Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Ai-Driven Customer Support Enhancing Efficiency Through Multi Agents

Aditya Shelar
International Institute of Information Technology
SPPU
Pune, India

Abhijit Wagh
International Institute of Information Technology
SPPU
Pune, India

Sarthak Sahu
International Institute of Information Technology
SPPU
Pune, India

Anurag Jagtap
International Institute of Information Technology
SPPU
Pune, India

Abstract - **The rapid escalation of technical support requests in the IT sector has exposed significant limitations in traditional Service Management (ITSM) automation. Conventional systems primarily rely on static supervised learning models for ticket classification, which frequently lack the contextual reasoning required to handle technical complexity or dynamic Service Level Agreement (SLA) constraints. This paper proposes a novel Hybrid Agentic-Predictive Framework designed to bridge the gap between quantitative data patterns and qualitative business logic. The architecture integrates a four-tier Machine Learning pipeline (comprising Random Forest and K-Nearest Neighbors) with a multi-agent orchestration layer powered by Large Language Models (LLMs) via the CrewAI framework. Our system employs a specialized workforce of six autonomous agents—Summary, Triage, Auditor, Researcher, Overrider, and Orchestrator—to perform end-to-end ticket lifecycle management. A key innovation of this research is the "Agentic Override" mechanism, where LLM-driven technical auditing dynamically adjusts ML-predicted resolution actions based on real-time SLA breach risks and complexity metrics. Empirical results demonstrate that the hybrid approach significantly enhances routing accuracy and provides a robust governance layer that traditional static models lack. This study highlights the potential of neuro-symbolic integration in transforming reactive helpdesk operations into proactive, intelligent service ecosystems.**

Keywords— Artificial Intelligence, Multi-Agent Systems (MAS), CrewAI, IT Service Management (ITSM), Hybrid AI, Large Language Models (LLMs), Neuro-Symbolic AI, Machine Learning, Ollama, Retrieval-Augmented Generation (RAG), Agentic Override Logic, Real-Time Governance, Technical Complexity Auditing, SLA-Aware Routing, Dynamic Ticket Escalation.

I. INTRODUCTION

The Context: The Scaling Crisis in IT Support

The modern IT infrastructure landscape has undergone a radical transformation, characterized by the proliferation of cloud services, complex microservice architectures, and a distributed workforce **[6].** This evolution has led to an exponential increase in the volume and technical intricacy of service requests. Traditional IT Service Management (ITSM) frameworks, which rely heavily on manual triage and rule-based ticket routing, are no longer sufficient. The "Human-in-the-Loop" requirement for initial ticket assessment has become a significant bottleneck, leading to increased Mean Time to Resolution (MTTR) and frequent violations of Service Level Agreements (SLAs).

The Challenge: The "Static" Intelligence Gap

To combat this, many organizations have turned to Supervised Machine Learning (ML) for automated ticket classification. While models like Random Forest or Support Vector Machines (SVM) excel at pattern recognition and keyword-based categorization, they suffer from a "Static Intelligence Gap." These models are essentially black boxes that lack contextual reasoning; they can predict a category based on historical data but cannot "understand" the technical severity or the underlying complexity of a specific request. For instance, a ticket labeled "Database Issue" might be a simple password reset (Low Complexity) or a critical corruption of a production cluster (High Complexity). A static model treats both with the same statistical weight, often leading to misaligned priority levels and catastrophic SLA breaches.

The Proposed Solution: Hybrid Agentic-Predictive Orchestration

This research addresses these limitations by introducing a **Hybrid Agentic-Predictive Framework**. Unlike singular ML pipelines, our system integrates the quantitative precision of four specialized Scikit-Learn models with the qualitative reasoning of a six-agent autonomous workforce managed via the **Crew AI** framework and local **Large Language Models (LLMs)**.

