Diksha Jagtap, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Autonomous Grievance Redressal System for E-Governance Using AI and Automation

[1]Diksha Jagtap, [2]Avantika Pachpute, [3]Prof. Shital Y. Mandlik, [4]Dr. Anand A. Khatri
Savitribai Phule Pune University

Abstract- Grievances systems for e-governance developed traditionally suffers the issues of duplication of complaints, faulty categorization, delays in resolution, heavy workload of manual intervention and others. The intelligent cloud-based complaint management system powered by artificial intelligence, natural language processing, on-device machine learning, and cloud computing is called AGRS-EG, or autonomous grievance redressal system for e-governance. Through an Android application, users will send complaints containing text, images, and location through GPS. A hybrid classification engine that uses NLP for semantic text understanding and TFLite for on-device image classification achieves a 96.3% combined accuracy. The image validation module rejects irrelevant uploads at 89.5% accuracy. The main contribution is the three-level duplicate detection pipeline category filtering, Haversine-based 50-metre geo-proximity filtering, and AI semantic similarity which results in 93.8% duplicate detection accuracy and 4.1% false duplicate. Any duplicate, based on its severity, number, and type, will cause an escalation in priority based on a dynamic system. AI chatbot solves 87.6% of user queries by itself. Using the Google Maps API the administrative dashboard shows possible hot-spots on the map. The processing was 6.9× faster than manual baseline.

Keywords: Grievance Redressal; E-Governance; AI Classification; Duplicate Detection; TensorFlow Lite; Google Gemini; Firebase; NLP; Image Validation; Dynamic Priority.

I. INTRODUCTION

Governance must be effective through speedy resolution of grievances In the urban e-governance realms such as municipal corporations, transport authorities, and utilities, daily complaints run into the thousands making it impossible to deal with manually [25]. The portals presently available facilitate submission and status tracking, but these portals do not intelligently classify, validate/deduplicate and priorities [1]. As a result, the duplicates add to the workload. Also, misclassified complaints reach the wrong departments. Further, static priority in queues results in delays in critical infrastructure issues. This leads to the erosion of the trust of the citizens [4]. Recent advancements in LLMs, on-device ML, and cloud services have made it possible for grievance systems to function as proactive intelligent decision-support platforms [7][9]. This paper presents the AGRS-EG that consists of six innovations, specifically hybrid AI classification, image validation, three-level location-aware duplicate detection, dynamic priority assignment, AI chatbot, and mapping GIS. Sections II to VIII are alternative on Related Work Architecture Methodology Results Comparison Discussion and Conclusion.

II. LITERATURE SURVEY

In [1], Suganya et al. designed domain classification with sentiment-based severity prioritization which lacked duplicate aware priority escalation as static queues still contain same location issues which is a redundant approach. Sidharth and others applied MobileNetV2 in their image-based routing in android but they did not provide any duplicate detection nor any dynamic priority. Balakrishnan et al. [3] applied YOLO [15] for image authenticity verification (real/fake only) but cannot gauge severity or merge associated reports. The authors Joy et al. [4] experimented with BERT [11] along with the methodology of keyword clustering for textual grouping but proximity-unaware textual grouping fails to segregate such reports which are expectedly different but are far apart geographically. Ramesh et al. [5] used SVM [14] and Random Forest [16] for multi-label classification with only static priority. Without validating or de-duplication of image, Chen et al. [6] took to LSTM [17] routing. AGRS-EG covers all 6 gaps.

III. SYSTEM ARCHITECTURE

The architecture of AGRS-EG has six functional layers which create a closed-loop pipeline from the submission of the complaint till the resolution [7][8]. Information moves up from the sensors to the admin dashboard, while actionable alerts and ML predictions come down.

USER LAYER (Android App)

Complaint Form · GPS Location · Image Upload · AI Chatbot

▼

AI PROCESSING LAYER

NLP (Text) + TFLite (Image) → Hybrid Category Detection and Image Validation Module

▼

DUPLICATE DETECTION LAYER

Step 1: Category Match → Step 2: Location ≤50m (Haversine) Step 3: Semantic Similarity

▼

PRIORITY & ROUTING LAYER

Priority = 0.4·Severity + 0.4·DupCount + 0.2·TypeWeight SLA Assignment: Critical(24h) · High(48h) · Medium(72h)

▼

BACKEND LAYER (Firebase)

Auth · Realtime DB · Cloud Storage

▼

ADMIN DASHBOARD LAYER

Complaint Mgmt. · Map Visualization (Google Maps) Status Tracking · Analytics · Hotspot Detection

Fig. 1. AGRS-EG Six-Layer System Architecture.

