Vol.9 Issue 4 Oct 2026

Multimedia Research

Received 5 September, Revised 11 September, Accepted 10 October



A Rule-Based NLP Pipeline for Classifying, Prioritizing, and Routing Railway Passenger Grievances with Geotagged Resolution Verification

Manjula K

Assistant Professor, Department of Computer Applications,
GM University, Davangere, Karnataka, India
manjulak.comp@gmail.com

Nisha G

Master of Computer Applications,
GM University, Davangere, Karnataka, India
nisha0708geetha@gmail.com

Abstract: Rail Madad grievances from Indian Railways passengers are still read and forwarded by hand, and the workflow does not separate safety hazards from routine service issues. This study examines whether a lightweight, interpretable pipeline can take over this triage step while making complaint closure accountable. The proposed system accepts typed or spoken complaints in English, Kannada, and Hindi and translates non-English input into English. It assigns one of six department categories through lexicon matching, sets a High, Medium, or Low priority by combining a sentiment score with an urgency-keyword override, screens attached photos using YOLO content detection, Error Level Analysis (ELA), and duplicate hashing, and routes the complaint automatically. A complaint can be closed only after staff uploads a geotagged photo of the completed repair. Implemented with FastAPI and PostgreSQL, the system was evaluated on 600 manually labeled complaints collected between July 2026 and September 2026, using a stratified 80:20 split with a held-out test set of 120 complaints. On the test set, the rule-based classifier reached 78.0% accuracy and a macro F1-score of 0.76, outperforming SVM (75.0%), matching Random Forest (78.0%), and trailing fine-tuned BERT (89.0%) trained on the same split. The priority and image-verification modules were implemented and demonstrated end-to-end on the deployed system, and their quantitative evaluation on labeled test sets is planned for future work. A transparent rule-based design therefore matches classical machine learning without any model training and is a practical and auditable starting point until enough labeled data exists for a transformer.

Keywords: Grievance Redressal; Rule-Based Text Classification; Complaint Prioritization; Error Level Analysis; YOLO Object Detection; Multilingual Voice Input

Nomenclature

Abbreviation	Description
AI	Artificial Intelligence
API	Application Programming Interface
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
CV	Computer Vision
ELA	Error Level Analysis
ER	Entity-Relationship
FIFO	First-In, First-Out
GPS	Global Positioning System
JPEG	Joint Photographic Experts Group (image format)
ML	Machine Learning
NLP	Natural Language Processing
PNR	Passenger Name Record
RF	Random Forest
RPF	Railway Protection Force
SVM	Support Vector Machine
TF-IDF	Term Frequency–Inverse Document Frequency
TP, TN, FP, FN	True positives, true negatives, false positives, false negatives
YOLO	You Only Look Once (object detector)

1. Introduction

Rail Madad is the grievance portal through which Indian Railways passengers report problems such as dirty coaches, poor food, faulty fans, theft, and medical emergencies. The volume is large and uneven, rising sharply during festival and holiday travel, yet the first step after submission is still manual: a staff

© Resbee Info Technologies Pvt Ltd

member reads each complaint, decides which department should handle it, and forwards it. This creates an operational problem with two visible symptoms. Complaints are sometimes sent to the wrong department and have to be forwarded a second time, and a complaint about a fire risk or a sick passenger waits in the same first-in, first-out queue as a complaint about a broken reading light unless someone happens to notice it. There is also a quieter problem at the end of the process. Passengers can attach photos, but the portal does not check whether those photos are genuine, and a department can mark a complaint as resolved without any evidence that the repair was actually carried out.

Earlier studies have shown that railway complaints posted on social media can be sorted automatically with standard text-mining and machine-learning methods [1][2][3], and that speech recognition for Indian languages has matured enough to support voice-based input [4]. Grievance systems for other public services have used NLP to categorize citizen reports [5][6][7], and complaint prioritization has been studied as a separate decision problem [8]. Image forensics offers ELA-based methods for spotting edited photos [9][10][11][12], while detectors from the YOLO family can quickly confirm whether an image shows the expected kind of scene [13][14]. The limitation is that each of these efforts solves one piece in isolation, usually on a dataset from a different domain. None of them checks whether a reported issue was actually fixed. Two considerations make an integrated solution timely. On the operational side, a single wrong routing decision or an overlooked safety complaint has a direct cost for passengers and staff. On the technological side, translation, speech input, and lightweight vision models are now easy to deploy without special hardware, while the main obstacle for a newly built railway system, the lack of a large labeled corpus, favors methods that do not depend on one.

This paper presents an AI-augmented triage layer for Rail Madad that brings these steps together in one deployed pipeline. Complaints arrive as text or voice in English, Kannada, or Hindi and are translated to English. A rule-based lexicon classifier assigns a department category; a sentiment score combined with an urgency-keyword override assigns priority; attached photos pass through YOLO, ELA, and duplicate checks, and the complaint is routed automatically. Department staff must then upload a geotagged photo before the complaint can be closed. The backend uses FastAPI and PostgreSQL. The study is guided by three research questions. RQ1: Can an interpretable lexicon-based classifier assign department categories on a small railway dataset with accuracy comparable to classical ML and transformer baselines trained on the same data? RQ2: Does an urgency-keyword override identify safety-critical complaints that sentiment alone would rank as routine? RQ3: Can a combined YOLO and ELA screen reliably separate genuine railway photos from edited or unrelated ones? The corresponding hypotheses are that the rule-based classifier performs at least as well as classical ML baselines (SVM and Random Forest) trained on the same data (H1), that the override raises recall on High-priority complaints compared with sentiment-only ranking (H2), and that the combined image screen correctly separates genuine photos from edited and unrelated ones (H3). The main contributions of this work are:

- An end-to-end railway grievance pipeline that combines text classification, urgency prioritization, image-evidence verification, and resolution auditing, four tasks that earlier work treats as separate problems (Table 1).
- A closed-loop accountability step in which a complaint cannot be closed without a geotagged resolution photo, which addresses false or unverified closures that classification and routing systems do not touch.
- An interpretable rule-based classification and prioritization design in which every routing decision can be traced to the keyword that triggered it, formalized in Eqs. (1) to (5) and suited to the low-resource setting in which the system was built.
- A benchmark of the rule-based classifier against SVM, Random Forest, and fine-tuned BERT baselines on an identical test split [15][16].
- Multilingual voice intake in English, Kannada, and Hindi, with translation to English so that one classification pipeline serves all three languages.

The method was evaluated on 600 manually labeled complaints collected from July to September 2026, with a held-out test set of 120 complaints used only for final scoring. The rule-based classifier achieved 78.0% accuracy and a macro F1-score of 0.76, higher than SVM (75.0%), equal to Random Forest (78.0%), and 11 percentage points below fine-tuned BERT (89.0%). Section 5 traces the likely sources of error to indirect phrasing and overlapping vocabulary, which affect the small Medical category most. Because classification and routing happen at the moment of submission, the manual reading step that currently delays triage is removed entirely, and every decision remains explainable to staff.

The rest of the paper is organized as follows. Section 2 reviews related work and identifies the research gap. Section 3 describes the existing system, the proposed architecture, the classification, prioritization, and verification methods, the dataset, the baselines, and the evaluation metrics. Section 4 reports the results. Section 5 discusses what the results mean, why misclassifications occur, the strengths and limits of the design, and its implications for railway administration. Section 6 concludes the paper and outlines future work.

2. Literature Review

We reviewed 34 studies for this work: 17 on NLP-based complaint and text classification, 1 on complaint urgency and prioritization, 14 on image forensics and object detection, and 2 on speech processing for Indian languages. They are grouped below into ten methodological directions.

