International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(21s) 2026 pp. 2007-2017
https://doi.org/10.70917/ijcisim-2026-5976
Article

From Classical Ensembles to Agentic Intelligence: Modernizing Automated Incident Ticket Routing in Telecommunications

Bhalchandra Shirole

Independent Researcher, USA
ORCID: 0009-0005-7731-6251

Abstract: Automated incident ticket routing in telecommunications, historically addressed through TF-IDF vectorization paired with XGBoost and Random Forest classifiers, has reached a performance ceiling defined not by model capability but by the data engineering architectures that feed those models. This paper examines a prior machine learning framework for classifying and routing Remedy tickets within a major telecommunications carrier, identifies the specific pipeline failures responsible for its structural limitations, and proposes a three-layer modernization comprising domain-adaptive Transformer fine-tuning (ModernBERT with Low-Rank Adaptation), Retrieval-Augmented Generation for resolution recommendation, and a constrained agentic orchestration layer for priority incidents. Each layer is specified at the data pipeline level, not only the model level, addressing ingestion, feature store, vector store, and MLOps infrastructure. The central contention is that model architecture is the secondary constraint in ITSM automation systems, and the binding factor is the quality and adaptability of the data pipeline surrounding any model. The data engineering investment required to sustain these layers in production feature stores, schema contracts, event-driven ingestion, RAGOps monitoring, and continuous parameter-efficient adaptation is where the real implementation work lies, and where prior literature on ITSM automation has been systematically underweighted. This paper presents a critical retrospective of a deployed production system and a modernization roadmap grounded in published empirical results.

Keywords: AIOps, ITSM ticket classification, data pipeline architecture, Transformer fine-tuning, Retrieval-Augmented Generation, MLOps

1. Introduction

Incident ticket routing in a telecommunications carrier is not a clean classification problem. What arrives in a Remedy queue is a hybrid artifact: part automated network alarm, part unstructured engineer narrative, generated under operational pressure with inconsistent documentation discipline and against the backdrop of continuously changing network infrastructure. Routing such a ticket to the correct resolver group, Optical Transport, IP Backbone, RAN Engineering, among dozens of others, requires understanding entity relationships and temporal context that no bag-of-words model can recover.

The machine learning framework this paper revisits solved this problem using TF-IDF vectorization feeding into XGBoost, with Random Forest providing feature-level interpretability and K-Means clustering serving as a drift proxy. At the time of deployment, the system reduced first-assignment ticket bounce rate by roughly a quarter over the deployment period, with a corresponding improvement in Mean Time to Repair absolute ticket volumes and business-specific performance data are withheld under employer confidentiality. The performance improvement, however, masked a structural fragility: the system's operational accuracy was determined less by its model choices than by the consistency or inconsistency of the data pipeline sustaining it. Recent scholarship confirms that classical

Copyright: © 2026 by the authors

methods underperform relative to fine-tuned Transformer models on incident ticket classification, and that the gap is widened by superior data infrastructure (Liu et al., 2025; Ticket-BERT, 2023).

Proposing better models without proposing better pipelines solves the wrong problem. A Transformer fed by a poorly governed pipeline will produce superior initial accuracy and degrade through the same failure modes, including schema drift, unchecked label noise, and training-serving skew as its classical predecessor. The argument of this paper is that model selection is the secondary constraint, and pipeline architecture, schema governance, and adaptation infrastructure are the primary ones.

Three contributions follow: a critical retrospective of the classical pipeline's specific data engineering failure modes, a three-layer modernized architecture grounded in current empirical literature, and a detailed specification of the data and ML pipeline infrastructure required to sustain that architecture in a production telecommunications environment. The analysis proceeds from the perspective of a practitioner who has built large-scale, production-grade data platforms for enterprise environments and an experience base that informs the infrastructure design recommendations throughout.

2. Data Engineering: The Foundation of Any ML System

Treating the model as the primary deliverable and the pipeline as a secondary implementation concern is the most common and most expensive error in ITSM automation projects. In the author's production experience, the model represents a small fraction of a working ML system, roughly five percent, with the remaining share consisting of the ingestion logic, schema validation, feature computation, serving architecture, monitoring, and adaptation mechanisms that collectively determine whether the model's theoretical capability translates to sustained operational performance, a proportion consistent with the technical-debt analysis of Sculley et al. (2015). Understanding where that infrastructure failed in the classical framework is necessary before any modernization can be specified.

