Advanced Sciences and Technology Journal

ASTJ vol. 3 no. 1 (2026) P1069
10.21608/astj.2025.392567.1069
https://astj.journals.ekb.eg/

Smart Serve: Redefining Customer Support with AI-Driven Ticketing Intelligence

Ahmed Taha[1*], *Doaa Mabrouk*[2], *Ashraqt Tamer*[2], *Mariam Tarek*[2], *Fatma Alaa*[2], *Mahmoud El-Fateh*[2], *Fady Romany*[2]

[1]*Computer Science Department, Faculty of Computers & Artificial Intelligence, Benha University, Benha, Egypt.*
[2]*Software Engineering Department, Faculty of Engineering, Egyptian Chinese University, Gisr el Suez, Cairo, Egypt*
[*] *ahmed.taha@fci.bu.edu.eg*

ARTICLE INFO

Article history:
Received 06 June 2025
Revised 06 July 2025
Accepted 08 August 2025
Available online 15 November 2025

Editor: Prof. Dr. Mohamed Talaat Moustafa

Keywords:
Ticketing Systems
Artificial Intelligence
Natural Language Processing
Large Language Model
Gemini-Based Model
Customer Support Workflow

ABSTRACT

With growing customer expectations and increasing volumes of support requests, traditional helpdesk systems often fall short, resulting in slow resolution times, misrouted tickets, and a less-than-ideal customer experience. This paper presents Smart Serve, a cutting-edge AI-based ticketing solution that revolutionizes the working model of customer support teams. By leveraging cutting-edge Natural Language Processing (NLP) and state-of-the-art Large Language Models (LLMs), such as GPT and Gemini, the system automatically evaluates incoming requests for urgency, complexity, and category, enabling real-time prioritization, precise routing, and efficient handling. What differentiates this solution is its hybrid architecture: mundane issues are handled automatically with AI-powered responses, while tough ones are passed on smoothly to human agents. The combination of automation and human interaction will not only boost efficiency but also deliver a responsive and personalized customer experience. This study brings out the potential of innovative ticketing systems to revolutionize customer care, making it quicker, wiser, and more proactive.

1. Introduction

Existing service providers typically receive a large number of customer requests, and there is an urgent need for effective action to deliver proper support. In such situations, machine learning algorithms play a vital role in optimizing support ticket processing procedures. However, existing techniques continue to employ traditional Natural Language Processing (NLP) techniques, which may fall short of the most recent advancements in the field [1]. The aim of this paper is to provide an overview to support Ticket Automation (TA), explore recent recommendations in this area, and demonstrate how high-level language models can be effectively used for better ticket management.

Efficient handling of support tickets is the backbone of modern customer service, serving as the primary gateway for resolving issues and service requests across various industries. These auto-generated or manually triggered text-based interactions are a critical process for maintaining customer satisfaction and achieving Service-Level Agreements (SLAs). Wasteful processing and routing of these tickets, however, result in the waste of resources and negative financial consequences. Various studies have been conducted on utilizing Artificial Intelligence (AI) to enhance ticket management. For instance, [2] presented an IT service management platform based on automation for managing service requests and resolving queries, indicating a possible increase in productivity at the organizational level. Similarly, [3][4] experimented with an AutoML-enabled model using Google Vertex AI [5][6] for classifying support tickets, demonstrating the promise of auto-classification while highlighting limitations such as the need for high-quality training data and regular model refreshes. Cloud-based ticketing platforms and AutoML [7][8][9][10]solutions have been made available to continue streamlining query resolution and auto-categorize tickets with a constant focus on the benefits of AI for greater operational efficiency and end-user satisfaction, but with ongoing calls for further improvement in scalability and security Building upon this existing research base, which has already demonstrated the capabilities of AI and machine learning across a range of ticketing areas, this paper outlines a new approach to automatic support ticket classification. At the center of our proposed Smart Serve ticketing system is the ability of Large Language Models (LLMs). Through a process of trial and experimentation with various methods of utilizing these more enriched models, we discovered an optimal method that yields a significant improvement in ticket classification accuracy. The empirical results of this paper will demonstrate the effectiveness of this chosen method. Our system, built on the Laravel framework with a MySQL database backend, automates key tasks such as ticket classification, SLA assignment, and response generation, based on conclusions drawn from our experience with large language models (LLMs). The evaluation, which includes baseline comparisons and various embedding methods, demonstrates the higher classification accuracy of our methods and provides interesting insights into document representation in support tickets.

In [11], the development of AI-based support systems, in the form of web-based platforms, was presented to streamline issue tracking and ticket management processes. This paper presents a thorough analysis of existing AI-based systems, highlighting the disruptive contributions that intelligent automation can make to ticketing processes. Through analysis of early adoption practices and future developments, stretching from natural language processing to classification techniques and system scalability. This context can be drawn up for the current discussion, as modern-day organizations seek to implement such types of smart solutions in an effort to automate IT support and provide more responsive customer service.

Smart Serve relies on its ability to preserve data and protect users' privacy. Data protection from malware attacks must be incorporated at every level of access. Administrators need to have advanced malware detection software, along with privileged access controls, to ensure that critical backend operations are safeguarded. Employee access must be secured by endpoint protection technologies that monitor and quarantine malicious behavior, ensuring internal systems are not compromised by standard attack techniques, such as phishing or malicious attachments. Customer interactions must be scrubbed by real-time malware scanning of form data and uploaded files to prevent injection of malicious scripts or executable payloads. Additionally, role-based access control must be implemented in a manner that allows only authorized personnel to administer or access confidential information. Also, data privacy acts must be achieved through visibility, user awareness, and the encryption of private data throughout the lifecycle. Scheduled security audits, as well as penetration testing, are essential in evaluating vulnerabilities and putting the security position in good standing.

The contributions of this work are summarized as follows:

- Identified the limitations of traditional ticketing systems utilized by support teams—manual tagging, language limitations, and delayed responses—and proposed a simple-to-use web-based smart ticketing system for organizations and end-users.
- Developed a smart ticketing process wherein users feed support forms and Large Language Models (LLMs) classify, prioritize, route, and generate auto-responses automatically, enhancing accuracy, supporting multilingual input, and relieving human workload.
- Integrated a middleware layer with support for role-based access and validation control for support agents, admins, and customers to deliver a secure and real-time pipeline for ticket processing.
- Results showed outstanding improvement in response time, ticket resolution efficiency, and customer satisfaction, with the system being scalable, flexible, and user-friendly for diverse organizational use cases

The rest of the paper is structured as follows: Section II discusses related work. Section III presents the proposed method. Section IV describes multilingual datasets “dataset-tickets-multi-lang-4-20k”, and Section V presents the experiments and results. Section VI concludes the paper, highlighting future research directions.

