

RESEARCH ARTICLE

OPEN ACCESS

SAMANVI: A Location-Aware Local Community Platform Using PostGIS-Based Spatial Filtering, NLP Urgency Triage, and Community Trust Scoring

Dabbara Nithish Bargav Chowdary^{1,*}, Gogu Arjun², Dudekula Gouse Peera³, and Chinthamanu Aravind Kumar⁴

^{1,2,3,4}Department of Computer Science & Engineering (AIML), Parul University, Vadodara, Gujarat, India

¹2403031467006@paruluniversity.ac.in, ²2403031467010@paruluniversity.ac.in,

³2403031467008@paruluniversity.ac.in, ⁴2303031460222@paruluniversity.ac.in,

Under the Guidance of: **Prof. Ritu Agrawal**, Department of Computer Science & Engineering (AIML), Parul University

* Corresponding Author: Dabbara Nithish Bargav Chowdary

Abstract - Residents who notice local hazards such as broken streetlights, open manholes, or waterlogging usually have only centralized complaint channels or general social networks to turn to, and neither connects the report with people who live nearby. This paper presents SAMANVI, a mobile platform for hyperlocal problem reporting and peer assistance that combines five mechanisms in one workflow. A PostGIS spatial query restricts every feed to a 1 km boundary at the database level. A TF-IDF and Naive Bayes classifier assigns each report a low, medium, or critical urgency. A weighted-average Community Trust Score is updated from feedback after each completed task. A four-digit arrival PIN prevents a task from being closed unless the helper and the requester meet in person, and a randomized offset of 10–20 m hides exact problem coordinates on the public map. The system uses a React Native client, a Python Flask REST backend, and a PostgreSQL database of 15 tables with WebSocket messaging. Functional testing of six workflows, including phone and Aadhaar verification, problem posting, the 1 km boundary rule, urgency triage, and arrival-PIN verification, passed in every case. The contribution is an integrated, working design rather than a new learning algorithm, and classifier accuracy and load-test latency are not yet reported. The work shows that database-level locality, lightweight triage, and in-person verification can be combined in one low-cost community platform.

Keywords - community trust score, location-based services, PostGIS, spatial filtering, text classification

1. Introduction

Local problems such as damaged streetlights, open manholes, waterlogging, and burst pipelines are usually reported through centralized complaint channels. These channels suit formal grievances, but they do not tell the people living nearby that something is wrong, and they give residents no easy way to help each other. General social platforms have the opposite weakness. Their feeds are global or follower-based, so a report about one street competes with everything else, and nothing confirms that a helper actually arrived or whether the helper can be trusted. Neighborhood research treats locality as both a geographic area and a network of everyday interaction [1,2], and studies of location-based social networks link physical proximity to social ties [3,4].

SAMANVI is a mobile platform built around this idea. A resident posts a problem with its location, and only users within 1 km receive it. A nearby volunteer offers help, the requester accepts, and the task closes after the helper enters a four-digit PIN that the requester gives in person. Feedback then updates the helper's Community Trust Score. Urgency is estimated from the report text, and public map positions are shifted by a small random offset.

The paper makes three contributions. First, it describes a design that enforces the 1 km rule inside the database with an indexed ST_DWithin query. Second, it combines urgency triage, arrival-PIN verification, and a trust-score update into one closed assistance loop, with a coordinate offset protecting the public map. Third, it reports an implementation and a functional validation of six workflows. None of the individual techniques is new, and the paper does not claim a new learning algorithm.

Section 2 reviews related work, Section 3 describes the architecture and methods, Section 4 reports the validation results and limitations, and Section 5 concludes.