2.1 Source Data Characteristics and Ingestion Challenges

Remedy tickets are structurally heterogeneous at scale. Three distinct creation pathways, automated network management systems, service desk agents, and self-service portals produce records with materially different field completion profiles. Short descriptions are reliably populated resolution notes, and affected configuration items may be absent in the majority of records, and this missingness is not random. It correlates with the ticket's source system and the documentation culture of the creating team, a systematic pattern that, when unaddressed in the training corpus, introduces class-conditional label noise and reduces classification confidence on the most diagnostically rich ticket categories (Breck et al., 2019).

Schema-aware validation at the point of capture is the architectural response: records falling below minimum field completion thresholds must be rejected or quarantined before they enter the training corpus. In the classical framework, this validation was applied inconsistently. A rigorous data validation framework specifying expected schema, value ranges, and field completion requirements as versioned artifacts updated alongside model code is a prerequisite for any ML system operating on enterprise operational data, regardless of the model architecture chosen (Breck et al., 2019).

2.2 Domain-Specific Preprocessing Pipeline

Generic NLP preprocessing destroys the signal that makes telecom ticket text classifiable. Standard tokenizers strip the acronyms that carry the most diagnostically significant content, BGP, DWDM, LTE, OTN, MPLS, and SDH, treating them as stopwords rather than as structured vocabulary with operational meaning. Node IDs, circuit references, and IP addresses must be preserved in normalized form because their structure encodes routing topology information that is among the most predictive features for resolver group assignment.

Four components constitute a production-grade preprocessing pipeline for this domain: a curated telecom-domain vocabulary protecting critical acronyms from stemming and stop-word removal, a named entity normalization

layer canonicalizing node IDs, circuit references, and IP ranges into consistent token forms, a text cleaning layer removing ephemeral auto-acknowledgement boilerplate without discarding engineer-authored diagnostic notes, and a metadata extraction layer structuring implicit ticket fields, severity escalation history, reassignment count, and time-to-first-response as explicit numerical features. Each of these components must be version-controlled alongside the model. Any preprocessing change not reflected consistently in both training and inference pipelines becomes a source of training-serving skew.

2.3 Feature Store Architecture

Computing features inline at training time and repeating that computation at inference time without a shared layer is the architectural origin of training-serving skew, one of the most common and most quietly damaging sources of ML production degradation (Sculley et al., 2015). The classical framework did not address this feature, and computation was decoupled from a shared store, and the training and serving environments diverged silently as upstream schema changes accumulated.

Three feature families require distinct partition strategies in the Remedy ticket use case: ticket-level text embeddings computed by the fine-tuned encoder and cached for retrieval, structured metadata features refreshed on each ticket update event, and network-context features derived from live topology data, device health scores, active maintenance windows, and adjacent incident density that change independently of the ticket itself. Partitioning these families by update cadence is the critical design decision: fast-changing network-context features must not trigger unnecessary recomputation of expensive text embeddings, and slow-changing metadata features must not be cached on the same refresh schedule as real-time topology signals. This partition-by-cadence pattern, directly analogous to how advertising ML platforms separate user session signals from long-horizon audience features in feature store design, eliminates the skew problem structurally rather than operationally.

2.4 Event-Driven Pipeline Architecture

Coupling inference latency to a batch export schedule is not a minor engineering trade-off for Priority-1 and Priority-2 - incidents it is an operational failure. The classical framework's batch-export architecture introduced routing latency of several hours between ticket creation and assignment decision, defeating the operational purpose of automation for the incident categories where speed matters most.

The replacement architecture is event-driven. Every ticket creates and updates an event that emits a message to a durable message broker. Containerized consumers subscribe to these streams, trigger preprocessing and feature extraction, invoke the inference service, and write routing decisions back to Remedy, all within a latency budget measured in seconds. Decoupling the inference service from the Remedy database transaction path has a secondary consequence as important as the latency reduction: the ticketing system's availability is never contingent on the ML inference service's availability. The inference service can be scaled, updated, or rolled back independently without disrupting service desk operations. This separation of concerns is a foundational data platform engineering principle absent from the original framework.

