT '14: 5th International Workshop on Web of Things, Cambridge, MA, USA, 8 October 2014; Association for Computing Machinery: New York, NY, USA, 2014; pp. 34–39.
4. Nițulescu, I.-V.; Korodi, A. Supervisory Control and Data Acquisition Approach in Node-RED: Application and Discussions. *IoT* **2020**, *1*, 76–91. [[CrossRef](#)]

5. Medina-Pérez, A.; Sánchez-Rodríguez, D.; Alonso-González, I. An Internet of Thing Architecture Based on Message Queuing Telemetry Transport Protocol and Node-RED: A Case Study for Monitoring Radon Gas. *Smart Cities* **2021**, *4*, 803–818. [\[CrossRef\]](#)
6. Ferrari, P.; Flammini, A.; Sisinni, E.; Rinaldi, S.; Brandão, D.; Rocha, M.S. Delay Estimation of Industrial IoT Applications Based on Messaging Protocols. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 2188–2199. [\[CrossRef\]](#)
7. Schmid, T.; Charbiwala, Z.; Friedman, J.; Cho, Y.H.; Srivastava, M.B. Exploiting manufacturing variations for compensating environment-induced clock drift in time synchronization. In Proceedings of the SIGMETRICS '08: 2008 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, Annapolis, MD, USA, 2–6 June 2008; Association for Computing Machinery: New York, NY, USA, 2008; pp. 97–108.
8. Barnes, J.A.; Chi, A.R.; Cutler, L.S.; Healey, D.J.; Leeson, D.B.; McGunigal, T.E.; Mullen, J.A.; Smith, W.L.; Sydnor, R.L.; Vessot, R.F.C.; et al. Characterization of Frequency Stability. *IEEE Trans. Instrum. Meas.* **1971**, *IM-20*, 105–120. [\[CrossRef\]](#)
9. Allan, D.W. Time and Frequency (Time-Domain) Characterization, Estimation, and Prediction of Precision Clocks and Oscillators. *IEEE Trans. Ultrason. Ferroelect. Freq. Contr.* **1987**, *34*, 647–654. [\[CrossRef\]](#)
10. Blackstock, M.; Lea, R. FRED: A Hosted Data Flow Platform for the IoT. In Proceedings of the MOTA '16: 1st International Workshop on Mashups of Things and APIs, Trento, Italy, 12–16 December 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1–5.
11. Steel, C.; Bejarano, C.; Carlson, J.A. Time Drift Considerations When Using GPS and Accelerometers. *J. Meas. Phys. Behav.* **2019**, *2*, 203–207. [\[CrossRef\]](#)
12. Baunach, M. Handling Time and Reactivity for Synchronization and Clock Drift Calculation in Wireless Sensor/Actuator Networks. In Proceedings of the 3rd International Conference on Sensor Networks, Lisbon, Portugal, 7–9 January 2014; SCITEPRESS—Science and Technology Publications: Lisbon, Portugal, 2014; pp. 63–72.
13. Hauweele, D.; Quoitin, B. Toward Accurate Clock Drift Modeling in Wireless Sensor Networks Simulation. In Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Miami Beach, FL, USA, 25–29 November 2019; ACM: Miami Beach, FL, USA, 2019; pp. 95–102.
14. Elsharief, M.; Abd El-Gawad, M.A.; Kim, H. FADS: Fast Scheduling and Accurate Drift Compensation for Time Synchronization of Wireless Sensor Networks. *IEEE Access* **2018**, *6*, 65507–65520. [\[CrossRef\]](#)
15. Wang, W.; Zhu, Q.; Gao, H. Drift Detection of Intelligent Sensor Networks Deployment Based on Graph Stream. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 1096–1106. [\[CrossRef\]](#)
16. Wang, Y.-L.; Han, Q.-L. Modelling and controller design for discrete-time networked control systems with limited channels and data drift. *Inf. Sci.* **2014**, *269*, 332–348. [\[CrossRef\]](#)
17. Manias, D.M.; Chouman, A.; Shami, A. Model Drift in Dynamic Networks. *IEEE Commun. Mag.* **2023**, *61*, 78–84. [\[CrossRef\]](#)
18. Žliobaitė, I.; Pechenizkiy, M.; Gama, J. An Overview of Concept Drift Applications. In *Big Data Analysis: New Algorithms for a New Society*; Japkowicz, N., Stefanowski, J., Eds.; Studies in Big Data; Springer International Publishing: Cham, Switzerland, 2016; Volume 16, pp. 91–114. [\[CrossRef\]](#)
19. Behnke, I.; Austad, H. Real-time performance of industrial IoT communication technologies: A review. *IEEE Internet Things J.* **2023**, *11*, 7399–7410. [\[CrossRef\]](#)
20. Hijazi, A.; Andó, M.; Pödör, Z. Data losses and synchronization according to delay in PLC-based industrial automation systems. *Heliyon* **2024**, *10*, e37560. [\[CrossRef\]](#)
21. Kopetz, H.; Steiner, W. Temporal Consistency of Data and Information in Cyber-Physical Systems. *arXiv*, 2024; arXiv:2409.19309.