2. Related work

In [14], an effective IT service management platform was proposed to automate query resolution and service request management. The system enhances organizational productivity and user satisfaction by effectively managing user issues and IT assets. The key functionalities include a library of pre-formulated answers for common questions, technical support for addressing outstanding issues, and a request module to manage user requests efficiently. The solution is deployed on a cloud-based virtual machine, such as AWS, to be made accessible within an organization. IT request tracking and asset management are integrated to collect all IT-related information in one repository, enhancing resource management and resolving issues at a faster pace. Although the system offers a good architecture, it lacks scalability for large organizations, integration with other IT management software, and provides no detailed strategies for user training and adoption. However, it doesn't provide comprehensive security features, nor does it address performance measures, indicating a need for further research to address these gaps.

A model based on AutoML using Google Vertex AI was presented to classify support tickets by escalation type and category in [15]. Despite its notable advantages, the system also presents several challenges, including the need for high-quality training data, model updates, and robust data security in the cloud. The small size of the dataset also restricts the research, as it relies on only "Title" and "Description" as features, and lacks benchmarking with common machine learning models. AutoML and machine learning have demonstrated significant promise in ticket classification, further advancements are essential to address key concerns such as data quality, security protocols, and scalability, ensuring these solutions are robust and viable for real-world applications.

As already noted, IT service ticket management often suffers from issues such as excessive query volumes, response latency, and challenges with task prioritization. To mitigate these, cloud-based ticketing.
AutoML-based solutions and systems have been proposed for optimizing query resolution and automating ticket categorization. However, the systems are confronted with challenges such as the need for quality training data, continuous updates, and robust data security. Despite these challenges, the use of AI and automation has been shown to have positive effects on improving operating efficiency, reducing delays, and enhancing user satisfaction, making future research vital for advancing scalability and security.

In [16], an automated support ticket system (STS) utilizing machine learning (ML) was presented. In [20], customer support requests were difficult to handle in companies, for instance, due to inefficiencies and longer resolution times resulting from manual ticket assignment. It identifies Support Vector Machine (SVM) and Random Forest (RF) as the top-performing classifiers for ticket classification, with accuracies ranging from 63% to 98%. High-quality training data plays a crucial role in ensuring the success of ML implementations in STSs, and further research is needed to address existing gaps, particularly in comparative studies, customer sentiment prediction, and automated ticket response. Large-scale studies were proposed, including real-world case studies, to validate theoretical findings and examine the impact of ML automation on customer trust and agent satisfaction.

In [17], an AI-powered ticketing system was presented to replace traditional ticketing procedures by eliminating issues such as manual ticket assignment, error vulnerability, and the lack of sentiment analysis to facilitate priority decisions. The system utilizes machine learning techniques, including nearest neighbors and sentiment analysis, as well as TF-IDF vectorization, to process the descriptions of tickets. Although the solution saves human effort and reduces errors, it still does not address issues like the need for high-quality datasets, integration into existing systems, or fairness in AI decision-making. Some future work that can be done includes addressing these issues through transfer learning, developing more efficient frameworks, and conducting fairness tests.

In [18], a Real-time Ticket Assignment Deep Auto Advisor system “TADAA” was proposed for Customer Support, Help Desk, and Issue Ticketing Systems, which leverages AI to optimize ticket assignment processes in organizations. The researchers identified limitations of traditional ticketing systems, including delays and errors resulting from manual interventions. Their proposed model automates ticket assignments using Transformer-based models, specifically BERT and RoBERTa, to assign tickets to the relevant group and the best resolver. Approximate Nearest Neighbor (ANN) is also used by the system for retrieving similar past tickets and assisting resolvers. It demonstrated impressive performance metrics, with RoBERTa achieving a top-3 accuracy of 95.5% on group classification, while an ensemble model on resolver classification achieved a top-5 accuracy of 79.0%. The dataset contained 144,600 cleaned tickets extracted from an initial pool of 203,300 records. Although the system significantly reduces manual effort and improves efficiency, the authors failed to address potential problems, including ethical concerns, biases in decision-making, scalability to larger datasets, compatibility with existing systems, and the long-term impact on job postings. Additionally, the study lacked comparative research with other ticketing systems and real-life validations to demonstrate practical feasibility. Future research directions should focus on mitigating biases, enhancing scalability, conducting real-world testing, and evaluating the broader implications of AI integration in ticketing workflows to ensure robust and sustainable implementations.

Providing technical support for own IT products is a core activity of software development or software-providing companies [10], [19], evaluated the usability of JIRA for issue tracking and project management in IT companies, highlighting its strengths in ticket tracking, transaction recording, and workflow automation through features like reporter and assignee. However, restricted field editing, scalability, and complexity of workload information were identified as challenges in the research. It also noted lacunas in addressing ethical concerns, integration with other software, and user satisfaction. The authors concluded that JIRA functions, but more tools and studies are needed to address its lacunas and increase scalability, integration, and usability.

In [13], a generic framework for automated ticket processing systems was proposed with emphasis on multi-level classification scenarios. It examined the use of machine learning and natural language processing techniques, such as BERT and its variants, to improve ticket categorization and routing precision. It used hierarchical models like ML-BERT and SupportedBERT. SupportedBERT demonstrated superior performance, achieving an impressive accuracy of 64.9% on a finance dataset and 61.1% on a Linux Bugs dataset. These results highlight the effectiveness of incorporating hierarchical information to enhance the precision and robustness of classification tasks across diverse domains. There are several limitations, including noisy data, scaling models to larger datasets, and exploring real-world applications. It provides a solid foundation for integrating AI-based solutions into ticket management systems and highlights the potential of hierarchical methods to enhance support efficiency. Various aspects of ticketing systems, from implementation to pros and cons, have been discussed, including the use of AutoML for customer support ticket classification, the utilization of AI for ticket automation assignment, and the development of AI-based systems like TaDaa for dynamic ticket assignment.

While there have been notable advancements in AI-based solutions, there are some critical areas that remain. Some of the issues that require more consideration include ethical concerns, scalability, compatibility with existing systems, the long-term impact on job roles, comparative analyses with other solutions, testing in the real world, and the user interface.

2.1. Thesis Statement

The customer service operation's support process is generally hindered by inefficiencies in email processing, particularly due to manual processing requirements for categorization, prioritization, and ticket assignment. These are laborious processes, prone to human error, and might not meet stringent service level agreements (SLAs). Process automation has been a key focus to help reduce response time and improve accuracy. Effective customer communication management relies on principles of operations management, natural language processing (NLP), and artificial intelligence (AI). For instance, AI technologies such as machine learning (ML) classifiers and deep learning models provide the basis for automating the classification and prioritization of emails. Work of [23], also extends these foundations by pointing out the contribution of deep learning models for sentiment and intent analysis that are essential for accurate classification of customer interactions and their estimated urgency, also contribute to this framework by showing how machine learning and AI models, particularly in ticketing systems, are playing an ever more essential role in reducing manual effort and automating ticket assignment and response times in dynamic environments.