3. Background and Related Work

3.1 Classical Ensemble Baseline

XGBoost served as the primary multi-class classifier in the original framework, with Random Forest providing interpretability through Gini importance scores and K-Means clustering on low-confidence ticket batches acting as a proxy for concept drift detection (Breiman, 2001; Chen & Guestrin, 2016). The architecture was operationally effective at deployment. Its structural constraints, however, were hard-coded: TF-IDF discards syntactic and semantic context, and tree-based models cannot capture long-range dependencies in multi-sentence descriptions, and K-Means cluster interpretation required manual review, introducing a human bottleneck into an otherwise automated workflow.

3.2 Transformer-Based Ticket Classification

Bidirectional contextual embeddings, established by BERT (Devlin et al., 2019), represent a qualitatively different approach to ticket text. A ticket reading "BGP session flap following maintenance window on PE-router in Dallas NOC" contains entity relationships, timing, device type, location, and event type that no bag-of-words representation captures. Transformer attention mechanisms learn these dependencies contextually, and their value on incident ticket corpora is empirically demonstrated: Ticket-BERT (2023) showed the superiority of domain-adaptive language models over baseline classifiers on Azure incident datasets, specifically for the machine-generated-versus-human-authored variability present in real-world Remedy data.

Principled evaluation of Transformer performance across enterprise-scale classification benchmarks has advanced alongside architectures (Chang et al., 2024). ModernBERT, a 2024 successor architecture trained on two trillion tokens with an 8,192-token context window, reduces training time by approximately three times compared to the original BERT while outperforming it by three percent on F1-score on 77-class classification benchmarks (Warner et al., 2024). Across a ticket corpus spanning 100-plus engineering groups, that performance differential is operationally material.

3.3 RAG for Resolution Recommendation

The architectural contribution of Lewis et al. (2021), grounding generative model outputs in retrieved external evidence rather than relying on parametric memory addresses the hallucination and knowledge-staleness problems that make purely generative approaches unsuitable for operational support systems. Applied to IT support, a RAG system combining an encoder classifier with a generative LLM resolved support cases from ticket subject and description alone, with recommendations cited to official documentation (Isaza et al., 2024). In the telecommunications domain, RAG augmented with dynamic knowledge graphs continuously updated with live network telemetry has been shown to reduce hallucination by grounding responses in verified, current topology data (Alabbasi et al., 2024). Production RAG system design principles, retrieval component selection, chunking strategy, index management, and faithfulness evaluation are well-characterized in the information retrieval literature (Gao et al., 2024).

3.4 Agentic AI in Operations

Multi-agent LLM systems coordinated around specialized detection, diagnosis, and mitigation roles have demonstrated measurable advantages over prior SRE automation benchmarks: STRATUS outperformed prior systems on AIOpsLab and ITBench by at least 1.5 times (Chen et al., 2025). Parameter-efficient fine-tuning via LoRA and Retrieval Augmented Fine-Tuning have emerged as the dominant adaptation strategies for domain-specific LLM deployment in operational environments (Zhang et al., 2025). The extension of LLM-based agentic architectures to security operations contexts that share the pattern detection and triage challenges of ITSM automation provides additional evidence that this architectural direction generalizes beyond research settings (Srinivas et al., 2025).

4. Proposed Framework: Architecture and Data Pipeline Design

Three modeling layers, each with a dedicated data pipeline and connected through a shared event-driven backbone, constitute the modernized framework. Each layer can be deployed independently, enabling incremental adoption rather than requiring a full system replacement.

4.1 Layer One Domain-Adaptive Transformer Classification

4.1.1 Training Data Pipeline

Three stages govern the fine-tuning pipeline for Remedy ticket data. A historical corpus extraction spanning a minimum of three years provides adequate representation of seasonal incident patterns and all active resolver groups. A data quality gate filters records failing minimum field completion thresholds, short description, assignment group, and resolution code, and applies the domain-specific preprocessing specified in Section 2.2. A label engineering step

then resolves historical reassignment chains to their terminal resolver group, ensuring training labels reflect correct routing rather than erroneous initial assignments.

