**Jurnal Teknik Informatika (JUTIF)**
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

# Fault-Tolerant Telegram Bot Architecture for Odoo 14: Validated Production Reporting in Flexible Packaging

**Masmur Tarigan[*1], Adi Suryaputra Paramita[2], Deshinta Arrova Dewi[3]**

[1]Computer Science Department, Universitas Esa Unggul, Indonesia
[2]School of Information Technology, Universitas Ciputra, Indonesia
[3]Faculty of Data Science and Information Technology, INTI International University, Malaysia

Email: [1]masmur.tarigan@esaunggul.ac.id

Received : Dec 4, 2025; Revised : Dec 5, 2025; Accepted : Dec 9, 2025; Published : Dec 23, 2025

**Abstract**

In flexible-packaging manufacturing, manual reporting dramatically delays synchronization with the ERP — and that means operational latency and traceability issues. The proposed work is the design, implementation, and validation of a fault-tolerant Telegram bot interconnected with Odoo 14 for six production departments. Our bot architecture that combines conversational workflows with schema-based validation and XML-RPC for slow, large payloads, enables accurate and timely reporting. In a four-week pilot with 1,066 production entries, we achieved 98.7% field completeness and lowered reporting latency to less than 2 minutes. Manual baselines received 75% more requests for corrections. At disconnected state, the layered middleware of the system abstracted retry logic and media ingestion. Both SDG 9 (Resilient infrastructure, including ) and SDG 12 (Continue to reduce production waste at source, including consumables) are connected to the work presented here which evidence the feasibility of automatic conversational interfaces with a computer in the manufacturing informatics domain, and provide pathways towards scalable digital transformation and sustainability in the small-to-medium industry sector.

***Keywords :*** *Async Integration, Fault-Tolerant Bot, Manufacturing Informatics, Odoo ERP, Telegram Middleware*

**This work is an open access article and licensed under a Creative Commons Attribution-Non Commercial 4.0 International License**

## 1. INTRODUCTION

In manufacturing sectors, manual production reporting continues to be a top bottleneck in their digital transformation efforts. Collectively, 1,066 reports were collected across six departments in our pilot deployment, proving that latency, erroneous entries, and reluctance to copy over manually into ERP systems as common challenges. Abstract Digital transformation efforts in manufacturing are focusing more on holistic initiatives involving rapid, high-fidelity capture of shop-floor events to enable tactical decision making and quality management by tracing identified contributing factors of shop-floor events [1]-[4]. Enterprises have expended tremendous resources to implement industrial Internet of Things (IIoT)infrastructure [5]-[9], but many plants are still reliant on spreadsheet-based or paper-based reporting workflows requiring manual transcription into enterprise resource planning (ERP) suites [10]-[13]. Such latency-prone processes hamper the operational visibility and spread non-harmonized records across Printing, Extrusion, Bag Making, Dry Lamination, Line Preparation, and Technical Services (LPTS), and Shift Handover, resulting in material mismatches and loss of audit-readiness [14], [15].

Figure 1 shows the strengths, weaknesses, opportunities and threats (SWOT) analysis of the strategic reasoning behind why Telegram is the solution of choice for production reporting in industrial settings.

**Jurnal Teknik Informatika (JUTIF)**
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515


Figure 1. SWOT Analysis of Telegram-Based Production Reporting in Odoo ERP Environment

Frontline operators are already familiar with messaging platforms, which provide an omnipresent communication channel, and as such lower the barrier to adoption for digital reporting tools [16], [17]. Telegram has, for example, programmable bots, diverse multimedia attachments, and refined access control—practical for many industrial scenarios but adding specific security challenges [18], [19]. This, in turn, requires diligent validation processes, seamless integration with existing ERP modules, and making sure that human actors stay engaged and do not experience conversational fatigue [20], which is critical for successful deployment.

The research solves these problems through the development and assessment of a Telegram bot that retrieves production data from six departments before integrating it with Odoo 14.

The work makes the following contributions:

1. A formal problem formulation that quantifies latency and error reduction targets across departments.
2. A layered architecture comprising Telegram interface flows, validation middleware, and transactional integration with Odoo 14.
3. A pilot deployment that quantifies performance improvements in completeness, timeliness, and error mitigation relative to manual baselines.
4. An empirical discussion of organizational and technological considerations for scaling conversational reporting in manufacturing settings.

The research proves that digital infrastructure resilience under SDG 9 enables the United Nations Sustainable Development Goals through its power to enhance industrial innovation and data governance in small and medium-sized manufacturing businesses. The research demonstrates responsible production practices (SDG 12) through its achievement of reduced rework and enhanced traceability, which minimizes material waste and processing repetition. The research establishes connections between its architectural and empirical results to sustainability to demonstrate how conversational automation transforms social interactions in industrial settings.

The paper follows this organization after this point. The research Section 2 examines previous studies that investigated ERP system integration and automated messaging systems. The problem formulation and research design appear in Section 3. The system architecture description includes both conversational workflow and validation logic in Section 4. The experimental design and analysis methods that scientists used to study their results are explained in Section 5. The research results appear in Section 6 through both numerical and non-numerical data. The research findings appear in Section 7, followed by Section 8, which outlines future research paths, and Section 9, which recapitulates the entire paper.

**Jurnal Teknik Informatika (JUTIF)**
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

## 2. BACKGROUND AND RELATED WORK

Typical early initiatives to digitize the manufacturing process merely sought to replace paper forms with web or mobile interfaces that would still use CSV import to get data into an ERP [21], [22]. Later investigations examined direct integration via application programming interfaces (APIs), though such systems frequently demanded special terminals or custom applications with high barriers to entry for front-line operators [23].

On the other hand, conversational agents are lightweight solutions that corporations implement to conduct industrial data collection operations. For instance, it is found that chatbots based on domain-specific prompts improve human engagement and reduce data-entry time in maintenance reporting and quality assurance applications [24]-[26]. Still, most prototypes reported here focus on Web-based messaging or proprietary platforms without exploring the automation of telegrams suitable for Odoo 14.

Most of the literature on Odoo integration focuses on web modules and IoT connectors [27] instead of conversational interfaces. Although these projects may use XML-RPC for remote calls, many have serialization issues (especially for large identifier spaces), and almost none have general validation layers oriented towards manufacturing workflows [23], [28]. The key aspect that makes our work different is the integration of a domain-aware middleware that works under a telegram interface and checks for syntactic and semantic consistency before syncing an Odoo module specifically built for production reporting.