In [24], highlights the use of deep learning models for classifying customer service tickets by intent and sentiment. They emphasize the usefulness of combining sentiment analysis with machine learning to gain a deeper understanding of customer inquiries, which enables more accurate ticket classification and priority assignment. This system has the potential to be particularly helpful in dynamic, high-volume environments where quick and accurate decision-making is critical.

In [25], explore the application of machine learning models in ticketing systems to automate the classification of customer inquiries and assign appropriate priorities. They discuss how the combination of AI-based classification and machine learning can be utilized to enhance the productivity of customer support teams, particularly when dealing with large volumes of tickets. Their article emphasizes the necessity for a dynamic ticket management system that adapts to evolving customer needs while improving SLA compliance and reducing response times. A work by [25] provides a detailed analysis of an AI-based ticketing system for IT service management. Their system utilizes NLP mechanisms to categorize incidents and requests with great accuracy and rank them in order of urgency. By utilizing AI for task assignments, the system minimizes human errors and optimizes the efficiency of task allocation.

A work by [26] discusses the application of NLP in customer email management. They emphasize how supervised learning models effectively categorize incoming emails into pre-determined categories such as "request" or "problem." The study also highlights the need to incorporate domain-specific language datasets for enhanced accuracy.

In [27], the authors address the usage of automation tools for customer service to reduce repetitive work. They develop an architecture that integrates task automation and email servers in a manner that makes the operation transparent, even for customers using traditional email communication. Their work fits within the motivation of enabling hybrid interaction models (email and form submission).

In [28], the issue of workload balancing for customer support teams is addressed. Their system dynamically balances tickets in real-time, analyzing the availability of team members and their historical performance data. This approach improves SLA adherence and evenly balances team workloads.

In [29], the authors present an end-to-end system for automating customer support tasks, including email classification and prioritization. Their paper focuses on the integration of AI in automating response and workload distribution, demonstrating how machine learning models can reduce SLA violations and improve service quality. The authors also mention the challenge of integrating email and chatbot systems for hybrid solutions.

2.2 Empirical Research Summary

Accuracy Improvements: AI-based classification systems have demonstrated a classification accuracy exceeding 85%, as shown in [26], validating their utility in real-world applications.

Efficiency: The system proposed by [27], reduced manual workload by 70%, ensuring quicker response times in customer interactions.

SLA Compliance Rates: In [29]**,** report significant improvements in SLA adherence when combining email automation and chatbot systems, highlighting the potential for hybrid solutions.

Workload Balancing: AI-based workload distribution approaches described by [28], effectively reduce employee burnout while maintaining consistent service delivery standards.

Sentiment Analysis and Intent Classification: [23]**,** demonstrates the impact of sentiment and intent analysis in ticket classification, improving both ticket categorization and response prioritization in customer service environments.

Strategy: Using the correct strategy makes the machine learning system (model) run more quickly and efficiently. Additionally, the model enhances the accuracy of the results.

Machine Learning System Integration: [24], explore the potential of machine learning for ticketing systems, but do not explore how such systems can be integrated with existing enterprise-level customer service software or multi-channel solutions.

Leveraging Hybrid Machine Learning: Hybrid machine learning (HML) is a powerful technique that combines multiple machine learning methods to overcome limitations and improve predictive capabilities. This approach has been widely adopted in various machine learning (ML) applications, including anomaly detection and regression. The process unfolds in two phases: first, a PCA-based anomaly detection algorithm is applied to the initial dataset to uncover patterns and anomalies; second, the resulting insights are merged with a second dataset to create a more comprehensive data foundation. A regression algorithm is then trained on this enriched dataset to predict ticket counts with greater accuracy and robustness. Our approach aims to surpass the predictive capabilities of standalone techniques, offering a more reliable and precise solution for ticket count forecasting.

The first algorithm is trained using the first dataset, and its output is then used to train a second algorithm with a second dataset. It is not possible to use anomaly detection by itself to predict ticket volumes, as the mere sum of anomalies discovered has no relation to tickets given. With the raw statistical measures of the log data, such as log rows per day, one could achieve some results with a single algorithm if the tickets obtained are highly correlated with the volume of log events. What we can do is supplement our ticket estimating algorithm with anomaly feature values. Because we first tally up the anomaly numbers with the anomaly detection algorithm and then use the statistical characteristics of the outcomes in another algorithm, such as the regression algorithm, we possess additional relative information when we make the final ticket number predictions.

2.3 Critical Analysis

Existing research has made significant strides in email management and automation of ticketing systems, yet critical limitations remain unaddressed. For instance, [25], [26] propose a pragmatic AI model or highlight the importance of domain-specific data; they often fail to consider broader applicability, data maintenance in dynamic environments, or scalability challenges. In [27], presents a strong case for automating email servers but does not consider potential latency issues in high-traffic environments. Furthermore, while workload distribution models, as described by [28], enhance SLA fulfillment, the utilization of historical data could lead to task assignment biases. In [29], an advanced Framework for automation is provided, but it does not discuss the scalability problems of using more than one AI-based tool integrated into one platform. Other works emphasize automation and workload distribution but overlook potential issues such as latency in high-traffic scenarios, biases from historical data, or integration with existing platforms. Additionally, as noted in [23], advancements in deep learning for intent analysis and machine learning for large-scale ticketing systems often lack adaptability for cross-industry variability and multi-channel support. In [24], the authors demonstrate how machine learning can be employed to optimize ticketing systems for large-volume customer interactions; however, they fail to describe how this system can be incorporated into existing customer service platforms or tailored for use in multi-channel settings.

3. Proposed Method

This section proposes a novel solution for automatically classifying support tickets in shallow hierarchical taxonomies, as part of an AI-driven ticketing system that aims to optimize customer and internal support processes. The method leverages state-of-the-art Transformer-based language models, renowned for their ability to generate semantically rich text embeddings, to produce highly meaningful document representations. More precisely, a multi-level classification framework is proposed that exploits the hierarchical nature of ticket categories to improve accuracy. Additionally, several alternative methods for generating consolidated document embeddings, drawing on recent developments in literature, were explored. These methods, while BERT-centric, are extensible to any language model capable of producing contextual embeddings. The system was developed using the Laravel PHP framework, with MySQL as the backend database, and incorporates Google's Gemini LLM to automate key tasks, including ticket classification, SLA assignment, and response generation. The empirical evaluation, encompassing baseline comparisons and various embedding strategies, demonstrates that the proposed methods achieve significantly superior classification performance while providing valuable insights into document representation in this domain.

3.1 System Design Overview

The system employs a standard three-tier architecture on the Laravel MVC framework:

- **Application Layer:** User interface web application built using Laravel Blade and Bootstrap, with dynamic HTML content rendering and seamless integration with backend logic.
- **Data Layer:** Persistent data, including user data, tickets, ticket history, and AI metadata,

is stored in the MYSQL database and controlled by Laravel's Eloquent ORM.

