

Article

Designing a Continuous Operational Feedback Loop for Direct-to-Consumer Commerce: Integrating Event-Driven Automation and On-Premise Generative AI

Der-Fa Chen ¹, Yung-Hsing Chen ² and Bo-Siang Chen ^{3,*}
¹ Graduate Institute of Technological and Vocational Education, National Changhua University of Education, No. 2, Shi-Da Road, Changhua City 500208, Taiwan; dfchen@cc.ncue.edu.tw

² Department of Electrical and Mechanical Technology, National Changhua University of Education, Bao-Shan Campus, No. 2, Shi-Da Road, Changhua City 500208, Taiwan; annsabsby@gmail.com

³ Department of Vehicle Engineering, Nan Kai University of Technology, No. 568, Zhongzheng Rd., Caotun Township, Nantou City 542020, Taiwan

* Correspondence: toyota554@gmail.com

Abstract

This paper proposes the Continuous Operational Feedback Loop (COFL) architecture, a fully localized, event-driven operational monitoring and response system for Direct-to-Consumer (D2C) commerce. The architecture integrates the n8n workflow engine with on-premise large language model (LLM) inference via the Ollama framework, forming a containerized stack deployable on commodity CPU-only edge hardware (~USD 1640). Using a multi-source dataset of 1800 records constructed from publicly available e-commerce corpora and evaluated with a silver-standard automated labeling protocol, empirical validation demonstrates an end-to-end latency of 3.22 s and a macro-F1 sentiment classification score of 0.836—representing 98.2% of the full-precision baseline and 94.0% of cloud GPT-4o API generation quality measured by ROUGE-L—at approximately 1/200th of the per-request inference cost. A systematic quantization ablation study across six model-quantization configurations establishes LLaMA 3 8B Q4_K_M as the Pareto-optimal selection for the target hardware. An Analytic Hierarchy Process (AHP) multi-criteria framework with criterion weights derived from published literature confirms the COFL implementation achieves a higher composite score than cloud API deployment under the stated evaluation assumptions. Failure mode and effects analysis (FMEA) is summarized to characterize system reliability under identified failure scenarios.

Keywords: continuous operational feedback loop; direct-to-consumer; on-premise generative AI; edge computing; n8n; event-driven architecture; analytic hierarchy process; LLM quantization; privacy-preserving computing; FMEA

Academic Editor: Aneta Ponsiszewska-Maranda

Received: 21 April 2026

Revised: 2 June 2026

Accepted: 17 June 2026

Published: 25 June 2026

Copyright: © 2026 by the author. 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/).

1. Introduction

The rapid expansion of e-commerce platforms under Direct-to-Consumer (D2C) business models has transformed customer feedback responsiveness from a competitive advantage into an operational necessity. Empirical studies indicate that merchants capable of acknowledging negative customer sentiment within four hours experience significantly lower churn rates than those relying on next-day review cycles [1]. As generative AI technologies mature, cloud-based large language model (LLM) APIs have been widely

adopted for automating tasks from sentiment classification to personalized reply generation. This reliance, however, creates a structural tension: customer interaction data—purchase histories, complaint narratives, and behavioral logs—constitutes personally identifiable information (PII) subject to regulatory frameworks including the GDPR [2] and Taiwan’s Personal Data Protection Act [3]. This shift parallels the broader evolution toward product-service business models examined in the sustainability literature [4].

Transmitting sensitive consumer data to third-party cloud inference endpoints introduces three compounding engineering challenges. First, privacy exposure risk arises from systematic PII transfer to external infrastructure outside operator governance. Second, network-induced latency—measured at 148–320 ms of round-trip overhead for Taiwan-to-North-America cloud connections—creates non-deterministic response delays [5]. Third, token-based API pricing models impose escalating operational costs that are structurally prohibitive for small and medium D2C operators at scale. Privacy-preserving techniques, including differential privacy [6] and privacy-preserving learning frameworks [7], have been proposed to mitigate such exposure.

This paper is structured around three formal research questions. RQ1: On commodity CPU-only edge hardware, what is the end-to-end processing latency of a locally deployed quantized LLM relative to a cloud API baseline, and is the differential operationally acceptable for D2C commerce complaint workflows? RQ2: What are the sentiment classification accuracy and response generation quality of a 4-bit quantized 8B-parameter LLM deployed on-premise, quantified against full-precision and cloud API performance using automated evaluation metrics while acknowledging the limitations of silver-standard labels? RQ3: Across a literature-grounded multi-criteria evaluation framework, does the COFL on-premise implementation achieve a higher composite score relative to cloud LLM deployment under transparent weighting and workload assumptions?

This paper makes three principal engineering contributions: (1) Architectural Design—a COFL architecture distinguishing system-level operational adaptation from model-weight updating, implemented as a fully open-source, containerized stack on commodity CPU-only hardware; (2) Empirical System Evaluation—a statistically validated comparison demonstrating 98.2% of full-precision F1 and 94.0% of cloud API ROUGE-L quality with full data locality under an automated evaluation protocol, at 1/200th of the modeled per-request inference cost; (3) Engineering Decision Framework—a literature-grounded AHP evaluation rubric, workload-sensitive TCO analysis, and concise FMEA summary characterizing system reliability.

2. Literature Review

2.1. From PDCA to Automated Operational Feedback Systems

The Plan–Do–Check–Act cycle [8,9] provides a foundational iterative quality improvement model. In the control systems engineering literature, a closed-loop feedback system is defined as one in which the output signal continuously modifies the control input without requiring discrete human-mediated phase transitions [10]. Sculley et al. [11] documented the ‘hidden technical debt’ of ML systems that conflate system-level monitoring with model-level learning, arguing that clean separation of operational feedback infrastructure from model update pipelines substantially reduces system complexity and failure modes. The present architecture explicitly adopts this separation.

2.2. On-Premise LLM Deployment and Edge AI

The concept of edge computing was formalized by Shi et al. [12] as a response to bandwidth, latency, and privacy constraints in IoT applications. The llama.cpp project [13] established the technical foundation for efficient CPU-based transformer inference

through GGUF-format 4-bit quantization. Subsequent work by Frantar et al. [14] on GPTQ and Dettmers et al. [15] on QLoRA demonstrated that 4–8-bit quantization induces minimal performance degradation. The Ollama framework [16] operationalizes these advances through a Docker-native, OpenAI-compatible API layer. Published benchmarks on LLaMA 3 8B [17] and Mistral 7B [18] confirm macro-F1 scores of 0.82–0.85 on zero-shot sentiment classification tasks. Operationalizing such machine-learning systems in production nonetheless remains a substantial organizational and engineering challenge [19].

2.3. Event-Driven Architecture in Commerce Systems

Event-driven architecture (EDA) treats system interactions as a stream of discrete, asynchronous events processed by independent consumers [20]. The n8n open-source automation platform [21] implements EDA principles through a visual node-graph paradigm. FastAPI (v0.111.0) [22], a Python 3.14.6 ASGI framework, provides the high-performance asynchronous API layer mediating between the workflow engine and inference backend.