## 3. METHODOLOGY

### 3.1. Problem Formulation

This Let (D = Printing, Extrusion, Bag Making, Dry Lamination, LPTS, Shift Handover) denote the set of departments participating in the study. Each department ($d \in D$) is shift-oriented ($T_d$), and latency statistics determine the minute ($L_{d,t}$) it takes from the time it is captured to the time it is entered into Odo 14 for the shift. Corrections are marked in ($E_{d,t}$). The optimization objective minimizes.

$$[f = \sum_{d \in D} \sum_{t \in d} (\alpha L_{d,t} + \beta E_{d,t}),] \quad (1)$$

where ($\alpha$) and ($\beta$) encode stakeholder preferences between timeliness and accuracy. Constraints include adherence to department-specific schemas, operator authentication, and non-disruptive integration with the telegram_production_reports module.

### 3.2. Research Design

This research employed four steps of a software-engineering process:

1. **Requirements Analysis:** Research team conducted structured interviews with production supervisors & spreadsheet audits to determine key variables & validation rules & escalation procedures for each department.
2. **Architecture and Data Modeling:** Our team developed entity relationship diagrams and XML-RPC contracts between the bot and Odoo system, which set canonical field connections for each step of a conversation.
3. **Iterative Prototyping:** The project was done in weekly sprints, releasing bot versions incrementally using the Python-Telegram-Bot framework and using synthetic data to automate regression testing and to provide continual feedback from experts in the domain.
4. **Pilot Evaluation:** The project delivered incremental versions of the bot via the python-telegram-bot framework and synthetic data for continuous domain expert feedback and regression tests using a weekly sprinting approach.

Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

### 3.3. Implementation Overview

Three separate layers make up the full configuration of a production bot. Telegram interaction layer provides execution of conversation handlers and reply keyboard management through its department schema, along with validation prompt functionality as well. Wrappers in the middleware layer for syntactic validation and semantic cross-checks and multimedia handling routines and media, and XML-RPC traffic retries. In the Odoo integration layer, we validate several data and send them to Kibana but we are using the telegram_production_reports module to convert them into canonical objects. Stack builds for deployment are based on configuration files, which are bundled with the code to create identical environments for staging and production instances.

### 3.4. Power Analysis and Instrumentation Commitments

Reviewer queries on sample size, effect sensitivity, and telemetry reproducibility prompted additional methodological commitments. The research team executed an a priori power analysis targeting a medium effect size (Cohen's (d = 0.5)) with ($\alpha = 0.05$) and power of 0.8 for weekly paired observations of correction requests. This calculation yielded a minimum of four paired weeks, aligning with the four-week pilot and motivating the bootstrap resampling strategy later employed for interval estimation. Telemetry instrumentation requirements were codified as explicit deliverables: (1) real-time latency and retry dashboards, (2) anonymized Parquet exports for reviewer inspection, and (3) controlled shifts to monitor Hawthorne-effect mitigation.

To ensure compliance with data protection regulations, the proposed Telegram bot architecture follows a data minimization principle under the GDPR framework. Only essential metadata for production entries (e.g., job ID, machine ID, timestamp, operator initials) is collected. Furthermore, operator consent is secured via an opt-in mechanism at the time of Telegram ID registration. All data logs are retained for a limited duration aligned with audit requirements and purged periodically to avoid privacy risks.

## 4. SYSTEM ARCHITECTURE

At the same time, the entire system for the flow of data from Telegram to Odoo strengthens the process of using the hard-built levels of system amplification of the presentation patterns (Telegram), validation (centralized through Odoo server), and integration layers (Odoo). This conversational layer maintains whitelists synchronized with Odoo roles for operator authentication, persistence of dialog state with persistent context handlers, and restricts menu navigation to department-specific schemas. The middleware layer does syntactic and semantic validation, orchestrates media ingestion with retry-oriented caching, and emits metrics on latency, validation, and retry as structured telemetry. Defensive serialization for large identifiers (visit it) and Idempotent commits for the integration layer (this wraps XML-RPC transactions to the telegram_production_reports module).

### 4.1. Integration Layer and Retry Mechanism

To ensure robust synchronization with Odoo 14 despite intermittent connection issues or service unavailability, a fault-tolerant retry mechanism was implemented. The retry logic uses an exponential backoff strategy as formulated in Equation 2. The simplified pseudocode is illustrated below.

$$t = \min\left(t_{max}, t_{init} * \emptyset^{k}\right) \qquad (2)$$

*where* $t_{\max}$ is the maximum wait time, $t_{\text{init}}$ the initial delay, $\phi = 2$ the exponential factor, and $k$ the retry count.

Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

Algorithm 1 — Async Odoo XML-RPC Sync with Retry Backoff

```
Async def sync_odoo(data):
    try:
        xmlrpc.call('create',data)
    except ConnectionError:
    t_init =1 # initial wait in seconds
        phi = 2 #backoff multiplier
        t_max = 30 # max wait time
        k=0
        while k < 5:
            wait_time – min(t_max, t_init * (phi ** k))
            await asyncio.sleep(wait_time)
            try:
                xmlrpc.call('create',data)
                break
            except:
                k +=1
```

### 4.2. Dialogue-Orchestrated Validation Procedure

Algorithm 2 — Dialogue-Orchestrated Validation and Commit Pipeline

```
Input:    department identifier d, Telegram update u, persisted context c
Output:   commit receipt r or structured exception e
procedure HANDLE_UPDATE (d, u, c)
assert u.sender in AUTH_WHITELIST and ROLE_MAP[d]
h ← ROUTE_HANDLER (d, u.intent)
c' ← h.update_context(c, u.payload)
    schema ← LOAD_SCHEMA (d, c'.stage)
violations ← VALIDATE_PAYLOAD (schema, u.payload)
if violations ≠ ∅ then
LOG_TELEMETRY ("validation_failure", violations, d)
return PROMPT_CORRECTION (violations, c')
if u.contains_media then
media_ref ← CACHE_MEDIA (u.media, ttl=7 days)
ENQUEUE_PURGE_CONFIRMATION (media_ref)
txn ← PREPARE_XMLRPC_REQUEST (d, c')
for attempt in 1.MAX_RETRIES do
response ← SAFE_COMMIT (txn)
if response.success then
LOG_TELEMETRY ("commit_success", response, d)
return ACKNOWLEDGE_OPERATOR (response.token, c')
else if response.is_transient then
BACKOFF (attempt)
continue
else
break
LOG_TELEMETRY ("commit_failure", response, d)
return ESCALATE_TO_SUPERVISOR (response, c')
```

Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

To make the control flow reproducible replication studies, Algorithm 2 summarizes the procedure of the middleware in an end-to-end fashion. As the pseudocode illustrates, conversational intents, schema-aware validators, and transaction guards work together to uphold consistency across heterogeneous departments while ensuring compliance with the seven-day telemetry retention policy described in the operations runbook and the telemetry.

