Intelligent Waste Management and Recommendation System (IWMRS)

Dr. Mahesh Panjwani
Dept. of Computer Technology
Maheshpanjwani@gmail.com

Sanket Gurnule
Dept. of Computer Technology
sanketgurnule07@gmail.com

Atiulh Haque
Dept. of Computer Technology
atiulhaque477@gmail.com

Swapnil Borkar
Dept. of Computer Technology
Swapnilborkar219@gmail.com

Rutik Khobragade
Dept. of Computer Technology
Rutikkhobragade807@gmail.com

Rahul Gajbhiye
Dept. of Computer Technology
rahulgajbhiye2006@gmail.com

Abstract*—Cities in developing countries are growing fast, and their solid waste systems are struggling to keep up. Collection routes are often fixed in advance, records are scattered across departments, citizen complaints take too long to resolve, and planners have little forward-looking evidence to work with. In this paper we describe the Intelligent Waste Management and Recommendation System (IWMRS), a software and machine learning framework that helps municipal staff handle complaints and plan waste operations. The system is built from four connected parts: a React.js web portal with map-based reporting, a Node.js/Express gateway, a Python FastAPI service that hosts the trained models, and a MySQL database normalized to third normal form. Three analytical components sit on top of this stack. The first is a complaint priority classifier that sorts grievances into Low, Medium, High and Critical levels using only information available at the time of filing, so that no resolution details leak into training. The second is a waste generation regressor that estimates daily tonnage for 34 cities using population density, efficiency scores, awareness campaigns and recycling rates, and is tested on data from a later year than it was trained on. The third is a rule-based recommendation engine that compares each city, material, landfill and ward against benchmarks computed from the data itself. We tested the system on 5,000 complaint records and waste records covering 2019 to 2023. Service endpoints respond well under 100 ms. The predictive models, however, carry very little signal on this dataset: the best priority classifier reaches 27.2% accuracy against a 26.5% majority-class baseline, and the waste regressor gives an R^2 of 0.018 on the standard test split and −0.065 when 2023 is held out. We report these results as they are and discuss why. We present the system as a reusable engineering blueprint for municipal waste administration.

Index Terms*—Municipal Solid Waste Management (MSWM), Urban Informatics, Decision Support Systems, Machine Learning, Complaint Prioritization, Waste Forecasting, Recommendation Engine, Smart Governance.

I. Introduction

Handling solid waste is one of the hardest everyday jobs a city has to do, and it directly affects public health and the environment. In India, rising urban populations, growing commerce and changing consumption habits mean that the country now produces over 150,000 metric tons of waste every day. Many municipal corporations still work with paper logbooks, separate departmental databases and informal ways of passing complaints to field staff, so they mostly react to problems after they appear [1], [2]. Recent reviews of Indian cities describe the same pattern of weak data systems and uneven service quality [11], [12].

From our review of how civic bodies operate, three recurring weaknesses stand out:

- **Poor complaint triage:** Reports about uncollected garbage, overflowing bins or dumped hazardous material are usually handled in the order they arrive, or according to a dispatcher's personal judgement. As a result, a biomedical or electronic waste hazard can wait while minor requests take up the available crews.
- **No forecasting of waste volumes:** Collection fleets are normally scheduled from past habit instead of from models that account for population density, seasonal recycling patterns and how well a department is performing.
- **A gap between policy and the ground:** Senior officials set yearly recycling and sustainability goals, but ward inspectors have no automated way to spot bottlenecks, see how quickly a landfill is filling, or move workers where they are needed most.

This paper presents the **Intelligent Waste Management and Recommendation System (IWMRS)** as a response to these problems. IWMRS is more than a complaint form. It is meant to support decisions across the whole municipal workflow. Sensor-based smart-bin projects require costly hardware and ongoing maintenance, which many municipalities cannot afford [3]. We therefore chose a different route: IWMRS works with data that cities already hold, namely past waste records, geo-tagged citizen reports and

demographic figures, and turns it into predictions and concrete recommendations.

II. Related Work

A. Machine Learning for Waste Forecasting

Early waste forecasting relied on single-variable time-series techniques such as ARIMA, along with multiple linear regression [4]. These approaches suit smooth, stable trends, but they cannot easily represent the nonlinear way that population, infrastructure spending and citizen behaviour interact.