Fig. 1 shows the end-to-end workflow of the Smart Serve System. The process begins at the User Interface, where users submit inputs such as queries or requests. These are transmitted to the Laravel Backend, which acts as the middleware to process and route data. The information is then stored in a MYSQL Database for structured persistence.

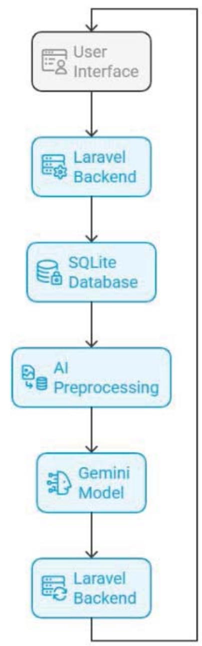


Fig.1 Smart Serve System Flowchart: Data Flow from User Input to AI Response

3.2 Backend Approach

The system's engine, where information is processed, forms are submitted, validated, and saved. The most important elements are

- **Routing System:** Laravel's routing system transforms routes into URLs by translating them into controller methods, determining which action should be performed based on the input received from the user.
- **Controller Layer:** Handles user requests, showing the form to enter a ticket, handling data input, input validation, and saving ticket details in a database.
- **Model Layer:** Specifies data structure and interfacing with MYSQL database through Laravel's Eloquent ORM, which delivers appropriate data fetching, insertion, and updating.
- Event System: The Laravel event system initiates second-level tasks upon ticket submission, such as notifying administrators or logging system processes.
- **Database Integration (MySQL):** Utilizing its minimal footprint and simplicity of implementation, MySQL is ideal for small to medium-sized applications. Laravel migrations make managing database tables easy.

3.3 Frontend Strategy

Frontend is established through Laravel's Blade templating engine with a user experience focus and dynamic content rendering:

- **User Interface Design:** With a plain, form-based interface, it's easy to open support tickets. Responsive and lightweight, the interface makes it easy to access.
- **Validation Feedback:** Users get instant feedback upon form submission. Success or failure messages prompt users to resubmit input where necessary.
- **Template and Layout Structure:** Consistency between pages is maintained through Blade templates. Redundancy is removed through layouts, partials, and components, resulting in minimal CSS for a clean and responsive layout.

3.4 Ticket Lifecycle and AI Automation

Automation of the ticket life cycle using Gemini LLM includes classification, SLA setting, priority, and response generation:

1. **Ticket Submission:** The user completes a form with a subject, description, and optional attachments.
2. **AI Processing:** The ticket is passed to Gemini LLM via a secure API call, which returns:
    - Relevant Department (e.g., IT, HR, Billing)
    - Priority Level (Low, Medium, High, Critical)
    - Ticket Type (Question, Complaint, Request, Bug Report)
    - Suggested SLA (e.g., 2 hours, 4 hours, 24 hours)

      AI-Generated Answer
3. **SLA Assignment:** SLA is recorded in the database, and timers and reminders are set to assist with compliance.
4. **Auto-Response:** Auto-response is executed by the system for straightforward questions, and the ticket status is changed. Escalated tickets are assigned to agents with AI-recommended suggestions.
5. **Agent Review & Follow-Up:** Agents override, review, or edit AI suggestions before manual response.
6. **Ticket Closure**: Archived closed tickets together with the SLA compliance status.

3.5 Integration of AI Using Gemini LLM

The AI module enhances ticket handling by:

- **Ticket Insight:** Retrieves structured information (department, priority, type, SLA) to guide backend processes.
- **SLA Forecasting:** Forecasts accurate SLAs depending on content, type, and urgency (e.g., 2-hour SLA for priority bugs, a 24-hour SLA for routine requests.
- **Auto-Responses and Suggestions:** They provide complete answers to simple questions or provide answer outlines for approval by an agent in complex cases.

3.6 Security and Privacy Measures

- The system imposes strict security and privacy procedures:

  Laravel Security Features: Role-based access control, hashed authentication, and CSRF protection.
  Secure Communications: Token-based authentication and HTTPS secure communication with the Gemini

LLM API.

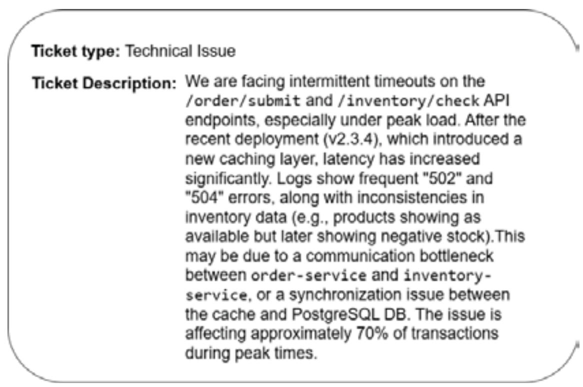
Data Anonymization: Sensitive user data is anonymized before processing by AI, ensuring compliance with GDPR or HIPAA regulations**.**

3.7 Justification of Technology Stack

Laravel: Offers secure, maintainable, and scalable backend logic through inherent functionality.

- **My SQL:** Light and stable, appropriate for single-instance or mid-scale deployment.
- **Gemini LLM:** Supports AI-driven automation of classification, SLA prediction, and response generation.

Fig.2. shows an example of the well-formatted input ticket for Natural Language Processing (NLP) and Large Language Models (LLMs) to be handled efficiently. Automated analysis is applied to tickets within the described system to pick useful features such as issue type, affected components, error codes, and indicators for urgency. Past research has indicated the use of NLP methodologies such as Named Entity Recognition (NER), TF-IDF, and BERT-based classifiers for routing and support ticket classification [13],[26].LLMs further enhance context sensing and facilitate few-shot or zero-shot classification capacity [27], allowing the system to accurately determine ticket type (e.g., "Technical Issue") and severity (e.g., "High"). It reduces the amount of human triage, expedites response time, and optimizes resource allocation in massive support systems [28].



Ticket type: Technical Issue

Ticket Description: We are facing intermittent timeouts on the `/order/submit` and `/inventory/check` API endpoints, especially under peak load. After the recent deployment (v2.3.4), which introduced a new caching layer, latency has increased significantly. Logs show frequent "502" and "504" errors, along with inconsistencies in inventory data (e.g., products showing as available but later showing negative stock).This may be due to a communication bottleneck between `order-service` and `inventory-service`, or a synchronization issue between the cache and PostgreSQL DB. The issue is affecting approximately 70% of transactions during peak times.

Fig.2. Ticket Details Overview: Description and type classification.