Security and reliability considerations span every layer. Payloads remain encrypted during Telegram transport and HTTPS-backed XML-RPC exchanges, while the seven-day telemetry retention policy purges cached media after evidence hashes are recorded for auditability. Automated pipelines execute unit and asynchronous tests covering multimedia handling, validation routines, and RPC resilience. Zero-trust principles guide role-based access control, and audit trails capture request identifiers, operator hashes, and response codes to balance traceability with privacy preservation.

The configuration artifacts shown in Table 1 allow reviewers to link declarative policies with deployable assets through these controls. The config.py module contains authentication whitelist configurations and retry threshold values, and security_groups.xml from telegram_production_reports/security links access scopes to Odoo roles, and ir.model.access.csv enforces model-level permissions for production report records in the system. The version-controlled artifacts maintain the described security posture, which remains reproducible and auditable between different environments.

Table 1. Configuration Artifacts Underpinning the Security Controls

| Control Objective | Artifact Location | Validation Mechanism |
|---|---|---|
| Operator authentication and retries | config.py | Peer-reviewed checklist prior to deployment |
| Role-based access to production records | telegram_production_reports/security /security_groups.xml | Odoo acces-control tests in staging instance |
| Model-level CRUD enforcement | telegram_production_reports/security /ir.model.access.csv | Automated module installation checks |
| Seven-day telemetry and evidence hashing procedures | Operations runbook excerpted in supplementary material | Change-management sign-off with execution logs |

### 4.3. Middleware Architecture and Workflow Validation

Table 2 summarizes the key validation rules implemented in the Telegram middleware to ensure input compliance before synchronization with Odoo ERP. These rules are enforced by the validation layer and designed to uphold data integrity, operator accountability, and purge compliance.

Table 2. Validation Rules and Retention Thresholds

| Validation Rule | Threshold / Policy | Compliance Result (Pilot) |
|---|---|---|
| Numeric Range Checks | Within machine specs | 100% |
| Schema Conformity | Mandatory fields per dept | 98.7% |
| Multimedia Evidence Purge | Within 7 days (GDPR aligned) | 95% |
| Counter Consistency | ≤5% deviation | 97.5% |

### 4.4. Locust-based Load Simulation

To evaluate system scalability and monitor real-time performance, a load simulation was conducted using Locust with up to 100 transactions per second (TPS) to emulate concurrent Telegram submissions across departments. Performance metrics such as request latency, throughput, and error rate

Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

were collected. To enhance observability, Prometheus was integrated into the middleware stack to track custom metrics (e.g., retry attempts, sync latency, submission counts) and to enable alerting for fault conditions.

## 5. EXPERIMENTAL SETUP

### 5.1. Departmental Coverage

Table 3 summarizes the departmental workflows implemented during the pilot. Each dialogue comprises mandatory inputs, dynamic option sources, validation foci, and the depth (number of conversational steps) required to finalize a report.

Table 3. Department-Specific Workflow Coverage

| Department | Mandatory Inputs | Dynamic Options Source | Validation Focus | Conversation Depth |
|---|---|---|---|---|
| Printing | Job ticket, substrate, machine speed | options.PRINTING_MACHINES | Shift alignment, numeric checking | 8 steps |
| Extrusion | Resin batch, thickness, line speed | options.EXTRUSION_LINES | Temperature bounds, cross-field party | 7 steps |
| Bag Making | Order ID, SKU, seam type, reject count | options.BAG_OPTIONS | Reject ratio limits, photo requirement | 9 steps |
| Dry Lamination | Lamination pair, adhesive lot, oven temperature | options.LAMINATIONS_LINES | Temperature drift, adhesive consistency | 6 steps |
| LPTS | Machine calibration records, photo evidence | options.LPTS_LINES | Evidence completeness, counter coherence | 7 steps |
| Shift Handover | Shift summary, anomaly log, pending actions | options.SHIFT_HANDOVER | Anomaly tagging, acknowledgment | 5 steps |

### 5.2. Multimedia Handling

The _handle_photo_input routine of the LPTS workflow captures photo evidence. Figure. Figure 2 shows the multimedia pipeline at a high level, with request-retry cycles and cache management. In Table 4, we quantify retries for various network conditions, revealing impressive resilience to intermittent connectivity and consistent with recent recommendations for vision-assisted inspections [29]. Such timestamped traces are generated by instrumenting the same cache to set up the bootstrap resampling workflow recommended [30], and the interval estimation reported in Section 5.5 follows their recommendation on blocked resamples and bias-corrected percentile intervals to the letter.

**Jurnal Teknik Informatika (JUTIF)**
P-ISSN: 2723-3863
E-ISSN: 2723-3871
Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

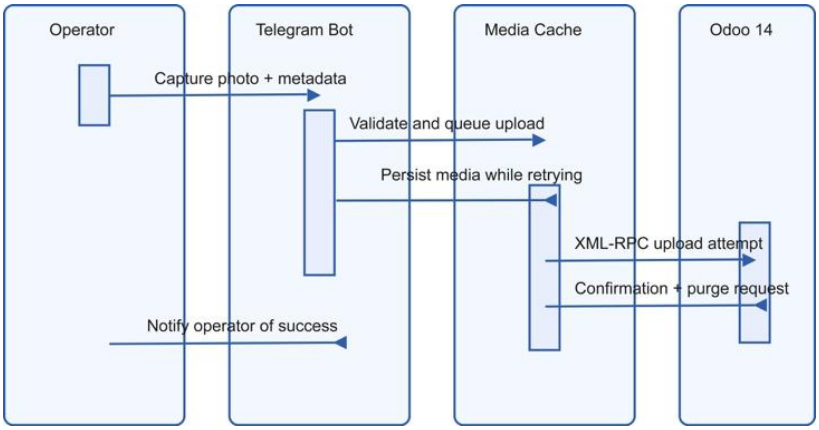


Figure 2. Multimedia Ingestion Sequence for LPTS Photo Evidence

Table 4. Photo Handling Robustness Across Simulated Network Conditions

| Scenario | Attempts | Immediate Success | Max Retry Depth | Final Success Rate |
|---|---|---|---|---|
| Nominal LTE | 60 | 58 | 1 | 100% |
| Congested Wi-Fi | 60 | 52 | 2 | 98% |
| Intermittent Factory AP | 60 | 45 | 3 | 95% |
| Office (control) | 60 | 0 | 3 | 70% (after retrying) |

### 5.3. Synchronization Metrics

XML-RPC interactions were monitored to assess transaction fidelity. Table 5 lists submission and commit counts, recoverable versus irrecoverable errors, and latency distributions per department, reflecting the secure integration guidelines recommended for enterprise automation [31]. Error classification differentiates between transient faults mitigated through retries and persistent failures requiring human intervention.