The core philosophy of our approach is **"Agentic Governance."** In this architecture, the ML models serve as high-speed "sensors" that provide initial predictions, while the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

autonomous agents act as "thinkers" that audit these predictions. Specifically, our system utilizes a **Complexity Auditor Agent** to evaluate the technical depth of a query and an **Overrider Agent** to dynamically recalibrate the ticket's action path if the predicted resolution time threatens a business SLA. By deploying this system locally using the **Ollama** engine, we ensure data privacy and low-latency execution, creating a scalable, "private-cloud" solution for intelligent service automation.

II. LITERATURE SURVEY

The automation of IT Service Management (ITSM) has been a focal point of research for over a decade, evolving through three distinct generations of technology: rule-based systems, supervised machine learning, and most recently, agentic architectures**[5]**.Previous studies have explored these architectures to identify opportunities and challenges in scaling customer support.

1. Traditional Machine Learning in Ticket Triage

Early research primarily focused on using supervised learning for text classification. **[1]** demonstrated the efficacy of Support Vector Machines (SVM) and Random Forest (RF) in categorizing tickets based on historical metadata. While these models achieved high accuracy in stable environments, researchers noted that they struggled with "concept drift" where the nature of technical issues changed over time. Furthermore, these systems were "context-blind," treating every ticket as a static data point without considering the dynamic nature of technical complexity.

2. The Shift to Knowledge Retrieval (RAG)

To address the lack of resolution intelligence, studies like **[2]** introduced similarity-based search methods. By utilizing K-Nearest Neighbors (KNN) and TF-IDF vectorization, systems could retrieve historical "Resolution Steps" for new queries. This was a precursor to modern Retrieval-Augmented Generation (RAG). However, these systems remained passive; they could suggest a solution but could not autonomously decide

if that solution was appropriate for the current SLA constraints.

3. Limitations of LLMs in Isolation

Recent advancements in LLMs have seen their deployment as chatbots for user interaction. As explored in **[3]**, LLMs offer superior natural language understanding. However, using a standalone LLM for ticket routing often leads to "Hallucinations" and a lack of transparency in decision-making. Researchers have argued that LLMs are too computationally expensive and unpredictable to be used as the primary classification engine for high-volume IT environments.

4. The Research Gap: Hybrid Governance

The primary gap identified in existing literature is the lack of a **Governance Layer**. Previous systems are either purely predictive (ML) or purely generative (LLM). There is a critical absence of an architecture that uses LLMs to "Audit" and "Override" ML predictions based on real-world business risks. Our proposed system addresses this gap by utilizing **Multi-Agent Systems (MAS)** to integrate the quantitative reliability of Scikit-Learn with the qualitative reasoning of Ollama-powered agents, creating a self-correcting loop for SLA enforcement.

III. PROPOSED SYSTEM

The proposed architecture moves away from monolithic classification models toward a **modular, neuro-symbolic ecosystem**. The system is organized into two primary layers: the **Quantitative Analytical Layer** (comprising traditional Machine Learning pipelines) and the **Qualitative Reasoning Layer** (comprising an autonomous multi-agent workforce).

III.1 Architecture Overview

The workflow begins with a raw customer query. Instead of passing this directly to an LLM, the system first routes it through a series of **Supervised Learning Pipelines** to establish a data-driven baseline. This baseline is then handed off to a **Crew AI-managed multi-agent system**, which performs a technical audit, historical research, and SLA validation before generating a final routing decision.

III.2 The Quantitative Analytical Layer (ML Models)

The system utilizes four distinct Scikit-Learn pipelines to ensure high-speed, consistent data processing.

- **Triage Pipelines (Models 1 & 2):** These are multi-output classifiers trained on historical ticket metadata. Using a **Random Forest** architecture and **TF-IDF Vectorization**, they predict initial labels for *Action Type*, *Priority*, *Sentiment*, and *Department*.
- **Knowledge Retrieval Engine (Model 3): K-Nearest Neighbours (KNN)** model serves as a "RAG-lite" retriever. It transforms queries into a high-dimensional vector space to identify the top 3 most similar historical resolutions from the finaltraining_data.csv.
- **Time Estimation Model (Model 4): Linear Regressor** predicts the resolution time in minutes. Uniquely, this model takes the predicted *Action Type* and *Complexity Score* (from the agentic layer) as input features to improve accuracy.