3.8. Other Methods Considered

Traditional and recent models have been employed for support ticket classification, including transformer-based architectures such as BERT, DistilBERT, and XLM-RoBERTa[29], facebook/bart-base, as well as conventional machine learning techniques such as TF-IDF coupled with classifiers like XGBoost, Support Vector Machines (SVM), and Naive Bayes. But all these techniques did much worse compared to our LLM-based Gemini approach [30] , Most traditional models struggled to capture the fine-grained semantics and context-based intents encoded in detailed ticket descriptions, leading to diminishing classification and priority prediction accuracy. For instance, while TF-IDF-based models were saddled with sparse representations, standard classifiers such as Naive Bayes and

Random Forest exhibited poor generalization across categories. Additionally, certain previously proposed methods in the literature could not be entirely assessed because either the code was inaccessible or the reported results were irreproducible. In contrast, Gemini's advanced contextual comprehension, capability for few-shot learning, and flexibility of integration enabled more robust ticket categorization and SLA management, resulting in a more practical solution that can be deployed in real-world scenarios within dynamic support settings.SO Transformers was very useful and a huge architecture for complex methods, as shown in Fig.3.

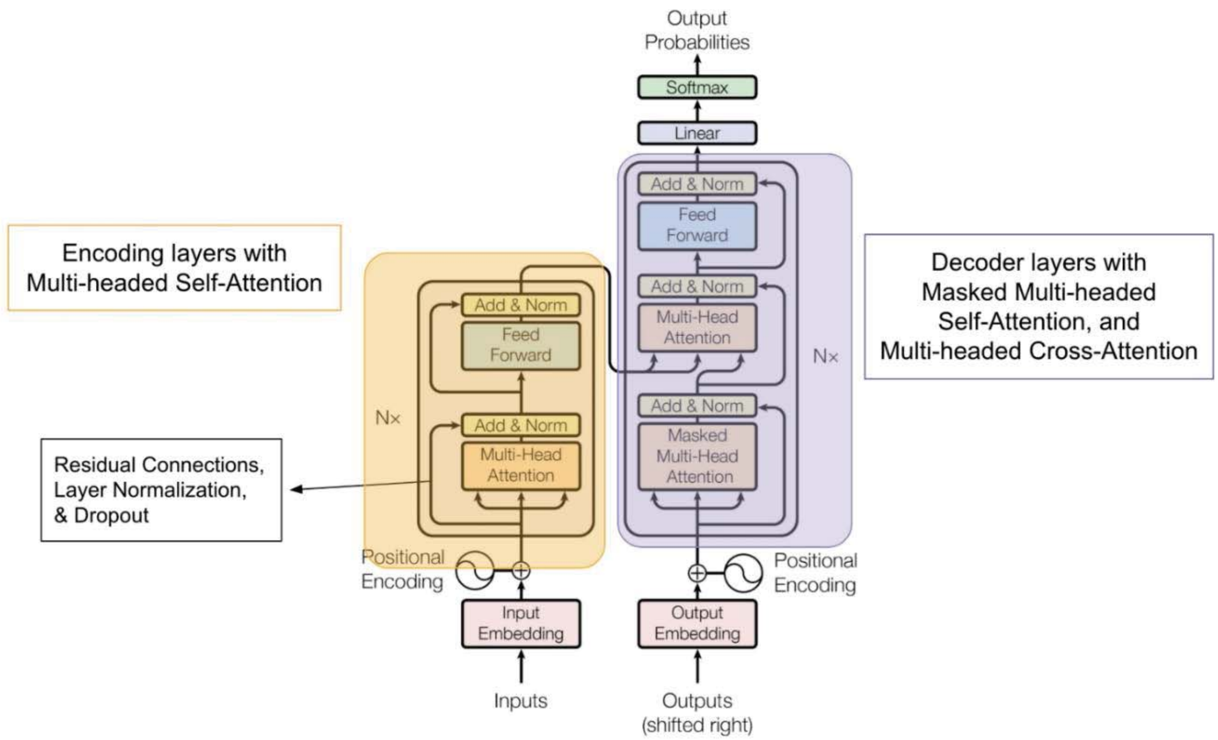


Fig.3. Overview of the architecture of transformers.

4. Datasets

There are two comprehensive datasets to advance intelligent helpdesk systems. The first one is A massive multilingual dataset called "dataset-tickets-multi-lang-4-20k". It consists of 20,000 real customer support tickets in English and German [31]. It is specifically designed to simulate enterprise-grade helpdesk interactions and is rich in both structured and unstructured fields. It has fields required such as the subject and body (customer message), response (provided by support), priority levels (Low, Medium, Critical), ticket type (Incident, Request, Problem, Change), department queues (such as Technical Support, Billing), business type, and multi-label tags specifying the problem. The ordered fields, such as assignment groups, short descriptions, and priority levels, are of particular interest to model performance and are typically not dropped after feature selection to optimize accuracy. It also supports response generation research and facilitates the construction of a Tag Co-Occurrence Network, as shown in Fig. 4, revealing semantic relationships among terms such as "Bug," "Feedback," and "Technical." This aids in uncovering latent topic hierarchies and validating classification results against real-world tag distributions.

The second dataset, from Suraj520, comprises over 8,000 customer support tickets for various tech products [32]. Its scheme includes client identity, product details, ticket type, source, status, and more, offering a rich foundation for multidimensional analysis and machine learning experimentation. These datasets enable the exploration of ticket triage, priority estimation, type classification, and contextual routing—key components of next-generation customer support automation.

To enhance predictive modeling, the combined fields have been used to create composite labels. For "dataset-tickets-multi-lang-4-20k," we merge ticket type and queue (e.g., "Incident Technical Support"), while Suraj520's dataset, ticket type, and product (e.g., "Hardware Issue Laptop") have been combined. This approach enables the model to simultaneously identify both the issue type and its context, thereby streamlining classification and enhancing the efficiency of automated support systems. These datasets collectively serve as valuable resources for developing and testing intelligent, scalable, and accurate customer support solutions.