2.1 Summary of Previous Research

2.1.1 Classical Machine Learning for Railway Complaints

Rawat and Kaushik built an NLP-based grievance framework for Indian Railways that uses classical classifiers, namely Naïve Bayes, SVM, and Random Forest, to separate posts addressed to the Railway Ministry into complaints, suggestions, and compliments [1]. Roy et al. applied machine-learning classifiers directly to railway complaint text [2], and Akhtar and Beg focused on identifying railway complaint tweets among general transport-related posts [3]. Together, these studies show that railway complaints can be classified automatically, but they stop at labeling text and do not assign priority or examine attached evidence.

2.1.2 NLP-Based Grievance Systems for Public Services

Several systems apply NLP to citizen grievances beyond railways. Qurat-Ul-Ain et al. proposed an autonomous model for classifying complaints [5], and Suganya et al. described an autonomous grievance-redressal system for government services [6]. Intani et al. automated the classification of public complaints received through Jakarta's JakLapor channel [17]. These works confirm the value of automated triage in public administration, although each is tied to its own category scheme and none verifies closure.

2.1.3 Contextual and Transformer-Based Classification

Madyatmadja et al. presented a contextual text-analytics framework for classifying citizen reports in Indonesian [7]. BERT introduced deep bidirectional pre-training that can be fine-tuned for classification [15], and Indonesian variants such as IndoBERT have been used for data augmentation [18] and multilabel classification together with multilingual BERT [19]. These models generally outperform keyword and TF-IDF methods when enough labeled data is available, but they need that data and offer little insight into individual decisions.

2.1.4 Fine-Tuning Strategies and Hybrid Deep Architectures

Bilal and Almazroi showed the effectiveness of a fine-tuned BERT model for separating helpful from unhelpful customer reviews [20], and Kaur and Kaur improved BERT for requirements classification by adding a bidirectional LSTM-CNN layer [21]. ULMFiT established transfer learning with gradual fine-tuning as a general recipe for text classification [22], and Jianan et al. applied deep learning to classify text in engineering-consulting standards [23]. These studies point toward the upgrade path for the present system once more data exists.

2.1.5 Text Preprocessing and Representation

Chai compared text-preprocessing methods and found that choices such as lowercasing, stop-word removal, and stemming measurably affect downstream accuracy [24]. Lan proposed a hybrid similarity measure that combines term semantics with TF-IDF weighting [25], and Xu surveyed deep learning in NLP more broadly [26]. These results inform the normalization step in our pipeline and the text features used by the classical baselines.

2.1.6 Analytics-Driven Complaint Prioritization

Vairetti et al. formulated complaint prioritization as an urgent-versus-routine decision and combined deep learning with multi-criteria decision-making to automate a labeling process that otherwise depends on expert judgement [8]. Our urgency override follows the same principle in a lighter, rule-based form: instead of learning an urgency function from labeled examples, a fixed lexicon of safety, fire, medical, and theft terms overrides a baseline sentiment-derived priority.

2.1.7 Error Level Analysis for Image Forgery Detection

ELA re-saves an image at a known JPEG quality and inspects the difference between the original and the re-compressed copy; regions that were edited tend to show a different error level from the rest of the image [9]. Sari and Fahmi reported that ELA preprocessing improves the accuracy of deep-learning forgery detectors [10]. Joshi et al. benchmarked several transfer-learning backbones, including VGG19, Inception-V3, ResNet, and Xception, on ELA-processed images [11], and Gupta et al. used ELA with a CNN to detect

images edited with common tools [12]. These methods apply directly to checking passenger evidence photos.

2.1.8 YOLO-Family Detectors for Fast Content Verification

YOLO treats object detection as a single regression pass, which makes it fast enough to act as a first-pass filter. YOLOv3 [13], YOLOv4 [27], and YOLOv6 [28] successively improved the balance between speed and accuracy, and Zong et al. demonstrated real-time object recognition with a YOLO model [14]. Hussain reviewed the family from YOLOv1 to YOLOv10 [29] and compared YOLOv5, YOLOv8, and YOLOv10 for real-time vision [30]. In our pipeline, such a detector checks that an uploaded photo actually shows a coach, platform, or station element before the more expensive forgery analysis runs.

2.1.9 Benchmarking of Object Detectors in Applied Domains

Applied comparisons help in choosing a detector version. Khanam et al. compared YOLOv5, YOLOv8, and YOLOv11 for solar-panel defect detection [31], Casas et al. compared YOLOv5 and YOLOv8 for wildfire and smoke detection [32], Mahendrakar et al. compared YOLOv5 with Faster R-CNN for navigation around non-cooperative targets [33], and Balasubramaniam and Pasricha surveyed object detection for autonomous vehicles [34]. A common finding is that the best version depends on the domain, which makes detector choice an important design decision for railway imagery.

2.1.10 Speech and Language Processing for Indian Languages

Priyamvada et al. studied multilingual speech recognition for Indian languages [4], and Nagaraja and Jayanna examined Kannada speaker identification under limited data [35]. Both highlight the scarcity of Kannada resources, which supports our choice to transcribe and translate regional-language input into English rather than building a separate classifier for each language.

2.2 Review Table

Table 1 summarizes the methodology, advantages, and disadvantages of the main studies reviewed above.

Table 1. *Comparative summary of related work*

Study	Methodology	Advantages	Disadvantages
[1]	NB, SVM, RF on railway social-media posts	Railway-specific; simple, reproducible models	Text labels only; no priority or evidence check
[2]	ML classification of railway complaints	Automates department sorting	No urgency handling or closure verification
[3]	Identification of railway complaint tweets	Filters complaints from general transport posts	Stops at detection; no routing
[5][6]	NLP-based autonomous grievance systems	End-to-end intent for public services	Generic categories; no image or closure checks
[7][17]	Citizen-report classification (Indonesia)	Real public-service datasets	Language- and city-specific; needs labeled data
[15][20][21]	Fine-tuned BERT and hybrid BERT models	High accuracy with enough data	Data-hungry; opaque decisions; GPU cost
[18][19]	IndoBERT augmentation and multilabel models	Handles low-resource languages better	Still needs a sizeable labeled corpus
[8]	Deep learning with multi-criteria decision-making for priority	Principled urgency scoring	Priority only; no classification or evidence
[9][10]	ELA combined with deep learning	Lightweight preprocessing; better accuracy	Sensitive to editing technique and compression
[11][12]	Transfer-learning CNNs on ELA images	Compares strong backbones	Not domain-specific; no content check
[13][14][27] [28][29][30]	YOLO-family real-time detection	Fast; suitable as a first-pass filter	Detects content only; cannot spot tampering
[4][35]	Speech processing for Indian languages	Supports regional-language access	Kannada resources are limited
Present work	Lexicon NLP + urgency override + YOLO/ELA + geotagged closure	Interpretable, integrated, low-cost, auditable	Lower accuracy ceiling; small dataset

2.3 Research Gap

Across the reviewed literature, complaint classification, urgency detection, and image forgery detection have been studied as separate problems, each evaluated on its own dataset. This leaves three specific gaps. First, no reviewed system combines text classification, priority assignment, and image verification in one

pipeline. Second, none is designed for railway grievances, where department structures and the meaning of urgency differ from generic customer-service settings. Third, none checks at closure time that a reported problem was actually fixed. There is also a methodological gap: studies that use transformers rarely report how a transparent, low-cost method compares on the same small dataset. The present work addresses all four points by integrating the three tasks in one railway-specific pipeline, adding mandatory geotagged resolution proof, and benchmarking the rule-based classifier against classical and transformer baselines on an identical split.