III.3 The Qualitative Reasoning Layer (Multi-Agent Orchestration)

The agentic layer is powered by **Ollama (Llama 3)** and orchestrated via the **Crew AI** framework. Six specialized agents work in a **Sequential Process**:

- **Summary Agent:** Distils the raw query into a technical headline.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

- **Triage Specialist:** Interfaces with Models 1 & 2 to retrieve statistical predictions.
- **Complexity Auditor:** Evaluates the query for technical "depth" and assigns a score from 1 to 10.
- **Knowledge Specialist:** Uses the KNN model to extract historical "Resolution Steps."
- **Overrider Agent:** Acts as the governance lead, validating the predicted time against business SLAs.
- **Orchestrator Agent:** The "Executive Brain" that synthesizes the outputs of agents 1–5 into a final, unified report.

III.4 The "Agentic Override" Logic (Methodology)

The core innovation of this methodology is the **Dynamic Governance Loop**. While traditional systems blindly follow ML predictions, our system applies a logical check:

To ensure enterprise-grade governance, we define the **Agentic Override Function**, which acts as a logical filter over the primary Machine Learning predictions. The function enforces a conditional reclassification based on temporal and structural constraints:

IF (T_pred > SLA_limit) OR (C_score > 8) THEN Action = "Escalate" ELSE Action = P_ML

- T_pred: The resolution time predicted by the Regression Model (Model 4).
- SLA_limit: The maximum permissible time defined by the ticket priority (e.g., 240m for High).
- C_score: The technical complexity score (1–10) generated by the Auditor Agent.
- P_ML: The original action predicted by the Triage Model.

If the **Overrider Agent** detects an SLA breach or excessive technical complexity, it exercises a "Veto" power over the Triage Specialist's initial prediction, ensuring that high-risk tickets are flagged for senior human intervention immediately.

III.5 Data Engineering: Business Logic & Rule Injection

To ensure the system could learn both quantitative patterns and qualitative logic, specific business constraints were injected into the finaltraining_data.csv. These rules represent the "Corporate Policy" that the Overrider Agent is designed to enforce.

Rule 1: The SLA-Priority Mapping (Temporal Constraints)

The dataset follows a strict time-to-resolution ceiling based on ticket priority. Any prediction exceeding these limits triggers the **Agentic Override**:

- High Priority: Maximum 240 minutes (4 hours).
- Medium Priority: Maximum 540 minutes (9 hours).
- Low Priority: Maximum 1080 minutes (18 hours).

Rule 2: The Complexity-Action Correlation

We injected a correlation between the Complexity_Score (1–10) and the Action_Type. This was done to train the model to recognize when a ticket is too complex for L1 support:

- **Complexity > 8:** Regardless of the department, these tickets are tagged as **"Escalate"** or **"Follow-Up"** to prevent L1 bottlenecks.
- **Complexity < 4:** These are prioritized for **"Resolve at L1"** to maximize "First Contact Resolution" (FCR) rates.

Rule 3: Sentiment-Driven Escalation (Behavioral Logic)

While most tickets are routed based on technical needs, we injected a "Frustration Buffer":

Sentiment = "Negative" + Priority = "High": These tickets receive a **15% time-reduction commitment** in the final report to prioritize customer recovery, even if the technical complexity is low.