22. Nelissen, G.; Pautet, L. Special issue on reliable data transmission in real-time systems. *Real-Time Syst.* **2023**, *59*, 662–663. [\[CrossRef\]](#)
23. Val, I.; Seijo, O.; Torrego, R.; Astarloa, A. IEEE 802.1 AS clock synchronization performance evaluation of an integrated wired-wireless TSN architecture. *IEEE Trans. Ind. Inform.* **2021**, *18*, 2986–2999. [\[CrossRef\]](#)
24. Lautenschlaeger, W.; Frick, F.; Christodoulopoulos, K.; Henke, T. A scalable factory backbone for multiple independent time-sensitive networks. *J. Syst. Archit.* **2021**, *119*, 102277. [\[CrossRef\]](#)
25. Schnierle, M.; Röck, S. Latency and sampling compensation in mixed-reality-in-the-loop simulations of production systems. *Prod. Eng. Res. Devel.* **2023**, *17*, 341–353. [\[CrossRef\]](#)
26. Ulagwu-Echefe, A.; Eneh, I.I.; Chidiebere, U. Mitigating the Effect of Latency Constraints on Industrial Process Control Monitoring Over Wireless Using Predictive Approach. *Int. J. Res. Innov. Appl. Sci.* **2021**, *6*, 82–87. [\[CrossRef\]](#)
27. Aslam, M.; Liu, W.; Jiao, X.; Haxhibeqiri, J.; Miranda, G.; Hoebeke, J.; Marquez-Barja, J.; Moerman, I. Hardware Efficient Clock Synchronization Across Wi-Fi and Ethernet-Based Network Using PTP. *IEEE Trans. Ind. Inform.* **2022**, *18*, 3808–3819. [\[CrossRef\]](#)
28. Bui, T.T.; Ngo, H.Q.T. Enhancing time synchronization between host and node stations using a fuzzy proportional-integral approach for EtherCAT-based motion control systems. *PLoS ONE* **2025**, *20*, e0324939. [\[CrossRef\]](#) [\[PubMed\]](#)
29. Hamma, F.; Venmani, D.P.; Singh, K.; Jahan, B. Synchronization in Industrial IoT: Impact of propagation delay on time error. In Proceedings of the 2023 IEEE Conference on Standards for Communications and Networking (CSCN), Munich, Germany, 6–8 November 2023; pp. 318–323. [\[CrossRef\]](#)

30. Hildebrandt, G.; Dittler, D.; Habiger, P.; Drath, R.; Weyrich, M. Data Integration for Digital Twins in Industrial Automation: A Systematic Literature Review. *IEEE Access* **2024**, *12*, 139129–139153. [\[CrossRef\]](#)
31. Hirsch, E.; Hoher, S.; Huber, S. An OPC UA-based industrial Big Data architecture. In Proceedings of the 2023 IEEE 21st International Conference on Industrial Informatics (INDIN), Lemgo, Germany, 18–20 July 2023; pp. 1–7. [\[CrossRef\]](#)
32. Mkiva, F.; Bacher, N.; Krause, J.; van Niekerk, T.; Fernandes, J.; Notholt, A. ICT-Architecture design for a Remote Laboratory of Industrial Coupled Tanks System. In Proceedings of the 13th Conference on Learning Factories (CLF 2023), Reutlingen, Germany, 9–11 May 2023. [\[CrossRef\]](#)
33. Patera, L.; Garbugli, A.; Bujari, A.; Scotece, D.; Corradi, A. A layered middleware for ot/it convergence to empower industry 5.0 applications. *Sensors* **2021**, *22*, 190. [\[CrossRef\]](#)
34. Adam, M.H.; Mahdin, H.; Mohd, H.A. Latency Analysis of WebSocket and Industrial Protocols in Real-Time Digital Twin Integration. *Int. J. Eng. Trends Technol.* **2025**, *73*, 120–135. [\[CrossRef\]](#)
35. Ma, L.; Wang, M.; Peng, K. A missing manufacturing process data imputation framework for nonlinear dynamic soft sensor modeling and its application. *Expert Syst. Appl.* **2024**, *237*, 121428. [\[CrossRef\]](#)
36. Noufel, S.; Maaroufi, N.; Najib, M.; Bakhouya, M. Hinge-FM2I: An approach using image inpainting for interpolating missing data in univariate time series. *Sci. Rep.* **2025**, *15*, 5389. [\[CrossRef\]](#)
37. Khattab, A.A.R.; Elshennawy, N.M.; Fahmy, M. GMA: Gap Imputing Algorithm for time series missing values. *J. Electr. Syst. Inf. Technol.* **2023**, *10*, 41. [\[CrossRef\]](#)
38. Zhang, Y.; Thorburn, P.J. Handling missing data in near real-time environmental monitoring: A system and a review of selected methods. *Future Gener. Comput. Syst.* **2022**, *128*, 63–72. [\[CrossRef\]](#)
39. Zhou, K.; Hu, D.; Hu, R.; Zhou, J. High-resolution electric power load data of an industrial park with multiple types of buildings in China. *Sci. Data* **2023**, *10*, 870. [\[CrossRef\]](#)
40. Yao, Z.; Zhao, C. FIGAN: A missing industrial data imputation method customized for soft sensor application. *IEEE Trans. Autom. Sci. Eng.* **2021**, *19*, 3712–3722. [\[CrossRef\]](#)
41. Gao, X.; Liu, Z.; Xu, L.; Ma, F.; Wu, C.; Zhang, K. Sensor Data Imputation for Industry Reactor Based on Temporal Decomposition. *Processes* **2025**, *13*, 1526. [\[CrossRef\]](#)
42. What is MongoDB?—Database Manual—MongoDB Docs. Available online: <https://www.mongodb.com/docs/manual/> (accessed on 18 October 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.