2. Related Work

2.1. Neighborhoods and location-based networks

Wellman and Leighton treat networks, neighborhoods, and communities as different ways of understanding local social organization [1], and the Project on Human Development in Chicago Neighborhoods community survey measured neighborhood-level conditions [2]. Neither is a software platform, but both support using proximity as the main condition for discovering problems. Scellato et al. studied the

socio-spatial properties of online location-based social networks [3], Wang et al. identified overlapping communities in them [4], and Chorley et al. examined personality in such networks [5]. SAMANVI takes a simpler path: it restricts discovery to a fixed radius and does not infer personality.

2.2. Trust and reputation

Ruan and Durrezi surveyed trust-management systems for online social communities, covering trust modeling, inference, and attacks [6]. Ahmadian et al. used deep representation learning for trust-aware recommendation [7], and Zarghami et al. proposed a T-Index approach [8]. SAMANVI uses trust as an interaction signal, but implements it as a weighted average that is easy to inspect rather than as a learned model.

2.3. Text classification for problem reports

Allam et al. reviewed machine-learning methods for text classification [9], Dwarakanath et al. applied automated machine learning to emergency response from social media after disasters [10], and Howard and Ruder showed that transfer learning improves text classification [11]. These methods are more powerful than TF-IDF with Naive Bayes, but they need more data and computing. SAMANVI uses the lighter method as a triage aid, not as an autonomous emergency decision system.

2.4. Spatial data, privacy, and moderation

Zhang and Yi discuss spatial data management with PostGIS [12], and PostgreSQL provides the relational platform [13]. Keeping the distance test inside the database avoids fetching candidate rows and computing distances in application code. For privacy, Shokri et al. proposed a unified framework for location privacy [14], Sun et al. studied privacy preservation for mobile users of location-based services [15], and Jia et al. proposed a blockchain-based incentive mechanism for crowd sensing [16]. SAMANVI uses a simpler application-level offset. Cullen and Kairam describe moderation as a reflective practice [17], which motivates the administrative moderation workflow. React Native, Flask, and Google Maps documentation [18–20] served as implementation references.

2.5. Research gap

Table 1 groups the reviewed work by area. Each area is well studied on its own, but the reviewed work does not combine them into one neighborhood-assistance workflow. Three needs remain: locality enforced at the database rather than the client, verification that a helper physically reached the location before a problem is closed, and a feedback-driven score that weighs past interactions.

Table 1. Research areas reviewed and their relevance to SAMANVI

Research area	Representative work	Relevance to SAMANVI
Neighborhood	Wellman and Leighton [1]; Earls et al. [2]	Locality-oriented social foundation
Location-based networks	Scellato et al. [3]; Wang et al. [4]; Chorley et al. [5]	Proximity as a discovery condition

Research area	Representative work	Relevance to SAMANVI
Trust	Ruan and Durrezi [6]; Ahmadian et al. [7]; Zarghami et al. [8]	Informs the Community Trust Score
Text classification	Allam et al. [9]; Dwarakanath et al. [10]; Howard and Ruder [11]	Supports urgency triage
Spatial data	Zhang and Yi [12]; PostgreSQL [13]	Database-level proximity filtering
Location privacy	Shokri et al. [14]; Sun et al. [15]; Jia et al. [16]	Motivates public-coordinate displacement
Moderation	Cullen and Kairam [17]	Administrative moderation workflow

3. Materials and Methods

3.1. System architecture

SAMANVI follows a three-tier architecture (Figure 1, Table 2). The client tier is a React Native application built with Expo that handles location access, reporting, maps, and chat. The application tier is a Python Flask REST service that handles authentication, problem management, spatial requests, urgency classification, trust-score updates, and the assistance workflow. Real-time messaging uses WebSockets, media files go to Cloudinary, and push notifications are sent through Firebase Cloud Messaging. The persistence tier is PostgreSQL with the PostGIS extension, hosted through Supabase, with 15 relational tables. Problem coordinates are stored as GEOGRAPHY(POINT,4326) and indexed with a GiST index. As Figure 1 shows, the client talks only to the Flask API, which in turn coordinates the identity services, media storage, real-time messaging, and the spatial database.

