ISSN: 3139-0862

TicketFlow Nexus: A Risk-Aware AI Framework for Intelligent Support Ticket Classification, Prioritization, Routing, and Resolution

Praneet Nagesh Poojary[#1], Sumeet Sudhir Tawte[#2], Pranay Dhirendra Pandey[#3], Aditya Dinesh Gawade[#4]

[#]*Department of Computer Engineering, Watumull Institute of Engineering and Technology, Mumbai, India*

[1]buildwithpraneet@gmail.com , [2]stawte16@gmail.com , [3]pandeypranaydm555@gmail.com , [4]adityagawade021@gmail.com

***Abstract*—**Customer support systems must process incoming tickets, determine their urgency, assign them to the appropriate teams, and resolve them while meeting service-level expectations. Manual ticket handling can lead to inconsistent prioritization, delayed routing, and unnecessary workload for support personnel. This paper presents TicketFlow Nexus, a proposed AI-assisted framework for intelligent support ticket classification, prioritization, routing, and resolution. The framework combines natural language processing for ticket analysis with a rule-based priority engine, department routing, service-level agreement monitoring, and a controlled AI-assisted resolution workflow. A retrieval-augmented generation component and restricted tool access are proposed to support knowledge-grounded responses, while response validation and human escalation provide safeguards for cases that cannot be resolved reliably through automation. The architecture separates model predictions from operational decisions and workflow state transitions to improve traceability and control. The current work describes the proposed design and implementation approach; quantitative performance claims are deferred until datasets and experiments are available.

Keywords*—*Artificial intelligence, natural language processing, ticket classification, ticket prioritization, intelligent routing, retrieval-augmented generation, human-in-the-loop systems.

I. INTRODUCTION

Customer support platforms are widely used to manage service requests, technical issues, account problems, and other customer concerns. As the volume of incoming tickets increases, support teams must identify the nature of each request, determine its urgency, assign it to an appropriate department, and decide whether it can be resolved automatically or requires human intervention. When these activities depend heavily on manual processing, ticket handling can become inconsistent and time-consuming.

Artificial intelligence and natural language processing offer opportunities to automate parts of this workflow. Text classification can help identify ticket categories, while sentiment analysis and entity extraction can provide additional context about customer concerns. Priority scoring and routing mechanisms can use this information to support operational decisions. However, individual predictions alone do not provide a complete ticket-management solution. A practical system must also account for service-level agreements, workflow states, resolution constraints, and escalation requirements.

TicketFlow Nexus is proposed as an AI-assisted framework that combines ticket analysis, risk-aware prioritization, department routing, and controlled resolution within a unified workflow. Its architecture separates machine-learning predictions from operational decision rules and workflow state transitions. This separation is intended to make ticket handling more traceable and to reduce the risk of allowing uncertain model outputs to directly trigger consequential actions.

The framework includes a natural language processing pipeline for extracting ticket information, a priority engine for assessing urgency and service-level risk, and a routing mechanism for directing tickets to suitable support teams. For tickets considered appropriate for automated assistance, a retrieval-augmented generation component and restricted tools are proposed to support knowledge-grounded responses. Response validation and human escalation are included to handle cases where the proposed response is unsuitable, the available knowledge is insufficient, or the issue requires specialist intervention.

The main contribution of this work is the proposed integration of ticket intelligence, deterministic operational decisions, and controlled AI-assisted resolution within a layered support architecture. The framework also identifies evaluation requirements for classification, prioritization, routing, response quality, and workflow reliability. Since the implementation and experimental evaluation are still in progress, this paper presents the system design and proposed methodology rather than claiming measured performance improvements.

II. RELATED WORK

Automated support-ticket management has been studied through text classification, priority prediction, information retrieval, and AI-assisted response generation. These approaches address different stages of the support process, but their results must be interpreted according to the task, dataset, and evaluation metric used.