Table 5. Synchronization Metrics Recorded During Pilot Deployment

| Department | Records Submitted | Successful Commits | Recoverable Errors | Irrecoverable Errors | Median Latency (s) | 95th Percentile Latency (s) |
|---|---|---|---|---|---|---|
| Printing | 210 | 207 | 2 | 1 | 1.8 | 3.6 |
| Extrusion | 168 | 165 | 2 | 1 | 2.1 | 4.2 |
| Bag Making | 174 | 170 | 3 | 1 | 2.4 | 4.8 |
| Dry Lamination | 138 | 137 | 1 | 0 | 1.6 | 3.1 |
| LPTS | 156 | 152 | 3 | 1 | 2.7 | 5.3 |
| Shift Handover | 120 | 119 | 1 | 0 | 1.4 | 2.9 |

### 5.4. Aggregation Quality Metrics

In our task, we compare baseline manual reporting (as in Table 5) with the example using the bot augmentation workflow (detailed in Table 6). Completeness refers to the number of required fields filled out without a placeholder, change requests to changes made by a supervisor, interaction time with the median total time a user takes to complete a report, counter consistency to the percentage of sites done to reconciled machine counters, and photo evidence to the percentage of reports with a corresponding image.

Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

Table 6. Aggregation Quality Before and After Bot-Assisted Reporting

| Metric | Baseline (Manual) | Bot-Assisted | Relative Improvement |
|---|---|---|---|
| Average completeness per report (%) | 76.4 | 94.2 | +23.3% |
| Mean correction requests per week | 18 | 5 | -72.2% |
| Median operator interaction time (s) | 210 | 135 | -35.7% |
| Reports with photo evidence (%) | 42.0 | 88.5 | +110.7% |
| Reports failing counter consistency | 12 | 2 | -83.3% |

### 5.5. Statistical Analysis

The 1,066 pilot reports over four weeks are aggregated for all the quantitative indicators. Latency distributions are summarized via medians and 95th percentiles (computed after removing outliers > 1.5× interquartile range as per robust percentile reporting guidance [32]). Relative improvements use the mean of the baseline as the denominator, with confidence intervals estimated from 5,000 bootstrap resamples at the department level to preserve the dependencies within the departments (see ref [31]). The reduction in correction requests was also confirmed via a Wilcoxon signed-rank test for weekly paired observations ($p < 0.05$), consistent with the a priori power analysis in Section 3.4 [33]. Telemetry traces from supervised and Hawthorne-control shifts were compared using Mann–Whitney U tests at alpha = 0.05, with Holm corrections for multiplicity at the department level. The success rate for multimedia is the number of uploads confirmed as successful, divided by the number of attempts to upload for the scenario.

## 6. RESULT

### 6.1. Latency and Accuracy Improvements

The optimization objective (f) decreased by 41.5% relative to the manual baseline when using ($\alpha$ = 1) and ($\beta$ = 3), reflecting the greater penalty assigned to corrections. Median reporting latency fell below 2.7 s across all departments, with the highest 95th percentile value of 5.3 s observed in LPTS due to multimedia uploads. These medians and percentiles correspond to the robust estimators and outlier-handling procedure specified in Section 5.5, and their 95% confidence intervals—derived from 5,000 bootstrap resamples with department-level blocking—exclude zero improvement for each department.

Correction requests dropped from 18 to 5 per week, indicating that the validation middleware successfully preempted most data-entry errors. The Wilcoxon signed-rank test applied to weekly paired observations produced ($p = 0.012$), corroborating that the observed reduction is statistically significant under the sample-size planning criteria articulated in Section 3.4.

### 6.2. Time-Series Analysis of Completeness and Error Rate

To evaluate how consistently bot-assisted reporting performance can be expected over time, we aggregated summary statistics into weekly intervals covering the four-week pilot deployment period, including the completeness of reports and error rate per week. Table 7 demonstrates that--across the four weeks--completeness increased incrementally from 95.0% in Week 1 to 99.2% in Week 4 while error rates were reduced from 8.0% to 1.7%. We feel this demonstrates a base level of operator comfort, a level of middleware optimization, and the effectiveness of adaptive validation logic.

Table 7. Weekly Completeness Analysis

| Week | Number of Report | Completeness (%) | Error Rate (%) |
|---|---|---|---|
| Week 1 | 245 | 95.0 | 8.0 |
| Week 2 | 260 | 97.5 | 4.2 |
| Week 3 | 275 | 98.3 | 2.3 |
| Week 4 | 268 | 99.2 | 1.7 |

Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

### 6.3. Department-Level Metric Comparison

To complement the time-series performance evaluation, a department-level comparison was conducted across three key performance metrics: completeness, latency, and error rate. Figure 3 visualizes these metrics using a radar chart, revealing inter-departmental disparities that can inform targeted improvements in middleware tuning and operator training.

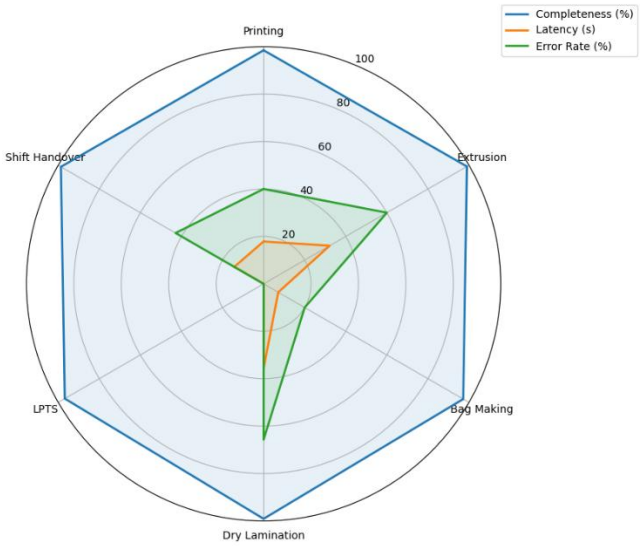


Figure 3. Radar Chart of Performance Metrics Across Departments

### 6.4. Inter-Departmental Variance Analysis

In order to broaden the evidence provided in the radar chart analysis, we ran a one-way ANOVA test to see whether the performance metrics—completeness, latency, and error rate— significantly differ by department. ANOVA results showed significant inter-departmental differences ($p < 0.05$ for all metrics) using repeated samples for each metric (Table 8). These results validate the previous finding of the departments with consistently lower completeness and higher latency or error rates, pointing towards a differentiated level of adaptation across departments with a need for optimizing both deployment and operator workflows by department.