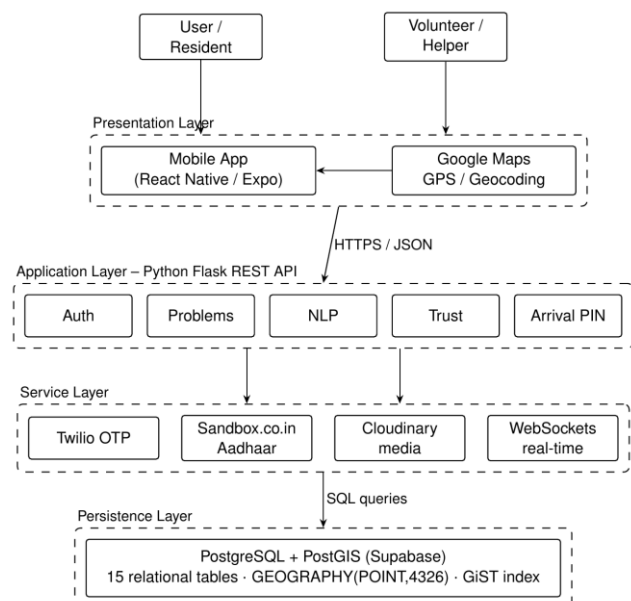


Fig. 1 Overall system architecture of the SAMANVI platform.

Table 2. SAMANVI technology stack

Tier	Technology	Responsibility
Client	React Native / Expo	Interface, GPS access, reporting, chat
Application	Python Flask REST	Authentication, APIs, urgency triage, trust updates
Persistence	PostgreSQL / PostGIS (Supabase)	Users, problems, coordinates, spatial filtering
Media	Cloudinary	Image and media storage
Real-time	WebSockets	Live communication and updates
Alerts	Firebase Cloud Messaging	Push alerts for nearby problems and offers
Maps	Google Maps	Map visualization
Identity	Twilio OTP; Sandbox.co.in	Phone and Aadhaar verification
Admin	React / Vite dashboard	Moderation, heatmap, flagged accounts

3.2. Roles and supporting features

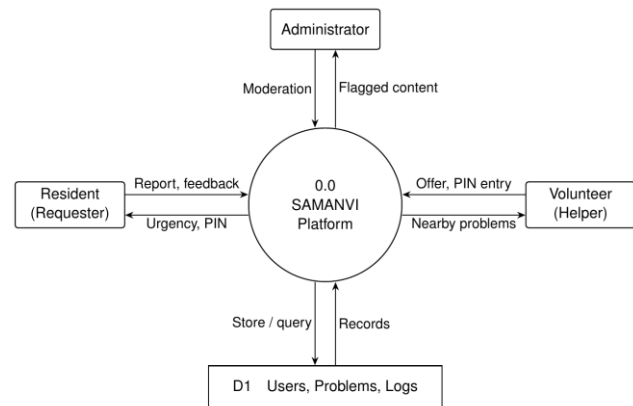
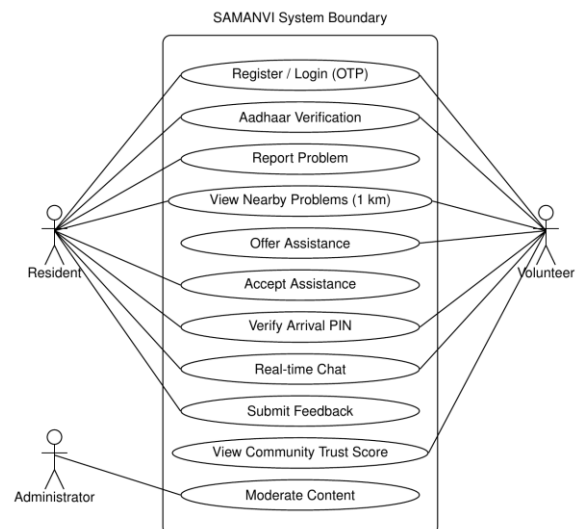
The platform has three roles. Residents report problems, accept offers, and rate the help they receive. Volunteers discover nearby problems, offer assistance, and enter the arrival PIN. Administrators moderate content through a separate React and Vite dashboard, where they view reports on a heatmap, filter them by urgency or date, remove inappropriate posts, and review flagged accounts. The nearby feed is anchored to each user's verified home coordinates rather than to live GPS. Volunteers can select skill tags that filter the requests they see, and verified users can generate and redeem invite codes.