Selvi et al. [5] investigated customer-support ticket categorization and prioritization using natural language processing and machine-learning techniques. Their approach includes topic modelling for department-oriented categorization and urgency and impact indicators for priority identification. *Kumar and Revathy [2]* proposed an IT-support system combining TF-IDF features and Logistic Regression for ticket classification and priority prediction. Their work also connects model predictions with ticket-management functions, including dashboards and SLA monitoring. These studies provide relevant

ISSN: 3139-0862

foundations for automated triage, although the effectiveness of a model depends on the dataset, class distribution, and evaluation procedure. Category prediction and priority estimation must also be distinguished from the operational decision of assigning a ticket to a suitable agent.

Suresh and Durga Devi [6] described a customer-support system that combines TF-IDF-based classification with rule-based response generation. Mapping predefined responses to predicted categories offers a predictable approach for routine requests. However, fixed response templates may be insufficient for complex issues or cases requiring current organizational knowledge. This motivates a resolution process that considers the available evidence and the suitability of automated handling before a response is delivered.

Information retrieval provides another important component of support-ticket management. *Benitez Pereira et al. [1]* compared multiple retrieval techniques for finding relevant historical IT-support tickets and reported that a multilingual Sentence-BERT model achieved the highest relevance among the evaluated approaches. *Reimers and Gurevych [4]* introduced Sentence-BERT, which generates sentence embeddings suitable for semantic similarity comparisons. These techniques can help identify related tickets even when their wording differs. Nevertheless, similarity does not establish that a historical solution is applicable to the current issue; retrieved information must be evaluated in its operational context.

Lewis et al. [3] introduced retrieval-augmented generation (RAG), which combines information retrieval with text generation for knowledge-intensive tasks. By supplying retrieved passages as context, RAG offers a way to generate responses informed by external knowledge sources. *Yao et al. [7]* presented ReAct, an approach that combines language-model reasoning with actions and observations from external tools. These methods provide foundations for knowledge-assisted responses and multi-step tool interaction. Their reported benchmark results, however, do not directly establish safe or effective automated resolution in a customer-support environment.

The reviewed studies address important individual capabilities, but the supplied literature does not establish the effectiveness of a single integrated workflow that combines contextual ticket analysis, priority and SLA-risk assessment, operational routing, semantic retrieval, restricted tool use, response validation, and human escalation. TicketFlow Nexus is proposed to investigate this system-level integration. Its design separates model predictions from deterministic operational decisions and workflow-state transitions, while directing uncertain or unsuitable cases to human review. The effectiveness of this integration remains to be established through implementation and task-specific evaluation.

III. Proposed Methodology

TicketFlow Nexus is proposed as a risk-aware, AI-assisted support-ticket management framework that combines ticket analysis, priority assessment, department routing, and controlled resolution. The methodology defines the sequence of operations performed on an incoming ticket and the conditions under which it proceeds toward automated assistance or human handling. Machine-learning predictions are used as inputs to operational decisions, while predefined rules govern priority thresholds, permissions, escalation, and workflow-state transitions.

A. Ticket Intake and Preprocessing

The process begins when a customer submits a support ticket containing a subject, description, and other relevant details. The system validates the submitted fields and checks whether the request satisfies the required input format. Once accepted, the ticket is assigned a unique identifier and recorded with an initial workflow state.

The ticket text is then prepared for analysis. Preprocessing may include handling missing or malformed text, removing irrelevant formatting, and normalizing the input where appropriate. The original ticket information is retained so that the submitted request remains available for review. These steps provide a consistent input for the subsequent analysis stages.

B. NLP-Based Ticket Intelligence

The preprocessed ticket is analysed to identify information that can support classification and operational decisions. The proposed NLP pipeline includes ticket intent and category classification, department identification, sentiment analysis, entity extraction, and urgency estimation. The precise set of tasks will depend on the availability of suitable training data and the final model implementation.

DistilBERT is considered a candidate model for contextual ticket classification, while other task-specific models or conventional machine-learning methods may be evaluated where appropriate. The resulting predictions and confidence information will be checked before they are used by downstream components. When predictions are uncertain or inconsistent, the system may request additional information or direct the ticket for manual review rather than relying on an unverified output.

