

Civic Complaint Priority Management System

Using a Hybrid NLP and Location-Aware Machine Learning Architecture

K. Karunakar

Department of Computer Science and Engineering
Sreenidhi Institute of Science and Technology
Hyderabad, India

K. Sathvik

Department of Computer Science and Engineering
Sreenidhi Institute of Science and Technology
Hyderabad, India

S. Anjan Saketh

Department of Computer Science and Engineering
Sreenidhi Institute of Science and Technology
Hyderabad, India

K. Vishwa Teja

Department of Computer Science and Engineering
Sreenidhi Institute of Science and Technology
Hyderabad, India

Abstract - Municipal government has many problems when it comes to manually managing citizen complaints. Issues include inconsistent prioritization of complaints, lack of timely responses to urgent events, and limitations on scale under a heavy volume of incoming submissions. This paper describes a Civic Complaint Priority Management System that enables the automated classification and priority assignment of citizen-generated complaints through the use of a hybrid Natural Language Processing (NLP) and location aware machine learning architecture. The Civic Complaint Priority Management System uses a two-level priority inference pipeline: a fuzzy rule based matching engine using a curated domain specific dictionary (RapidFuzz at 80% threshold) as its primary classifier and a TF-IDF and Logistic Regression as the fallback machine learning pipeline for unclassified complaints. The priority assignment process integrates a weighted composite score that combines: (1) the severity of the text of the complaint (70% weight), and (2) the density of geographic locations of the complaints derived from the OpenStreetMap (Overpass API) (30% weight), to produce three levels of operational priority (P1 = Critical, P2 = Moderate, and P3 = Low). The system is designed as a Full-stack Web Application with citizen and administrator dashboards, complaint status tracking, and routing of complaints to appropriate departments. Evaluation using a holdout set of complaints produced overall classification accuracy of 91.3%, with a P1 class recall of 94.2%, indicating that the system exhibits a high degree of sensitivity to high severity civic incidents. The proposed hybrid architecture outperforms pure ML and pure rule-based baselines on minority-class recall, validating the design of the two classification stages.

Keywords - Complaint Priority Classification, Natural Language Processing, TF-IDF; Logistic Regression, Fuzzy Matching, Civic Tech, Location-Aware Classification, Overpass API, Flask, Complaint Management System

I. INTRODUCTION

In contemporary urban governance, managing citizen complaints represents an essential function of delivering services. A citizen may submit a report about a civic issue from a life-threatening situation such as a building collapse, gas leak, or flood to a non-urgent civic issue like general queries or lost property. To successfully operate emergency response and

routine maintenance functions, municipalities require the accurate and timely identification of incoming complaints to ensure they are directed to the appropriate department and resources are allocated based on an urgency of need.

Traditional systems for managing citizen complaints rely significantly on human triage by administrative employees, creating three types of issues within the process; inconsistent assessments of the severity of the complaints being triaged by human assessors; inability to scale manual reviews of complaints as the number of complaints submitted increases, creating a bottleneck in initiating responses to critical incidents; lack of real-time contextual information, such as geographic population density of the area being assessed, which has a significant impact on predicting the impact of complaints made regarding infrastructure issues or public safety.

By utilizing Natural Language Processing (NLP) as the mechanism for automating the semantic understanding of free-text complaint descriptions, in addition to Geographic Information Systems (GIS) for providing the necessary contextual information required to correctly assess the severity of the complaints being evaluated, municipalities can automate the complaint management system. This paper presents a Civic Complaint Priority Management System (CCPMS) that integrates both dimensions into a unified, operationally deployable web application, demonstrating that hybrid rule-based and machine learning classification significantly outperforms either approach alone.

A. Problem Statement

This study's primary research problem relates to automating the assignment of operational priority tiers (OT) to free-text civic complaints. In addition to reflecting the severity (intensity) of complaint content, the OT must reflect the geographic context (scope) of that intensity. A complaint of a pothole in a densely populated urban commercial area should be assigned a higher OT than similar complaints in low-density rural areas; because of the larger number of residents impacted (scope) and the potential for future incidents because of damage to other property or injury, respectively.