A. User Layer — Android Application

The mobile frontend built with Java/XML on Android Studio for Android 8.0+ provides a login based on Firebase Authentication, a form to submit complaints with an image capture, and GPS coordinates capture of the geospatial duplicate pipeline [8].

B. AI Processing Layer — Hybrid Classification

Classification employs a parallel architecture consisting of two streams. The text is passed to the Gemini API [7] for zero-shot semantic classification into six road categories: Road Damage, Garbage, Water Leakage, Electrical, Sanitation, Infrastructure Damage. The image stream makes use of TFLite MobileNet [10] in-device (47 ms inference) Most tasks observe higher accuracy or precise semantic effects when combining image and text.

Text Input (Description)	▶	Gemini NLP	▶	TFLite Image	▶	Hybrid Fusion	▶	Category

Fig. 2. Hybrid AI Classification Pipeline

Image Validation Module

The category predicted by the image is cross-verified to the user-selected category. A mismatch will lead to rejection and prompt a re-upload to prevent fraud. The validation resulted in an accuracy of 89.5%; false rejections occurred due to low-resolution images from poor illumination [8].

Duplicate Detection Layer

The three-level pipeline (Fig. 3) result in higher computational cost at each level. Haversine formula calculates geo-distance. Only complaints that pass Steps 1 and 2 are subject to semantic similarity [20], with per-complaint processing remaining below 200 ms [7].

New Complaint Submitted

▼

Step 1: Category Match? (Same complaint category only)

NO → *Skip to UNIQUE path* **YES ↓**

▼

Step 2: Geo-Proximity Filter (Within 50-metre radius — Haversine Formula)

NO → *Skip to UNIQUE path* **YES ↓**

▼

Step 3: Semantic Text Similarity (Gemini AI cosine similarity > 0.85)

▼

All 3 Steps Match?

YES ↓ **NO ↓**

DUPLICATE	**UNIQUE**
Link to parent complaint	Calculate priority score
Increment duplicate count	Assign SLA deadline
Recalculate priority score	Store in Firebase DB
Notify user	Send notification

Fig. 3. Three-Level Duplicate Detection Pipeline.

Priority & Routing Layer

Priority P is calculated as (0.4 • Severity + 0.4 • DupCount + 0.2 • TypeWeight). On a scale from 1 to 5; if total duplicates linked; the critical high medium, weightage 3-2-1. Deadlines SLA Critical 24h, High 48h, Medium 72h, does automatic tracking, overdue escalation alerts.

IV. METHODOLOGY

Figure 4 shows the complete end-to-end flow. The complaints pass through various stages like validation, classification, deduplication, and storage, in that order. Duplicate decision and image-validity decision nodes branch out from these stages.

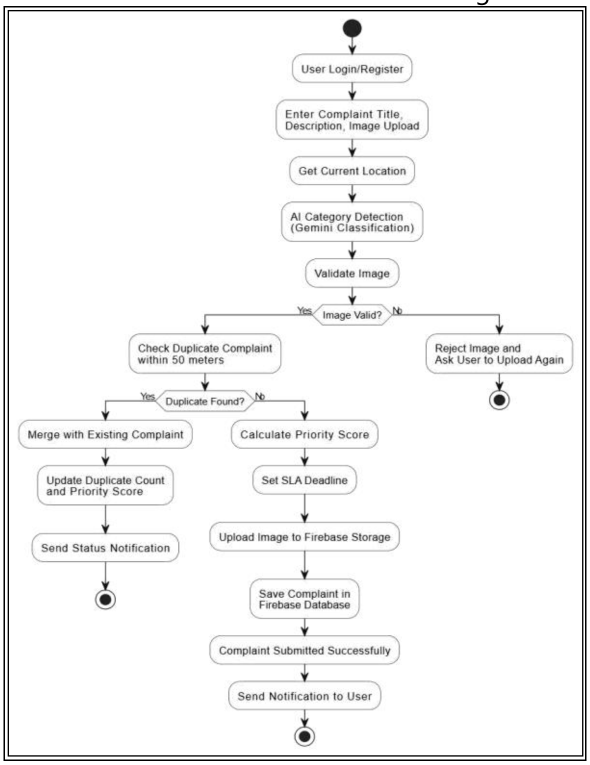

Fig. 4. AGRS-EG System Operational Flow.