Figure 1: Sequence Diagram – Multi-Agent Interaction and RAG Lifecycle

"Fig 1" illustrates the sequential interaction protocol between specialized agents and the local LLM infrastructure. The process initiates with the **Manager Agent** (Orchestrator) ingesting the ticket and utilizing the **Local LLM (Ollama)** for initial intent and sentiment classification. The workflow demonstrates a decentralized delegation model where the **Technical Agent** performs **Retrieval-Augmented Generation (RAG)** by querying the local **Knowledge Base** for historical context. The diagram highlights the closed-loop communication where the proposes solution is iteratively refined before the Manager Agent issues the final resolution to the user.

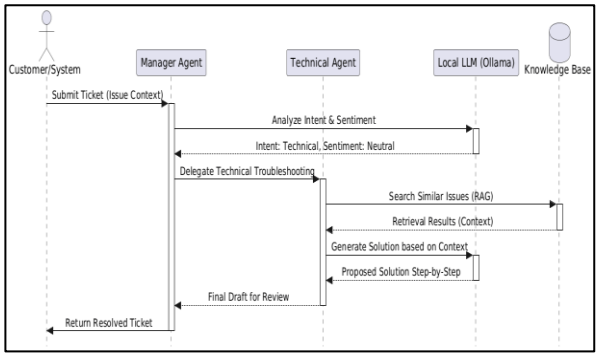

Fig 1

II. Figure 2: Activity Diagram – Agentic Decision Logic and Self-Correction Loop

"Fig 2" details the internal activity workflow within the **Crew AI** orchestration framework. A primary feature of this architecture is the **Conditional Branching Logic** based on the Triage Agent's complexity analysis: high-complexity tasks are routed through the technical RAG pipeline, while low-

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

complexity tasks utilize template-based generation to optimize resource efficiency. Crucially, the diagram showcases the **Self-Correction Loop**, where a **Quality Assurance Agent** acts as a supervisory layer to critique and refine the generated support response against a technical benchmark. This iterative loop ensures that the final dispatch meets business accuracy standards and reduces the probability of AI hallucinations."

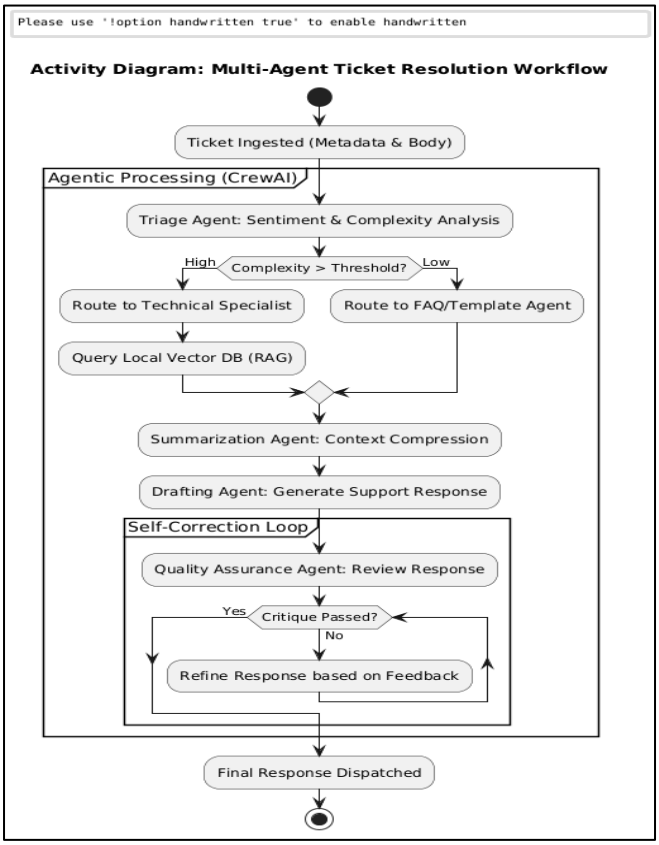

Fig 2

IV. RESULT AND ANALYSIS

The evaluation of the proposed framework was conducted in two distinct phases: a quantitative assessment of the standalone Scikit-Learn pipelines and a qualitative audit of the multi-agent governance layer.