Current complaint routing systems utilize rule-based assignments that are not sensitive to the contextual nature of complaints. Further, traditional machine learning classifiers do not possess the domain vocabulary consistency required for distinguishing between Critical and Moderate complaints in the civic safety domain.

B. Research Contributions

The primary contributions of this work are:

- Design and implementation of a two-stage hybrid complaint classifier combining rule-based fuzzy dictionary matching (RapidFuzz) with a TF-IDF and Logistic Regression ML fallback pipeline, achieving 91.3% overall accuracy on civic complaint classification.
- Development of a location-aware priority scoring function that integrates complaint text severity scores with real-time geographic Point-of-Interest density from the OpenStreetMap Overpass API, with configurable weighting (70:30).
- Deployment of the full system as a production-ready Flask web application with role-based access control for citizens and administrators, complaint status tracking, department routing, and SQLite persistence.
- Empirical evaluation demonstrating that the hybrid architecture achieves 94.2% P1-class recall, substantially outperforming pure TF-IDF ML (82.1%) and pure rule-based (88.6%) baselines on critical complaint detection.

II. LITERATURE SURVEY

Traditional complaint management systems in urban governance have primarily relied on database-driven routing mechanisms without any form of automated classification or prioritization. Afolabi et al. demonstrated the deployment of such rule-based complaint management systems for urban governance, highlighting their operational utility while also exposing their limitations in scalability, adaptability, and intelligent prioritization as complaint volumes grow in modern municipal environments [1].

Early research in text classification for complaint and opinion-based data established the foundational use of term-frequency representations and supervised learning. Pang and Lee provided a comprehensive survey of opinion mining and sentiment analysis techniques, laying the groundwork for applying natural language processing to subjective text such as user complaints, and demonstrating the viability of lexical feature extraction for short-text classification tasks [2].

Subsequent studies specifically examined the application of text classification to complaint routing in service-oriented systems. Kataria and Singh conducted a comprehensive study of text classification techniques applicable to complaint routing, evaluating approaches ranging from Bag-of-Words with Naive Bayes to Support Vector Machines with TF-IDF features, and observed that while such methods achieve reasonable accuracy on constrained datasets, they degrade

significantly when faced with domain-specific jargon or unseen terminology [3].

To address the brittleness of exact-match keyword dictionaries in service routing, fuzzy string matching has been proposed as a robust alternative. Bachmann and Ott examined the application of fuzzy string matching for short-text classification in service routing, demonstrating that edit-distance-based similarity scoring using libraries such as RapidFuzz can substantially extend the coverage of rule dictionaries to near-match and misspelled complaint descriptions without requiring exhaustive manual vocabulary maintenance [4].

The integration of geographic context into complaint classification and priority estimation represents an important advancement for civic applications. Vaidya and Bhatt proposed a location-aware priority scoring framework for urban complaint management leveraging GIS data, showing that incorporating spatial features such as population density, proximity to critical infrastructure, and historical incident frequency significantly improves priority classification accuracy for public safety complaint datasets [5].

Hybrid architectures combining rule-based engines with learned machine learning classifiers have been explored to overcome the individual limitations of each approach. Wang et al. proposed a hybrid rule-based and machine learning classification system for customer complaint routing, demonstrating that a cascade design — wherein the rule engine handles high-confidence known categories and the ML model resolves residual unmatched inputs — consistently outperforms either component in isolation, particularly for minority complaint class detection [6].

Among machine learning algorithms suitable for complaint classification, Logistic Regression paired with TF-IDF feature representations has been shown to offer a strong balance of accuracy, probability calibration, and interpretability. Pedregosa et al. introduced Scikit-learn, the widely adopted Python machine learning library that provides efficient implementations of Logistic Regression and TF-IDF vectorization, and which serves as the computational backbone for the ML classification component in the proposed system [7].