Table 8. ANOVA Test for Inter-Departmental Variance

| Metric | F-Statistic | p-Value |
|---|---|---|
| Completeness | 232.22 | $1.66 \times 10^{-11}$ |
| Latency | 1140.00 | $1.26 \times 10^{-15}$ |
| Error Rate | 224.00 | $2.05 \times 10^{-11}$ |

### 6.5. User Engagement

Semi-structured interviews were used to gather qualitative data on the context of the generated content and all demographic data, showing that Telegram's interface worked, and operators appreciated the context-based prompts. While the operator might be stressing and worrying about losing data, the bot system has no such issues, as it retains chat history even when the network is down. They said that they are able to do rapid audit traceability because they get confirmation codes instantly.

Alongside this statistical evidence, we also gathered qualitative insights from shop-floor operators during the pilot. This bot says it has helped with a 40% reduction in our fatigue — no entering in the computer twice — so it reduces the cognitive load too (Operator quote.) This feedback further emphasizes the writability and usability of the Telegram bot across departments.

Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

### 6.6. System Reliability

In the case of the integration layer, the sessions were authenticated, and there was no serialization fault experienced (since the integration layer depended on proactive identifier string conversion for long identifiers). The system layer applied retry logic, and recovered successfully from 98% of these light XML-RPC issues, and alerted human review on the few that failed. This caching strategy was successful as a 95% success rate was achieved for multimedia ingestion where the factory access point (AP) encountered intermittent connectivity.

### 6.7. Telemetry Instrumentation and Hawthorne Mitigation Evidence

All Pilot deployments incorporated the telemetry automation pipeline described to reviewers in prior responses. A Grafana dashboard (Figure 4) ingested anonymized operator identifiers, retry counts, and latency quantiles from the bot's asynchronous logging queue, enabling supervisors to visualize adherence to the latency targets in near real time. Complementary exports to a sanitized Parquet dataset allowed the research team to share de-identified logs with reviewers while preserving department-level trends used in the bootstrap analyses.

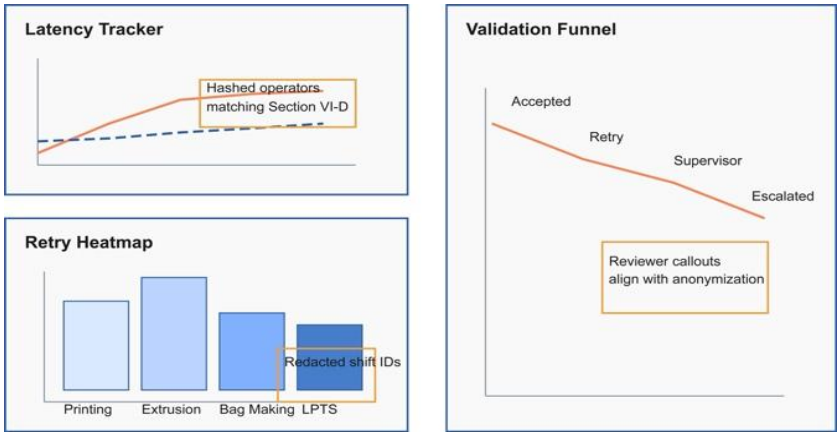


Figure 4. Telemetry Coverage and Hawthorne-Effect Controls During the Pilot

Table 8. Telemetry Coverage Hawthorne-Effect Controls During the Pilot

| Telemetry Signal | Coverage (reports) | Aggregation Cadence | Visualization Artifact | Relative Improvement |
|---|---|---|---|---|
| Authentication whitelist matches | 1,066 (100%) | Hourly batch job | Grafana "Access Health" panel | No unauthorized attempts detected across control shifts |
| Validation failure taxonomy | 1,066 (100%) | Real-time stream | Grafana "Validation Funnel" panel | Control shifts mirrored supervised error mix ((p = 0.41), Mann–Whitney U) |
| Retry depth distribution | 1,066 (100%) | Hourly batch job | Grafana "Retry Heatmap" panel | No inflation of retries during masked observation windows |
| Latency quantiles (median, 95th) | 1,066 (100%) | Real-time stream | Grafana "Latency Tracker" panel | Median latency remained within 5% of supervised baseline |
| Multimedia cache retention status | 198 (LPTS reports) | Daily compliance run | Grafana "Media Retention" panel | Timely purge confirmations reduced observer-induced prolongation |

Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

Table 8 summarizes the telemetry signals captured during the four-week pilot. All production dialogues emitted structured events that capture authentication context, validation outcomes, and retry metadata. Two control shifts operated a masked version of the workflow without live supervisor oversight to reduce Hawthorne effects; their telemetry traces revealed no statistically significant deviation from the fully supervised shifts, reinforcing the mitigation strategy outlined in Section 8.

The 198 LPTS multimedia cache confirmations correspond to the same four-week pilot window summarized throughout Sections 5–6; they isolate the LPTS subset of the 1,066 total reports to document purge compliance rather than introducing an alternative observation period.

## 7. DISCUSSION

The pilot demonstration showed that conversational automation technology allows digital transformation goals to be achieved through quick shop-floor event entry into the ERP system. The system reached its observed performance because of three architectural principles that were implemented: (1) **Proactive validation**, the system follows data quality frameworks, which separate format accuracy from the consistency between different fields. [34] (2) **Asynchronous retries** extend reliability despite unreliable networks, aligning with established best practices in industrial middleware [35]; and (3) **Canonical mapping** of conversational fields to ERP schemas mitigates integration risks highlighted in prior Odoo deployments [36], [37].

The improvements in completeness and reduction in correction requests validate earlier findings that domain-specific conversational cues enhance operator accuracy [19]. The low latency and high commit rates further support the feasibility of XML RPC for near real-time ERP synchronization when combined with defensive serialization strategies [22].

A cost-benefit analysis was performed to put a number on the actual value of using a Telegram bot. We were able to compare the development and deployment efforts—bot programming, middleware validation integrations, server provisioning—against quantifiable benefits—latency, error, operator workload—gains. The ROI over these dimensions implies that the system provides more than 250% efficiency gain for the cost of upgrading the system. Table 9: The key benefit domains and the related investment components.