3. Methodology

3.1 Existing System and Problem Statement

The current Rail Madad platform accepts complaints through web and mobile channels, but sorting, prioritization, and verification are all manual. A staff member reads each complaint, decides which department is responsible, and forwards it. There is no automated flag that lifts a safety-critical complaint above routine ones, attached photos are stored without any authenticity check, and a complaint can be marked resolved without proof that corrective action was taken at the reported location. Table 2 contrasts the existing workflow with the proposed one.

Table 2. *Existing Rail Madad workflow compared with the proposed system*

Function	Existing Rail Madad	Proposed system
Complaint sorting	Manual, staff-reviewed	Automatic lexicon-based classification
Priority assignment	Not available (FIFO queue)	Sentiment score with urgency-keyword override
Image verification	None; any image accepted	YOLO content check and ELA forgery detection
Department routing	Manual forwarding	Automatic routing; escalation for High priority
Duplicate detection	Not available	Hash-based duplicate image flagging
Resolution proof	Not available	Geotagged photo required before closure
Analytics	Basic counts	Dashboard by category, priority, department, and status

Consider a passenger who reports a fan that is not working in a coach. Under the manual workflow, the complaint sits in an undifferentiated queue until someone reads and forwards it, and once it is marked resolved, there is nothing to show that the fan was repaired. Under the proposed system, the same complaint is classified as Electrical, given Medium priority, routed to Electrical Maintenance, and cannot be closed until a geotagged photo of the repaired fan is uploaded in Fig. 10.

3.2 System Architecture

The system follows a three-tier design: a client layer for passengers and staff, a FastAPI backend layer, and an AI processing layer made up of an NLP engine and a computer-vision engine, with PostgreSQL as the data store in Fig. 1.

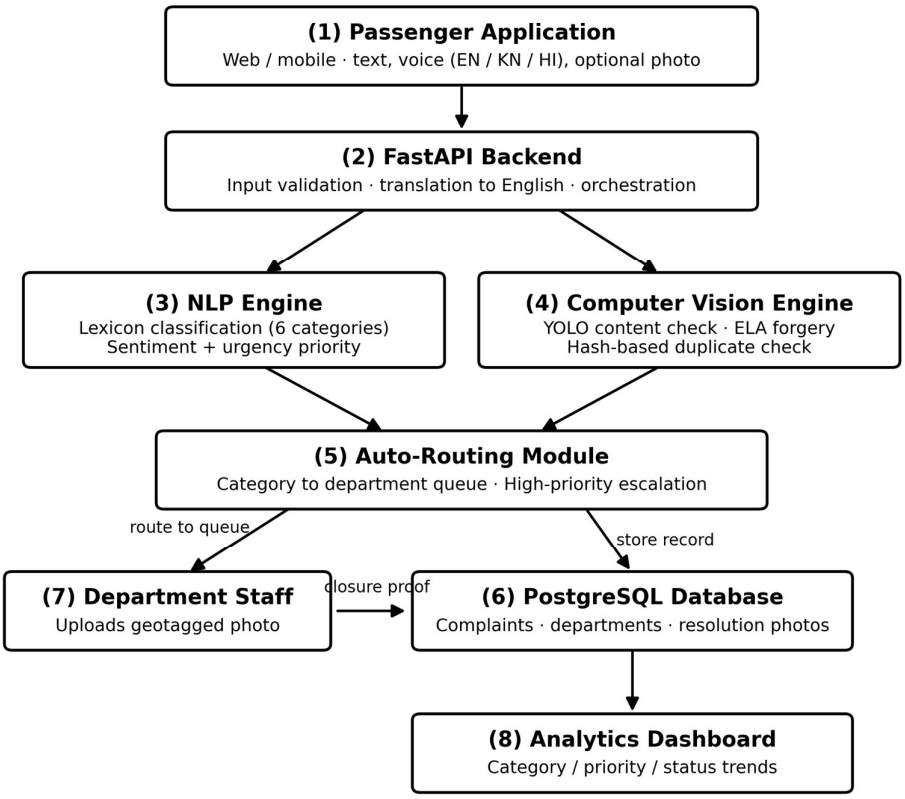

Fig. 1. *High-level architecture of the proposed system.*

Passenger input passes through the FastAPI backend to parallel NLP and computer-vision engines, whose outputs drive automatic routing; the loop closes only when department staff upload geotagged proof of resolution, which also feeds the analytics dashboard.

The flow through Fig. 1 proceeds in eight steps:

- **Step 1 (Passenger Application):** The passenger enters train, coach, seat, and PNR details and describes the problem by typing or speaking in English, Kannada, or Hindi, optionally attaching a photo.
- **Step 2 (FastAPI Backend):** The request is validated, non-English text is translated to English, and the complaint is passed to both AI engines.
- **Step 3 (NLP Engine):** the translated text is normalized, matched against six category lexicons to choose a department category, and given a priority based on its sentiment score and urgency keywords.
- **Step 4 (Computer Vision Engine):** If a photo is attached, YOLO checks that it shows railway content, ELA checks for signs of editing, and a hash check flags reuse of an earlier photo.
- **Step 5 (Auto-Routing):** The category is mapped to its operational department, and High-priority complaints are escalated to the top of that department's queue.
- **Step 6 (PostgreSQL Database):** The complaint, its category, priority, verification result, and status are stored, and a complaint ID with a verification code is returned to the passenger.
- **Step 7 (Department Staff):** Staff resolve the issue and upload a geotagged photo; only then can the status change to Closed.
- **Step 8 (Analytics Dashboard):** Officials see live distributions by category, priority, department, and resolution status.

The six operational modules are Complaint Intake, NLP Classification, Sentiment and Priority, Image Verification, Auto-Routing, and Resolution Verification with Analytics. Fig. 2 traces the messages exchanged during one complaint's lifetime, and Fig. 3 shows the underlying data model.

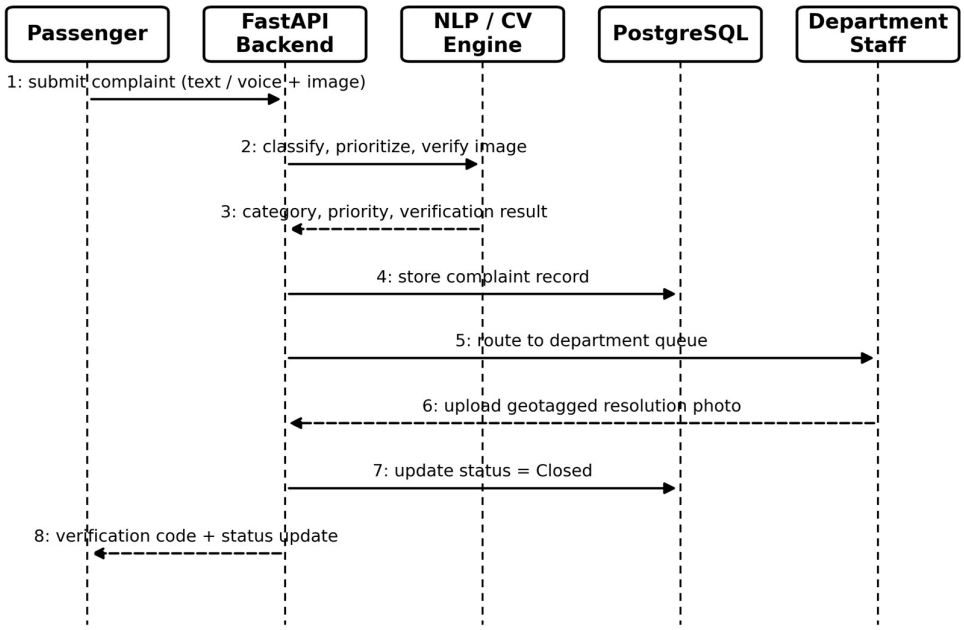

Fig. 2. *Complaint lifecycle sequence diagram.*

Classification, prioritization, and image checks happen in a single request before the complaint is stored and routed, and the status can change to Closed only after the department uploads a geotagged resolution photo.