Algorithm Description

The user submits a complaint through text, image, and GPS. Gemini NLP and TFLite classify; fusion picks category [7][9]. Image validation, invalid images rejected. The Firebase retains complaints of the same category. Use haversine filters to have 50 m [12]. Gemini similarity on filtered subset; score >0.85 = duplicate [20]. (7a) Duplicate: increase count, recalculate priority, notify user. (7b) Unique: determine priority, assign SLA, store message, send FCM [21]. Reword this (8 words): “Admin dashboard updated in real time”.

Dataset & Training

The TFLite model was trained on 34,927 images in 6 categories. Transfer learning was used to fine-tune MobileNetV2 [10]. The top-1 accuracy of 91.2%, with a 20% held-out test set, was achieved with data augmentation (rotation ±15°, horizontal flip, brightness ±20%).

API Roles

The Gemini API (7) is used (i) for zero-shot type classification by way of structured prompts, (ii) for scoring the semantic similarity between pairs of complaint descriptions (on the lines of Sentence-BERT (20)), and (iii) to power the conversational AI chatbot that assists users. to save on latency and costs, API calls are batched and cached.

V. EXPERIMENTAL RESULTS

Hardware & Software Configuration

Table I provides an overview of the AGRS-EG development testing configuration.

Table I: HARDWARE & SOFTWARE CONFIGURATION

Parameter	Specification
Processor	Intel Core i3 or higher
Memory	4 GB RAM or above
Android Version	Android 8.0 (Oreo) or higher
Internet	2 Mbps minimum bandwidth
Storage	500 GB Hard Disk
IDE	Android Studio, VS Code
Language	Java, XML

Parameter	Specification
Database	Firebase Realtime DB

B. Performance Metrics

Table II: PERFORMANCE EVALUATION RESULTS
Metric AGRS-EG Baseline

Metric	AGRS-EG	Baseline
Text Classification Accuracy	94.7%	81.3%
Image Classification Accuracy	91.2%	78.6%
Hybrid Classification Accuracy	96.3%	83.5%
Duplicate Detection Accuracy	93.8%	67.4%
Image Validation Accuracy	89.5%	N/A
Avg. Processing Time	1.8 sec	12.4 sec
False Duplicate Rate	4.1%	18.7%
Priority Assignment Accuracy	95.2%	N/A
Chatbot Query Resolution	87.6%	N/A
System Uptime (30-day)	99.4%	—

Accuracy Comparison

Table III shows comparisons of metric accuracy. The baseline refers to the combination of keyword matching and Jaccard deduplication. The hybrid classification (96.3%) approach yielded considerable advances as compared to the text-only (81.3%) and image-only (78.6%) approach.

Table III: Accuracy Comparison — AGRS-EG vs. Baseline

Metric	Proposed AGRS-EG	Baseline System	%
Text Classification			94.7%
Image Classification			91.2%
Hybrid Classification			96.3%
Duplicate Detection			93.8%
Image Validation		— No baseline	89.5%
Priority Assignment		— No baseline	95.2%
Chatbot Resolution		— No baseline	87.6%

█ = AGRS-EG (Blue), █ = Baseline (Red)

Processing Time

Table IV indicates the latency before the stage. The manual baseline took 12.4 seconds while the average end-to-end processing took 1.8 seconds. This is an overall 6.9× improvement. According to [7][9], Gemini API (680 ms) is the main contributor to costs, TFLite adds only 47 ms via on-device inference [7][9].

Table IV. Processing Time Breakdown (seconds)
Stage Time (0–14 sec scale) Sec

Stage	Time (0–14 sec scale)	Sec
Gemini API (NLP)		0.68

Stage	Time (0–14 sec scale)	Sec
TFLite Inference		0.05
Image Validation		0.12
Duplicate Detection		0.20
Firebase Write		0.45
FCM Notification		0.30
AGRS-EG Total		1.80
Manual Baseline		12.4

12.4

Green=sub-stages, Blue=AGRS-EG total (1.8s), Red=Manual baseline (12.4s).