Class imbalance is addressed through stratified sampling at the batch construction stage. Stratified sampling is reported in the calibration literature to produce more consistent confidence scores in multi-class classification than post-hoc oversampling, a property expected to matter operationally when the system must decide between automated routing and human escalation.

4.1.2 Model Architecture and Fine-Tuning

Low-Rank Adaptation (LoRA) restricts gradient updates to decomposition matrices injected into the attention layers, reducing trainable parameters by up to 99 percent and making domain adaptation feasible on standard GPU infrastructure without full-cluster access (Hu et al., 2025). QLoRA extends this approach by applying quantization to base model weights before adapter injection, reducing memory requirements with minimal accuracy cost (Dettmers et al., 2023). A hierarchical classification head above the [CLS] token enables two-stage routing first to a domain (Transport, Access, Core IP), then to a specific resolver group, a structure expected to improve performance on low-frequency classes, consistent with findings in the hierarchical text-classification literature on flat single-stage classifiers underperforming on long-tail label distributions.

Interpretability is maintained through attention visualization (BertViz), replacing the Gini importance mechanism of the original Random Forest shadow model with a representation native to the Transformer architecture and amenable to operational audit.

4.2 Layer Two RAG-Enhanced Resolution Recommendation

4.2.1 Vector Store Pipeline

Resolved historical tickets, engineering runbooks, network equipment documentation, and post-incident reports are embedded and indexed in a persistent vector database, following production RAG design principles for retrieval component selection, index management, and provenance tracking (Gao et al., 2024). Document lifecycle management is a distinct pipeline concern: superseded runbooks are deprecated from the index, newly resolved tickets are ingested within minutes of resolution closure, and network topology updates are reflected in knowledge graph nodes within the same cadence.

Every document in the vector store carries a provenance record source system, ingestion timestamp, and last-verified date. Retrieved results surfaced to the LLM are filtered to exclude documents older than a configurable staleness threshold. Network equipment documentation ages faster than in most enterprise domains. Procedures written for a hardware generation three years old may actively mislead a generative model attempting to synthesize a resolution recommendation.

4.2.2 Inference and Discovery

High-confidence matches at inference time trigger a resolution recommendation synthesized by a fine-tuned LLM and cited to retrieved evidence (Isaza et al., 2024) appended to the Remedy record alongside the routing decision. Low-confidence tickets, for which no sufficiently similar historical resolution exists, enter an embedding-space clustering pipeline that groups them by semantic similarity and surfaces emerging incident patterns to engineering leads. RAG's grounding mechanism (Lewis et al., 2021) ensures that what the system retrieves, rather than what the model infers parametrically, drives the recommendation, a constraint that is not optional in a regulated operational environment.

4.3 Layer Three Agentic Orchestration for Priority Incidents

Pre-diagnostic work for Priority-1 and Priority-2 incidents, device state queries, open change ticket cross-referencing, adjacent infrastructure health checks, and Remedy record pre-population is performed autonomously before the assigned engineer opens the ticket. The design follows the STRATUS multi-agent architecture (Chen et al.,

2025), constrained by a Transactional No-Regression (TNR) specification prohibiting any action that cannot be atomically reversed. All action outputs are written to the Remedy record through the same event-driven pipeline that drives ticket ingestion, ensuring that every agentic action is logged, auditable, and replayable.

The agentic layer's value is not autonomous remediation. Its value is the quality of context it delivers to the human engineer: a pre-populated ticket reduces the time from assignment to first productive diagnostic action, and that reduction, compounded across thousands of priority incidents annually, is where the operational impact of this layer materializes.

5. Comparative Analysis: Classical vs. Modern Framework

Table 1 maps the original framework's design decisions to their modernized counterparts across eight operational and data engineering dimensions. What the comparison makes visible is that modernization is not the substitution of one model for another. Every layer from text representation through adaptation mechanism requires a redesign grounded in different architectural principles.

Table 1. Dimension-by-dimension comparison of the original and proposed frameworks.