C. Risk-Aware Priority and SLA Assessment

After ticket analysis, the proposed priority engine estimates the operational priority of the ticket. Instead of relying exclusively on the predicted category, the framework considers urgency, impact, severity, business impact, and the risk of missing a service-level agreement (SLA) deadline.

An initial weighted scoring model is proposed as follows:

$$P = 0.25U + 0.25I + 0.20S + 0.15B + 0.15R \quad (1)$$

where P represents the priority score, U represents urgency, I represents impact, S represents severity, B represents business impact, and R represents SLA risk. Each input is normalized to a scale from 0 to 100 before the weighted score is calculated.

The weights in the proposed priority score represent an initial design choice intended to combine immediate urgency, service impact, issue severity, business consequences, and the

ISSN: 3139-0862

risk of violating service-level agreements (SLAs). Urgency and impact receive the highest individual weights because they directly influence the need for timely intervention. Severity represents the seriousness of the technical or service issue, while business impact captures the consequences for business operations. SLA risk accounts for the remaining time before a deadline and the likelihood that the ticket will breach its service target. Each factor is assigned a normalized score between 0 and 100 using predefined criteria. The resulting weighted score is mapped to the proposed priority bands. These weights and thresholds are heuristic and require validation using labelled tickets, operational requirements, or expert review before they can be considered effective.

The initial priority bands are defined as Low (0–24), Medium (25–49), High (50–74), and Critical (75–100). These thresholds and weights are preliminary design choices and have not yet been experimentally validated. SLA rules will additionally determine relevant deadlines and whether warnings or escalation actions are required. Critical incidents may be handled through predefined override rules where the final implementation supports them.

D. Department Routing and Workflow Management

The routing component uses the ticket's predicted category and department, together with its assigned priority and applicable operational constraints, to identify an appropriate destination. Where agent-level assignment is implemented, the system may also consider agent eligibility, workload, and availability.

The routing decision will be governed by predefined application rules rather than by an unconstrained model output. The selected department or agent, the resulting assignment, and relevant decision information may be recorded for traceability. The workflow controller will maintain valid ticket states and prevent unauthorized transitions, ensuring that routing, escalation, and closure follow the configured rules.

E. AI-Assisted Resolution and Human Escalation

After classification and routing, the system determines whether the ticket is suitable for AI-assisted resolution. For eligible requests, a proposed retrieval-augmented generation (RAG) component will retrieve relevant information from an approved knowledge base and provide that context to a language model for response generation. Restricted tools or authorized business APIs may be used where the request requires an approved action and the necessary integration is available.

Before a generated response is accepted, it should be checked for relevance, consistency with the retrieved information, and compliance with the permitted actions. A response must not be treated as a successful resolution merely because the model has generated an answer.

Tickets involving insufficient supporting information, uncertain predictions, failed response validation, unsupported requests, or specialist requirements will be directed to a human agent or review queue. The final resolution and closure will be reflected in the ticket record according to the configured workflow rules.

F. Data Persistence, Monitoring, and Model Improvement

Throughout the workflow, relevant ticket information, predictions, priority scores, assignment decisions, SLA-related events, and state transitions are intended to be recorded for authorized access and subsequent analysis. These records will support ticket tracking, operational auditing, and evaluation of the proposed framework.

Model improvement will be handled as a separate process from live ticket processing. Where suitable labelled data become available, the data can be prepared for training and evaluation using separate training, validation, and test sets. Candidate models will be assessed using appropriate metrics before any approved update is deployed. Monitoring, version control, and rollback procedures may be incorporated according to the final implementation.

The proposed methodology therefore establishes a controlled sequence from ticket submission to resolution or human escalation. Its effectiveness will be determined through implementation, functional testing, and task-specific evaluation rather than assumed in advance.