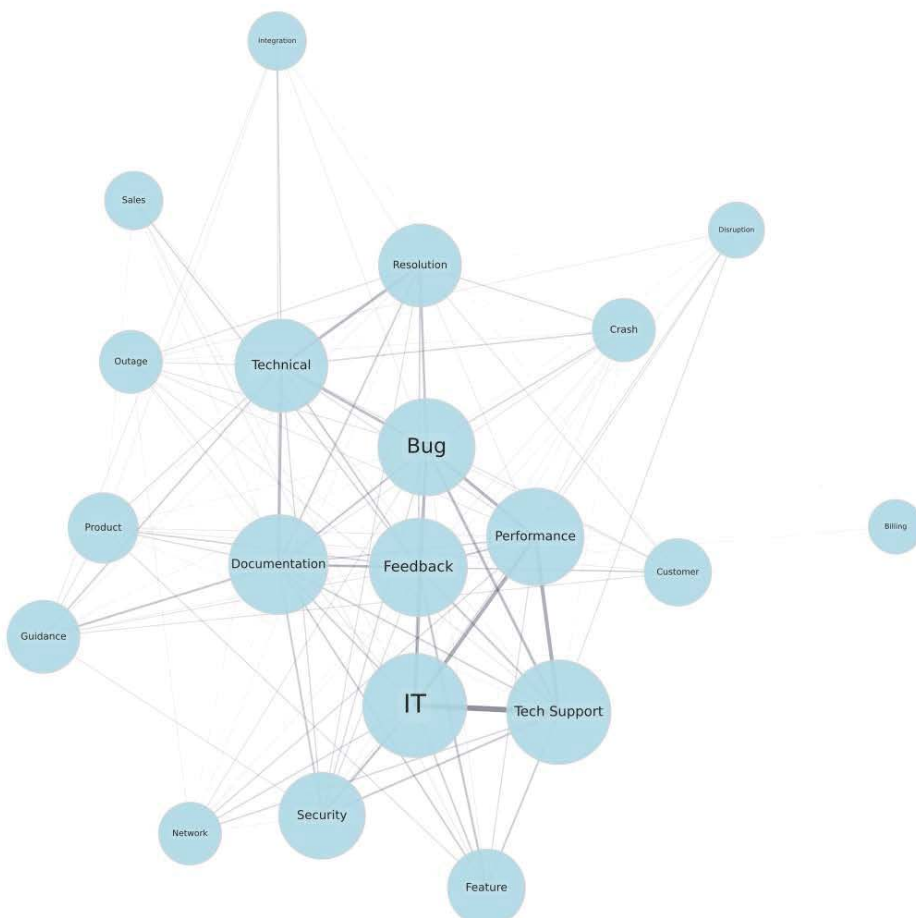


Fig.4. Tag Co-Occurrence Network (Relative Weights, Normalized by Number of Tickets)

5. Results and Discussion

To compare model performance, 5-fold cross-validation with two iterations was used. The dataset is divided into 20% for testing and 80% for training. The 5-fold setup is balanced between CPU time and good performance estimation, offering more consistent results. This approach provides more reliable estimates compared to 3-fold cross-validation, without significantly increasing runtime. Stratified sampling was employed to maintain the original class balance in every fold. Additionally, 20% of every training split was utilized as a hyperparameter tuning validation set. The results are presented as the average metric over all 10 test runs (2 repetitions × 5 folds), estimating both performance and variability.

5.1.Binary Classification

In the initial phase, a binary classification task was performed to distinguish between two primary ticket categories, as a precursor to multi-class classification. The process involved machine learning pipelines. Starting with Term Frequency-Inverse Document Frequency (TF-IDF) to convert unstructured text into weighted numeric feature vectors. To address the imbalance, Synthetic Minority Over-sampling Technique (SMOTE) was used to generate Synthetic minority classes by interpolating between existing minority samples. This gives a more balanced ratio, thereby indirectly raising the learning power of classifiers.

Three algorithms, Support Vector Machine (SVM), XGBoost, and Multinomial Naive Bayes (MNB), were evaluated in parallel on the same preprocessed, SMOTE-balanced datasets under the same parameters to have an equal footing. These models were chosen for their effectiveness in handling sparse, high-dimensional text data and their ability to generalize with limited datasets. The results of these algorithms used in this classification task are summarized in Table 1.

TABLE 1.

Binary Classification Machine Learning Models Performance

Model	Acc	F_1	Prec	Rec
SVM	92%	92%	93%	92%
XGBOOST	90%	90%	91%	90%
Naive Bayes	89%	89%	90%	89%

5.2 Multi-Class Classification:

In the second experimental phase [33], multi-class classification of support tickets was performed across four pre-defined classes: Incident, Request, Change, and Problem. A pipeline combining TF-IDF vectorization, SMOTE [34] for class balancing, and GridSearch for hyperparameter tuning was implemented. The experiments were conducted using Stratified 5-Fold Cross-Validation to prevent overfitting and ensure balanced evaluation across all classes.

Three classifiers, Support Vector Machines (SVM) [35], XGBoost [36], and Naive Bayes [37], were evaluated under identical conditions. SVM achieved the best performance, with strong F1-scores and balanced precision-recall trade-offs, followed by XGBoost for stability and accuracy. Naive Bayes was faster but less accurate overall. As shown in **Table 2.**

TABLE 2

Multiclassification Machine Learning Models Performance.

Model	Acc	F_1	Prec	Rec
SVM	85%	85%	85%	85%
XGBOOST	88%	88%	88%	88%
Naive Bayes	82%	82%	82%	85%

To evaluate the performance of the preceding ticket classification scheme, the XGBoost classifier algorithm, a gradient boosting algorithm, was employed due to its simplicity and effectiveness for structured data. The model is designed to predict ticket priority (low, medium, high) based on a set of textual features derived from the ticket subject and body via TF-IDF vectorization, as well as categorical metadata such as department, type, and language. To eliminate the class imbalance problem in the training set, SMOTE Tomek was applied, which applies both oversampling and undersampling methods to balance the classes. The model performed at a mean accuracy of 96.81% on the test set. All priority classes consistently recorded high precision, recall, and F1-score values. Specifically, the macro and weighted average F1-scores were 0.97, indicating the model's performance at every instance in multi-class classification. The outcome confirms that XGBoost, under adequate text preprocessing and class balancing, can make highly predictive outcomes for a real customer support issue in the case of ticket priority classification, **as shown in Table 3**

TABLE 3

Performance of ticket priority.

Priority Class	Precision	Recall	F_1-Score	Support
Low (0)	0.98	0.93	0.96	553
Medium (1)	0.98	0.97	0.97	851
High (2)	0.95	0.99	**0.97**	981
Accuracy			**0.97**	2385
Macro Avg	0.97	0.96	0.97	2385
Weighted Avg	**0.97**	**0.97**	**0.97**	2385

5.3 Multi-Class Classification Using LLMS:

For the second experimental configuration, multi-class ticket classification was performed, assigning ticket to one of four categories: Incident, Request, Change, or Problem. Four transformer models have been fine-tuned beforehand, including DistilBERT [38] and facebook/bart-base [39], on a labeled ticket corpus. Minimal text preprocessing was applied to ensure the maintenance of contextual integrity. All models were trained under identical configurations to facilitate an unbiased comparison. The pre-trained models were downloaded from Hugging Face's model hub, which provides standardized model card metadata detailing each model's architecture, training data, purpose, and limitations. The performance of type and priority classifications is shown in Table 4. The Facebook/bart-base model achieved good performance accuracy, with great generalizability to both abstract and classification tasks. DistilBERT, meanwhile, offered a tantalizing speed–performance trade-off and was highly suitable for applications where inference time is a significant bottleneck. This shows the performance of each one, which helps to reinforce the usefulness of transformer-based models for real-world ticket triaging tasks. The table presents the performance of various models (Bart-base, Distilbert, and Deberta-v3-base) on the queue classification task, specifically determining to which department a customer service ticket should be routed. Percentages indicate accuracy of each model in routing tickets to different departments. Our findings show that deberta-v3-base had highest accuracy in overall terms, indicating its greater ability to classify tickets for departmental allotment**.** While classic transformer models such as DistilBERT, facebook/bart-base, and microsoft/deberta-v3-base had performed well, we discovered certain limitations that made them unsuitable for use in more dynamic real-world ticketing systems. For one, these models possess limited understanding of long-range dependencies and context beyond their fixed input length, which is relevant in support tickets with multi-sentence or conversational content. Second, they cannot perform multi-task reasoning or interleave and respond, and thus are unsuitable for adaptive comprehension or task-switching tasks. Although these models also require task-specific fine-tuning and have no cross-goal generalizability, in a few instances, they involve additional engineering overhead. To overcome these challenges, we transitioned to a more advanced large language model (LLM), a Gemini-based architecture that offers deeper contextual comprehension, few-shot and zero-shot learning, and greater flexibility in accommodating diverse ticket classification and triaging scenarios. The transition enabled us to handle more complex and nuanced inputs with less retraining, increased system flexibility, and significantly improved performance. To address these problems, we incorporated Gemini, a Google LLM known for its profound contextual semantics and task reasoning features, which was an extremely complex and advanced large language model that integrates a deep transformer-based structure with sophisticated reasoning and comprehension layers, as shown in Table **4.**