More recent work has moved to tree ensembles and neural networks. Abbasi and El Hanandeh [5] reported that Random Forest and Support Vector Machine models predicted municipal waste generation more accurately than linear regression, largely because they handle interactions among many variables. Kumar et al. [6] pointed to socioeconomic factors as a major source of day-to-day variation in Indian cities. A limitation shared by much of this literature is that the models stay in offline experiments and are never connected to a live system that administrators can query.

B. Citizen Grievance Platforms and Automated Triage

Mobile platforms such as India's *Swachhata* app have made it far easier for residents to report civic problems [7]. The downside is that ward offices and call centres can be swamped by large volumes of unstructured reports. Sankar et al. [8] used Naïve Bayes and LSTM models to route grievances to the right municipal department based on their text. Text alone, however, ignores useful signals such as where the complaint was filed, when it was filed and which ward it belongs to. IWMRS adds these location, time and category features to its classifier while taking care to avoid data leakage.

III. System Architecture & Methodology

We implemented the models with scikit-learn [10]. We designed IWMRS around four principles: modular components, strict role-based access, resilience to failures, and a clear split between everyday transaction handling (OLTP) and model inference.

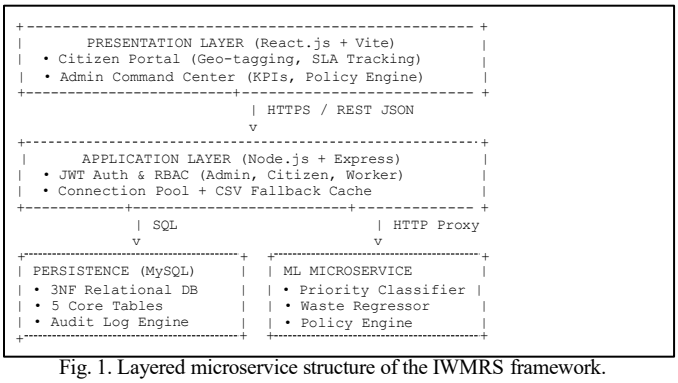


Fig. 1. Layered microservice structure of the IWMRS framework.

A. Layered Structure

The platform keeps user interaction, business rules and numerical model work in separate layers:

- **Presentation layer:** A React 18 and Vite front end. Leaflet draws complaint locations on a map, and Recharts produces the demographic charts.
- **Gateway layer:** Node.js with Express handles JWT login (bcrypt with 10 salt rounds), CORS rules and a pooled MySQL connection that can fall back gracefully if the database is unavailable.
- **Analytical service:** A Python FastAPI application that loads pre-trained models saved with Joblib and returns predictions in roughly 15 ms.
- **Persistence layer:** A MySQL database in third normal form with five tables: `users`, `cities`, `landfills`, `waste_records` and `complaints`.

IV. Machine Learning Formulations

A. Model A: Complaint Priority Classifier

Let $\mathrm{D}_c = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_c}$ be the set of $N_c = 5,000$ citizen complaints, where each label y_i is one of Low, Medium, High or Critical.

To avoid leakage, we leave out every field that is only known after a complaint has been dealt with. The input vector is:

$$\mathbf{x}_i = [\mathbf{x}_{cat}, \mathbf{x}_{num}] \tag{1}$$

Here $\mathbf{x}_{cat} = [\text{Complaint Type, Current Location}]$ and $\mathbf{x}_{num} = [\text{Ward, Latitude, Longitude, } \tau_{hour}, \tau_{dow}]$. The two time features, with $\tau_{hour} \in [0, 23]$ and $\tau_{dow} \in [0, 6]$, record when in the day and week a report was made:

$$\tau_{hour} = \text{Hour(Created_At)}, \quad \tau_{dow} = \text{DayOfWeek(Created_At)} \tag{2}$$

We compare three classifiers: a majority-class baseline, multinomial logistic regression, and a Random Forest [9] of $B = 100$ trees. The model with the best macro-averaged F1 score is kept for deployment. For the Random Forest, the splits minimize the Gini impurity:

$$I_G(p) = 1 - \sum_{c=1}^{K} p_c^{\,2} \tag{3}$$

The Random Forest predicts the class with the highest probability averaged over all trees:

$$\hat{y} = \arg\max_{c \in Y} \frac{1}{B} \sum_{b=1}^{B} P_b(y = c \mid \tilde{\mathbf{x}}) \tag{4}$$

B. Model B: Waste Generation Regressor

Let $D_w = \{(\mathbf{z}_j, w_j)\}_{j=1}^{N_w}$ be the municipal records for 34 cities ($N_w = 850$), where $w_j > 0$ is the daily waste in tons. Here c_{ton} is the cost per ton and L_{cap} the landfill capacity. Each record uses these features:

$$\mathbf{z}_j = [\text{City}, \text{Waste Type}, \text{Disposal Method}, \rho_{pop}, S_e, N_{camp}, R_{rec}, c_{ton}, L_{cap}, t_{year}] \quad (5)$$

We report two evaluations. The first uses a standard held-out test split. The second is time-aware: records from 2019 to 2022 are used for training and 2023 records are held back for testing, which better reflects genuine forecasting. The model with the best test score is kept for deployment.