IV. SYSTEM ARCHITECTURE AND TECHNOLOGY

TicketFlow Nexus is proposed as a layered software system that integrates a web interface, application services, persistent storage, and AI-based processing components. Each layer has a defined responsibility and communicates with the other layers through controlled interfaces. This modular design is intended to support maintainability, traceability, and independent testing of system components. The architecture distinguishes application-level decisions from machine-learning inference and keeps workflow control within the backend.

A. Frontend Technologies

React.js with Vite is proposed for developing the user interface. The frontend will provide separate views for customers, support agents, supervisors, and administrators according to their roles. React Router may be used to manage navigation between these views, while form-handling libraries may support ticket input and validation. The interface will communicate with backend services through REST APIs and display ticket information returned by the application layer.

ISSN: 3139-0862

B. Backend and Application Services

Node.js and Express.js are proposed for the backend application. This layer will expose REST endpoints, validate requests, enforce access permissions, and coordinate communication between the frontend, database, and AI services. JSON Web Tokens (JWT) may be used for session authentication, subject to the final implementation. Role-based access control will restrict operations according to user permissions.

The backend will also maintain application-level rules and workflow state. Keeping these responsibilities in the application layer allows model services to return predictions without directly controlling authorization, ticket closure, or other restricted operations.

C. Database Technology

MongoDB with Mongoose is proposed for persistent data storage. MongoDB will store ticket records, classification outputs, priority information, department assignments, SLA-related fields, workflow states, and audit events. Mongoose will provide schema definitions and validation for database operations.

The database layer will maintain the information required for ticket tracking, operational traceability, and subsequent performance analysis. Access to stored records will be governed by backend authorization rules so that users can access only the information permitted by their roles.

D. Machine Learning and Natural Language Processing

Python is proposed as the primary language for the machine-learning components. Hugging Face Transformers and PyTorch may be used to implement and execute contextual language models. DistilBERT is a candidate model for ticket classification because of its relatively compact architecture. Supporting libraries such as scikit-learn, pandas, and NumPy may be used for data preparation, baseline modelling, and evaluation.

The AI layer is intended to expose predictions and analytical features to the application layer. Depending on dataset availability and experimental findings, these may include ticket category, intent, department, sentiment, urgency, and extracted entities. The final model selection and configuration will depend on task requirements, computational constraints, and evaluation results. These technologies remain proposed components until their implementation is confirmed.

E. AI-Assisted Resolution and Knowledge Retrieval

A retrieval-augmented generation (RAG) component is proposed for knowledge-assisted response generation. It will connect a language model with an approved knowledge base containing relevant support documents or historical information. A retrieval mechanism will identify potentially useful material and provide it as context for response generation.

An LLM provider API or SDK may be used to connect the application to a language model. A retrieval library, embedding model, and vector-search component may be selected according to the final implementation. Access to external tools and business APIs will be restricted through explicit permissions. Response validation and escalation decisions will remain subject to application-level controls. The final retrieval and orchestration technologies must be confirmed through implementation and testing.

F. AI Service Integration and Workflow Control

The Python-based AI components may be deployed as a separate service using FastAPI. This arrangement would allow the Node.js backend to communicate with model inference endpoints without embedding model execution directly in the application server. Communication may use REST requests and JSON payloads, with appropriate input validation, error handling, and response checks.

The backend will remain responsible for authorization, operational rules, and workflow-state transitions. The AI service will provide analytical outputs, while the application layer will determine how those outputs may be used. This separation is intended to improve modularity and make it easier to test or replace individual models without redesigning the entire application.

G. Monitoring, Evaluation, and Model Lifecycle

The proposed architecture includes supporting mechanisms for logging, evaluation, and controlled model updates. Ticket records, model outputs, routing decisions, and workflow events may be retained for authorized review and analysis. Classification performance can be evaluated using precision, recall, and F1-score, while routing, prioritization, SLA handling, and response quality require task-specific evaluation procedures.