Unlike older models, Gemini excels at reading full conversational cues, intent, tone, and subtle variation, making it extremely flexible for ticket analysis. Gemini demonstrated better generalization and decision capability in our multi-class classification task, outperforming earlier work by providing smoother distinctions among classes [40]. The ability to accommodate diverse sentence lengths and abstract ticket descriptions significantly enhanced the credibility of the classification for all four types of tickets.

Moreover, Gemini was employed to anticipate ticket priority levels—High, Medium, or Low—based on linguistic cues of urgency, business impact, or user frustration. Auto-prioritization is another feature that facilitates the easy creation of SLA workflows and enables rapid response to high-priority tickets.Fig.5 shows the architecture of the Gemini-based model. In addition to that, the model also delivered dynamic department routing, which automatically routed every ticket to the most suitable support team, such as Technical Support, Customer Service, Billing, or IT Operations. Smart routes reduced manual overhead and speed up resolution times. Overall, having Gemini deployed throughout our pipeline resulted in a context-aware and highly scalable ticketing system that not only accurately classified and prioritized tickets but also drove internal processes more efficiently through smart automation.

TABLE 4

Performance of type, priority& queue classifications model transformers.

Model	Acc	F_1	Prec	Rec
		Type classification		
Bart-base	81%	79%	82%	79%
Distilbert	81.9%	81.3%	82%	81.2%
deberta-v3-base	81%	82%	82%	82%
		Priority classification		
Bart-base	51.2%	41.4%	53.2%	43.8%
Distilbert	52.2%	41.5%	50.9%	46.2%
deberta-v3-base	50.7%	43.1%	49.5%	44.2%
		Queue classification		
Bart-base	45.3%	33.6%	43.2%	31.7%
Distilbert	44.4%	32.8%	49.3%	31.2%
deberta-v3-base	72.3%	47.5%	78.9%	43.7%

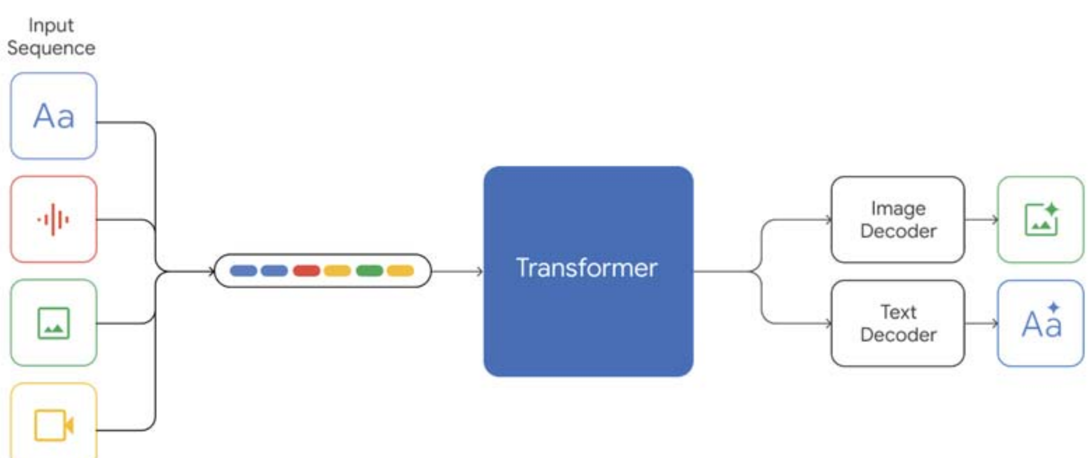


Fig5. An overview of the architecture of the Gemini-based model [40].

7. Conclusions

Traditional helpdesk systems struggle to meet the growing demands of customers and the increasing volume of support requests, often resulting in slow responses, misallocated tickets, and subpar customer experiences. To address these challenges, this paper introduces "Smart Serve," an innovative AI-based ticketing system designed to revolutionize support team operations. By leveraging the strengths of Natural Language Processing (NLP) and Large Language Models (LLMs), such as GPT and Gemini, Smart Serve streamlines core operations, including ticket creation, classification, and prioritization, through a simple-to-use web interface. The core strength of Smart Serve lies in its hybrid architecture, which generates intelligent differentiation between routine issues handled automatically by AI-based responses and complex cases escalated intuitively to human operators. This strategic integration of automation and human insight promises not only to improve efficiency and rationalize support processes dramatically but also to deliver an enhanced, responsive, and personalized customer experience. Ultimately, this study highlights the revolutionary potential of intelligent ticketing systems, such as Smart Serve, to transform customer care into a faster, more innovative, and more responsive service that meets emerging needs.

References

[1] S. B. M Truss, "AI-based Classification of Customer Support Tickets: State of the Art and Implementation with AutoML," in *Fifth and Sixth International Workshop on Entrepreneurship, Electronic and Mobile Business*, 2022. doi: 10.48550/arXiv.2406.01789.

[2] G. S. A Mandal, N Malhotra, S Agarwal, A Ray, "Automated Dispatch of Helpdesk Email Tickets: Pushing the Limits with AI," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. doi: 10.1609/aaai.v33i01.33019381.

[3] and S. M. Jeff Berg, Eric Buesing, Paul Hurst, Vivian Lai, "The state of customer care in 2022," 2022. https://www.mckinsey.com/capabilities/operations/our-insights/the-state-of-customer-care-in-2022#/

[4] J. S. M Jyotheeswar, M Kapse, "Issue Categorization Machine Learning Model using Azure ML," in *Alternatives of Compact Sedan Cars in India*, 2020, pp. 42–47. [Online]. Available: https://auth.services.adobe.com/en_US/deeplink.html#/jump/complete

[5] Cloud Google, "Innovate faster with enterprise-ready AI, enhanced by Gemini models." https://cloud.google.com/vertex-ai/?utm_source=google&utm_medium=cpc&utm_campaign=emea-eg-all-en-dr-bkws-all-all-trial-b-gcp-1605140&utm_content=text-ad-none-any-DEV_c-CRE_691961150784-ADGP_Hybrid+%7C+BKWS+-+MIX+%7C+Txt+-+AI+And+Machine+Learning+-+Vertex+

[6] W. H. Fuchs S., Drieschner C., "Improving Support Ticket Systems Using Machine Learning: A Literature Review," in *55th Hawaii International Conference on System Sciences*, 2022. doi: 10.24251/HICSS.2022.238.