For the development of the web-based interface enabling non-technical municipal personnel to interact with the classification system, a lightweight and extensible backend framework is essential. Grinberg provided a comprehensive treatment of Flask-based web application development in Python, establishing Flask as a widely adopted micro-framework for building RESTful APIs and full-stack interfaces that integrate seamlessly with Python-based machine learning pipelines [8].

Real-time geographic density scoring in the proposed system relies on open-source spatial data retrieval rather than proprietary GIS infrastructure. The OpenStreetMap Foundation's Overpass API enables structured querying of Point-of-Interest density, road network data, and infrastructure

proximity from the OpenStreetMap database, providing a cost-free and openly accessible alternative to commercial spatial data services for municipal complaint priority estimation [9].

Deep contextual language models have significantly advanced the state of the art in text classification by capturing bidirectional semantic dependencies across complaint descriptions. Devlin et al. introduced BERT, a pre-trained deep bidirectional transformer model that achieves superior performance on a wide range of NLP benchmarks, and which represents a natural direction for future enhancement of the classification component beyond the TF-IDF and Logistic Regression baseline employed in the current system [10].

Distributed word representation models provide an alternative approach to encoding semantic similarity between complaint terms in low-dimensional vector spaces. Mikolov et al. proposed efficient methods for learning word vector representations from large text corpora, and the resulting Word2Vec embeddings have been widely applied to short-text classification tasks to capture semantic proximity between complaint keywords that share contextual meaning but differ in surface form [11].

Interpretability of automated classification systems is a critical concern for deployment in civic complaint management, where accountability and transparency are required. Ribeiro et al. proposed LIME, a model-agnostic explanation framework that identifies the input features most responsible for any individual classifier prediction, and which can be applied to the proposed system to provide municipal operators with human-readable justifications for automated priority assignments [12].

III. SYSTEM ARCHITECTURE AND DESIGN

A. Overall Architecture

The Civic Complaint Priority Management System was initially created as an independent full stack web application composed of three different tiers: (i) a static HTML/CSS/JavaScript front-end which separates the citizen and the admin interface; (ii) a Python Flask RESTful API back-end used for authentication, storing complaints, and performing ML classification of closed cases; and (iii) an SQLite database used to store complaints, users, and complaints' status.

The ML classifications from the back-end will be provided through a modular python service (`priority_classifier.py`) which will be loaded at the time of the server's start-up. This will enable the back-end to be able to support different classification methods independently of the web layer.

The back-end contains authenticated REST endpoints for submitting complaints, obtaining status of submitted complaints, filtering complaints by administrator purposes, and updating the complaint's priority level. Role based access control is implemented through JWT authentication, which enables a citizen to submit complaint(s) and retrieve his/her history of submitted complaint(s) while allowing an administrator to retrieve all submitted complaints with their associated status(es), sort, and filter the complaints based on the current assignment to them.

The SQLite database will contain a record of every submitted complaint, which includes a UUID for the primary key, timestamp of submission, GPS location, department, status, and computed priority score.

B. Complaint Classification Pipeline

The classification pipeline uses five stages to classify all incoming complaint descriptions. Stage 1 is dictionary setup. It brings in a merged priority mapping dictionary that contains more than 80 different terms representing complaint categories. Each of the complaint categories in the dictionary is organized into a three-tiered priority system: P1 (Critical), P2 (Moderate), and P3 (Low). The P1 tier contains emergency and life-safety complaint categories such as Gas Leak, Collapsed Building, Fire Emergency, Flooding and Explosion. The P2 tier contains significant but not life-threatening complaint categories such as Noise Pollution, Illegal Parking and Traffic Signal Condition. The P3 tier contains low urgency complaint categories such as Lost Property, General Inquiry and Miscellaneous.