Authentication uses JSON Web Tokens with BCrypt-hashed passwords. For Aadhaar verification, only a SHA-256 hash and the last four digits are stored, never the full number. Consent, data-export, and erasure routes support the requirements of India's Digital Personal Data Protection Act.

3.3. Operational workflow

A typical assistance cycle has twelve steps: (1) the user registers and authenticates; (2) the user's verified home location is obtained; (3) the user reports a local problem; (4) the problem location is stored in PostGIS; (5) an urgency category is assigned; (6) nearby users are identified through spatial filtering; (7) a nearby volunteer offers assistance; (8) the requester accepts; (9) a four-digit arrival PIN is used for physical verification; (10) the problem is resolved; (11) the requester gives feedback; and (12) the Community Trust Score is updated.

Figure 2 shows the Level 0 data flow diagram, which documents how requester, volunteer, and administrator interactions cross the platform boundary and reach the data store. Figure 3 gives the use cases for each role. Figure 4 shows the message passing among the subsystems for one complete cycle, from report to trust-score update. Figure 5 gives the lifecycle states of a reported problem, and Figure 6 the activity flow, including the PIN decision. Figure 7 outlines the core entities of the relational schema: users (carrying the trust score), problems (with the spatial location and urgency), assistance logs (with the arrival PIN), feedback, and media.