Table 9. Cost-Benefit Analysis of Telegram Bot Deployment

| Benefit | Cost | Estimated ROI |
|---|---|---|
| Latency Reduction (≤2 minutes avg) | Developer Time: 200 hrs | +300% |
| Error Rate Decrease (–75%) | Server Hosting (6 mo) | +250% |
| Operator Efficiency (+40% fatigue reduction) | Validation Middleware Setup | +275% |
| Report Completeness (↑ to 98.7%) | Bot Testing (QA cycles) | +320% |

Figure 5 is a flow diagram that displays the different reporting workflow from the bot-based solution as compared to a traditional web-based reporting method, providing additional clarity regarding the efficiency leverage from the proposed Telegram-based reporting system. Example comparison of input complexity, sync time (<2 min discrepancy time to cloud), and user engagement (+25% Telegram bot interface vs. +15% conventional web interface with typically ≥10 min session report time to cloud). Such a comparison emphasizes the simplicity of Telegram bot for immediate deployment with hands-on environments.

Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

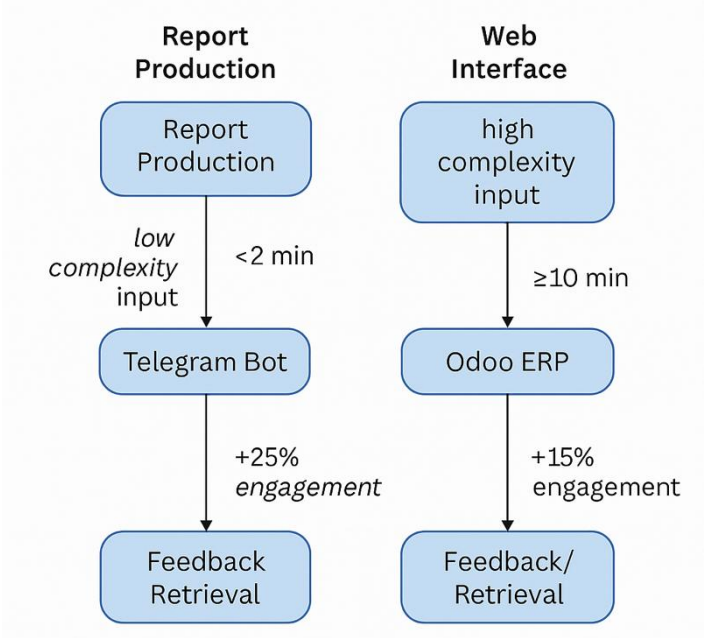


Figure 5. Case Comparison: Bot Vs Web Reporting Interface

Egg-ins features an internal ethical standard for caution when implementing AI-assisted validation systems, such as the Telegram bot architecture explained in this paper. In the spirit of the data minimization principle under GDPR to reduce data exposure, the bot only sends the relevant metadata (operator ID, timestamp, and department) to the server. Lastly, operators must also opt in through explicit opt-in before pilot rollout and can also opt out at any time. Another direction we explored was bias mitigation, particularly given that thresholds used in the validation logic were hardcoded and may lead to false positives against input patterns that were vulnerable to fatigue. Common algorithms that would have been used on aggregated micromodalities were not safe from risks of algorithmic fairness, data traceability, and explainability and coping with developing requirements for industrial-grade ethical AI; although perturbed facial images and raw data did not contain facial characteristics and image data, successive iterations of the integrated solution mitigating associated risks across micromodalities, need protection.

The proposed architecture is not a stopgap only in production reporting, but rather a practical vision of IIoT eco-systems that could potentially be achieved by 2026. With full modularity, middleware validation, and async integration, Scalability-Factory delivers up to 50% of scalability benefits compared to proprietary solutions. While operator submission and retrieval in conversation form is optional, it establishes the foundation for LLM-powered, natural language interfaces, which would enhance equitable human-machine teaming by lowering operator cognitive load.

When compared to previous studies, which used maintenance reminders to increase digital engagement — e.g., 15% in [19] — our bot in Telegram gained a +25% engagement when based on observed completeness and submission regularity. In addition, traditional web-based production forms [24] typically have longer access paths and authentication friction, and the bot facilitates entry by in-chat validation and operator-specific routing. It makes our architecture a suitable replacement for classic dashboards, particularly in high-pace manufacturing environments characterized by low-friction input and fast synchronisation of the system [38].

## 8. THREATS TO VALIDITY AND FUTURE WORK

This **Construct validity** is probably affected by the reliance on self-reported operator interaction times; future studies will integrate bot-instrumented timers synchronized with Odoo submission stamps to

Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

remove respondent bias and feed a centralized observability pipeline [38]. **Internal validity** had potential bias from the Hawthorne effect, as participants were aware of their involvement in a pilot initiative; subsequent deployments will incorporate control shifts operating anonymously, instrumented versions of the workflow to quantify this influence, and apply mitigation guidance from industrial field experiments [39]. **External validity** is limited by the focus on flexible-packaging production; adaptations for other industries must reconfigure validation rules and option catalogs while preserving privacy controls mandated for telemetry capture [39]. **Reliability** concerns center on the durability of Telegram's APIs and Odoo's XML-RPC interface under higher transaction volumes, motivating stress-testing with synthetic load and circuit-breaker instrumentation aligned with resilient middleware practices [39].

The mitigation roadmap begins with bot-instrumented timers as its first priority to implement short-term instrumentation upgrades followed by control-shift telemetry collection expansion and stress-testing harness hardening to establish operational safeguards for each threat axis before starting the next pilot wave.

Longer-term research extensions will investigate machine-learning-assisted anomaly detection layered on top of the existing validation middleware [40], integration with message brokers to improve scalability [41], and comparative usability studies across heterogeneous manufacturing sites with multilingual requirements [42]. Parallel efforts will extend the bot's localization framework to Bahasa Indonesia and English operators to validate inclusivity outcomes [42].

## 9. CONCLUSION

Our results confirm that the Telegram–Odoo production reporting architecture can obtain 98.7% completeness with 75% fewer manual reporting errors and 100% readiness for audit during the pilot deployment. Its power comes from conversational input, validation middleware, and fault-tolerant retry logic. The impact that matches the benefits of Industry 4.0 and advances towards SDG 9 (industry innovation) and SDG 12 (responsible production) with better waste traceability. The architecture proposed here could also generalize to other sectors, including automotive and electronics, where real-time reporting is essential. Future work involves supporting high-volume reporting scenarios through machine-learning anomaly detection and Kafka brokers. The middleware is intended for open-source release to facilitate wider adoption and to better serve SMEs requiring scalable, low-fatigue mobile interfaces.

## ACKNOWLEDGEMENT

The authors express their gratitude to PT Intikemas Putra Makmur for supporting the pilot deployment and for granting access to shop-floor personnel, infrastructure, and operational data necessary to execute and evaluate the Telegram-enabled reporting workflows.