Dimension	Original Framework	Proposed Modern Framework
Text representation	TF-IDF bag-of-words	ModernBERT contextual embeddings (8,192-token window)
Classification model	XGBoost + Random Forest ensemble	Fine-tuned Transformer with hierarchical classification head
Feature computation	Inline at training and inference time (no shared layer)	Centralized feature store partitioned by cadence
Inference latency	Batch export from Remedy database (several hours)	Event-driven pipeline ticket create/update events (seconds)
Schema validation	Inconsistent application post-hoc during preprocessing	Schema registry + versioned data contracts at event broker
Drift detection	K-Means cluster review (manual, human-bottlenecked)	Embedding-space clustering (automated surfaces emerging patterns)
Resolution support	None routing only	RAG layer with provenance-controlled vector store
Adaptation mechanism	Full retraining: 3–4-week data preparation cycle	Continuous LoRA fine-tuning triggered by drift alerts (hours)

The two dimensions with the greatest operational consequence are the training data pipeline and the adaptation mechanism. Training-serving skew in the classical system manifested as a gradual routing accuracy decline correlated with upstream schema changes, a failure mode never formally diagnosed precisely because it looked like model drift rather than infrastructure drift. The feature store architecture eliminates this structurally by ensuring the same computation graph governs both training-time feature generation and inference-time feature retrieval. The shift from a three-to-four-week full retraining cycle to continuous LoRA fine-tuning with automated drift alerting converts adaptation from a scheduled event into a continuous operational property.

In the absence of a pilot evaluation of the proposed system, Table 2 synthesizes published quantitative results for each of the modernization steps this paper proposes, allowing the existing literature to carry the empirical burden of the thesis while the paper's own contribution remains the pipeline-level architecture that ties these components together.

Table 2. Published quantitative evidence for each proposed modernization step.

Approach	Dataset / Deployment	Published Metric	Citation
Ticket-BERT (Transformer vs. classical baselines)	Azure Cognitive Services incident tickets (human-authored subset)	F1-score of 85.90 vs. 84.58 for the strongest classical baseline (Logistic Regression + BoW)	Liu et al. (2023)
TickIt (LLM-based ticket escalation)	ByteDance Volcano Engine cloud platform, production deployment	39% reduction in Mean Time to Repair versus manual escalation	Liu et al. (2025)
ModernBERT vs. BERT (encoder architecture)	77-class customer-support classification benchmark	Approximately 3x reduction in training time, F1 improvement of roughly 3 percentage points (0.93 vs. 0.90)	Warner et al. (2024)
STRATUS (agentic SRE orchestration)	AIOpsLab and ITBench SRE benchmark suites	At least 1.5x improvement in failure-mitigation success rate over prior SRE agent systems	Chen et al. (2025)

6. MLOps and RAGOps: Infrastructure for Sustained Performance

Moving from classical ML to LLM-based systems does not reduce the infrastructure burden but transforms it. MLflow model versioning (Zaharia et al., 2022), Airflow-orchestrated retraining, and F1-score threshold monitoring remain relevant but must be extended with components addressing failure modes absent from the ensemble era. Table 3 summarizes the full infrastructure stack.

Table 3. MLOps and RAGOps components for the modernized framework.

Component	Implementation	Data Engineering Role
Model versioning	MLflow (Zaharia et al., 2022)	Track experiments, parameters, metrics, and artifact lineage across fine-tuning runs
Training orchestration	Apache Airflow	DAG-based pipeline scheduling and dependency management for preprocessing and fine-tuning jobs
Feature store	Custom partitioned store	Separate cadence layers for text embeddings, structured metadata, and live network-context features
Schema registry	Apache Avro / Protocol Buffers	Enforce versioned data contracts at the event broker to catch breaking upstream schema changes before inference
Vector store	Pinecone / pgvector	Persistent embedding index for RAG retrieval provenance metadata for staleness filtering
Drift monitoring	Evidently AI	Statistical tests for embedding distribution shift triggers LoRA fine-tuning workflow on threshold breach
RAG faithfulness	RAGAS (Es et al., 2023)	Evaluate faithfulness, answer relevance, and context precision on sampled resolution recommendations