Model updates are intended to follow a controlled lifecycle involving data preparation, validation, evaluation, approval, deployment, and monitoring. Where supported by the implementation, model versioning and rollback mechanisms may be used to manage changes. Feedback can inform offline analysis and future improvements rather than automatically modifying production models.

ISSN: 3139-0862

Overall, the proposed architecture combines a React-based frontend, Node.js and Express application services, MongoDB persistence, and Python-based AI components. Optional language-model, retrieval, and model-serving integrations can extend the system while preserving a clear separation between inference, application rules, and workflow control. The final component selection and integration will be established through implementation and testing.

V. EVALUATION METHODOLOGY

The evaluation of TicketFlow Nexus is planned to examine the reliability of ticket classification, prioritization, department routing, and AI-assisted resolution. Since the implementation and experimental evaluation are still in progress, this section defines the proposed evaluation procedure and metrics rather than reporting measured results. The evaluation will consider both individual component performance and the consistency of the complete ticket-management workflow.

A. Dataset Preparation

The evaluation will require a labelled support-ticket dataset containing ticket descriptions and, where available, their categories, assigned departments, priority levels, and resolution information. Before training and testing, the data will be checked for missing values, duplicate records, inconsistent labels, and irrelevant text. Preprocessing will be applied consistently to prevent differences between training and evaluation inputs.

The dataset will be divided into training, validation, and test subsets where model training is required. The test subset will be kept separate from model development and parameter selection to reduce data leakage. The availability, size, and suitability of the dataset must be established before quantitative evaluation can begin.

B. Ticket Classification Evaluation

Ticket classification will be evaluated by comparing predicted categories with the corresponding ground-truth labels. Precision, recall, and F1-score will be used to measure classification performance. Accuracy may also be reported when it provides a meaningful summary of the results. For datasets with imbalanced categories, class-wise metrics and macro-averaged scores will be considered to avoid allowing frequent categories to dominate the evaluation.

The classification results will be examined to identify categories that are frequently confused. This analysis will help determine whether the selected model and preprocessing procedure are suitable for the ticket categories represented in the dataset.

C. Priority Assessment and SLA Evaluation

The priority engine will be evaluated by comparing assigned priority levels with reference labels or predefined operational rules. The assessment will examine whether urgent, high-impact, and deadline-sensitive tickets receive appropriate priority levels. Where suitable labelled data are available, classification metrics may be used to assess agreement between predicted and reference priorities.

SLA-related evaluation will examine whether tickets are assigned appropriate deadlines and whether warning or escalation rules are triggered under the defined conditions. Test cases will cover different priority levels, approaching deadlines, overdue tickets, and critical incidents. The proposed priority thresholds and weighting scheme will be treated as initial design choices until their suitability is assessed.

D. Department Routing Evaluation

Routing performance will be assessed by comparing the department selected by the system with the expected department in the evaluation data. Routing accuracy and department-wise errors will be examined where reference assignments are available. Additional functional tests will check whether routing rules respect department eligibility, priority requirements, and applicable assignment constraints.

If workload-aware agent assignment is implemented, further tests may examine whether eligible agents are selected according to the configured workload and availability rules. Such results will be reported only after the required assignment logic and evaluation data are available.

E. AI-Assisted Resolution and Human Escalation

The AI-assisted resolution component will be evaluated using test tickets with relevant knowledge-base material and reference solutions, where available. Assessment criteria may include factual correctness, relevance to the ticket, consistency with retrieved evidence, and compliance with tool permissions. Retrieved documents will also be examined for their relevance to the original request.

Human-escalation tests will cover insufficient knowledge, uncertain predictions, unsupported requests, failed response validation, and issues requiring specialist intervention. The evaluation will verify whether these cases are directed to the appropriate review path rather than being marked as successfully resolved without sufficient evidence.

ISSN: 3139-0862

F. Workflow Reliability and Testing

Functional and integration tests will examine the interaction between ticket creation, classification, prioritization, routing, resolution, escalation, and closure. Tests will verify that invalid state transitions are rejected, access permissions are enforced, and ticket updates are recorded consistently. Additional cases will cover malformed inputs, unavailable AI services, failed API requests, and repeated ticket submissions.