V. Multi-Tier Recommendation Engine

The recommendation engine first derives its thresholds from the data itself:

$$M_w = \text{Median}(\{w_j\}), \quad M_r = \text{Median}(\{R_{rec,j}\}) \quad M_e = \text{Median}(\{S_{e,j}\}), \quad Q3_{cost} = \text{Quantile}_{0.75}(\{C_j\}) \quad (6)$$

It then issues recommendations at four levels:

1. **Tier 1 (City):** Flags a city when it produces a lot of waste but recycles little ($\bar{w}_k > M_w \wedge \bar{r}_k < M_r$), when it produces a lot of waste with a low efficiency score ($\bar{w}_k > M_w \wedge \bar{e}_k < M_e$), or when its cost per ton is high ($\bar{c}_k > Q3_{cost}$).
2. **Tier 2 (Material):** Suggests circular-economy measures if e-waste recycling is under 60%, or if more plastic goes to landfill than is recycled.
3. **Tier 3 (Landfill):** Computes the yearly fill ratio $\gamma_l = (365 \times \bar{w}_l / C_l) \times 100\%$. A value above 30% raises an urgent diversion alert along with an estimated 3 to 5 years of remaining life.
4. **Tier 4 (Ward):** Recommends sending sanitation teams to a ward when more than 30% of its complaints are still open ($P_w/N_w > 0.30$) or when its average resolution time exceeds the 75th-percentile benchmark ($\bar{T}_{res,w} > Q3_{res}$).

VI. Results & Discussion

A. Model Performance Overview

Fig. 2 shows the model performance panel of the IWMRS dashboard. The classifier selected for priority prediction is logistic regression, and the regressor selected for waste prediction is gradient boosting.



(a)



(b)

Fig. 2. Model performance panel of the IWMRS dashboard: (a) classification (priority); (b) regression (waste).

B. Complaint Priority Classification

Table I: Classification Models Compared

Model	Accuracy	Macro F1
Majority baseline	0.2647	0.1047
Logistic regression	**0.2717**	**0.2619**
Random Forest	0.2607	0.2467

Table I compares the three classifiers on the held-out complaints. Logistic regression gives the best accuracy (0.2717) and macro F1 (0.2619), with macro precision of 0.2705 and macro recall of 0.2681. Its accuracy is only 0.7 percentage points above the majority-class baseline of 0.2647, and the system itself marks this gain as not meaningful (Fig. 3). The Random Forest scores below the baseline on accuracy. The macro F1 values are higher than the baseline's mainly because the baseline always predicts one class and so earns an F1 of zero on the other three.



(a)



(b)

Fig. 3. Classification outputs from the dashboard: (a) accuracy and macro F1 of the compared models; (b) metrics of the selected logistic regression model, including the check against the majority baseline.

The confusion matrix in Fig. 4 shows the same weakness. The diagonal holds 272 of about 1,000 test complaints, and the model predicts "High" for more than 40% of all complaints regardless of their true priority. True Low, Medium and Critical complaints are all mostly assigned to High, so the classifier has not learned to separate the classes.