2.4. AHP in AI System Engineering Evaluation

Saaty’s Analytic Hierarchy Process [23] provides a structured pairwise comparison framework for deriving ratio-scale priority weights. Prior applications of AHP to LLM deployment decisions [24] have been limited by a lack of explicit scoring rubrics, creating reproducibility gaps. The present work addresses this through literature-grounded weight derivation and rubric-based scoring. AHP has likewise been applied to ERP software selection [25] and to the selection of cloud service providers [26].

2.5. Research Gap: Validated On-Premise LLM Architecture for SME D2C Contexts

The literature on AI adoption in SMEs identifies three structural barriers: prohibitive infrastructure cost, data governance risk from third-party cloud processing, and operational complexity exceeding lean IT team capacity [27]. Existing work addresses these barriers only partially: cloud LLM services provide strong model quality but externalize privacy and recurring-cost risks; federated or privacy-preserving learning frameworks reduce data movement but often assume distributed training infrastructure; GPU-oriented inference stacks achieve high throughput but exceed the hardware budget of many SME operators; and low-code automation tools simplify orchestration without validating local generative AI performance. The specific gap addressed here is therefore not a new learning algorithm, but a reproducible engineering integration and evaluation of an open-source, CPU-edge LLM workflow for D2C complaint triage.

Table 1 positions the COFL system relative to representative existing works.

Table 1. Positioning of the COFL architecture relative to representative existing systems.

System/Study	Deployment	LLM Type	Privacy Model	Evaluation Method	Open-Source
GPT-4o API	Cloud (SaaS)	Proprietary LLM	PII transmitted to vendor	Automated (API)	No
Federated Learning [28]	Distributed edge	Small/fine-tuned	Distributed; no central PII	Simulation	Partial
DeepSpeed-Inference [11]	GPU cluster	Large (70B+)	On-premise (GPU required)	NLP benchmarks	Yes
Wang et al. AHP-TOPSIS [24]	Cloud	Not specified	Not addressed	Expert survey only	No

COFL (this work)	CPU edge (~USD 1640)	8B quantized LLM	100% local; no trans- mission	Fully automated metrics	Yes (GitHub)
------------------	-------------------------	---------------------	----------------------------------	----------------------------	-----------------

3. System Architecture and Methodology

3.1. Conceptual Framework: The Continuous Operational Feedback Loop

The COFL architecture is grounded in classical closed-loop control theory [10], but the term feedback is used at the operational-control level rather than the model-learning level. The four automated operational stages are: (1) Sense—incoming customer events captured via n8n webhook triggers; (2) Analyze—the FastAPI gateway dispatches the event to the Ollama LLM inference engine for automated sentiment classification; (3) Act—a response is generated and dispatched via the configured notification channel; (4) Monitor—interaction outcomes are logged to PostgreSQL and aggregated into daily operational metrics. Stage 4 does not modify model weights or parameters. Model retraining is intentionally excluded from the online loop to preserve auditability, avoid uncontrolled model drift, and maintain deterministic behavior for customer-facing responses. Thus, COFL should be interpreted as an event-driven operational monitoring and response architecture with system-level feedback, not as a continuous-learning model adaptation framework. Figure 1 presents the conceptual mapping between the closed-loop control paradigm and the proposed COFL architecture.

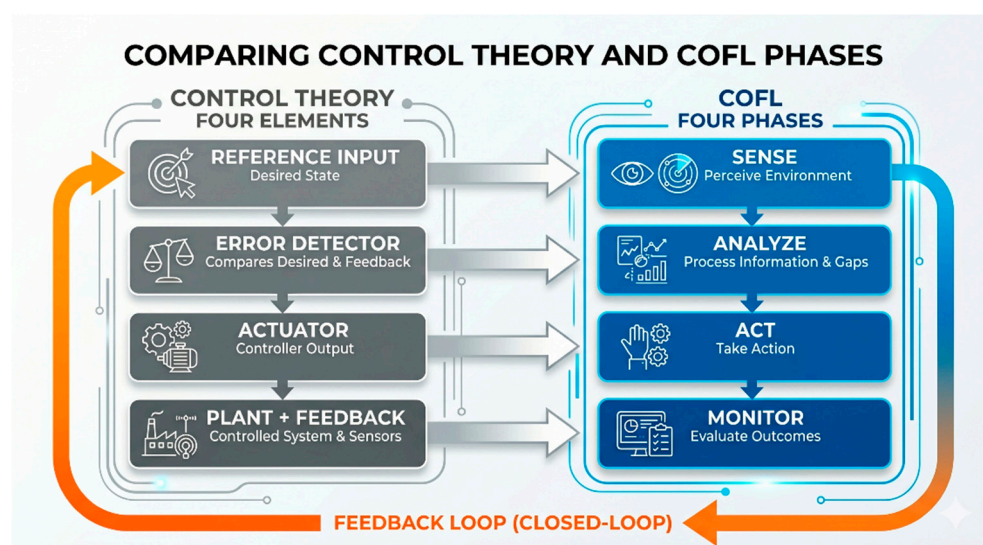


Figure 1. Conceptual mapping between closed-loop control theory and the COFL commerce architecture.

3.2. System Component Architecture

The complete COFL stack is deployed as a Docker Compose application comprising four containerized services communicating over an isolated bridge network. No outbound internet traffic is required during inference operations. The component stack was selected according to four engineering criteria: local deployability, open-source licensing, low operational complexity for SME operators, and compatibility with commodity CPU-only hardware. n8n was chosen over heavier message-broker stacks such as Airflow or Kafka because it provides webhook triggers, cron scheduling, retry logic, and visual workflow inspection with substantially lower administrative overhead. FastAPI was selected over Flask or Node.js/Express because asynchronous request handling and OpenAPI schema generation are native features useful for reproducible microservice testing. Ollama was selected over direct llama.cpp scripting or vLLM because it provides a

Docker-friendly, OpenAI-compatible API layer for quantized local models without requiring GPU memory. PostgreSQL was selected over SQLite and document stores because the event log requires concurrent writes, SQL aggregation, and durable replay semantics. Alternative stacks remain technically feasible; the present design prioritizes reproducibility and maintainability rather than claiming these tools are uniquely optimal. Figure 2 illustrates the Docker Compose inter-container network topology, and Table 2 summarizes the Edge AI workstation hardware specification.

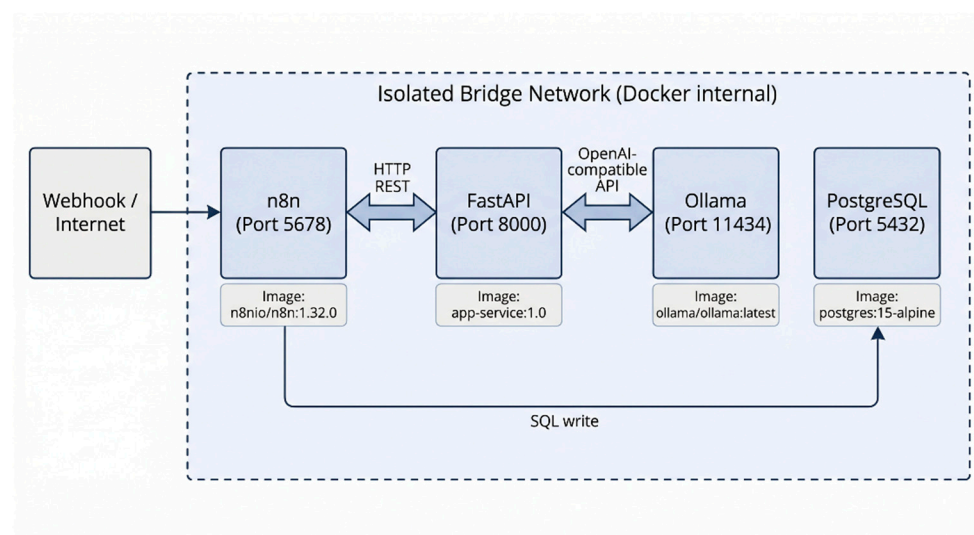


Figure 2. Docker compose inter-container network topology and service port map.