The second stage is rule-based priority assignment. The complaint text is compared to all entries in the dictionary through the use of RapidFuzz token-set ratio matching. In order for a match to exist, it must meet or exceed an 80% similarity threshold. If a match exists that meets this similarity threshold, the associated priority tier will be returned along with a human-readable explanation string indicating the matched category and similarity score. The second stage provides a deterministic process to classify a large percentage of clear complain categories quickly and efficiently without the use of the machine learning model.

The third stage of processing is the machine learning fallback classification of complaints which did not reach the threshold of 80% matched using rule-based matching. Complaint descriptions that fall into this category are processed using a machine learning pipeline (implemented using the scikit-learn Library) that includes a TfidfVectorizer followed by a Logistic Regression (with `max_iter = 1000`) classifier. This model has been trained using a representative sample of labelled complaint descriptions from all three tiers (priority) and has been serialized (using Joblib) for fast loading at the time of inference. The machine learning stage classifies complaint descriptions that are stated in informal terms, use synonyms specific to the domain, or consist of multiple word (multi-word) contextual descriptions that are not located in the rule dictionary.

The fourth stage of processing involves determining the density of public amenities located within a radius of 1,000m from the complained location. This is achieved by sending the GPS coordinates of the location to the `count_pois` function which uses the OpenStreetMap Overpass API to query for a count of all public Points of Interest (such as public amenities, recreational facilities, commercial premises, bus stop locations, and public transport nodes) located within that area. This count will then be categorized into one of three tiers - High Density (>50), Medium Density (15-50), or Low Density (<15).

The fifth stage is composite priority scoring: the text-derived priority score (P1=3, P2=2, P3=1) and the location density score (High=3, Medium=2, Low=1) are combined using a weighted linear function:

$$Priority_Score = (Text_Score \times 0.7) + (Density_Score \times 0.3) \dots (1)$$

where F is the final composite priority score, D is the description severity score (P1=3, P2=2, P3=1), and L is the location density score (High=3, Medium=2, Low=1). The final priority category is assigned as: Priority 1 (final_score >= 2.5), Priority 2 (final_score >= 1.5), or Priority 3 (otherwise). The 0.7:0.3 weighting reflects the domain judgment that complaint content severity is the primary determinant of urgency, while geographic density provides a secondary but meaningful contextual adjustment.

C. Web Application Interface

The citizen interface provides an easy way for citizens to submit complaints using a structured form to provide the title of the complaint, a written description of the complaint, and the location of the complaint using GPS or manual entry. When a complaint is submitted, it is immediately classified and saved. The department that will receive the information is determined based on where the complaint was predicted (complaint category). Using the need to be able to track all of their complaints, citizens will be able to review and filter by status (registered/in progress/resolved) and delete their own complaints as needed.

The administrator can review all complaints submitted by the citizens in a table view format (both sortable and filterable). Among other things, the administrator can see the complaint number, title of the complaint, GPS coordinates of the complaint, department, citizen submitting the complaint, estimated priority score of the complaint, registered date and resolved date and current status. Using an inline drop-down menu, the administrator will be able to update the status of the complaint. This allows the administrator to manage the workflow of the complaints in real-time without having to maintain a separate case management system.

IV. SYSTEM METHODOLOGY

A. TF-IDF and Logistic Regression Training

The ML classifier utilizes a training dataset consisting of 14 labeled complaint samples that represent all three tiers: 5 P1 samples (first tier), 5 P2 samples (second tier), and 4 P3 samples (third tier). The TfidfVectorizer is configured for English stop-word removal by default. The LogisticRegression is instantiated with a maximum number of iterations (max_iter) equal to 1000 in order to ensure convergence on the training data. The Pipeline encapsulates both vectorization and classification so that inference inputs can be transformed consistently with no chance of a discrepancy between train and test features. The trained LogisticRegression object is serialized using Joblib and loaded back into the application at startup; the latencies for inferring over complaint(s) will remain under 10ms.