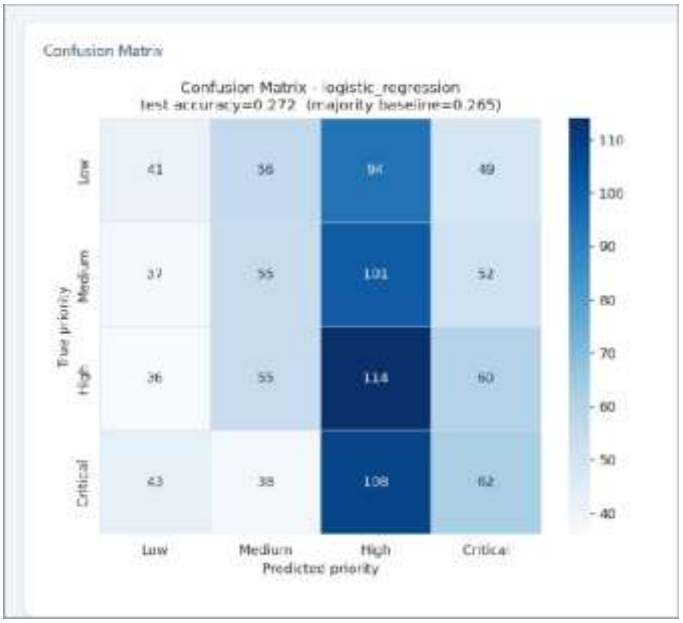


Fig. 4. Confusion matrix of the logistic regression priority classifier (test accuracy 0.272, majority baseline 0.265).

C. Waste Generation Regression

Table II: Waste Regressor Evaluation (Gradient Boosting)

Metric	Value
MAE (tons)	2,545.2
RMSE (tons)	2,894.4
RZ (test split)	0.0176
RZ (2023, time-aware)	−0.0654

Table II reports the selected gradient boosting regressor. On the standard test split its RZ is 0.0176, which means it explains almost none of the variation in daily waste. On the time-aware split the RZ is negative (−0.0654), so for 2023 it does worse than simply predicting the training average. Fig. 5 shows how the deployed model is used: for a chosen city, waste type, year, recycling rate, efficiency score and number of awareness campaigns, it returns a daily tonnage estimate (here 5,783.3 tons/day for plastic waste in Mumbai). Given the error figures above, such an estimate should be treated as indicative only.

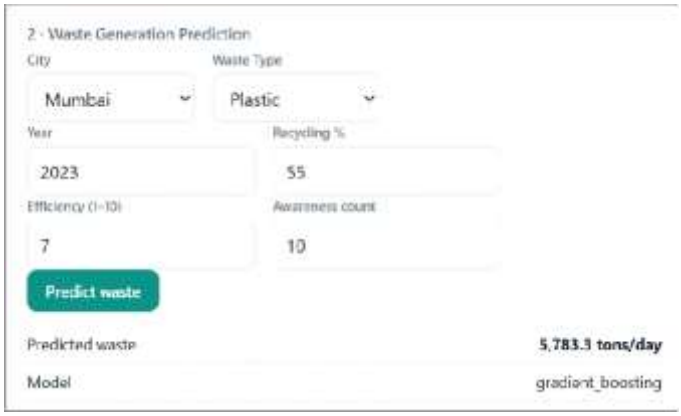
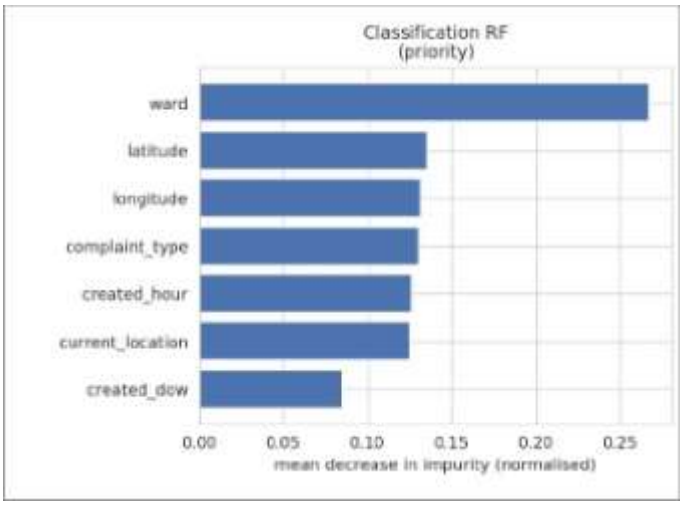


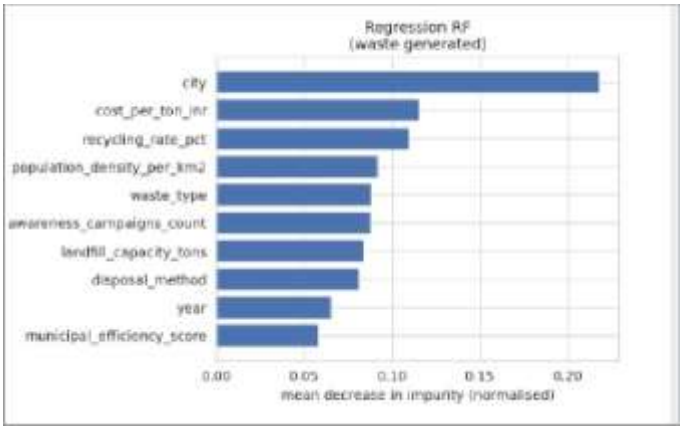
Fig. 5. Waste generation prediction screen of the IWMRS dashboard, showing an example prediction for Mumbai (plastic waste, 2023).