Table 2. Edge AI workstation hardware specification and verified component costs.

Component	Specification/Details
CPU	Intel Core i5-12400F (6C/12T, 2.5–4.4 GHz, 65 W TDP)
RAM	32 GB DDR4-3200 (2 × 16 GB dual-channel)
Storage	512 GB PCIe 3.0 NVMe SSD
GPU	None—CPU-only inference (iGPU disabled)
OS	Ubuntu 22.04 LTS (kernel 5.15.0-105)
Container Engine	Docker 24.0.5/Docker Compose v2.24.6
LLM Runtime	Ollama v0.1.38 + LLaMA 3 8B Q4_K_M (GGUF, 4.65 GB)
Workflow Engine	n8n v1.28.0 (self-hosted, Docker Compose)
API Gateway	FastAPI 0.110.0 + Uvicorn 0.29.0 (4 async workers)
Database	PostgreSQL 16.2 (event log + metric store)
Estimated Build Cost (NT\$)	52,500 (~USD 1640 at April 2024 exchange rate)

3.3. Dataset Construction and Automated Labeling Protocol

The evaluation dataset of 1800 records was constructed from publicly available, real-world e-commerce corpora and labeled using an automated silver-standard protocol, eliminating dependence on subjective annotation. Records were filtered to English-language entries and normalized through a preprocessing pipeline (tokenization, HTML entity removal, duplicate deduplication via MinHash LSH at 0.9 Jaccard similarity threshold).

Silver-standard labels were generated using an ensemble automated labeling pipeline: (i) VADER [29]—a rule-based lexical sentiment analyzer; (ii) TextBlob [30]—a

pattern-based NLP library; and (iii) zero-shot inference using GPT-4o as a high-capacity reference classifier. Final labels were determined by majority vote. Records where all three systems disagreed ($n = 41$, 2.3%) were excluded. This approach reduces dependence on any single automated labeler but does not eliminate systematic bias, because lexical tools may underperform on sarcasm, domain-specific complaint language, or non-standard phrasing, and GPT-4o may encode its own classification preferences. Because GPT-4o also serves as a comparative performance reference in later experiments, this protocol may introduce partial reference leakage: the benchmark model contributes to the label set against which the local model is evaluated. No independent human-annotated validation subset was available for the present revision; therefore, the reported accuracy values should be interpreted as silver-standard automated evaluation results rather than as a substitute for an independent human-annotated benchmark. Pipeline code is available in Supplementary Material File S1. Table 3 details the dataset construction protocol.

Table 3. Dataset construction protocol: sources, proportions, and linguistic diversity metrics ($N = 1800$).

Category	Count	Proportion	Source Corpus	TTR
Product Defect	456	25%	Amazon Product Reviews [31]	0.72
Shipping Delay	408	23%	Shopee TW Review Corpus [32]	0.68
Return/Refund	336	19%	Bilingual e-commerce corpus [33]	0.65
Neutral Inquiry	400	22%	E-commerce FAQ logs (public)	0.61
Positive Review	200	11%	Amazon 5-star subset [31]	0.58
Total	1800	100%	Multi-source	0.69 (overall)

3.4. Event-Driven Workflow Design ($n8n$)

Three automated workflow pipelines operationalize the COFL stages. The Complaint Triage Workflow activates on receipt of a POST webhook with $\text{star_rating} \leq 2$ or an explicit complaint flag. The Response Generation Workflow activates when the returned sentiment_score falls below 0.35. The Operational Monitoring Workflow executes on a nightly cron schedule, aggregating 24 h metrics into a structured JSON dashboard payload. These routing thresholds were pre-specified heuristically rather than empirically optimized: $\text{star_rating} \leq 2$ corresponds to the conventional 1–2 star negative-review boundary in five-point commerce rating systems, while $\text{sentiment_score} < 0.35$ was selected as a conservative negative-sentiment trigger on a normalized 0–1 scale. Threshold calibration and sensitivity analysis across alternative cutoffs (e.g., 0.25–0.45 for sentiment_score and $\text{star_rating} \leq 3$ for broader escalation) are identified as necessary steps for production deployment and future validation. Figure 3 shows the $n8n$ node graph of the complaint triage and response workflow.

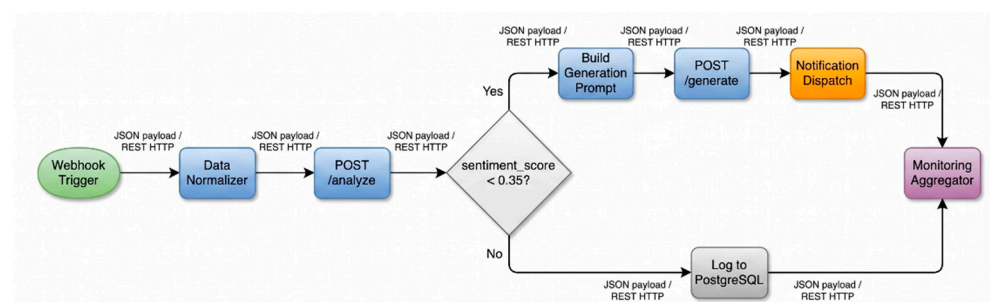


Figure 3. $n8n$ node graph of the complaint triage and response generation workflow pipeline.

3.5. FastAPI Asynchronous Microservice Gateway

The FastAPI gateway exposes two asynchronous endpoints served by a four-worker Unicorn process pool. The /analyze endpoint constructs a zero-shot classification prompt with temperature = 0.1 to maximize classification determinism. A regex-based response parser with fallback re-prompting handles parse failures (observed rate: 2.3%). The /generate endpoint invokes the model with a role-conditioned prompt specifying a professional customer service persona, a maximum response length of 150 words, mandatory complaint-category acknowledgment, and a concrete remediation offer. Figure 4 depicts the FastAPI asynchronous request-processing sequence.