The selection of such a small training set is intentional; the rule-based first stage covers the vast majority of clearly categorized complaints, leaving solely semantically neutral and/or colloquially phrased complaints for the ML stage. A small, high-quality training set is preferred for this residual classification because the ML classifier is designed specifically for edge-case input(s), where exactness supersedes breadth.

B. Priority Dictionary Design

The merged priority mapping dictionary was developed through a systematic review of municipal service classification taxonomies from the NYC311 complaint dataset; the UK National Complaint Classification Standard; and published onto logical complaint routing schema. The merged dictionary is a collection of 85 unique entries to establish coverage for commonly submitted civic complaints while still maintaining sufficient semantic separation between categories shown in Table I.

Table I: Priority Tier Classification Scheme

Priority Tier	Category Count	Representative Categories
P1- Critical	38	Gas Leak, Collapsed Building, Fire Emergency, Flooding, Explosion, Hazardous Materials, Carbon Monoxide
P2- Moderate	22	Noise Pollution, Illegal Parking, Traffic Signal, Streetlight Condition, Restaurant Hygiene
P3- Low	15	Lost Property, General Inquiry, Pet Lost/Found, Information Request, Miscellaneous

C. Fuzzy Matching Configuration

Various comparative search methods exist such as the simple ratio method, partial ratio method, and token-set ratio, but the token-set ratio has been chosen because it normalizes for the possibility of having another word or phrase with a similar meaning but with a different order or an overlapping substring, thereby allowing the token-set ratio to perform better than the other methods on the cases where words in the complaint have additional contextual words. For example, in the prior case, when someone would say "There is a gas leak near my kitchen", the token-set ratio would allow for a high degree of similarity to be found with the sample usage of the category "Gas Leak", despite the additional contextual words. The 80% threshold was empirically determined on a validation set of 200 manually labeled complaint descriptions to optimize the precision-recall trade-off for the rule-based stage.

D. Location Density Integration

The total counts of five OpenStreetMap node categories i.e. amenity, leisure, shop, highway=bus stop, public transport, are queried from the Overpass API within 1,000 metres of the complaint location. The total counts of these categories represent a proxy for the functional density of the area, correlating with both the population density as well as the number of individuals that will likely be affected by the reported problem. The 1,000-metre radius was selected to encompass the immediate neighbourhood's impact zone while also being tractable enough for real time API queries. The total count of elements returned in API responses is parsed, and any error handling will return -1 in the event of a failed request so that unsuccessful attempts to classify due to lack of API availability will not fail.

V. EXPERIMENTAL RESULTS AND EVALUATION

A. Classification Performance

The hybrid classification system was evaluated on a held-out test set of 240 manually labeled civic complaint descriptions, comprising 96 P1 samples (40%), 96 P2 samples (40%), and 48 P3 samples (20%). The balanced test set was assembled from real-world complaint submissions augmented with adversarial phrased variations to test robustness to informal language, misspellings, and multi-topic descriptions. Table II presents per-class and overall classification metrics for the hybrid system and both baseline configurations.

Table II: Classification Performance: Hybrid vs. Baselines

System	P1 Recall	P2 Recall	P3 Recall	Overall Acc.
Rule-based only	88.6%	79.4%	81.3%	84.2%
TF-IDF+LR only	82.1%	85.7%	87.5%	84.9%
Hybrid (proposed)	94.2%	88.5%	89.6%	91.3%

The hybrid system achieves 91.3% overall accuracy, outperforming both the rule-only baseline (84.2%) and the ML-only baseline (84.9%). Critically, the hybrid architecture achieves 94.2% P1-class recall, the most operationally important metric. The rule-based stage provides high P1 precision for clearly phrased emergency complaints, while the ML fallback recovers P1 complaints that use informal or indirect language not directly present in the dictionary. The complementary failure modes of the two stages are the architectural basis for the hybrid design's superiority over either approach alone.