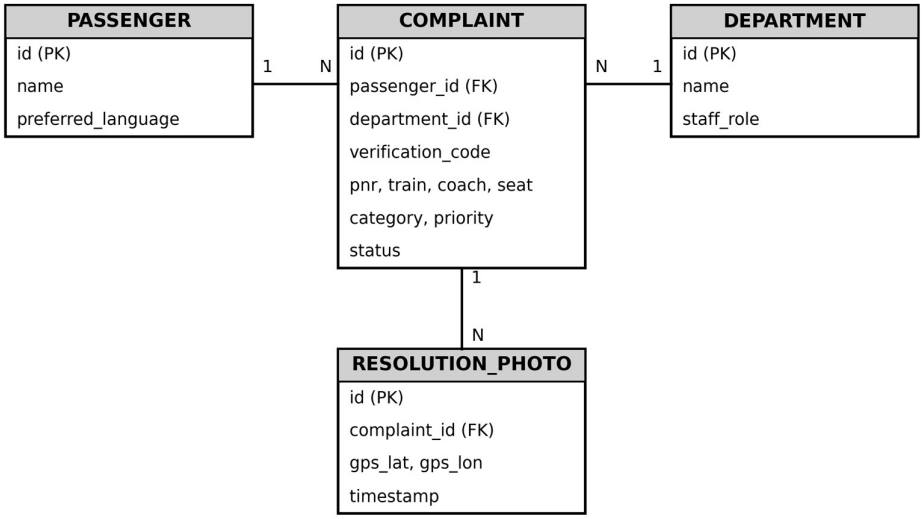

Fig. 3. *Simplified entity-relationship diagram.*

Each complaint belongs to one passenger and one department and may hold several resolution photos, so every closure can be traced back to time-stamped, located evidence.

3.3 Text Intake and Multilingual Handling

A complaint can be typed or spoken through a microphone control on the submission form, which also has a language selector for English, Kannada, and Hindi. Spoken input is transcribed in the browser using the Web Speech API, and non-English text is translated to English using the Deep Translator Python library. For example, the Hindi input "fan nahi chal raha hai" is translated to "The fan/air cooling is not working in the coach." Staff sees the translation next to the original, so nothing the passenger said is lost, while

classification always runs on a single English representation. A Track Complaint page lets passengers check status by entering their complaint ID (for example, RM-000001). Formally, the text passed to the classifier is

$$T' = norm(translate(T)) \quad (1)$$

where translate (·) is the identity for English input and norm (·) converts text to lowercase and removes punctuation.

3.4 Rule-Based Category Classification

Category assignment uses keyword and lexicon matching rather than a trained statistical classifier. This gives up some of the accuracy a fine-tuned transformer might reach in exchange for full interpretability, since every decision can be traced to the keywords that triggered it, and for independence from a large labeled corpus. Each category k ∈ {Cleanliness, Catering, Security, Medical, Electrical, Others} has a lexicon L_k of representative terms built from the training portion of the dataset. The match score of category k is

$$S_k(T') = \Sigma\,(w \in L_k)\; I[\, w \text{ occurs in } T' \,] \quad (2)$$

and the predicted category is

$$\hat{C} = argmax_k\, S_k(T') \text{ if } max_k\, S_k(T') > 0;\ \text{otherwise } \hat{C} = Others \quad (3)$$

When two categories tie, the one listed first in the order above is selected. The category is then mapped to an operational department: Cleanliness to Housekeeping, Catering to Catering, Security to the RPF, Medical to the Medical Team, Electrical to Electrical Maintenance, and Others to General Support. Fig. 4 shows the complete text pipeline, and Algorithm 1 gives the procedure.

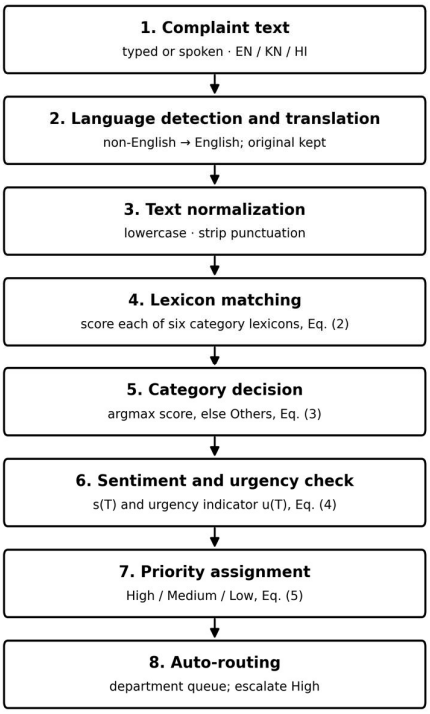

Fig. 4. *NLP and priority-assignment pipeline.*

Raw text in any of the three languages is reduced to one normalized English form before lexicon scoring, so a single set of rules produces both the department category and the priority.

Algorithm 1: Rule-based category and priority assignment
Input: complaint text T; Output: category C, priority P
1: $T' \leftarrow \text{norm}(\text{translate}(T))$ ▷ Eq. (1)
2: for each category k in {CL, CA, SE, ME, EL, OT}:
3: $S_k \leftarrow$ number of terms of L_k found in T' ▷ Eq. (2)
4: $C \leftarrow \text{argmax}_k S_k$; if all $S_k = 0$ then $C \leftarrow$ Others ▷ Eq. (3)
5: $s \leftarrow \text{sentiment_score}(T')$
6: if T' contains any term of U then $P \leftarrow$ High ▷ Eqs. (4)–(5)
7: else if $s \leq \tau$ then $P \leftarrow$ Medium
8: else $P \leftarrow$ Low
9: return C, P

3.5 Priority Assignment

Priority combines two signals. The first is a sentiment polarity score $s(T') \in [-1, 1]$, computed as the TextBlob polarity score, where more negative values indicate stronger dissatisfaction. The second is an urgency indicator based on the urgency lexicon U, a fixed word list that contains safety, fire, medical, theft, and accident terms:

$$u(T') = 1 \text{ if } \exists\, w \in U \text{ such that } w \text{ occurs in } T';\ \text{otherwise } u(T') = 0 \quad (4)$$

Priority is then assigned by the piecewise rule.

$$P(T') = \text{High if } u(T') = 1;\ \text{Medium if } u(T') = 0 \text{ and } s(T') \leq \tau;\ \text{Low otherwise} \quad (5)$$

where τ is a negative threshold chosen on the training portion. Because the urgency check comes first, a calmly worded but genuinely dangerous report, such as "there is a small fire near the pantry", is still ranked High instead of being downgraded for its neutral tone.