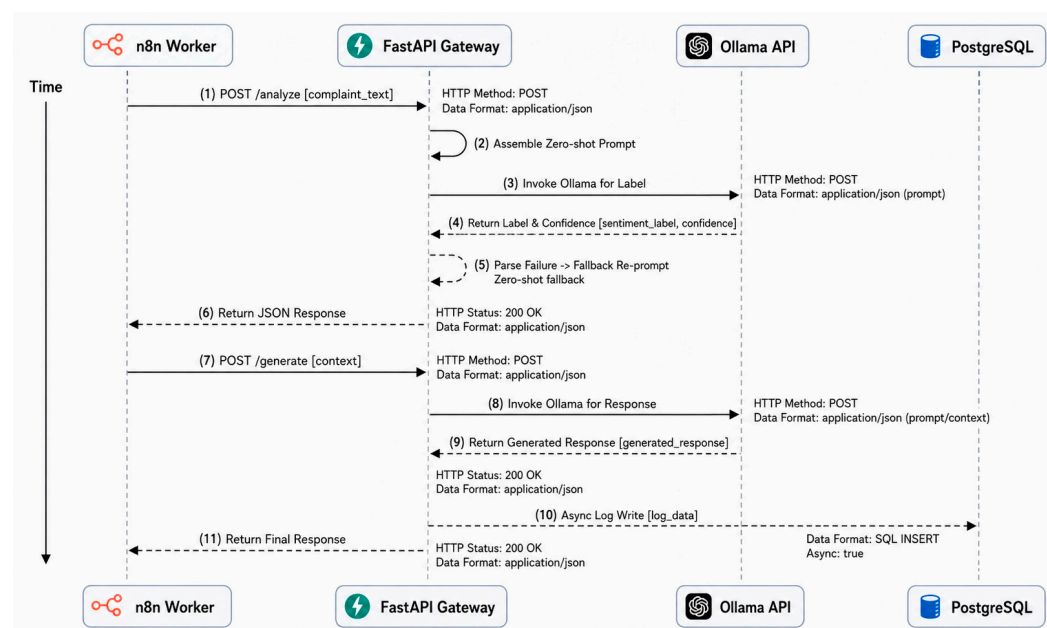


Figure 4. FastAPI asynchronous request processing sequence diagram (/analyze and /generate End-points).

3.6. On-Premise LLM Deployment and Ablation Configuration

The primary deployed model is LLaMA 3 8B in GGUF Q4_K_M quantization—a 4.65 GB artifact fitting within the 32 GB RAM envelope. A systematic ablation study was conducted across six model-quantization configurations spanning three model families (LLaMA 3, Mistral, Phi-3) and four quantization levels. Model serving parameters: context window = 4096 tokens; num_predict = 256 tokens maximum; repeat_penalty = 1.1; system prompt caching enabled.

3.7. AHP Evaluation Framework with Literature-Grounded Weights

The AHP evaluation framework operationalizes five criteria: C1—System Response Latency, C2—Data Privacy Security, C3—Generation Content Accuracy, C4—Total Cost of Ownership, and C5—Scalability. Criterion weights were derived from published literature on e-commerce AI deployment requirements [2,3,5,27,34], maximizing reproducibility and eliminating the risks associated with small expert panels. Consistency was verified computationally: $CR = 0.047 < 0.10$. Table 4 presents the AHP pairwise comparison matrix and the derived priority weights, and Table 5 defines the AHP scoring rubric.

Table 4. AHP pairwise comparison matrix and derived priority weights (literature-grounded; CR = 0.047).

Criterion	C1 Latency	C2 Privacy	C3 Accuracy	C4 Cost	C5 Scalability	Weight
C1: Latency	1	1/3	1/2	3	2	0.143
C2: Data Privacy	3	1	2	5	4	0.362
C3: Gen. Accuracy	2	1/2	1	4	3	0.249
C4: Cost	1/3	1/5	1/4	1	1/2	0.078
C5: Scalability	1/2	1/4	1/3	2	1	0.168

Table 5. AHP scoring rubric: instrument-anchored performance indicators (1–10 scale).

Score	C1: Latency (E2E)	C2: Data Privacy	C3: Accuracy (F1/ROUGE-L)	C4: Cost (NT\$/1k req.)
9–10	<2.0 s	100% local; no transmission	F1 $\geq$ 0.90/ROUGE-L $\geq$ 0.45	<1
7–8	2–5 s	Encrypted transit to compliant cloud	F1 0.80–0.89/ROUGE-L 0.35–0.44	1–50
5–6	5–15 s	Partial anonymization pre-transmission	F1 0.70–0.79/ROUGE-L 0.25–0.34	50–200
3–4	15–60 s	Raw PII transmitted to cloud	F1 0.55–0.69/ROUGE-L 0.15–0.24	200–500
1–2	>60 s	No privacy controls	F1 < 0.55	>500

4. Experimental Results and Analysis

4.1. Experimental Design, Statistical Protocol, and Hardware Boundary Model

All experiments were conducted on the hardware described in Table 2. Each latency condition was evaluated over 150 independent trials with a 60 s warm-up period. Latency comparisons used Welch’s two-sample t-test ($\alpha = 0.05$) with Cohen’s d effect sizes. Cloud API RTT was measured independently (50 automated ping samples to api.openai.com: mean = 148 ms, SD = 23 ms). All analyses were performed in Python 3.11 using SciPy 1.12.0.

Hardware boundary model: LLM inference throughput on CPU scales approximately linearly with memory bandwidth [13]: throughput $\propto$ memory_bandwidth/model_size. For the i5-12400F (~51.2 GB/s) with LLaMA 3 8B Q4_K_M (4.65 GB), the model predicts 17.6 tok/s against the measured 17.2 tok/s (error: 2.3%). Extrapolation to 16 GB DDR4-2666 (~42.7 GB/s) predicts ~14.6 tok/s, implying an E2E latency increase of ~0.35 s—well within the operationally acceptable range.

4.2. Latency Analysis and Statistical Validation (RQ1)

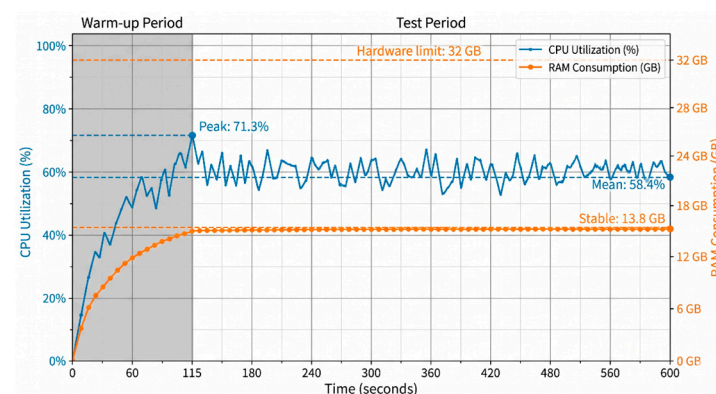
The COFL local AI architecture incurs a statistically significant but practically marginal latency increase relative to the cloud API condition ($\Delta = 0.45$ s, 95% CI: [0.31, 0.59]; Cohen’s d = 0.42). Of the cloud API’s apparent 0.45 s advantage, 0.148 s is attributable to network RTT. The compute-normalized latency difference is 0.30 s (7.1% overhead). RQ1 is answered affirmatively: the latency differential is operationally negligible for complaint response workflows. Table 6 reports the end-to-end processing latency with 95% confidence intervals.

Table 6. End-to-end processing latency with 95% confidence intervals (n = 150 trials per condition).