B. Composite Priority Scoring Validation

The location density weighting was validated through a controlled experiment comparing pure text-only priority scores against composite scores on 50 complaints with known high-impact locations (city center, dense commercial zones) and 50 complaints from low-density rural coordinates. For complaints where text classification yielded P2 priority, the density weighting correctly elevated 14 high-density complaints to P1 (final_score ≥ 2.5) and downgraded 9 low-density complaints to P3 (final_score < 1.5), representing a 23% net re-prioritization rate that aligns with domain expert assessments of relative urgency. This validates the 0.7:0.3 weighting as an operationally calibrated parameter for the civic complaint domain.

C. System Performance

End-to-end complaint submission latency, measured from form submission to priority score persistence, averages 340 ms under standard API conditions, with the Overpass API query accounting for approximately 290 ms of this latency. When the Overpass API is unavailable or returns an error, the system falls back to a default medium density classification, ensuring

complaint processing continues uninterrupted. The ML classification stage alone completes in under 10 ms. The Flask application supports concurrent requests through its development server, with production deployment on a Unicorn WSGI server supporting 50+ concurrent users without classification latency degradation.

VI. SYSTEM INTERFACE

A. Citizen Dashboard

The user-friendly application interface presents users with an easy to complete complaint submission form that provides the citizen with three entry fields: a short title for indexing purposes, a free-text description field for providing detailed content about the specific complaint, and a location field with an optional "Use My Location" button that uses browser Geolocation API to populate GPS coordinates for the user's current location via GPS. At the time the citizen submits the complaint on the backend, it is immediately assigned into the proper complaint type category, assigned to the appropriate department, and confirmed with an assigned complaint priority tier. Once a citizen submits a complaint, they can view all complaints submitted by them and have been documented on the complaint history dashboard in a sortable table format that has separate columns for the title, description, GPS location, assigned department, complaint status, and date. The citizen can filter the complaints they have submitted by the status of the complaint, as well as delete their complaint records.

B. Administrator Dashboard

One component of the administrator interface is a comprehensive complaint management console, which displays all complaints submitted by citizens across the entire complaint management system. Each record contains the following information: the UUID, title, GPS coordinates, assigned department, the e-mail address used by the citizen submitting the complaint, the computed priority score, the registration date of the complaint, the date the complaint was resolved, and the current status of the complaint within its workflow. Administrators have the ability to view and sort complaints by priority score and by date. Administrators have the ability to change the status of a complaint in its workflow by using an inline dropdown (Registered, In Progress, Resolved), as well as delete complaints that have been resolved. The administrator view of priority score directly corresponds to the final score from the composite function and is therefore transparent to administrators as to how the automated prioritization logic works.

VII. DISCUSSION

The results of the experiment validate the notion that a hybrid two-stage classification model has a significantly greater recall for P1 calls when compared to the performance of either rule-based or ML classification models individually. This directly impacts the operation of municipal emergency call dispatch; i.e. missing a call classified as P1 (false negative) has much more severe ramifications than over-prioritizing a call classified as P3 (false positive), so the primary objective for a call classification system's performance should be to maximize recall. In this case, the hybrid system obtains a P1 recall of 94.2% which is considerably better than the 88.6%

and 82.1% returns achieved by the rule-based and ML classification systems respective to their baseline performance levels.

The composite scoring function of 0.7:0.3 weighting ratio for complaint description severity versus location density reflects a sound calibration to the domain. Although complaint content is the foremost driver of urgency (i.e. a gas leak will always be a P1 emergency irrespective of the location), the location density is a significant secondary indicator of potential urgency for borderline cases between P2 and P3 classifications for complaints reported within high density areas, as these types of incidents collectively affect a larger portion of the population and subsequently have an increased likelihood of having secondary incident(s) as a result. Future work on this subject should develop methods of adaptive weighting schemes that use historical incident(escalation) response time and number of responses to determine an optimal description-severity to location-density ratio after considering history.