3.6 Image Verification

An attached photo X passes through three checks. First, a YOLOv11 detector, the most recent version among those compared in [31], checks content relevance:

$$R(X) = 1 \text{ if some detection } d \text{ has class}(d) \in \Omega \text{ and conf}(d) \geq \gamma;\ \text{otherwise } R(X) = 0 \quad (6)$$

where Ω is the set of railway-relevant classes, and γ is the detection-confidence threshold. Second, ELA re-compresses the image as a JPEG at a fixed quality q to obtain a copy J and computes the error map.

$$E(x, y) = |\, X(x, y) - J(x, y)\, | \quad (7)$$

The image is flagged as possibly edited when a summary statistic $\varphi(E)$ of the error map exceeds a threshold θ:

$$F(X) = I[\, \varphi(E) > \theta\,] \quad (8)$$

Third, a hash H(X) is compared with the hashes of previously stored photos, and the photo is marked as a duplicate when the distance to any of them is at most δ:

$$Dup(X) = I[\, \exists\, X' \text{ stored such that } dist(H(X), H(X')) \leq \delta\,] \quad (9)$$

The photo is accepted as supporting evidence only when all three checks pass:

$$V(X) = R(X) \cdot (1 - F(X)) \cdot (1 - Dup(X)) \quad (10)$$

3.7 Resolution Verification

Before a complaint c can move to Closed, the assigned department must upload a photo r of the resolved issue. The photo carries GPS coordinates, train and coach identifiers, and a timestamp, which gives auditors a verifiable record that the work was done rather than simply marked complete. The closure rule enforced by the backend is

$$Close(c) \text{ is allowed} \Leftrightarrow \exists\, r \text{ linked to } c \text{ with } GPS(r) \neq \emptyset \text{ and } t(r) > t_o(c) \quad (11)$$

3.8 Dataset

The lexicons and priority rules were developed and tested on a custom dataset of 600 complaints collected between July 2026 and September 2026. Complaints were short free-text descriptions, typed or voice-transcribed in English, Kannada, and Hindi, and each was manually labeled with a category and a priority by the authors. The data were split 80:20 with stratification by category into a training portion, used only to build the lexicons and choose τ, and a held-out test portion used only for final scoring in Table 3.

Table 3. *Dataset composition by category*

Category	Total	Train	Test
Cleanliness	180	144	36
Catering	90	72	18
Security	120	96	24
Medical	45	36	9
Electrical	105	84	21
Others	60	48	12
Total	600	480	120

The category distribution is imbalanced. Cleanliness is the largest category with 180 complaints (30%), followed by Security (120) and Electrical (105), while Medical is the smallest with only 45 complaints (7.5%), leaving just 9 Medical complaints in the test set. This mirrors the skew in real railway complaints, where hygiene problems are reported far more often than medical emergencies. For this reason, the macro-averaged F1-score, which weights every category equally, is reported alongside overall accuracy.

3.9 Baseline Models

To answer RQ1, three baselines were trained on the same training split and scored on the same test split as the rule-based classifier. The classical baselines were implemented with scikit-learn [16], and the transformer baseline follows the original BERT fine-tuning procedure [15]. Configurations are listed in Table 4.

Table 4. *Baseline configurations (identical train/test split)*

Model	Features	Model type
SVM	Vectorized complaint text	Margin-based linear classifier
Random Forest	Vectorized complaint text	Ensemble of decision trees
BERT	WordPiece tokens	Pre-trained transformer fine-tuned on the training split
Proposed	Keyword lexicons	Eqs. (1)–(5); no training beyond lexicon construction

3.10 Evaluation Metrics

Overall accuracy is the share of the N_t test complaints assigned the correct category:

$$Accuracy = (1 / N_t) \Sigma_i I[\hat{C}_i = C_i] \quad (12)$$

For each category k, precision, recall, and F1-score are computed from that category's true positives, false positives, and false negatives:

$$Precision_k = TP_k / (TP_k + FP_k) \quad (13)$$

$$Recall_k = TP_k / (TP_k + FN_k) \quad (14)$$

$$F1_k = 2 \cdot Precision_k \cdot Recall_k / (Precision_k + Recall_k) \quad (15)$$

Macro-averaged scores give each category equal weight regardless of size:

$$Macro\text{-}F1 = (1 / K) \Sigma_k F1_k \quad (16)$$

Overall accuracy is reported for all four models so that they can be compared directly on the same test split, and the macro-averaged F1-score is reported for the rule-based classifier.

3.11 Implementation

The backend is written in Python with FastAPI, chosen for its low request overhead and native asynchronous I/O, which suits a request that chains translation, classification, image processing, and database writes. PostgreSQL stores passengers, complaints, departments, and resolution-photo metadata, with foreign-key constraints enforcing the relationships in Fig. 3. The browser-based frontend serves Passenger, Department Staff, and Admin roles, with role-based views controlling which fields and actions are visible. Table 5 lists the reference hardware and software. Fig. 5, Fig. 6 and Fig. 7 show screens from the deployed system.

Table 5. *Hardware and software used for development and testing*

Category	Specification
Processor	Intel Core i5 / AMD Ryzen 5 or higher
RAM	8 GB minimum (16 GB recommended)
Storage	50 GB SSD minimum
Backend	FastAPI (Python)
NLP libraries	Deep Translator (translation), TextBlob (sentiment)
Database	PostgreSQL
Image verification	YOLOv11, ELA, image hashing
Operating system	Ubuntu 20.04 LTS / Windows 10 or higher

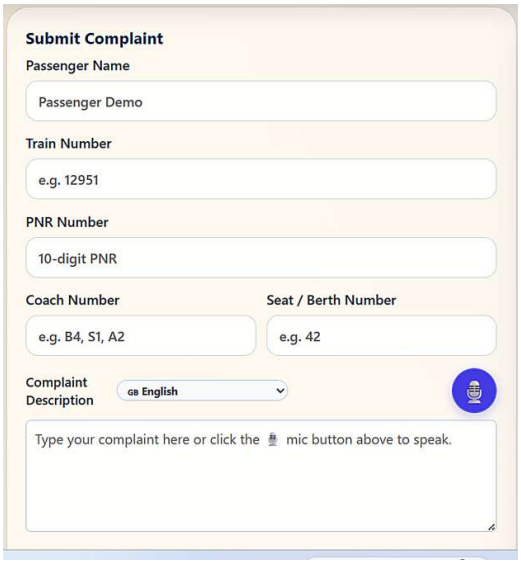

Fig. 5. *Complaint submission page with the voice-input button and the English/Kannada/Hindi language selector.*

- Passengers can describe a problem in their own language without typing.

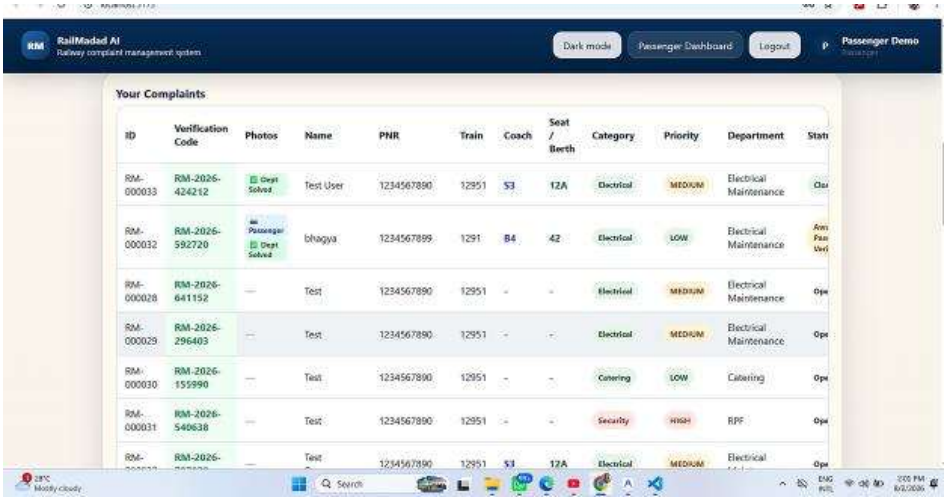

Fig. 6. *Passenger dashboard listing each complaint with its verification code, category, priority, assigned department, and status.*

- Classification results are visible to the passenger as soon as the complaint is submitted.