Prompt versioning	CI/CD prompt registry	Version, evaluate against the holdout set, and deploy instructional context changes as gated artifacts

6.1 The Data Contract Between Pipeline and Model

Schema changes are routine in large enterprise environments: a resolver group is renamed, a ticket category taxonomy is reorganized, a field definition is modified. Without a formal data contract or specification, versioned alongside model code, defining the schema, statistical properties, and quality constraints of the data, the model expects these routine upstream changes to silently corrupt model inputs (Breck et al., 2019). Schema registries enforce these contracts at the event broker level, catching breaking changes before they reach the inference service. The data contract is the instrument through which the pipeline's reliability guarantee is made formal rather than assumed.

6.2 Continuous Adaptation Without Full Retraining

LoRA-based adaptation is operationally significant precisely because it does not require retraining. When the embedding-space clustering pipeline surfaces a new incident pattern, a novel hardware generation producing an unrecognized failure signature, for example, a targeted LoRA fine-tuning job can be triggered on the relevant ticket subset within hours, updating routing behavior without disturbing base Transformer weights or disrupting inference for established categories (Zhang et al., 2025). QLoRA reduces the memory footprint of this process further, making continuous adaptation feasible without dedicated GPU cluster infrastructure (Dettmers et al., 2023).

Two monitoring requirements have no analogue in the classical MLOps stack. Faithfulness monitoring, confirming that LLM-generated resolution recommendations are grounded in retrieved evidence rather than parametric memory, is addressed through RAGAS evaluation metrics applied to sampled recommendation outputs (Es et al., 2023). Prompt versioning, managed through a CI/CD-gated registry, treats instructional context changes as code: versioned, evaluated against holdout sets, and deployed only when quality thresholds are met.

7. Discussion

7.1 Data Engineering Remains the Critical Path

Model architecture was not the binding constraint in the classical framework, and it will not be the binding constraint in the modernized one. The original system's failure modes, schema-unchecked label noise, training-serving skew, batch-coupled inference latency, and manual adaptation were entirely infrastructure problems. Upgrading to a Transformer without addressing the pipeline means repeating those failure modes with a more expensive model. The modernized framework invests as heavily in pipeline architecture as in model architecture because that investment is what converts superior theoretical model capability into sustained production performance.

Applying this same scrutiny to the proposed architecture, rather than only the classical one, surfaces three risks the modernized framework does not yet resolve. LoRA adapter proliferation is the first: as the embedding-space clustering pipeline triggers targeted fine-tuning jobs for new incident patterns, the adapter registry grows without an inherent retirement mechanism, and without a formal adapter lifecycle policy, an inference service can accumulate dozens of overlapping adapters whose combined routing behavior on edge-case tickets becomes difficult to reason about or roll back cleanly. Second, vector store staleness controls are calibrated against a fixed threshold, but telecommunications hardware turnover is not uniform across domains, and an aggressive network modernization cycle in one resolver group can invalidate runbooks faster than the staleness filter is tuned to detect, producing confidently wrong resolution recommendations rather than an absence of recommendations. Third, RAGAS faithfulness evaluation at production sampling rates carries a nontrivial compute and latency cost when applied continuously across a high-volume ticket stream sampling too sparsely reintroduces the undetected-hallucination risk the monitoring exists to prevent, while sampling densely enough to catch drift promptly competes for the same inference capacity the routing and recommendation services depend on. None of these risks invalidates the architecture, but a modernization proposal that does not name them has not been stress-tested against its own failure modes.

7.2 Governance and Privacy in Enterprise Fine-Tuning

Remedy ticket data is rich precisely because it contains information that creates compliance obligations. Personally identifiable information, customer-sensitive network topology details, and proprietary infrastructure descriptions are common in ticket text, and the same properties that make them useful for fine-tuning create legal exposure under GDPR and CCPA. Differential privacy applied during LoRA fine-tuning is a promising mitigation, but it introduces accuracy trade-offs that require empirical characterization before production deployment. The trade-off surface is not yet well-characterized for domain-specific ITSM corpora specifically. A formal data classification policy governing which ticket fields are eligible for model training must be established before any fine-tuning pipeline is built. This is not a compliance formality but a risk management prerequisite with direct consequences for model accuracy if violated through overly aggressive field exclusion.