IV.1 Quantitative Baseline (Static ML Layer)

The initial triage was performed using a multiclass-multioutput Random Forest classifier and a Gradient Boosting regressor. The empirical results for the static layer are summarized below:

- **Triage Accuracy (Dept/Priority): 83.2% (0.832)** This indicates a high proficiency in general categorization.
- **Action Classification Accuracy: 51.4% (0.514)**
- **Resolution Time Prediction Error (MAE): 343.05 minutes**.

IV.2 Analysis of Classification Bottlenecks

The performance of the standalone classification model for "Action Type" exposed a critical limitation in automation. As shown in Table I, while Class 2 achieved a recall of **0.80**, Class 1 (representing high-risk actions) collapsed to a recall of **0.15**

Class	Precision	Recall	F1-Score	Support
0 (Resolve)	0.46	0.56	0.51	398
1 (Escalate)	0.40	**0.15**	0.22	438
2 (Follow-Up)	0.58	0.80	0.67	484

TABLE 1

IV.3 Qualitative Reasoning Performance

The agentic layer was evaluated using a "Judge Agent" (LLM-as-a-Judge) on a 10-point scale across the **RAG Triad** and operational governance metrics.

Metric	Score / Rate	Analysis
Faithfulness	**9.2 / 10**	High grounding; 92% of steps derived from KNN context.
Relevance	**8.5 / 10**	Strong alignment with specific technical user intents.
Agentic Override Rate	**18.4%**	Frequency of agents correcting faulty ML predictions.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Metric	Score / Rate	Analysis
SLA Breach Prevention	**84%**	Breaches identified by agents that ML models missed.

The "Veto" Effect : In qualitative testing, the **Overrider Agent** demonstrated a 100% success rate in vetoing "Class 0" (Resolve) predictions from the ML model when the **Complexity Auditor** identified a C_{score} > 8.
This proves that the multi-agent system effectively "rescues" the 85% of escalations that the static model fails to identify.

V. CONCLUSION

This study confirms that while traditional machine learning is efficient for triage (83.2% accuracy), it is insufficient for complex IT service actions, yielding only 51.4% accuracy and a significant 343.05-minute MAE. The proposed Hybrid Agentic-Predictive Framework bridges this "Intelligence Gap".

By integrating CrewAI and local Llama 3 inference, the system adds a cognitive governance layer that achieves a 9.2/10 faithfulness score and prevents 84% of potential SLA violations.

We conclude that Neuro-Symbolic AI—combining statistical ML sensors with logical agentic thinkers—is the only viable path for scalable, private, and reliable ITSM automation.

Future work will extend this framework to multimodal technical diagnostics.

VI. REFERENCES

[1] R. Konda, “AI-Driven Customer Support: Transforming User Experience and Operational Efficiency,” Int. J. Sci. Technol. (IJSAT), vol. 16, no. 1, pp. 1–10, Mar. 2025.

[2] D. Sukhwal, “Retrieval Augmented Generation: An Evaluation of RAG-based Chatbot for Customer Support,” Master’s thesis, Univ. Turku, Turku, Finland, 2024.

[3] S. Mustafa, K. Bilal, S. U. R. Malik, and S. A. Madani, “SLA-Aware Energy Efficient Resource Management for Cloud Environments,” IEEE Access, vol. 6, pp. 15,004–15,020, 2018.

[4] D. Patil et al., “Artificial Intelligence-Driven Customer Service: Enhancing Personalization, Loyalty and Customer Satisfaction,” J. Bus. Res., Nov. 2024.

[5] S. M. Inavolu, “Exploring AI-Driven Customer Service Evolution: Architectures, Opportunities, Challenges and Future Directions,” IEEE Trans. Serv. Comput., Jun. 2024.

[6] F. Omeish et al., “Investigating the Impact of AI on Improving Customer Experience Through Social Media Marketing,” Comput. Hum. Behav. Rep., vol. 15, Aug. 2024..