The final evaluation will report the metrics supported by the implemented system and available test data. Quantitative results, limitations, and comparisons will be added after experiments have been completed. Until then, the proposed procedure provides a basis for systematic evaluation without implying that the expected outcomes have already been achieved.

VI. CONCLUSION AND FUTURE WORK

TicketFlow Nexus presents a proposed AI-assisted framework for intelligent support-ticket classification, prioritization, department routing, and resolution. The framework brings together natural language processing, risk-aware priority assessment, service-level agreement monitoring, and controlled AI-assisted responses within a unified ticket-management workflow. Its design separates machine-learning predictions from operational decisions and workflow-state management, allowing predefined rules to govern permissions, escalation, and ticket closure.

The proposed methodology incorporates ticket preprocessing, contextual classification, priority scoring, routing constraints, knowledge-assisted response generation, and human escalation. The system architecture further defines the responsibilities of the frontend, backend, database, and AI services. Retrieval-augmented generation and restricted tool access are proposed to support knowledge-grounded responses, while response validation and escalation provide safeguards for uncertain or unsuitable cases.

The current work primarily establishes the system design and proposed evaluation procedure. Implementation and experimental validation remain necessary to determine the effectiveness of the framework. In particular, the classification models, priority thresholds, routing logic, knowledge retrieval process, and response-validation mechanisms must be tested using suitable datasets and representative test cases. No quantitative performance improvement is claimed at this stage.

Future work will focus on completing the implementation, obtaining and preparing a suitable support-ticket dataset, and evaluating the system using task-specific metrics. Further work will include assessing classification precision, recall, and F1-score; validating priority and SLA decisions; measuring department-routing performance; and testing the relevance and correctness of AI-generated responses. Integration testing will also examine workflow reliability, access control, failure handling, and human-escalation behaviour. Based on the evaluation findings, the framework can be refined through controlled model updates and improvements to its operational rules.

In conclusion, TicketFlow Nexus provides a proposed foundation for integrating AI-based ticket intelligence with rule-governed support operations. Its practical value will depend on successful implementation, reliable integration, and systematic experimental evaluation.

REFERENCES

[1] L. S. Benitez Pereira, R. Pizzio, and S. Bonho, "*Comparison of Information Retrieval Techniques Applied to IT Support Tickets*," arXiv preprint arXiv:2508.05654, 2025.

[2] T. S. Kumar and D. Revathy, "*AI-Based IT Support Ticket Classification and Priority Prediction System*," IRE Journals, vol. 9, no. 9, pp. 2298–2304, Mar. 2026, doi: 10.64388/IREV9I9-1715498.

[3] P. Lewis et al., "*Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*," in Advances in Neural Information Processing Systems, vol. 33, pp. 9459–9474, 2020.

[4] N. Reimers and I. Gurevych, "*Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*," in Proc. 2019 Conf. Empirical Methods in Natural Language Processing and 9th Int. Joint Conf. Natural Language Processing (EMNLP-IJCNLP), pp. 3982–3992, 2019, doi: 10.18653/v1/D19-1410.

[5] C. S. K. Selvi, S. J. Shri, R. Prashanthini, Neelamegan, and R. Sanjay, "*Customer Support Ticket Categorization and Prioritization Using Natural Language Processing*," in Proc. 3rd Int. Conf. Futuristic Technology (INCOFT), vol. 3, pp. 690–698, 2025, doi: 10.5220/0013640400004664.

[6] M. V. N. Durga Sai Suresh and A. Durga Devi, "*An Intelligent AI-Based Customer Support System for Automated Ticket Classification and Response Generation,*" International Journal of Engineering Sciences and Advanced Technology, vol. 26, no. 4, pp. 1406–1415, 2026.

[7] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "*ReAct: Synergizing Reasoning and Acting in Language Models*," in Proc. Int. Conf. Learning Representations (ICLR), 2023.