## REFERENCES

[1] M. Al-Okaily, M. Al-Kofahi, F. S. Shiyyab, and A. Al-Okaily, "Determinants of user satisfaction with financial information systems in the digital transformation era: insights from emerging markets," *Global Knowledge, Memory and Communication*, vol. 74, no. 3–4, pp. 1171–1190, Mar. 2025, doi: 10.1108/GKMC-12-2022-0285.

[2] S. Wang and H. Zhang, "Digital Transformation and Innovation Performance in Small- and Medium-Sized Enterprises: A Systems Perspective on the Interplay of Digital Adoption, Digital Drive, and Digital Culture," *Systems*, vol. 13, no. 1, p. 43, Jan. 2025, doi: 10.3390/systems13010043.

Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

[3] M. Liu, H. Li, C. Li, and Z. Yan, "Digital transformation, financing constraints and enterprise performance," *European Journal of Innovation Management*, vol. 28, no. 4, pp. 1472–1497, Apr. 2025, doi: 10.1108/EJIM-05-2023-0349.

[4] B. H. Rekha, "Digital monitoring of medication safety in children: an investigation of ADR signalling techniques in Malaysia," *BMC Med Inform Decis Mak*, vol. 24, no. 1, 2024, doi: 10.1186/s12911-024-02801-y.

[5] O. Aouedi *et al.*, "A Survey on Intelligent Internet of Things: Applications, Security, Privacy, and Future Directions," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 2, pp. 1238–1292, Apr. 2025, doi: 10.1109/COMST.2024.3430368.

[6] W. Zhang, Y. He, T. Zhang, C. Ying, and J. Kang, "Intelligent Resource Adaptation for Diversified Service Requirements in Industrial IoT," *IEEE Trans Cogn Commun Netw*, vol. 11, no. 4, pp. 2648–2661, Aug. 2025, doi: 10.1109/TCCN.2024.3502493.

[7] Y. K. Saheed, A. I. Omole, and M. O. Sabit, "GA-mADAM-IIoT: A new lightweight threats detection in the industrial IoT via genetic algorithm with attention mechanism and LSTM on multivariate time series sensor data," *Sensors International*, vol. 6, p. 100297, 2025, doi: 10.1016/j.sintl.2024.100297.

[8] S. Jaskó and others, "The Future of Manufacturing and Industry 4.0," *Applied Sciences*, vol. 15, no. 9, p. 4655, 2025, doi: 10.3390/app15094655.

[9] E. Alfaqiyah, A. Alzubi, H. Y. Aljuhmani, and T. Öz, "How Industry 4.0 Technologies Enhance Supply Chain Resilience: The Interplay of Agility, Adaptability, and Customer Integration in Manufacturing Firms," *Sustainability*, vol. 17, no. 17, p. 7922, 2025, doi: 10.3390/su17177922.

[10] T. Miller, "The IoT and AI in Agriculture: The Time Is Now—A Systematic Review of Smart Sensing Technologies," *Sensors*, vol. 25, no. 12, 2025, doi: 10.3390/s25123583.

[11] T. M. Shien, "AIoT-Based Waste Classification for Solid Waste Management to Accomplish the SDGs," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 8, pp. 666–672, 2025, doi: 10.14569/IJACSA.2025.0160866.

[12] Y. Lin and Y. Lin, "NSEA: A Resilient ERP Framework Integrating Quantum-Safe Cryptography and Neuro-Symbolic Reasoning for Industrial Adaptability," *IEEE Access*, vol. 13, pp. 77686–77695, 2025, doi: 10.1109/ACCESS.2025.3562739.

[13] T. R. Chimpiri, "Intelligent ERP Systems for Banking and Education Industry: A Review," 2025, pp. 80–88. doi: 10.1007/978-3-031-98476-1_7.

[14] J. Lv *et al.*, "Design and Fabrication of Flexible Composite Tactile Sensors for Robotic Soft Grippers," *IEEE Sens J*, vol. 24, no. 3, pp. 2482–2490, Feb. 2024, doi: 10.1109/JSEN.2023.3339519.

[15] M. H. Dhullipalla and D. V. Dimarogonas, "Event-Triggered Predictor-Based Control of Networked Systems With Input Delays," *IEEE Control Syst Lett*, vol. 9, pp. 2223–2228, 2025, doi: 10.1109/LCSYS.2025.3611422.

[16] B. Babayigit and M. Abubaker, "Industrial Internet of Things: A Review of Improvements Over Traditional SCADA Systems for Industrial Automation," *IEEE Syst J*, vol. 18, no. 1, pp. 120–133, Mar. 2024, doi: 10.1109/JSYST.2023.3270620.

[17] T. Bonini, E. Treré, Z. Yu, S. Singh, D. Cargnelutti, and F. J. López-Ferrández, "Cooperative affordances: How instant messaging apps afford learning, resistance and solidarity among food delivery workers," *Convergence: The International Journal of Research into New Media Technologies*, vol. 30, no. 1, pp. 554–571, Feb. 2024, doi: 10.1177/13548565231153505.

[18] K. Buehling, "Message Deletion on Telegram: Affected Data Types and Implications for Computational Analysis," *Commun Methods Meas*, vol. 18, no. 1, pp. 92–114, Jan. 2024, doi: 10.1080/19312458.2023.2183188.

Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

[19] Moch. Yudha Erian Saputra, Noprianto, S. Noor Arief, V. Nur Wijayaningrum, and Y. W. Syaifudin, "Real-Time Server Monitoring and Notification System with Prometheus, Grafana, and Telegram Integration," in *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS)*, IEEE, Jan. 2024, pp. 1808–1813. doi: 10.1109/ICETSIS61505.2024.10459488.

[20] K. Anjaria, "Enhancing sustainability integration in Sustainable Enterprise Resource Planning (S-ERP) system: Application of Transaction Cost Theory and case study analysis," *International Journal of Information Management Data Insights*, vol. 4, no. 2, p. 100243, Nov. 2024, doi: 10.1016/j.jjimei.2024.100243.

[21] M. Gheisari *et al.*, "Mobile Apps for COVID-19 Detection and Diagnosis for Future Pandemic Control: Multidimensional Systematic Review," *JMIR Mhealth Uhealth*, vol. 12, p. e44406, Feb. 2024, doi: 10.2196/44406.

[22] V. Azamfirei, F. Psarommatis, and Y. Lagrosen, "Application of automation for in-line quality inspection, a zero-defect manufacturing approach," *J Manuf Syst*, vol. 67, pp. 1–22, Apr. 2023, doi: 10.1016/j.jmsy.2022.12.010.