Color bars represent each category's proportion of total submissions.

Duplicate Detection Analysis

The three-tiered pipeline achieved a duplicate detection accuracy of 93.8%. The location filter (Step 2) removed 71.4% of the candidate-category matches, reducing the number of similarity computations by approximately 70%. The false duplicate rate of 4.1%, where different issue with the same location uses near-identical wording, can be brought down by tightening the threshold from 0.85 to 0.90 at small recall loss.

System Latency & Uptime

More than one thousand timed trials have been conducted: median end-to-end latency is 1.8 seconds while 95th percentile is 2.4 seconds. Gemini API averages 680 ms, TFLite 47 ms, Firebase write 450 ms, and FCM 300 ms. Overall, the system uptime over 30 days was 99.4%, with a 0.6% downtime cause by Firebase scheduled maintenance [8][21].

VI. COMPARATIVE ANALYSIS

Table V shows the comparison of AGRS-EG against the best capacity of earlier systems [1]-[5] in 4 columns. AGRS-EG is the sole system providing all six innovations simultaneously. The most impactful differentiators are: (i) location-aware duplicate detection via Haversine [12], not performed by any prior system; (ii) the chatbot [7], resolving 87.6{e2 80 93} of queries without human intervention; and (iii) dynamic priority, ensuring community-impact-weighted issue escalation [25].

TABLE V. AGRS-EG VS. EXISTING SYSTEMS

Feature AGRS-EG Best Prior Gap?

Feature	AGRS-EG	Best Prior	Gap?
Hybrid Classification (T+I)	96.3%	Single-modal	Yes
Location-aware Dup. Detect.	93.8%	Text-only [4]	Yes
Image Validation	89.5%	Partial [3]	Yes
Dynamic Priority	95.2%	Severity only	Yes
AI Chatbot	87.6%	None	Yes
Map Hotspot Visualization	✓ Google Maps	Partial [3]	Yes
Real-time Notifications	✓ Firebase FCM	Partial	Yes
Avg. Processing Time	1.8 sec	12.4 sec	6.9× faster
False Duplicate Rate	4.1%	18.7%	4.5× lower
System Uptime (30-day)	99.4%	N/A	Yes

Routing issues have been dealt with more efficiently in [1][2][3]. However, these systems do treat each complaint independently. AGRS-EG's priority of being aware of community impact is a paradigm shift from grievance management of an individual to issue intelligence of a collective [25].

VII. DISCUSSION

A. Strengths

Only about 5% of complaints proceed to the costly similarity stage in the cascaded duplicate detection pipeline. TFLite inference on-device [9][10] avoids image upload round-trips and remains responsive on 2G/3G networks, critical for M-Governance

applications in semi-urban India. The chatbot resolves 87.6% of help-desk queries independently.

B. Limitations

Using firebase will require the internet, so no offline use. The copy radius of 50 meters may be too constricted in high-density areas of differing bona fides. Model accuracy degraded to ~78% if trained only on daytime images in TFLite [10][23]. The current version does not support voice input or multiple languages.

Scalability

The Firebase Realtime Database scales horizontally to handle concurrent access without the need for changes on the server side [8]. Gemini API rate limits are the overriding bottleneck with >10,000 complaints/day; future work will look at local LLM deployment or API request batching [7] The existing architecture can support up to 5000 concurrent users on a standard Firebase plan.

CONCLUSION

This paper proposed a fully integrated AI-powered autonomous grievance redressal system, AGRS-EG. It produced hybrid classification accuracy of 96.3%, duplicate detection accuracy of 93.8%, image validation accuracy of 89.5%, and autonomous chatbot resolution accuracy of 87.6%. It also had a system uptime of 99.4%. The average processing time has improved 6.9× that of the manual baseline. AGRS-EG platform, through grievance management considered as a collective intelligence problem wherein duplicate counts escalate community-impact priority, establishes a new AI e-governance benchmark. The modular architecture of the system allows it to be gradually deployed in existing municipal governance portals. In the future, the following will be explored: (i) complaint submission through voice in multiple languages, (ii) predictive hotspot analytics using historical data [22], (iii) audit trail immutability using blockchain, (iv) automatic complaint generation from IoT-based smart city devices, and (v) multilingual NLP using mBERT [11] for Indian regional languages.