7.3 Agentic Risk and Safety Engineering

Autonomous agents acting on live operational systems operate at machine speed. The TNR constraint provides formal safety guarantees at the individual-action level but does not address compound risk across multi-step decision sequences (Srinivas et al., 2025). An agent that individually satisfies TNR at each step may still produce a sequence of reversible actions whose aggregate effect is operationally harmful. Practitioners should deploy the agentic layer in read-only diagnostic mode for a minimum of 90 days, accumulating an action audit log before enabling write-back operations. The immutable, replayable audit log is the trust-building mechanism, not a logging artifact, but the evidence base through which the agentic system's behavioral consistency is demonstrated to operational stakeholders.

7.4 Societal and Operational Implications

Telecommunications infrastructure underpins services whose failure has consequences well beyond operational SLAs. Emergency response coordination, financial transaction processing, and healthcare communications depend on network availability that, at the margins, is determined by how quickly incidents are routed to the right resolver group. Each percentage-point improvement in first-assignment accuracy compounds through MTTR into availability improvements that have downstream public safety consequences. Reduced ticket bounce also eliminates redundant diagnostic cycles and unnecessary field technician dispatches, a carbon efficiency benefit that accumulates at scale from more precise routing decisions.

8. Conclusion

Two generations of automated ticket routing have now operated in telecommunications environments, separated by a decade of fundamental change in what language models can do. The classical framework delivered real value, but the field has moved past it. The movement, as this paper has argued, is not only in model architecture. It is in the data engineering discipline that makes architecture actionable: event-driven ingestion, feature stores, schema contracts, vector store provenance, RAGOps monitoring, and continuous parameter-efficient fine-tuning.

Three layers constitute the proposed modernization path: domain-adaptive Transformer classification, RAG-enhanced resolution recommendation, and constrained agentic orchestration. Their value is not realized independently of the pipeline infrastructure specified in Sections 2 and 6. The data contract, the feature partition strategy, the vector store staleness controls, and the LoRA adaptation triggers are not implementation details to revisit after architecture approval. They are the architecture.

Practitioners in enterprise data engineering across telecommunications, digital advertising, streaming platforms, and any high-volume operational event domain encounter the same structural question this system raises: whether to invest in model sophistication or pipeline quality when resources are constrained. The historical record of both generations examined here gives the same answer. The model is five percent. The pipeline governs everything else.

Future work should empirically characterize the accuracy trade-offs of differential privacy in LoRA fine-tuning on enterprise ITSM corpora, develop formal compound-action risk specifications for agentic layers beyond single-

action TNR constraints, and evaluate the portability of this three-layer architecture to ITSM environments beyond the Remedy platform.

9. Ethics Statement

Conflict of Interest: The author declares no conflict of interest.

Statement of Human and Animal Rights: This study did not involve human participants, animal subjects, or identifiable personal data requiring ethics board review.

Informed Consent: Not applicable.

AI Assistance Disclosure: Generative AI tools were used to support language editing, citation formatting, and structural revision during manuscript preparation. All technical content, architectural claims, and conclusions represent the author's own analysis and were independently verified by the author prior to submission.

Author Contributions: Bhalchandra Shirole is the sole author of this work and is responsible for the conceptualization, analysis, and writing of the manuscript in its entirety.

Acknowledgments: The author thanks colleagues in telecommunications network operations and data engineering for informal discussions that informed the practitioner perspective in this paper. No funding was received for this work.

About the Author: Bhalchandra Shirole is an independent researcher and data engineering practitioner with experience designing production-grade data pipelines and machine learning infrastructure for enterprise operational systems, including large-scale telecommunications environments. His work focuses on the intersection of data engineering discipline and applied machine learning, with particular emphasis on feature store architecture, event-driven pipeline design, and MLOps/RAGOps infrastructure for sustaining model performance in production.