Fig. 7. *Admin dashboard with live totals and category, priority, and department distributions.*

- Officials can see where complaints are concentrating without compiling reports by hand.

4. Results

This section reports the classification results on the held-out test set of 120 complaints, the composition of the evaluated data, and an end-to-end example from the deployed system. The priority-assignment and image-verification modules were implemented and are shown working in the end-to-end example, but their quantitative evaluation on labeled test sets is planned for future work, so no accuracy figures are reported for them here. Interpretation is left to Section 5.

4.1 Classification Performance and Baseline Comparison

Table 6 compares the rule-based classifier with the three baselines on the identical test split of 120 complaints, and Fig. 8 shows the same accuracies as a chart.

Table 6. *Classification accuracy on the held-out test set (120 complaints)*

Model	Features	Training required	Accuracy (%)
SVM	Vectorized complaint text	Yes	75.0
Random Forest	Vectorized complaint text	Yes	78.0
BERT (fine-tuned)	WordPiece tokens	Yes	89.0
Proposed (rule-based)	Keyword lexicons	No	78.0

The rule-based classifier achieved a macro-averaged F1-score of 0.76 across the six categories. Its accuracy was 3.0 percentage points higher than SVM, equal to Random Forest, and 11.0 percentage points lower than fine-tuned BERT. Of the four models, it is the only one that required no model training.

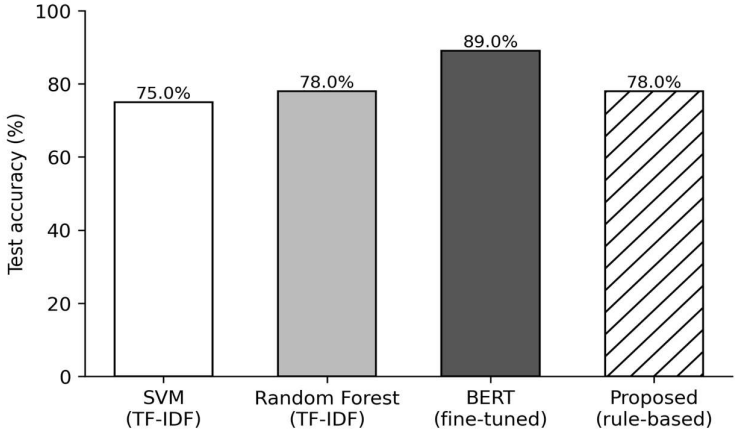

Fig. 8. *Test accuracy of the four classifiers on the same 120-complaint test set.*

- The rule-based classifier matches Random Forest and exceeds SVM without any model training, while fine-tuned BERT leads by 11 percentage points.

4.2 Composition of the Evaluated Data

Table 7 gives the share of each category in the test set, and Fig. 9 shows the train and test counts for all 600 complaints.

Table 7. *Test-set composition by category*

Category	Test complaints	Share of test set (%)
Cleanliness	36	30.0
Security	24	20.0
Electrical	21	17.5
Catering	18	15.0
Others	12	10.0
Medical	9	7.5
Total	120	100.0

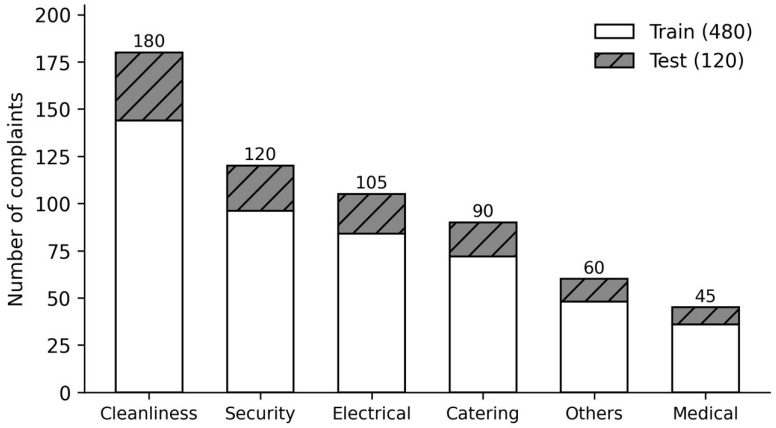

Fig. 9. *Train and test counts for each category across the 600 complaints.*

- Cleanliness makes up 30% of the data while Medical makes up only 7.5%, so Medical results rest on just 9 test complaints.

4.3 End-to-End Example from the Deployed System

Table 8 traces one real complaint through the deployed pipeline, and Fig. 10 shows its resolution audit. At the time the admin dashboard in Fig. 7 was captured, the system held 33 complaints, of which 21 were pending, 12 were resolved, and 3 were marked High priority.

Table 8. *Processing trace of a sample complaint in the deployed system*

Stage	Output
Passenger input (Hindi)	fan nahi chal raha hai
English translation	The fan/air cooling is not working in the coach.
Category	Electrical
Priority	Medium
Assigned department	Electrical Maintenance
Image verification	Verified; not a duplicate.
Resolution evidence	Geotagged photo uploaded by Electrical Maintenance
Final status	Closed

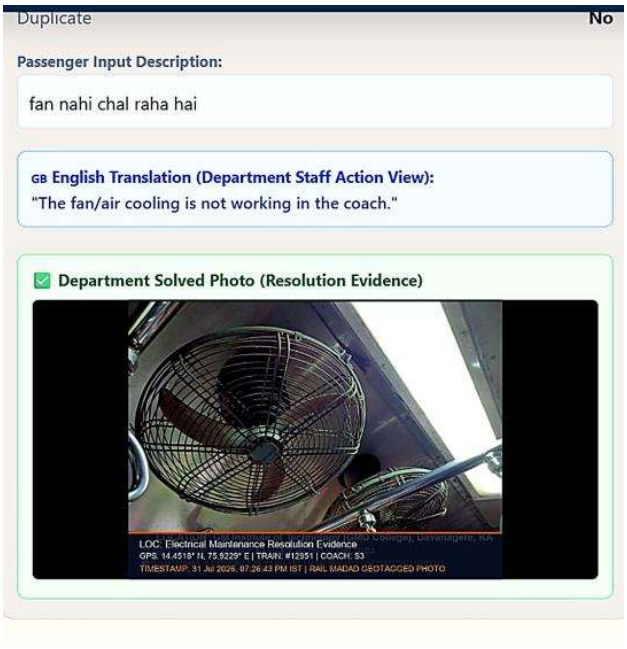

Fig. 10. *Resolution audit for the complaint in Table 8.*

- The department's repair photo carries GPS coordinates, train and coach details, and a timestamp, so the closure can be checked independently rather than taken on trust.

5. Discussion

5.1 Answers to the Research Questions

For RQ1, the rule-based classifier reached 78.0% accuracy, three points above SVM (75.0%) and on par with Random Forest (78.0%), so H1 is supported: hand-built lexicons did as well as classical models that learned from the same 480 training complaints. Fine-tuned BERT was clearly stronger at 89.0%, an 11-point lead that likely reflects its ability to understand indirect phrasing and context that keyword matching misses. The gap is the price of interpretability. The rule-based system needs no GPU, no retraining, and explains every decision, whereas BERT offers higher accuracy but opaque decisions and higher running cost. With only 600 complaints, the rule-based design is a sensible first deployment, and the BERT result indicates a clear upgrade path as labeled data grows, consistent with earlier findings on transformer classifiers for public complaints [7][18]. RQ2 and RQ3 were not measured quantitatively in this study. The deployed system does apply the urgency override and the image checks to every complaint, as the example in Table 8 shows, but testing H2 and H3 requires a labeled priority set and a labeled image set, which are planned as the next stage of evaluation.