Acknowledgements

The authors sincerely thank the J.C.O.E. Computer Engineering Department. Google is acknowledged for providing API for google map and Firebase services used in this research.

REFERENCES

1. Suganya S., Kirubakaran N., and Senthil Kumar S., "Autonomous Grievance Redressal System for Government Services," in Proc. IEEE ICDSBS, 2025.
2. Sidharth G., Vasra A., and Deisy C., "Automation of Grievance Registration Using Transfer Learning," in Proc. IEEE ICIPTM, 2023.
3. Balakrishnan S., Janet J., Rohith T., and Sakthivel R., "Online Complaint Management System using Image Recognition," in Proc. IEEE ICCES, 2023.
4. Joy S. P., Vishwa Vignesh N. K., Shreyas K. N., and Roy A. L., "Smart Grievance System using BERT and Keyword Clustering," in Proc. IEEE SMART GENCON, 2021.
5. Ramesh K., Nair P., and Sharma V., "AI-Driven Public Complaint Platform using SVM and Random Forest," J. Comput. Eng. Appl., vol. 14, no. 3, pp. 45-58, 2022.
6. Chen X., Liu Y., and Wang H., "Mobile Grievance Application with LSTM-based NLP for Municipal Governance," IEEE Access, vol. 8, pp. 112233-112245, 2020.
7. Google LLC, "Gemini API Documentation," Google AI, 2024. [Online]. Available: https://ai.google.dev
8. Google LLC, "Firebase Realtime Database Documentation," 2024. [Online]. Available: https://firebase.google.com
9. TensorFlow Team, "TensorFlow Lite — On-Device Machine Learning," Google, 2024. [Online]. Available: https://www.tensorflow.org/lite
10. Howard A. G. et al., "MobileNets: Efficient CNNs for Mobile Vision Applications," arXiv:1704.04861, 2017.
11. Devlin J., Chang M.-W., Lee K., and Toutanova K., "BERT: Pre-training of Deep Bidirectional Transformers," in Proc. NAACL-HLT, 2019, pp. 4171-4186.
12. Sinnott R. W., "Virtues of the Haversine," Sky Telescope, vol. 68, no. 2, p. 159, 1984.
13. Vaswani A. et al., "Attention Is All You Need," in Proc. NeurIPS, 2017, pp. 5998-6008.
14. Cortes C. and Vapnik V., "Support-vector networks," Mach. Learn., vol. 20, no. 3, pp. 273-297, 1995.
15. Redmon J. et al., "You Only Look Once: Unified Real-Time Object Detection," in Proc. IEEE CVPR, 2016, pp. 779-788.
16. Breiman L., "Random Forests," Mach. Learn., vol. 45, no. 1, pp. 5-32, 2001.

17. Hochreiter S. and Schmidhuber J., "Long Short-Term Memory," Neural Comput., vol. 9, no. 8, pp. 1735-1780, 1997.
18. Google LLC, "Google Maps Platform Documentation," 2024. [Online]. Available: https://developers.google.com/maps
19. Zhang Y., Jin R., and Zhou Z.-H., "Understanding bag-of-words model," Int. J. Mach. Learn. Cybern., vol. 1, pp. 43-52, 2010.
20. Reimers N. and Gurevych I., "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. EMNLP, 2019, pp. 3982-3992.
21. Google LLC, "Firebase Cloud Messaging," 2024. [Online]. Available: https://firebase.google.com/docs/cloud-messaging
22. Agrawal R. and Srikant R., "Fast algorithms for mining association rules," in Proc. VLDB, 1994, pp. 487-499.
23. LeCun Y., Bengio Y., and Hinton G., "Deep learning," Nature, vol. 521, pp. 436-444, 2015.
24. [Mikolov T. et al., "Distributed representations of words and phrases," in Proc. NeurIPS, 2013, pp. 3111-3119.
25. MeitY, "National e-Governance Plan (NeGP)," Govt. of India, 2023. [Online]. Available: https://meity.gov.in