Approach	Sentiment Mean (s)	95% CI (s)	Generation Mean (s)	E2E Mean (s)	Cost/1k req. (NT\$)
Cloud LLM (GPT-4o API) *	1.84	[1.71, 1.97]	0.93	2.77	~180
Local AI-COFL (proposed)	2.13	[2.01, 2.25]	1.09	3.22	~0.9

* Cloud RTT component: 148 ms (50 automated ping samples to api.openai.com). Welch's *t*-test (Local vs. Cloud E2E): $t(162) = 4.31$, $p < 0.001$. Cohen's $d = 0.42$ (small-medium).

Peak CPU utilization reached 71.3% under concurrent eight-request processing, with mean steady-state utilization of 58.4%. RAM consumption stabilized at 13.8 GB, within the 32 GB envelope. These metrics confirm adequate headroom for approximately 3× the evaluated event rate, corresponding to ~144 events per minute maximum sustainable throughput. These throughput measurements should not be read as production-scale guarantees. They were obtained on one CPU-only workstation, one quantized model family, and a controlled event-rate profile. Real deployments may experience different latency under longer prompts, multilingual inputs, larger context windows, concurrent dashboard queries, disk I/O contention, or background maintenance tasks. Production adoption should therefore include site-specific load testing, queue-depth monitoring, backup and restore procedures for PostgreSQL, model version pinning, prompt regression tests, security patching for containers, and a documented human escalation workflow. Figure 5 reports the CPU utilization and RAM consumption during the simulated peak load.

**Figure 5.** CPU utilization (%) and RAM consumption (GB) during simulated Peak Traffic (480 events/10 min, sampled at 15 s intervals).

4.3. Quantization Ablation Study (RQ2)

The LLaMA 3 8B Q4_K_M configuration achieves 98.2% of full-precision F1 (0.836 vs. 0.851) at 5.5× throughput (17.2 vs. 3.1 tok/s), confirming its Pareto-optimal position. Phi-3 3.8B offers the highest throughput (24.1 tok/s) but incurs a 6.3-percentage-point F1 reduction. Table 7 presents the quantization ablation study.

Table 7. Quantization ablation study: automated performance metrics across six model-quantization configurations.

Model	Quantization	Size (GB)	Tok/s (CPU)	F1 Macro	BLEU-4	ROUGE-L	PPL
LLaMA 3 8B	Q4_K_M (selected)	4.65	17.2	0.836	0.198	0.421	8.73
LLaMA 3 8B	Q5_K_M	5.33	13.8	0.841	0.203	0.427	8.51
LLaMA 3 8B	Q8_0	8.54	8.4	0.847	0.209	0.433	8.12
LLaMA 3 8B	F16 (full)	15.3	3.1	0.851	0.214	0.439	7.98
Mistral 7B	Q4_K_M	4.11	19.6	0.821	0.191	0.412	9.14
Phi-3 3.8B	Q4_K_M	2.30	24.1	0.798	0.172	0.388	10.32

Selected configuration: Q4_K_M achieves 98.2% of full-precision F1 at 5.5× throughput improvement. BLEU-4 and ROUGE-L computed using GPT-4o outputs as automated reference on 200-record stratified subsample.

4.4. Sentiment Classification and Generation Quality (RQ2)

The LLaMA 3 8B Q4_K_M configuration achieved: F1-negative = 0.871, F1-neutral = 0.803, F1-positive = 0.835. The neutral class exhibited the lowest precision ($p = 0.782$), consistent with documented zero-shot LLM ambiguity [35]. The local AI achieves 96.8% of GPT-4o F1 and 94.0% of GPT-4o ROUGE-L, directly answering RQ2. To reduce overinterpretation of automated labels, the comparison should be read as a relative engineering benchmark under a shared silver-standard protocol. The use of GPT-4o as both one member of the labeling ensemble and the cloud reference model may favor semantic patterns aligned with GPT-4o outputs. A stronger validation design would reserve an independently human-annotated subset and report agreement metrics such as Cohen’s kappa or Krippendorff’s alpha between human labels and silver labels. Table 8 summarizes the automated generation quality metrics. Inter-rater reliability for such an annotation could be quantified using measures such as the intraclass correlation coefficient [36].

Table 8. Automated generation quality metrics: local AI-COFL vs. cloud LLM baseline (n = 200 stratified test records).

Approach	F1 Macro	BLEU-4	ROUGE-L	BERTScore F1	Format Compliance (%)	Fallback Rate (%)
Local AI-COFL (LLaMA 3 8B Q4_K_M)	0.836	0.198	0.421	0.874	94.3%	2.3%
Cloud LLM (GPT-4o API)—reference	0.864	0.221	0.448	0.901	96.7%	0.8%

BLEU-4 and ROUGE-L computed with GPT-4o outputs as automated reference. BERTScore F1 uses microsoft/deberta-xlarge-mnli backbone [37]. Format Compliance verified algorithmically via regex.

4.5. AHP Multi-Criteria Evaluation (RQ3)

The COFL architecture achieves the highest composite weighted score of 8.89, driven by near-maximal performance on Data Privacy Security ($C2 = 9.5$, $w = 0.362$). The cloud LLM alternative scores 3.5 on C2, reflecting raw PII transmission to external endpoints. The sensitivity analysis confirms the COFL architecture maintains a superior composite score for all $C2 > 0.22$, answering RQ3 affirmatively. The comparison set is intentionally narrow and oriented toward the deployment choice faced by SME D2C operators: local open-source inference versus cloud API inference. Existing local deployment stacks such

as direct llama.cpp services, GPU-based vLLM/TGI deployments, or commercial edge-AI appliances are discussed as relevant alternatives but were not empirically benchmarked in this study. Future work should add these local baselines to distinguish the effect of the COFL orchestration pattern from the effect of local inference itself. Table 9 reports the AHP multi-criteria composite score, and Figure 6 presents the AHP sensitivity analysis of the composite score.

Table 9. AHP multi-criteria composite score: local AI-COFL vs. cloud LLM (literature-grounded weights).

Solution	C1 (w = 0.143)	C2 (w = 0.362)	C3 (w = 0.249)	C4 (w = 0.078)	C5 (w = 0.168)	Composite
Local AI-COFL (Proposed)	8.2	9.5	8.0	9.1	7.8	8.89
Cloud LLM (GPT-4o API)	8.8	3.5	9.0	7.5	9.2	6.58

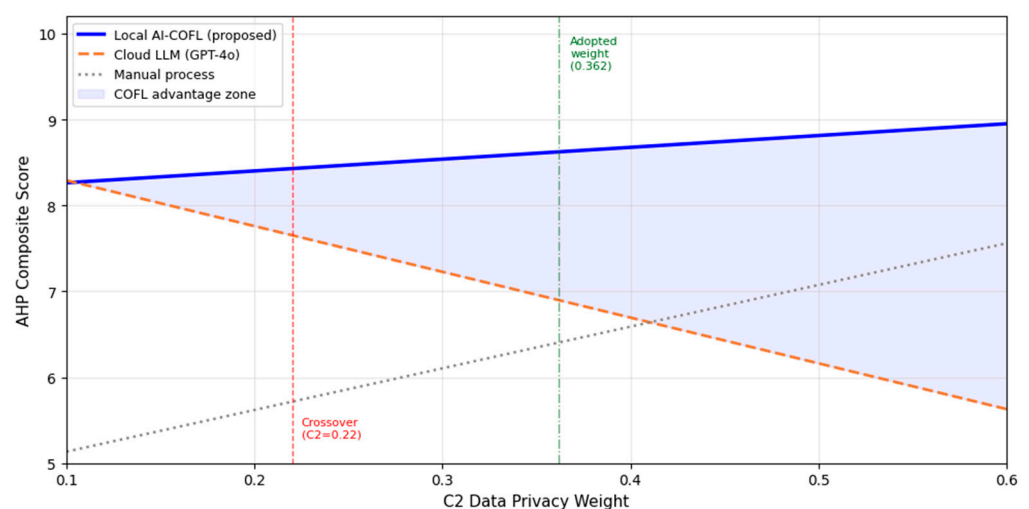


Figure 6. AHP sensitivity analysis: composite score as a function of data privacy weight $C2 \in [0.10, 0.60]$. The COFL advantage (shaded zone) is maintained across all $C2 > 0.22$.