REFERENCES

1. Alabbasi, N., Erak, O., Alhussein, O., Lotfi, I., Muhaidat, S., & Debbah, M. (2024). TeleOracle: Fine-tuned retrieval-augmented generation with long-context support for network operations. ArXiv.org. https://arxiv.org/abs/2411.02617
2. Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324
3. Breck, E., Polyzotis, N., Roy, S., Whang, S., & Zinkevich, M. (2019). Data validation for machine learning. Proceedings of the 2 nd SysML Conference, Palo Alto, CA, USA. https://mlsys.org/Conferences/2019/doc/2019/167.pdf
4. Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, M., & Wang, H. (2024). Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997v5 [cs.CL]. https://arxiv.org/pdf/2312.10997
5. Chang, Y., Wang, X., Wang, J., Yuan, W., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2024). A survey on the evaluation of large language models. ACM Transactions on Intelligent Systems and Technology, 15(3), 1–45. https://doi.org/10.1145/3641289
6. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785
7. Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs, arXiv. https://doi.org/10.48550/arXiv.2305.14314
8. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1. https://doi.org/10.18653/v1/N19-1423
9. Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2023). RAGAS: Automated evaluation of retrieval augmented generation. arXiv preprint arXiv:2309.15217. https://arxiv.org/abs/2309.15217
10. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., & Chen, W. (2025). LoRA: Low-rank adaptation of large language models. Proceedings of ICLR 2022. https://openreview.net/forum?id=nZeVKeeFYf9
11. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2021). Retrieval-augmented generation for knowledge-intensive NLP tasks. ArXiv.org, https://doi.org/10.48550/arXiv.2005.11401
12. Liu, F., He, X., Zhang, T., Chen, J., Li, Y., Yi, L., Zhang, H., Wu, G., & Shi, R. (2025). TickIt: Leveraging Large Language Models for Automated Ticket Escalation. ArXiv.org. https://doi.org/10.1145/3696630.3728558

13. Isaza, P. T., Nidd, M., Zheutlin, N., Ahn, J. W., Bhatt, C. A., Deng, Y., Mahindru, R., Franz, M., Florian, H., & Roukos, S. (2024). Retrieval augmented generation-based incident resolution recommendation system for IT support. arXiv preprint arXiv:2409.13707. https://arxiv.org/abs/2409.13707
14. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. Advances in Neural Information Processing Systems 28 (NIPS 2015), 28. https://proceedings.neurips.cc/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html
15. Srinivas, S., Kirk, B., Zendejas, J., Espino, M., Boskovich, M., Bari, A., Dajani, K., & Alzahrani, N. (2025). AI-augmented SOC: A survey of LLMs and agents for security automation. Journal of Cybersecurity and Privacy, 5(4), 95. https://doi.org/10.3390/jcp5040095
16. Liu, Z., Benge, C., & Jiang, S. (2023). Ticket-bert: Labeling incident management tickets with language models. arXiv preprint arXiv:2307.00108. https://arxiv.org/abs/2307.00108
17. Chen, Y., Pan, J., Clark, J., Su, Y., Zheutlin, N., Bhavya, B., Arora, R., Deng, Y., Jha, S., & Xu, T. (2025). Stratus: A multi-agent system for autonomous reliability engineering of modern clouds. Advances in Neural Information Processing Systems, 38, 50119-50165. https://proceedings.neurips.cc/paper_files/paper/2025/hash/47a6e9e2c3019f13ad94a0f259fe4970-Abstract-Conference.html
18. Zhang, L., Jia, T., Jia, M., Wu, Y., Liu, A., Yang, Y., Wu, Z., Hu, X., Yu, P. S., & Li, Y. (2025). A survey of AIOps in the era of large language models. arXiv preprint arXiv:2507.12472. https://arxiv.org/abs/2507.12472
19. Zaharia, M., Chen, A., Davidson, A., Ghodsi, A., Hong, S. A., Konwinski, A., Murching, S., Nykodym, T., Ogilvie, T., Parkhe, M., Xie, F., & Zumar, C. (2022). Accelerating the machine learning lifecycle with MLflow. IEEE Data Engineering Bulletin, 41(4). https://doi.org/10.48550/arXiv.1807.10272
20. Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., Gallagher, A., Biswas, R., Ladhak, F., Aarsen, T., Cooper, N., Adams, G., Howard, J., & Poli, I. (2024). Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. ArXiv.org. https://arxiv.org/abs/2412.13663