The current algorithm has multiple shortfalls; one example is that it requires a smaller training set to support the larger training sample size of the ML fallback classifier. In instances where there are segregated categories of complaints, the ML stage fails to generalize due to the limited number of classified and labeled complaint instances. The integration of public complaint datasets (NYC 311, UK National Complaint Repository) will greatly improve the completeness and coverage of the ML stage and reduce the dependence of marginal complaint descriptions on the rule dictionary.

VIII. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper presented the Civic Complaint Priority Management System, a full-stack web application that automates the classification and priority assignment of citizen-submitted civic complaints using a hybrid NLP and location-aware machine learning architecture. A two-stage pipeline incorporating RapidFuzz's rule-based matching with a TF-IDF and Logistic Regression machine-learning fallback, combined with a composite overpass API density score/output has produced an overall accuracy of 91.3% and a P1-class recall of 94.2% on a held-out set of 240 complaint descriptions. This system outperformed both pure rule-based and pure ML baselines on key complaint detection, demonstrating the synergistic efficacy of a combined approach. The deployed web application (Flask) will provide operational dashboards for both citizens and administrators, making it possible (and desirable) for all municipal complaint management departments to adopt it.

B. Future Work

Future work includes implementing a large scale labelled civic complaints corpus (NYC 311 or equivalent) will create a more robust machine-learning classifier; using BERT-based

sentence embeddings as an alternative to the TF-IDF version in the machine-learning fallback stage, thereby creating a more contextually semantic embedded representation; developing integrations with real-time notifications (to the citizen via SMS/email) when the status of their complaint has changed by an administrator; and developing explainability capabilities for administrators through LIME or attention-based visualizations to provide citizens with greater transparency in the rationale behind the/administrator's automated priority assignments, ultimately enhancing accountability to citizens for the processes of municipal governance.

ACKNOWLEDGMENT

The authors express gratitude to the Department of Computer Science and Engineering at Sreenidhi Institute of Science and Technology for providing computational resources and research support. Special thanks to project guide Mr. K. Karunakar (Assistant Professor, CSE) and project coordinator Mr. V. Satheesh Kumar (Assistant Professor, CSE) for their technical guidance and feedback throughout the development and evaluation of this system. The authors also acknowledge the use of AI tools like ChatGPT and Claude for language improvement and grammar refinement. All technical concepts and evaluations are solely the work of the authors.

REFERENCES

- [1] A. Afolabi et al., "Automated complaint management system for urban governance," *International Journal of Computer Applications*, vol. 182, no. 8, pp. 1-7, 2018.
- [2] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 1-135, 2008.
- [3] R. Kataria and M. Singh, "A comprehensive study of text classification techniques for complaint routing systems," *Journal of Information Processing Systems*, vol. 15, no. 3, pp. 555-573, 2019.
- [4] M. Bachmann and A. Ott, "Fuzzy string matching for short-text classification in service routing applications," in *Proc. ACM SIGIR Workshop on Matching Short Text*, 2020.
- [5] S. Vaidya and V. Bhatt, "Location-aware priority scoring for urban complaint management using GIS data," *Smart Cities*, vol. 4, no. 2, pp. 612-628, 2021.
- [6] Y. Wang et al., "Hybrid rule-based and machine learning classifiers for customer complaint routing," *Expert Systems with Applications*, vol. 121, pp. 96-105, 2019.
- [7] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.
- [8] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, O'Reilly Media, 2018.
- [9] OpenStreetMap Foundation, "Overpass API: Query the OpenStreetMap database," <https://overpass-api.de>, 2024.
- [10] J. Devlin et al., "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL-HLT 2019*, pp. 4171-4186.
- [11] T. Mikolov et al., "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
- [12] M. Ribeiro et al., "Why should I trust you? Explaining the predictions of any classifier," in *Proc. ACM SIGKDD*, 2016, pp. 1135-1144.