5.2 Why Misclassifications Occur

Because every decision follows Eqs. (2) and (3), the sources of error can be read directly from the design. Five mechanisms can cause misclassifications, and the first four are likely to affect the Medical category most, since it is the smallest in the dataset.

- Indirect phrasing and lexicon sparsity. Medical complaints are often described through symptoms or situations, such as "my father fainted" or "a child has been vomiting since Pune", rather than words like "doctor" or "medical". Medical also has the fewest training examples (36 of 480), so its lexicon L_k is the smallest, and more of its complaints can score zero under Eq. (2) and fall through to Others by Eq. (3). With only 9 Medical complaints in the test set, each such miss lowers Medical recall by about 11 percentage points.
- Vocabulary overlap between categories. Words such as "injured" or "hurt" occur both in assault reports (Security) and in health emergencies (Medical), and "food made me sick" matches both Catering and Medical. When scores are equal, the fixed tie-break order selects Catering or Security before Medical, which pulls borderline cases away from Medical.

- Translation drift. Regional expressions such as "tabiyat kharab hai" are often translated into general English phrases like "not feeling well", which may contain none of the Medical lexicon terms even though the original was clearly a health complaint.
- Exact matching without morphology or negation. The lexicon matches surface forms, so "fever" matches but "feverish" may not, and "no doctor available" and "doctor was helpful" count the same way.
- Others as a default. Since Others also collects every zero-score complaint, it absorbs errors from all other categories, which tends to lower its precision.

Two points soften the practical impact. Medical terms are also in the urgency lexicon, so a misclassified medical complaint is still marked High priority and escalated whenever it contains an urgency term, which limits the safety cost of the error. And each of these mechanisms has a targeted fix, such as symptom terms added to L_k, lemmatization, a negation window, or a small medical phrase list in Kannada and Hindi applied before translation, which a black-box model would not reveal as clearly.

5.3 Comparison with Previous Studies

Earlier railway studies classified complaints with classical models [1][2][3] but did not assign priority, check evidence, or verify closure. Public-service systems [5][6][17] automated sorting for other domains, and transformer-based work [7][19][20] reports strong accuracy when larger labeled datasets are available. The present results place the rule-based classifier at or above the classical baselines on the same small dataset and below the transformer, which agrees with the general pattern in that literature, while it remains the only approach among those compared that explains each decision. The image-verification stage builds on established ELA methods [9][12], and its quantitative comparison with that work is left for future evaluation.

5.4 Advantages and Limitations of the Rule-Based Design

The main advantages are interpretability, low cost, and ease of maintenance. Every category and priority can be explained by pointing to the triggering word, which matters for a public body that may have to justify how a complaint was handled. Processing needs no GPU, and staff can update the lexicons without retraining. The geotagged closure step adds accountability that is missing from the existing workflow, and multilingual voice input extends access beyond Rail Madad's current English and Hindi support to include Kannada.

The limitations are equally clear. Accuracy depends heavily on lexicon coverage and has a lower ceiling than a fine-tuned transformer on unusual or indirect phrasing. The dataset of 600 complaints is small and was collected over three months, so it may not reflect the full variety of real Rail Madad traffic. Classification depends on translation quality for Kannada and Hindi input. Per-category precision, recall, and F1-score with a confusion matrix for the rule-based classifier, as well as routing accuracy, response time, priority assignment, and image verification, have not yet been measured on labeled test sets; the image stage has not been tested against adversarially crafted forgeries, and the system has not been load-tested under peak holiday volumes or subjected to a formal security audit.

5.5 Implications for Railway Administration

For railway administrators, the results have several practical implications. Automatic triage removes the manual reading step from the critical path, so staff time moves from sorting complaints to resolving them. At the measured accuracy of 78%, some complaints will still be misclassified, so staff should keep the ability to reassign a complaint, and these corrections can be fed back into the lexicons. The urgency override gives control rooms a short, explicit list of High-priority complaints to act on first, which is especially useful during peak travel when queues are longest. Mandatory geotagged proof changes how departments are held accountable: closure figures can be audited by sampling photos, and unusually fast or repeated closures can be spotted from the dashboard. Because the classifier is transparent, zonal offices can adjust lexicons for local terms and languages without data-science support. Finally, the category and priority breakdowns give management evidence for where to direct maintenance budgets and staff, for example toward routes that generate repeated Electrical or Cleanliness complaints.

5.6 Scalability, Security, and Privacy

Since classification is rule-based, the per-complaint cost is low and does not grow with GPU demand; the main scaling concern is PostgreSQL write throughput during peak periods, which remains to be stress-tested. The system stores names, PNR numbers, and geotagged photos, all of which are personally identifiable or location-sensitive. Role-based access separates passenger, staff, and admin views, but

encryption at rest for stored images, access logging for the admin dashboard, and a formal security review are needed before any deployment with real passenger data.

6. Conclusion

This paper presented a grievance-triage pipeline for Indian Railways' Rail Madad platform that brings together rule-based NLP classification, sentiment and urgency-based prioritization, YOLO and ELA image verification, multilingual voice input, and geotagged resolution proof in one working FastAPI and PostgreSQL system. On 600 labeled complaints, the rule-based classifier reached 78.0% accuracy and a macro F1-score of 0.76, outperforming SVM (75.0%), matching Random Forest (78.0%), and trailing fine-tuned BERT (89.0%), without any model training. Classification and routing now happen at submission rather than after a manual read, and every closure carries geotagged proof.

In practical terms, the system gives every complaint a traceable category and priority the moment it is submitted and makes every closure auditable. Its advantages are transparency, ease of updating, and low deployment cost; its limitations are dependence on lexicon coverage, a small dataset, and untested behavior under peak load and adversarial images. Future work will fine-tune BERT and multilingual Indian-language models such as IndicBERT or MuRIL once a larger labeled dataset is available, classify Kannada and Hindi text directly instead of relying on translation, extend support to more regional languages, measure priority accuracy, routing accuracy, response time, and image-verification accuracy on labeled test sets, stress-test the image pipeline against adversarial forgeries, load-test the backend under holiday-scale traffic, and carry out a security review ahead of a pilot with a railway division.

Acknowledgment

The authors thank the School of Computer Applications, GM University, Davangere, for its guidance and resources during this project, and the faculty reviewers who commented on earlier drafts of the system design.

Appendix

Screenshots in Figures. 5,6,7 and 10 were captured from the deployed system and are included as evidence of implementation rather than as design mockups. Source code for the classification lexicons, priority rules, and geotag-verification logic is available from the authors on request. Category-to-department mapping: Cleanliness → Housekeeping; Catering → Catering; Security → RPF; Medical → Medical Team; Electrical → Electrical Maintenance; Others → General Support.

Compliance with Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows the ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

References

[1] M. Rawat and N. Kaushik, "NLP-based grievance redressal system for Indian Railways," arXiv preprint arXiv:2111.08999, 2021, doi: 10.48550/arXiv.2111.08999.

[2] T. S. Roy et al., "Automatic classification of railway complaints using machine learning," E3S Web of Conferences, vol. 477, p. 00085, 2024, doi: 10.1051/e3sconf/202447700085.

[3] N. Akhtar and M. M. S. Beg, "Railway complaint tweets identification," in Data Management, Analytics and Innovation: Proc. ICDMAI 2020, Advances in Intelligent Systems and Computing, vol. 1174. Springer, 2021, pp. 195–207, doi: 10.1007/978-981-15-5616-6_14.