**Fig. 2 Level 0 data flow diagram (DFD) of the SAMANVI platform.****Fig. 3 Use case diagram of the SAMANVI system.**

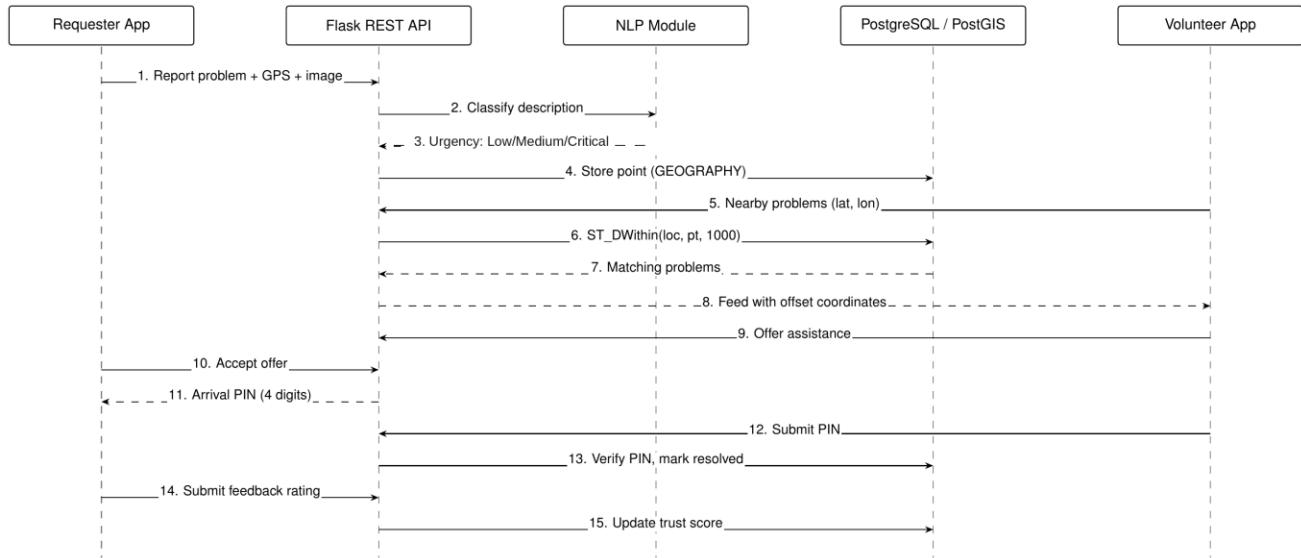


Fig. 4 Sequence diagram for problem reporting, nearby discovery, assistance, and trust update.

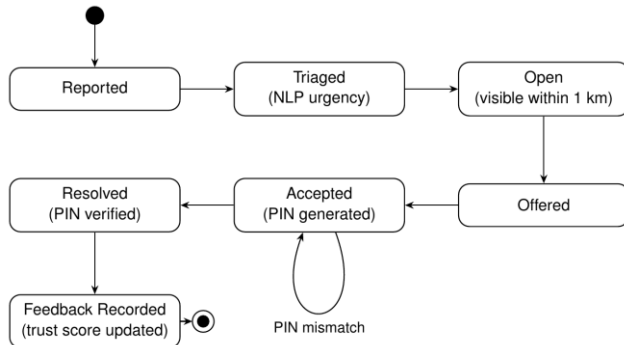


Fig. 5 Lifecycle states of a reported problem in SAMANVI.

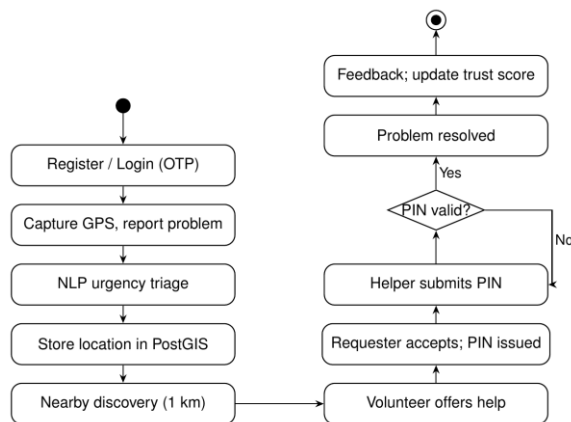


Fig. 6 Activity diagram of the SAMANVI assistance workflow.

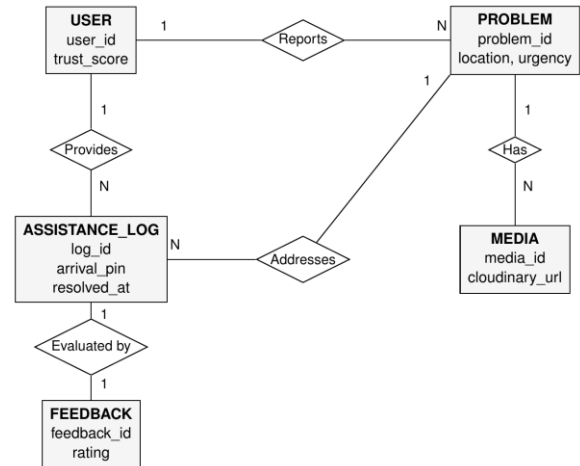


Fig. 7 Entity-relationship diagram of the core entities (15 tables in total).