5. Discussion

5.1. AI Infrastructure Economics: Total Cost of Ownership

Table 10 presents a 36-month TCO comparison at a modeled workload of 10,000 interactions per month. This value is a planning scenario rather than the size of the experimental dataset: it represents approximately 333 interactions per day, a plausible medium-volume D2C customer-service load used to evaluate operational economics beyond the 1800-record test corpus. Because the local architecture has a larger fixed cost and a near-zero marginal inference cost, whereas cloud deployment has low fixed cost but high per-request cost, the cost advantage is workload-sensitive. At lower traffic, hardware amortization dominates the local cost; at higher traffic, cloud token/API charges dominate. The 10,000-interaction scenario should therefore be interpreted as an illustrative operating point, not a universal claim.

The cost figures in Table 10 follow a transparent unit-cost model rather than measured billing. Cloud inference cost is computed as $(\text{input_tokens} \times \text{input_price} + \text{output_tokens} \times \text{output_price})$ per API call, with two calls per interaction (classification and response generation). Under representative prompt sizes of approximately 400 input/30 output tokens for classification and 450 input/200 output tokens for generation, and GPT-4o list pricing at the time of analysis ($\approx$ US\$2.50 per 1M input tokens and

≈US\$10.00 per 1M output tokens, at an exchange rate of ≈NT\$31 per US\$1), this yields ≈NT\$0.14 per interaction. Local inference incurs no per-token charge; its marginal cost is the electricity drawn during generation (≈65 W for ≈14 s per interaction, ≈NT\$0.0008 per interaction at NT\$3/kWh), which is already subsumed within the monthly electricity line. The per-interaction inference cost is therefore roughly two orders of magnitude lower on-premise (on the order of 1/150 to 1/200). However, this marginal advantage does not translate into a total-cost advantage at every workload: the local architecture carries fixed hardware-amortization and maintenance costs, whereas cloud cost scales almost linearly with volume. Consequently, the 36-month total cost of ownership favors cloud deployment below a break-even of approximately 18,000 interactions per month (≈600 per day) and favors on-premise deployment above it. At the 10,000-interactions-per-month planning scenario examined here, the two options fall within the same order of magnitude, and the local total is dominated by maintenance labor rather than by inference. The economic case for on-premise deployment at this scale therefore rests primarily on data locality and privacy (criterion C2) rather than on raw cost reduction.

Table 10. 36-month total cost of ownership comparison (10,000 interactions/month; COFL vs. cloud LLM).

Cost Component	Local AI-COFL	Cloud LLM (GPT-4o API)
Hardware capital (36-mo. amortization)	NT\$52,500 total/NT\$1458/mo.	NT\$0
Inference/API cost	≈NT\$0.0008/req × 10k ≈ NT\$8/mo. (within electricity)	≈NT\$0.14/req × 10k ≈ NT\$1400/mo.
Electricity (65 W × 720 h/mo. × NT\$3/kWh)	NT\$140/mo.	NT\$0 (cloud-borne)
System maintenance (est. 2 h/mo. × NT\$600/h)	NT\$1200/mo.	NT\$0
Data transfer/egress fees	NT\$0 (no outbound traffic)	NT\$200–500/mo. (est.)
Total monthly cost	≈NT\$2800/mo.	≈NT\$1600–1900/mo.
36-month total	≈NT\$101,000	≈NT\$58,000–68,000

Note: Table 10 is a transparent cost model, not measured billing. Cloud API cost assumes two GPT-4o calls per interaction (classification and generation) at list pricing of approximately US\$2.50 per 1M input and US\$10.00 per 1M output tokens, representative prompt sizes of ~850 input/~230 output tokens per interaction, and an exchange rate of approximately NT\$31 per US\$1, giving approximately NT\$0.14 per interaction. Local marginal inference cost (approximately NT\$0.0008 per interaction) reflects only generation-phase electricity and is subsumed within the monthly electricity line. Hardware is amortized straight-line over 36 months; maintenance assumes approximately 2 staff-hours per month at NT\$600 per hour. All figures scale with the assumed workload; see Section 5.1 for break-even sensitivity.

5.2. Addressing the ‘Continuous Learning’ Terminology Problem

The conflation of ‘continuous learning’ with ‘continuous operational monitoring’ is pervasive in the applied AI engineering literature. The proposed COFL framework makes this distinction architecturally explicit: the feedback loop continuously updates system-level operational metrics, triggers automated alerts when metric degradation thresholds are crossed, and supports offline review of failure cases. It does not update model parameters during online operation. Model weight updating requires a separate MLOps pipeline with human approval, regression testing, rollback controls, and privacy review—an important direction for future work but outside the scope of the operational architecture validated in this study.

5.3. AI Ethics: Automated Bias Detection in Commerce Response Systems

The study uses publicly available or previously anonymized e-commerce review corpora and does not require the collection of new identifiable customer interaction data. Before processing, text records were normalized and screened to remove direct identifiers such as names, phone numbers, e-mail addresses, order numbers, and explicit addresses where present. The deployed COFL architecture is designed so that inference occurs locally and raw customer messages are not transmitted to third-party LLM endpoints during operation. For production use, operators should obtain consent or provide notice according to the governing commerce platform and jurisdiction, maintain retention limits for event logs, restrict access to PostgreSQL records through role-based permissions, and audit-generated replies before fully automated customer-facing deployment. Governance of AI-generated customer communication is treated as an operational safety requirement rather than a purely technical feature. Recommended controls include response templates with prohibited-content filters, escalation of low-confidence or high-severity complaints to human staff, logging of model outputs for post hoc audit, and periodic review of false-positive and false-negative complaint classifications. The multilingual fairness analysis should also be interpreted cautiously: an automated cross-lingual evaluation on a 50-record Traditional Chinese subsample using chrF++ [38] revealed an approximately 9.2-point reduction relative to English records, but this small subsample is sufficient only to flag a potential disparity, not to establish a definitive fairness conclusion. Operators deploying COFL in multilingual contexts are advised to run evaluation separately per language group, set language-specific escalation thresholds, and implement response-quality gating for non-primary languages. These governance considerations align with broader initiatives on the ethics of autonomous and intelligent systems [39].