[4] R. Priyamvada, S. S. Kumar, H. B. B. Ganesh, and K. P. Soman, "Multilingual speech recognition for Indian languages," in Advanced Machine Intelligence and Signal Processing, Lecture Notes in Electrical Engineering, vol. 858. Springer, 2022, doi: 10.1007/978-981-19-0840-8_41.

[5] Qurat-Ul-Ain, A. Shaukat, and U. Saif, "NLP-based model for classification of complaints: Autonomous and intelligent system," in Proc. 2nd Int. Conf. Digital Futures and Transformative Technologies (ICoDT2), 2022, doi: 10.1109/ICoDT255437.2022.9787456.

[6] S. Suganya, N. Kirubakaran, S. Senthil Kumar, C. Selvaganesan, B. Sundarambal, and G. Senthil Kumar, "Autonomous grievance redressal system for government services," in Proc. Int. Conf. Data Science and Business Systems (ICDSBS), 2025, doi: 10.1109/ICDSBS63635.2025.11031909.

[7] E. D. Madyatmadja, B. N. Yahya, and C. Wijaya, "Contextual text analytics framework for citizen report classification: A case study using the Indonesian language," IEEE Access, vol. 10, pp. 31432–31444, 2022, doi: 10.1109/ACCESS.2022.3158940.

[8] C. Vairetti, I. Aránguiz, S. Maldonado, J. P. Karmy, and A. Leal, "Analytics-driven complaint prioritisation via deep learning and multicriteria decision-making," European Journal of Operational Research, vol. 312, no. 3, pp. 1108–1118, 2024, doi: 10.1016/j.ejor.2023.08.027.

[9] I. B. K. Sudiatmika, F. Rahman, Trisno, and Suyoto, "Image forgery detection using error level analysis and deep learning," Telkomnika, vol. 17, no. 2, pp. 653–659, 2019, doi: 10.12928/TELKOMNIKA.v17i2.8976.

[10] W. P. Sari and H. Fahmi, "The effect of error level analysis on the image forgery detection using deep learning," Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, vol. 6, no. 3, 2021, doi: 10.22219/kinetik.v6i3.1272.

[11] R. Joshi, A. Gupta, N. Kanvinde, and P. Ghonge, "Forged image detection using SOTA image classification deep learning methods for image forensics with error level analysis," arXiv preprint arXiv:2211.15196, 2022, doi: 10.48550/arXiv.2211.15196.

[12] A. Gupta, R. Joshi, and R. Laban, "Detection of tool-based edited images from error level analysis and convolutional neural network," arXiv preprint arXiv:2204.09075, 2022, doi: 10.48550/arXiv.2204.09075.

[13] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018, doi: 10.48550/arXiv.1804.02767.

[14] C. Zong, K. Meng, J. Sun, and Q. Zhou, "Real-time object recognition based on YOLO model," in Proc. 3rd Int. Conf. Electronic Information Engineering and Computer Science (EIECS), 2023, pp. 197–202, doi: 10.1109/EIECS59936.2023.10435392.

[15] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, doi: 10.18653/v1/N19-1423.

[16] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011, doi: 10.5555/1953048.2078195.

[17] S. M. Intani et al., "Automating public complaint classification through JakLapor channel: A case study of Jakarta, Indonesia," in 2022 IEEE Int. Smart Cities Conf. (ISC2), 2022, pp. 1–6, doi: 10.1109/ISC255366.2022.9922346.

[18] Muftie and M. Haris, "IndoBERT-based data augmentation for Indonesian text classification," in 2023 Int. Conf. Inf. Technol. Res. Innov. (ICITRI), 2023, pp. 128–132, doi: 10.1109/ICITRI59340.2023.10250061.

[19] G. Z. Nabiilah, I. N. Alam, E. S. Purwanto, and M. F. Hidayat, "Indonesian multilabel classification using IndoBERT embedding and MBERT classification," Int. J. Elect. Comput. Eng., vol. 14, no. 1, pp. 1071–1078, 2024, doi: 10.11591/ijece.v14i1.pp1071-1078.

[20] M. Bilal and A. A. Almazroi, "Effectiveness of fine-tuned BERT model in classification of helpful and unhelpful online customer reviews," Electron. Commer. Res., vol. 23, no. 4, pp. 2737–2757, 2023, doi: 10.1007/s10660-022-09560-w.

[21] K. Kaur and P. Kaur, "Improving BERT model for requirements classification by bidirectional LSTM-CNN deep model," Computers and Electrical Engineering, vol. 108, p. 108699, 2023, doi: 10.1016/j.compeleceng.2023.108699.

[22] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in Proc. 56th Annu. Meet. Assoc. Comput. Linguist., 2018, pp. 328–339, doi: 10.18653/v1/P18-1031.

[23] G. Jianan, R. Kehao, and G. Binwei, "Deep learning-based text knowledge classification for whole-process engineering consulting standards," Journal of Engineering Research, 2023, doi: 10.1016/j.jer.2023.07.011.

[24] C. P. Chai, "Comparison of text preprocessing methods," Nat. Lang. Eng., vol. 29, no. 3, pp. 509–553, 2023, doi: 10.1017/S1351324922000213.

[25] F. Lan, "Research on text similarity measurement hybrid algorithm with term semantic information and TF-IDF method," Adv. Multimed., vol. 2022, pp. 1–11, 2022, doi: 10.1155/2022/7923262.

[26] Z. Xu, "Research on deep learning in natural language processing," Adv. Comput. Commun., vol. 4, no. 3, pp. 196–200, 2023, doi: 10.26855/acc.2023.06.018.

[27] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020, doi: 10.48550/arXiv.2004.10934.

[28] C. Li et al., "YOLOv6: A single-stage object detection framework for industrial applications," arXiv preprint arXiv:2209.02976, 2022, doi: 10.48550/arXiv.2209.02976.

[29] M. Hussain, "YOLOv1 to YOLOv10: A comprehensive review of YOLO variants and their application in the agricultural domain," arXiv preprint arXiv:2406.10139, 2024, doi: 10.48550/arXiv.2406.10139.

[30] M. Hussain, "YOLOv5, YOLOv8 and YOLOv10: The go-to detectors for real-time vision," arXiv preprint arXiv:2407.02988, 2024, doi: 10.48550/arXiv.2407.02988.

[31] R. Khanam, T. Asghar, and M. Hussain, "Comparative performance evaluation of YOLOv5, YOLOv8, and YOLOv11 for solar panel defect detection," Solar, vol. 5, no. 1, p. 6, 2025, doi: 10.3390/solar5010006.

[32] E. Casas, L. Ramos, E. Bendek, and F. Rivas-Echeverria, "YOLOv5 vs. YOLOv8: Performance benchmarking in wildfire and smoke detection scenarios," J. Image Graph., vol. 12, no. 2, pp. 127–136, 2024, doi: 10.18178/joig.12.2.127-136.

[33] T. Mahendrakar et al., "Performance study of YOLOv5 and Faster R-CNN for autonomous navigation around non-cooperative targets," in 2022 IEEE Aerospace Conference (AERO), 2022, pp. 1–12, doi: 10.1109/AERO53065.2022.9843537.

[34] A. Balasubramaniam and S. Pasricha, "Object detection in autonomous vehicles: Status and open challenges," arXiv preprint arXiv:2201.07706, 2022, doi: 10.48550/arXiv.2201.07706.

[35] B. G. Nagaraja and H. S. Jayanna, "Kannada language parameters for speaker identification with the constraint of limited data," Int. J. Image Graph. Signal Process., vol. 5, no. 9, pp. 14–20, 2013, doi: 10.5815/ijigsp.2013.09.03.