3.4. PostGIS spatial filtering

SAMANVI leaves the proximity test to PostGIS instead of fetching candidate problems and computing distances in the mobile application. A problem at position P is returned to a user at position U only if Equation 1 holds,

$$D(P, U) \leq R, \quad R = 1000 \text{ m} \quad (1)$$

where $D_{P,U}$ is the geographic distance between the two points. The great-circle distance can be written with the Haversine formulation (Equations 2 and 3),

$$a = \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cos\phi_2 \sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (2)$$

$$c = 2 \arctan2(\sqrt{a}, \sqrt{1-a}), \quad d = r_E c \quad (3)$$

where ϕ and λ are latitude and longitude and r_E is the Earth radius. This formulation explains geographic distance, but the actual rule is enforced inside PostGIS by

ST_DWithin(location,

ST_MakePoint(lon, lat)::geography, 1000)

and the GiST index narrows the candidate rows before any distance is computed. Algorithm 1 gives the discovery procedure, and Figure 9 illustrates the boundary rule.

Algorithm 1. Nearby problem discovery

Input: user home location (λ_u, ϕ_u) , radius $R = 1000$ m

Output: set P_{near} of nearby problems

```
1: Build geography point  $U \leftarrow \text{ST\_MakePoint}(\lambda_u, \phi_u)$ 
2: Run ST_DWithin(location,  $U$ ,  $R$ ) over the GiST-indexed column
3: for each returned problem  $p$  do
4:   Replace the coordinate of  $p$  by its public (offset) coordinate  $P'$ 
5:   Add  $p$  to  $P_{\text{near}}$ 
6: end for
7: return  $P_{\text{near}}$ 
```

3.5. Urgency classification

Problem descriptions are converted to feature vectors with Term Frequency–Inverse Document Frequency. For a term t in document d , Equations 4 to 6 apply,

$$TF(t, d) = \frac{\text{occurrences of } t \text{ in } d}{\text{total terms in } d} \quad (4)$$

$$IDF(t) = \log\left(\frac{N}{1 + DF(t)}\right) \quad (5)$$

$$TFIDF(t, d) = TF(t, d) \times IDF(t) \quad (6)$$

where N is the number of documents and DF_t is the number of documents that contain t . The vector X is passed to a Naive Bayes classifier (Equations 7 and 8),

$$P(C | X) \propto P(C) \prod_{i=1}^n P(x_i | C) \quad (7)$$

$$\hat{C} = \arg\max_C P(C | X), \quad C \in \{\text{Low, Medium, Critical}\} \quad (8)$$

which selects one of three urgency classes: Low, Medium, or Critical. Figure 8 shows the processing chain. The class is a triage aid. It labels each report so that administrators can filter by urgency, and it does not decide on its own whether an emergency exists.

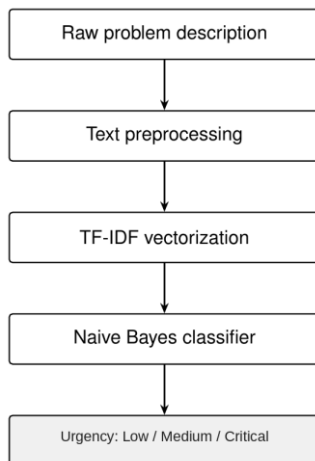


Fig. 8 Urgency triage workflow.

3.6. Community Trust Score

The Community Trust Score of user u is a weighted average of feedback ratings (Equation 9),

$$CTS_u = \frac{\sum_{i=1}^n w_i r_i}{\sum_{i=1}^n w_i} \quad (9)$$

where r_i is the rating from completed interaction i on a 1–5 scale, w_i is the trust weight of the rating source, and n is the number of completed feedback interactions. The weight reflects the source account's activity and verified interactions, so feedback from established accounts counts for more than feedback from new ones. After each new interaction the same average can be maintained incrementally (Equation 10),

$$CTS_{\text{new}} = \frac{W_{\text{old}} CTS_{\text{old}} + w_{\text{new}} r_{\text{new}}}{W_{\text{old}} + w_{\text{new}}} \quad (10)$$

where W_{old} is the accumulated historical weight, so earlier verified interactions stay in the score.