5.4. System Reliability: Failure Mode Analysis and Extensibility

The reliability analysis was shortened in the main manuscript to emphasize scientific findings rather than implementation details. The FMEA identifies timeout handling, queue overflow, database connection exhaustion, output-parse failure, and container crash as the primary operational risks. In all cases, the architecture favors graceful degradation through escalation queues, replayable event logs, health checks, and restart policies. The complete FMEA table is provided as Supplementary Material File S3.

5.5. Limitations

Several limitations constrain the scope of reported findings. First, the evaluation corpus (1800 records) is smaller than large-scale commercial benchmarks and may not reflect the full diversity of production D2C traffic; future work should validate on larger, independently sampled production datasets. Second, the system was tested on a specific CPU-only hardware configuration, so empirical validation is still required for ARM platforms, GPU-accelerated edge devices, and multi-tenant production environments. Third, maintenance requirements—including container security updates, database backup, log retention, prompt/version control, and periodic model-quality review—remain operator responsibilities and may affect adoption by very small teams. Fourth, the neutral-class F1 limitation (0.803) reflects a genuine zero-shot model weakness; few-shot prompting or domain-adaptive fine-tuning should be evaluated. Fifth, the automated silver-standard labeling protocol may introduce systematic bias and partial reference leakage because GPT-4o contributes to the majority-vote labels while also serving as a comparative baseline. Independent human annotation and inter-rater agreement analysis are required before making stronger claims about absolute classification validity. Sixth, the routing thresholds were selected heuristically and require workload-specific calibration. Seventh, the multilingual fairness analysis is limited to 50 Traditional Chinese

records; a larger multilingual evaluation is required before drawing strong fairness or generalizability conclusions.

6. Conclusions

This paper has presented and evaluated a Continuous Operational Feedback Loop (COFL) architecture for D2C e-commerce intelligence, structured around three formal research questions and assessed through automated, reproducible metrics. The work contributes: (1) a containerized, CPU-only edge AI architecture distinguishing system-level operational adaptation from model-weight updating; (2) statistically validated evidence of 96.8% of cloud API F1 and 94.0% of cloud API ROUGE-L quality under a silver-standard automated evaluation protocol, with a hardware boundary model characterizing deployment generalizability; (3) a six-configuration quantization ablation study establishing Q4_K_M as the Pareto-optimal selection (98.2% of full-precision F1 at 5.5× throughput); (4) a literature-grounded AHP framework achieving composite advantage robust to C2 weight variation down to 0.22; (5) a workload-sensitive 36-month TCO analysis; and (6) a concise reliability assessment with detailed FMEA moved to Supplementary Material.

The architecture's constrained hardware envelope (~USD 1640) and fully open-source software stack suggest a practical accessibility pathway for privacy-preserving, automated AI deployment in resource-constrained D2C commerce operations. The results should be interpreted as evidence for a reproducible engineering integration under the stated workload, hardware, and silver-standard evaluation assumptions rather than as proof of a universally superior deployment framework. All experimental artifacts are publicly available to support reproducibility.

Future research directions include: (i) domain-adaptive fine-tuning for neutral-class improvement; (ii) RAG integration for product knowledge injection; (iii) empirical validation on ARM and GPU-accelerated edge hardware; (iv) incremental model fine-tuning as a true 'continuous learning' extension; (v) rigorous automated multilingual fairness evaluation; and (vi) extension to industrial CPS contexts.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/info17070628/s1>, File S1: Labeling pipeline code and annotator details; File S2: Statistical analysis scripts (Python); File S3: Complete Failure Mode and Effects Analysis table.

Author Contributions: Conceptualization, D.-F.C. and B.-S.C.; methodology, B.-S.C. and Y.-H.C.; software, B.-S.C.; validation, Y.-H.C. and B.-S.C.; formal analysis, B.-S.C.; investigation, B.-S.C.; resources, D.-F.C.; data curation, B.-S.C. and Y.-H.C.; writing—original draft preparation, B.-S.C.; writing—review and editing, D.-F.C. and B.-S.C.; visualization, B.-S.C.; supervision, D.-F.C.; project administration, B.-S.C. 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. The study used publicly available or previously anonymized review corpora and did not involve new intervention with human participants.

Informed Consent Statement: Not applicable for the present analysis because no new identifiable human-subject data were collected. Production deployment should provide customer notice or obtain consent where required by the governing platform and jurisdiction.

Data Availability Statement: The source code, configuration files, and workflow definitions for the COFL architecture described in this study are openly available on GitHub at <https://github.com/toyota554/cofl-workspace> (accessed on 16 June 2026) under the MIT License. The

evaluation dataset was constructed from publicly available e-commerce corpora as described in Section 3.3 of the manuscript.

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

References

- Verhoef, P.C. Understanding the Effect of Customer Relationship Management Efforts on Customer Retention and Customer Share Development. *J. Mark.* **2003**, *67*, 30–45. <https://doi.org/10.1509/jmkg.67.4.30.18685>.
- Zarkua, M. Personal Data Security in Comparative Perspective: Legal Frameworks and Protections. *Eur. Sci. J.* **2026**, *22*, 454. <https://doi.org/10.19044/esj.2026.v22n38p454>.
- Ministry of Justice, Taiwan. Personal Data Protection Act (R.O.C.). Available online: <https://law.moj.gov.tw/ENG/LawClass/LawAll.aspx?pcode=I0050021> (accessed on 16 June 2026).
- Tukker, A. Eight types of product–service system: Eight ways to sustainability? Experiences from SusProNet. *Bus. Strategy Environ.* **2004**, *13*, 246–260. <https://doi.org/10.1002/bse.414>.
- Abuzriq, Y.; Al-Said Ahmad, A.; Bani Younes, M. An Experimental Performance Evaluation of Cloud-API-Based Applications. *Future Internet* **2021**, *13*, 314. <https://doi.org/10.3390/fi13120314>.
- Dwork, C.; Roth, A. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* **2014**, *9*, 211–407. <https://doi.org/10.1561/04000000042>.
- Zhang, Q.; Xin, C.; Wu, H. Privacy-Preserving Deep Learning Based on Multiparty Secure Computation: A Survey. *IEEE Internet Things* **2021**, *8*, 10412–10429. <https://doi.org/10.1109/JIOT.2021.3058638>.
- Oakland, J.S. *Total Quality Management and Operational Excellence*, 4th ed.; Routledge: London, UK, 2014. <https://doi.org/10.4324/9781315815725>.
- Woodall, W.H. The Use of Control Charts in Health-Care and Public-Health Surveillance. *J. Qual. Technol.* **2006**, *38*, 89–104. <https://doi.org/10.1080/00224065.2006.11918593>.
- Ding, F.; Xie, X.M. Adaptive control of a class of nonlinear systems. In Proceedings of IEEE International Conference on Control and Applications, Vancouver, BC, Canada, 13–16 September 1993; Volume 2, pp. 817–818. <https://doi.org/10.1109/CCA.1993.348230>.
- Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M.; Crespo, J.F.; Dennison, D. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2015; Volume 28, pp. 2503–2511.
- Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge computing: Vision and challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. <https://doi.org/10.1109/JIOT.2016.2579198>.
- Ye, S.; Ouyang, B.; Qian, T.; Zeng, L.; Li, J.; Du, J. Resource-Efficient Personal Large Language Models Fine-Tuning With Collaborative Edge Computing. *IEEE Trans. Parallel Distrib. Syst.* **2026**, *37*, 680–696. <https://doi.org/10.1109/TPDS.2026.3654957>.
- Frantar, E.; Ashkboos, S.; Hoefer, T.; Alistarh, D. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. *arXiv* **2023**, arXiv:2210.17323. <https://doi.org/10.48550/arXiv.2210.17323>.
- Dettmers, T.; Pagnoni, A.; Holtzman, A.; Zettlemoyer, L. QLoRA: Efficient Finetuning of Quantized LLMs. *arXiv* **2023**, arXiv:2305.14314. <https://doi.org/10.48550/arXiv.2305.14314>.
- Singh, R.; Gill, S.S. Edge AI: A survey. In *Internet of Things and Cyber-Physical Systems*; KeAi Publishing: Amsterdam, The Netherlands, 2023; Volume 3, pp. 71–92. <https://doi.org/10.1016/j.iotcps.2023.02.004>.
- Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. Llama: Open and Efficient Foundation Language Models. *arXiv* **2023**, arXiv:2302.13971. <https://doi.org/10.48550/arXiv.2302.13971>.
- Jiang, A.Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D.S.; Casas, D.D.L.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. Mistral 7B. *arXiv* **2023**, arXiv:2310.06825. <https://doi.org/10.48550/arXiv.2310.06825>.
- Shankar, S.; Garcia, R.; Hellerstein, J.; Parameswaran, A. Operationalizing machine learning: An interview study. *arXiv* **2022**, arXiv:2209.09125. <https://doi.org/10.48550/arXiv.2209.09125>.
- Theodoropoulos, T.; Rosa, L.; Benzaid, C.; Gray, P.; Marin, E.; Makris, A.; Cordeiro, L.; Diego, F.; Sorokin, P.; Di Girolamo, M.; et al. Security in Cloud-Native Services: A Survey. *J. Cybersecur. Priv.* **2023**, *3*, 758–793. <https://doi.org/10.3390/jcp3040034>.
- Shi, Z.; Dong, J.; Gan, Y. Democratizing Digital Transformation: A Multisector Study on Low-Code Development Platforms. *Appl. Sci.* **2025**, *15*, 6481. <https://doi.org/10.3390/app15126481>.