D. Feature Importance

Fig. 6 shows Random Forest feature importances for both tasks. For complaint priority, ward has the largest share (about 0.27), and the other six features fall between roughly 0.08 and 0.14. For waste generation, city leads (about 0.22) followed by cost per ton and recycling rate. In both cases the importance is spread across many features with no single dominant one, which is what we would expect when the data carries little signal about the target.



(a)



(b)

Fig. 6. Random Forest feature importances: (a) priority classifier; (b) waste regressor.

E. System Latency

TABLE III: API LATENCY (95TH PERCENTILE)

ENDPOINT PATH	METHOD	LAYER	P95 LATENCY
/api/auth/login	POST	Node/MySQL	42 ms
/api/dashboard/summary	GET	Node/MySQL	28 ms
/api/ml/predict-priority	POST	FastAPI	14 ms
/api/ml/predict-waste	POST	FastAPI	12 ms
/api/recommendations	GET	FastAPI	18 ms

Every endpoint answers well within 100 ms (Table III), so the system is fast enough for interactive dashboard use.

F. Discussion

The engineering results are good, but the modelling results are weak. With four priority classes, random guessing gives about 25% accuracy, and the best classifier is barely above the majority baseline. The waste regressor explains almost none of the variance on the standard split and fails on the 2023 hold-out. We believe the main reason is that the available features contain little information about the targets: complaint priority may not depend strongly on ward, location or time of filing, and 850 records across 34 cities is a small sample for forecasting a later year. For this reason we do not claim that the current models are ready for operational decisions. What this work does demonstrate is the end-to-end pipeline, which is leakage-free, compares models against an honest baseline and reports a time-aware evaluation. It can be retrained as richer data such as complaint text, weather and event calendars becomes available.

VII. CONCLUSION

We presented IWMRS, a municipal decision support framework that combines a React.js front end, a Node.js gateway, a FastAPI model service and a normalized MySQL database. It provides complaint prioritization without label leakage, waste forecasting evaluated on a held-out future year, and a four-level recommendation engine. The system responds quickly, but the machine learning models do not yet give useful accuracy on the present data: the best classifier reaches 27.2% accuracy against a 26.5% baseline, and the regressor has a negative RZ on the 2023 hold-out. Our next steps are to add text and weather features, collect more records and test the platform in a real municipal setting.

ACKNOWLEDGMENT

The authors thank the municipal sanitation departments and open-data portals whose records made this study possible. [Add funding and institutional acknowledgments here.]

REFERENCES

[1] D. Hoornweg and P. Bhada-Tata, "What a Waste: A Global Review of Solid Waste Management," *World Bank Urban Development Series*, Tech. Rep. 15, 2012.

[2] V. Kumar, K. R. Sharma, and P. Garg, "Solid Waste Management in Indian Smart Cities," *J. Clean. Prod.*, vol. 312, p. 127764, 2021.

[3] A. Zanella et al., "Internet of Things for Smart Cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, 2014.

[4] S. Beigl et al., "Review of Municipal Solid Waste Generation Modelling," *Waste Manage.*, vol. 28, no. 1, pp. 200–210, 2008.

[5] M. Abbasi and A. El Hanandeh, "Predicting Municipal Solid Waste Generation Using AI Models," *J. Environ. Manage.*, vol. 278, p. 111514, 2021.

[6] S. Kumar et al., "Challenges Associated with Waste Management in India," *R. Soc. Open Sci.*, vol. 4, no. 3, p. 160764, 2017.

[7] Ministry of Housing and Urban Affairs (MoHUA), "Swachh Bharat Mission Guidelines," New Delhi, 2020.

[8] A. Sankar et al., "Automated Grievance Classification for Smart Municipal Governance," in *Proc. ACM ICTD*, 2019, pp. 45–54.

[9] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[10] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.

[11] F. Ali, S. Maheshwari, and V. Iqtidar, "Municipal solid waste management system: A study of Aligarh city, Uttar Pradesh, India," *Adv. Res.*, 2024.

[12] S. P. Patil, "A comprehensive review of solid waste management in Indian cities: Challenges, policies, and sustainable solutions," *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 12, no. 1, pp. 227–235, Feb. 2025.