3.7. Location privacy

The exact problem coordinate $P = (\text{lat}, \text{lon})$ is not shown on the public map. A randomized displacement is applied first (Equation 11),

$$P' = P + \Delta P, \quad 10 \leq \|\Delta P\| \leq 20 \text{ m} \quad (11)$$

so that P' , not P , is displayed (Figure 9). One realization draws $u, v \sim U(0, 1)$, sets $\rho = 10 + 10u$ m and $\theta = 2\pi v$, and applies $\Delta\phi = \rho \cos \theta / r_E$ and $\Delta\lambda = \rho \sin \theta / (r_E \cos \phi)$. The result hides the exact address but still shows roughly which street the problem is on, which the feed needs. This is an implementation choice, not a formal location-privacy guarantee.

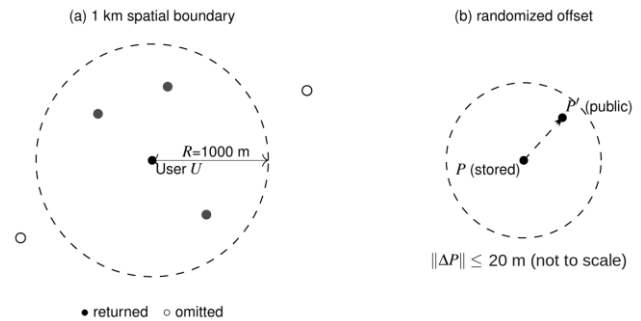


Fig. 9 (a) Nearby-problem boundary enforced by ST_DWithin; (b) randomized offset applied before public display.

3.8. Arrival PIN verification

A four-digit PIN is generated when the requester accepts an offer (Equation 12),

$$PIN \in \{0000, 0001, \dots, 9999\} \quad (12)$$

and the task can be resolved only if the submitted PIN matches the stored one (Equation 13),

$$PIN_{\text{submitted}} = PIN_{\text{stored}} \quad (13)$$

The helper obtains the PIN from the requester on arrival, so a match indicates that the two met in person before the problem could be closed (Figure 6). Because the PIN space has 10^4 values, an attacker who is allowed k distinct guesses succeeds

with probability $k/10^4$, so the number of attempts must be limited.

4. Results and Discussion

4.1. Functional validation

Functional testing covered authentication, problem reporting, spatial filtering, urgency triage, and the arrival-PIN workflow. The cases include phone registration through Twilio OTP and Aadhaar verification through Sandbox.co.in. Table 3 lists the six cases, and all of them passed. Beyond these cases, the test plan used PyTest unit tests for API routes and validation logic, integration tests with the external services, end-to-end runs on physical Android and iOS devices, and an Apache JMeter load test that simulates 500 concurrent connections against the spatial endpoints with a 250 ms response-time target. Measured load-test figures are not reported in this paper.

Table 3. System test cases and validation

ID	Test case	Expected result	Status
TC01	Register via phone	Twilio OTP dispatched	Pass
TC02	Aadhaar verification	Sandbox.co.in validated	Pass
TC03	Post problem	Image uploaded, database updated	Pass
TC04	1 km boundary rule	ST_DWithin restricted results	Pass
TC05	Urgency triage	TF-IDF classified urgency	Pass
TC06	Arrival PIN	Trust score updated	Pass

4.2. Spatial query

Validation confirmed that the spatial query enforced the 1 km boundary: problems outside the specified distance were omitted from the nearby feed. The outcome is the indicator in Equation 14,

$$\text{Result}(P, U) = \begin{cases} 1, & D(P, U) \leq 1000 \text{ m} \\ 0, & D(P, U) > 1000 \text{ m} \end{cases} \quad (14)$$

which gives a database-level mechanism for limiting nearby problem discovery.

4.3. Urgency triage, trust score, and arrival PIN

The functional test confirmed that the chain problem text $\rightarrow$ TF-IDF $\rightarrow$ Naive Bayes $\rightarrow$ {Low, Medium, Critical} produced an urgency category for the evaluated reports. After a completed assistance interaction, the trust score is updated by combining the feedback rating with the weight of its source, using the incremental form in Section 3.6. The arrival-PIN test confirmed that the workflow proceeds through verification to the trust update.