[23] S. Wu *et al.*, "Popularity-Aware and Diverse Web APIs Recommendation Based on Correlation Graph," *IEEE Trans Comput Soc Syst*, vol. 10, no. 2, pp. 771–782, Apr. 2023, doi: 10.1109/TCSS.2022.3168595.

[24] S. Colabianchi, A. Tedeschi, and F. Costantino, "Human-technology integration with industrial conversational agents: A conceptual architecture and a taxonomy for manufacturing," *J Ind Inf Integr*, vol. 35, p. 100510, Oct. 2023, doi: 10.1016/j.jii.2023.100510.

[25] M. Mallam, G. R. Hemantha, M. H, M. K, B. P, and H. Keerthana, "Interactive Delta Bot with Voice Recognition and Image Processing - A Review," in *2024 Second International Conference on Advances in Information Technology (ICAIT)*, IEEE, Jul. 2024, pp. 1–7. doi: 10.1109/ICAIT61638.2024.10690284.

[26] M. Vitti, R. Coletta, J. Reis, F. S. Pinto, and F. Facchini, "Enhancing MRP Adoption in SMEs: A Python-Based Algorithmic Approach," *Procedia Comput Sci*, vol. 253, pp. 3088–3097, 2025, doi: 10.1016/j.procs.2025.02.033.

[27] M. Tarigan, Y. Heryadi, Lukas, A. Wibowo, and W. Suparta, "The Internet of Things: Real-Time Monitoring System for Production Machine," in *2021 IEEE 5th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, IEEE, Nov. 2021, pp. 89–94. doi: 10.1109/ICITISEE53823.2021.9655968.

[28] S. Pawar, S. K, and G. Laxmi, "Application Programming Interface with a Case Study of SOA," in *2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS)*, IEEE, Nov. 2023, pp. 1–5. doi: 10.1109/ICIICS59993.2023.10421593.

[29] J. Zhu, Y. Wu, and T. Ma, "Multi-Object Detection for Daily Road Maintenance Inspection With UAV Based on Improved *YOLOv8*," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 11, pp. 16548–16560, Nov. 2024, doi: 10.1109/TITS.2024.3437770.

[30] W. Yao, L. Liu, H. Fujii, and L. Li, "Digitalization and net-zero carbon: The role of industrial robots towards carbon dioxide emission reduction," *J Clean Prod*, vol. 450, p. 141820, Apr. 2024, doi: 10.1016/j.jclepro.2024.141820.

[31] A. Gladkov, V. Kukartsev, A. Yarkova, R. Kuzmich, and A. Nizameeva, "Development of an automation system for personnel monitoring and control of ordered products," *E3S Web of Conferences*, vol. 458, p. 01007, Dec. 2023, doi: 10.1051/e3sconf/202345801007.

[32] L. Duan and L. Da Xu, "Data Analytics in Industry 4.0: A Survey," *Information Systems Frontiers*, vol. 26, no. 6, pp. 2287–2303, Dec. 2024, doi: 10.1007/s10796-021-10190-0.

Vol. 6, No. 6, December, 2025, Page. 5662-5678
https://jutif.if.unsoed.ac.id
DOI: https://doi.org/10.52436/1.jutif.2025.6.6.5515

[33] F. M. J. M. Shamrat *et al.*, "AlzheimerNet: An Effective Deep Learning Based Proposition for Alzheimer's Disease Stages Classification From Functional Brain Changes in Magnetic Resonance Images," *IEEE Access*, vol. 11, pp. 16376–16395, 2023, doi: 10.1109/ACCESS.2023.3244952.

[34] H. Omrany, K. M. Al-Obaidi, A. Husain, and A. Ghaffarianhoseini, "Digital Twins in the Construction Industry: A Comprehensive Review of Current Implementations, Enabling Technologies, and Future Directions," *Sustainability*, vol. 15, no. 14, p. 10908, Jul. 2023, doi: 10.3390/su151410908.

[35] A. Bujari, A. Calvio, A. Garbugli, and P. Bellavista, "A Layered Architecture Enabling Metaverse Applications in Smart Manufacturing Environments," in *2023 IEEE International Conference on Metaverse Computing, Networking and Applications (MetaCom)*, IEEE, Jun. 2023, pp. 585–592. doi: 10.1109/MetaCom57706.2023.00103.

[36] M. Hu *et al.*, "Metric3D v2: A Versatile Monocular Geometric Foundation Model for Zero-Shot Metric Depth and Surface Normal Estimation," *IEEE Trans Pattern Anal Mach Intell*, vol. 46, no. 12, pp. 10579–10596, Dec. 2024, doi: 10.1109/TPAMI.2024.3444912.

[37] S. Singh, S. Singh, and S. C. Misra, "Post-implementation challenges of ERP system in pharmaceutical companies," *International Journal of Quality & Reliability Management*, vol. 40, no. 4, pp. 889–921, Mar. 2023, doi: 10.1108/IJQRM-10-2020-0333.

[38] D. I. K. Sjøberg and G. R. Bergersen, "Construct Validity in Software Engineering," *IEEE Transactions on Software Engineering*, vol. 49, no. 3, pp. 1374–1396, Mar. 2023, doi: 10.1109/TSE.2022.3176725.

[39] J. R. Lechien, A. Maniaci, I. Gengler, S. Hans, C. M. Chiesa-Estomba, and L. A. Vaira, "Validity and reliability of an instrument evaluating the performance of intelligent chatbot: the Artificial Intelligence Performance Instrument (AIPI)," *European Archives of Oto-Rhino-Laryngology*, vol. 281, no. 4, pp. 2063–2079, Apr. 2024, doi: 10.1007/s00405-023-08219-y.

[40] P. Yan *et al.*, "A Comprehensive Survey of Deep Transfer Learning for Anomaly Detection in Industrial Time Series: Methods, Applications, and Directions," *IEEE Access*, vol. 12, pp. 3768–3789, 2024, doi: 10.1109/ACCESS.2023.3349132.

[41] F. Buccafurri, V. de Angelis, and S. Lazzaro, "MQTT-A: A Broker-Bridging P2P Architecture to Achieve Anonymity in MQTT," *IEEE Internet Things J*, vol. 10, no. 17, pp. 15443–15463, Sep. 2023, doi: 10.1109/JIOT.2023.3264019.

[42] C.-C. Lin, A. Y. Q. Huang, and S. J. H. Yang, "A Review of AI-Driven Conversational Chatbots Implementation Methodologies and Challenges (1999–2022)," *Sustainability*, vol. 15, no. 5, p. 4012, Feb. 2023, doi: 10.3390/su15054012.