[7] "Service Excellence Research Group." https://www.techra.com/category/s7-about-us/

[8] J. S. Y Wahba, NH Madhavji, "Evaluating the effectiveness of static word embeddings on the classification of it support tickets," in *30th Annual International Conference on Computer Science and Software Engineering*, 2020, pp. 198–206.

[9] J. V. Prabhant Singh1, "Automated Imbalanced Learning," *Springer Nat.*, 2021, doi: 10.48550/arXiv.2211.00376.

[10] Sever IscenM. Zahid Gurbuz, "A Comparison of Text Classifiers on IT Incidents Using WEKA," in *4th International Conference on Computer Science and Engineering (UBMK)*, 2019. doi: 10.1109/UBMK.2019.8907063.

[11] A. Taha *et al.*, "AI Web-Based Smart Ticketing System," vol. 2, no. 1, pp. 1–22, 2025.

[12] Florika GohilMr. Vikash Kumar, "Ticketing System," *Int. J. Trend Sci. Res. Dev.*, 2019, doi: 10.31142/ijtsrd23603.

[13] A..Zangari, M. Marcuzzo, M. Schiavinato, A. Gasparetto, and A. Albarelli, "Ticket automation: An insight into current research with applications to multi-level classification scenarios," Expert Syst. Appl., vol. 225, pp. 1–22, 2023, doi: 10.1016/j.eswa.2023.119984.

[14] H. Fuchs, S., Drieschner, C., & Wittges, "Improving Support Ticket Systems Using Machine Learning: A Literature Review," in *Proceedings of the 55th Hawaii International Conference on System Sciences*, 2022. doi: (Fuchs, S., Drieschner, C., & Wittges, H. (2022). Improving Support Ticket Systems Using Machine Learning: A Literature Review. Proceedings of the 55th Hawaii International Conference on System Sciences. DOI: Link provided in the paper.).

[15] Naveen Koka, "Enhancing a Ticketing System with AI Intelligence," *Int. J. Eng. Res. Technol.*, vol. Vol. 13, 2024.

[16] B. L. Leon Feng, Jnana Senapati, "TaDaa: real time Ticket Assignment Deep Learning Auto Advisor for customer support, help desk,

and issue ticketing systems," 2022, doi: 10.48550/arXiv.2207.11187.

[17] Monika GuptaMonika GuptaAllahbaksh AsadullahAllahbaksh AsadullahSrinivas PadmanabhuniSrinivas PadmanabhuniAlexander Serebrenik, "Reducing user input requests to improve IT support ticket resolution process," *Empir. Softw. Eng.*, 2018, doi: 10.1007/s10664-017-9532-2

[18] G. P. A, M.-C. T. B, M. J. C, and S. Kitsiou, "Synthesizing information systems knowledge: A typology of literature reviews," *Inf. Manag.*, vol. Volume 52, no. Issue 2, pp. 183–199, 2015, doi: 10.1016/j.im.2014.08.008.

[19] K. T. Jacob Devlin, Ming-Wei Chang, Kenton Lee, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1*, 2019. doi: 10.18653/v1/N19-1423.

[20] A. R. Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, "Transformers: State-of-the-Art Natural Language Processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020. doi: 10.18653/v1/2020.emnlp-demos.6.

[21] J. et al. Romero, "Deep Learning-Based Sentiment and Intent Analysis for Improved Ticket Classification," 2022.

[22] A. et al. Niemi, "Machine Learning for Ticketing System Automation," 2023.

[23] S. et al. Suresh, "AI-Based Ticketing System for IT Service Management," 2022.

[24] H. et al. Lee, "NLP Applications in Customer Email Management," 2020.

[25] W. Li, J. & Zhang, "Automation in Customer Support," 2021.

[26] "Automatic Ticket Classification Case Study (NLP)," Kaggle. https://www.kaggle.com/code/abhishek14398/automatic-ticket-classification-case-study-nlp.

[27] T. B. Brown et al., "Language models are few-shot learners," Adv. Neural Inf. Process. Syst., vol. 2020-December 2020.

[28] N. Arici, L. Putelli, A. E. Gerevini, L. Sigalini, and I. Serina, "LLM-based Approaches for Automatic Ticket Assignment: A Real-world Italian Application," CEUR Workshop Proc., vol. 3551, 2023.

[29] Hadi, Giachanou, A., and Bagheri, A., "Automatic Ticket Classification Case Study (NLP)," Kaggle, https://www.kaggle.com/code/abhishek14398/automatic-ticket-classification-case-study-nlp.

[30] Arici, N., Putelli, L., Gerevini, A. E., Sigalini, L., and Serina, I., "Automatic Ticket Classification Case Study (NLP)," Kaggle, https://www.kaggle.com/code/abhishek14398/automatic-ticket-classification-case-study-nlp.

[31] Attamer, A., "Ticket Classification," Kaggle, https://www.kaggle.com/datasets/ashrakattamer/ticket-classification.

[32] S. Suraj, Customer Support Ticket Dataset, Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/suraj520/customer-support-ticket-dataset.

[33] ACL 2022 - 60th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), vol. 1. 2022.

[34] N. V Chawla, K. W. Bowyer, and L. O. Hall, "SMOTE: Synthetic Minority Over-sampling Technique," vol. 16, pp. 321–357, 2002.

[35] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "Training algorithm for optimal margin classifiers," Proc. Fifth Annu. ACM Work. Comput. Learn. Theory, pp. 144–152, 1992, doi: 10.1145/130385.130401.

[36] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min., vol. 13-17-August-2016, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[37] A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," AAAI/ICML-98 Work. Learn. Text Categ., pp. 41–48, 1998, doi: 10.1.1.46.1529.

[38] Hadi, A. Giachanou, and A. Bagheri, "Towards Robust Online Sexism Detection: A Multi-Model Approach with BERT, XLM-RoBERTa, and DistilBERT for EXIST 2023Tasks," CEUR Workshop Proc., vol. 3497, pp. 1000–1010, 2023.

[39] M. Lewis *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *Proc. Annu. Meet. Assoc. Comput. Linguist.*, pp. 7871–7880, 2020, doi: 10.18653/v1/2020.acl-main.703.

[40] Gemini Team et al., "Gemini: A Family of Highly Capable Multimodal Models," pp. 1–90, 2023, [Online]. Available: http://arxiv.org/abs/2312.11805