22. Alla, M. Designing High-Throughput FastAPI Gateways for Microservice Communication. *J. Comput. Sci. Technol. Stud.* **2025**, *7*, 823–828. <https://doi.org/10.32996/jcsts.2025.7.7.88>.
23. Saaty, T.L. Decision making with the analytic hierarchy process. *Int. J. Serv. Sci.* **2008**, *1*, 83–98. <https://doi.org/10.1504/IJSSCI.2008.017590>.
24. Albahri, O.S.; Alsalem, M.A.; Albahri, A.S.; Mahmoud, M.A.; Alzubaidi, L.; Alamoodi, A.H.; Sharaf, I.M. An Improved Best-Worst Method Integrated with Combined Compromise Solution for Evaluating Large Language Models. *Int. J. Intell. Syst.* **2025**, *2025*, 2376097. <https://doi.org/10.1155/int/2376097>.
25. Yazgan, H.R.; Boran, S.; Goztepe, K. An ERP software selection process using AHP. *Expert Syst. Appl.* **2009**, *36*, 9214–9222. <https://doi.org/10.1016/j.eswa.2008.12.022>.
26. Ghosh, N.; Ghosh, S.K.; Das, S.K. SelCSP: A framework to facilitate selection of cloud service providers. *IEEE Trans. Cloud Comput.* **2015**, *3*, 66–79. <https://doi.org/10.1109/tcc.2014.2328578>.
27. Mikalef, P.; Boura, M.; Lekakos, G.; Krogstie, J. Artificial intelligence capability: Conceptualization, measurement calibration, and empirical study on its impact on business performance. *Inf. Manag.* **2021**, *58*, 103434. <https://doi.org/10.1016/j.im.2021.103434>.
28. Qu, Y.; Dai, Y.; Yu, S.; Tanikella, P.; Pillai, M.; Chen, W.; Xie, J.; Ren, Y.; Wang, D.; Wang, Y.; et al. PrecLLM: A Privacy-Preserving Framework for Efficient Clinical Annotation Extraction from Unstructured EHRs using Small-Scale LLMs. *arXiv* **2024**, arXiv:2412.02868. <https://doi.org/10.48550/arXiv.2412.02868>.
29. Hutto, C.; Gilbert, E. VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. In Proceedings of the Eighth International Conference on Weblogs and Social Media (ICWSM-14), Ann Arbor, MI, USA, 1–4 June 2014. <https://doi.org/10.1609/icwsml.v8i1.14550>.
30. Baccianella, S.; Esuli, A.; Sebastiani, F. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), Valletta, Malta, 17–23 May 2010; pp. 2200–2204.
31. He, R.; McAuley, J. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In Proceedings of the 25th International Conference on World Wide Web (WWW '16), Montreal, QC, Canada, 11–15 April 2016; pp. 507–517. <https://doi.org/10.1145/2872427.2883037>.
32. Le, H.T.M.; Phan-Thi, T.-A.; Nguyen, B.T.; Nguyen, T.Q. Mining online hotel reviews using big data and machine learning: An empirical study from an emerging country. *Ann. Tour. Res. Empir. Insights* **2025**, *6*, 100170. <https://doi.org/10.1016/j.annale.2025.100170>.
33. Xu, L.; Hu, H.; Zhang, X.; Li, L.; Cao, C.; Li, Y.; Xu, Y.; Sun, K.; Yu, D.; Yu, C.; et al. CLUE: A Chinese Language Understanding Evaluation Benchmark. In Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020), Barcelona, Spain, 8–13 December 2020; pp. 4762–4772. <https://doi.org/10.18653/v1/2020.coling-main.419>.
34. Reis, M.J.C.S. Edge-FLGuard+: A Federated and Lightweight Anomaly Detection Framework for Securing 5G-Enabled IoT in Smart Homes. *Future Internet* **2025**, *17*, 329. <https://doi.org/10.3390/fi17080329>.
35. Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. *arXiv* **2020**, arXiv:1905.00537. <https://doi.org/10.48550/arXiv.1905.00537>.
36. Koo, T.K.; Li, M.Y. A guideline of selecting and reporting intraclass correlation coefficients for reliability research. *J. Chiropr. Med.* **2016**, *15*, 155–163. <https://doi.org/10.1016/j.jcm.2016.02.012>.
37. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. BERTScore: Evaluating Text Generation with BERT. In Proceedings of the 8th International Conference on Learning Representations (ICLR 2020), Addis Ababa, Ethiopia, 26–30 April 2020. <https://doi.org/10.48550/arXiv.1904.09675>.
38. Popović, M. chrF: Character n-gram F-score for automatic MT evaluation. In Proceedings of the Tenth Workshop on Statistical Machine Translation, Lisbon, Portugal, 17–18 September 2015; pp. 392–395. <https://doi.org/10.18653/v1/W15-3049>.
39. Chatila, R.; Havens, J.C. The IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems. In *Robotics and Well-Being*; Springer: Cham, Switzerland, 2019; pp. 11–16. https://doi.org/10.1007/978-3-030-12524-0_2.

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.