4.4. Comparison with common channels

Table 4 contrasts the general characteristics of civic complaint portals and social networks with SAMANVI. The advantages follow from the design. Proximity is enforced in the database, so the feed contains only nearby problems. Reporting, triage, offer, acceptance, arrival verification,

feedback, and trust update form one closed loop. Resolution is gated on the arrival PIN, and public map positions are displaced from the stored coordinates.

Table 4. Feature comparison of SAMANVI with common channels

Feature	Civic portals	Social networks	SAMANVI
Spatial scope	Central desk	Unbounded feed	1 km PostGIS boundary
Assistance	Office-mediated	Informal replies	Nearby volunteers
Verification	Typically none	Typically none	Arrival PIN
Urgency triage	Manual	Not designed for it	TF-IDF and Naive Bayes
Reputation	Not typical	Likes and follows	Community Trust Score
Location privacy	Complaint-specific	Platform-specific	Offset of 10–20 m

4.5. Implementation experience

Three problems shaped the design. The nearby-problem query was slow at first because the distance search was not using a spatial index, and adding a GiST index to the geography column fixed it. Map rendering was slow on some Android devices when many markers were shown, so marker clustering and viewport debouncing were added. Occasional duplicate chat messages appeared when network delays caused retries, so each outgoing message now carries a unique identifier that the client checks against its local cache.

4.6. Limitations

The evaluation is functional. Classifier accuracy, training-data size, query latency under load, and user studies are not reported, and the JMeter results are missing from this paper, so the results show that the integrated workflow works and should not be read as a benchmark of performance, community effectiveness, or production reliability. The PIN confirms an in-person exchange rather than GPS presence, and its small space needs attempt limiting. A 10–20 m offset is a simple mechanism that does not hide the neighborhood of a report, and if a fresh random offset were drawn on every request, repeated observations could be averaged, so a stable per-problem offset is preferable. The weighted-average trust score gives a new account little influence, but a group of established accounts could still inflate a score. Identity verification, media storage, and hosting rely on third-party services.

5. Conclusion

SAMANVI presents a location-aware approach to coordinating community assistance around local problems. It combines PostGIS spatial filtering, TF-IDF and Naive Bayes urgency triage, a weighted-average Community Trust Score, real-time messaging, arrival-PIN verification, and a coordinate offset in a three-tier mobile system. Functional tests of the authentication, reporting, spatial-filtering, triage, and assistance workflows passed. The contribution lies in

combining established techniques from location-based services, trust management, text classification, spatial databases, and location privacy into one hyperlocal workflow, not in a new algorithm.

Future work includes measuring classifier accuracy and query latency under load; expanding the search radius when no helper is found within 1 km; comparing helper-uploaded photographs to indicate whether a problem appears resolved; forwarding reports to municipal systems where appropriate; adding Hindi, Gujarati, and Telugu support; offering optional in-app payments for skilled services; comparing the weighted average with learning-based reputation models on a larger interaction dataset; and investigating an append-only ledger that makes unauthorized changes to trust records easier to detect.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

Funding Statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Acknowledgments

The authors thank Prof. Ritu Agrawal, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Parul University, for continuous mentorship, project supervision, and guidance throughout the design, implementation, and evaluation of this work.

References

- [1] B. Wellman and B. Leighton, "Networks, Neighborhoods, and Communities: Approaches to the Study of the Community Question," *Urban Affairs Quarterly*, 1979.
- [2] F. Earls, J. Brooks-Gunn, S. Raudenbush, and R. Sampson, "Project on Human Development in Chicago Neighborhoods: Community Survey, 1994–1995," *ICPSR Data Holdings*, 1999.
- [3] S. Scellato, A. Noulas, R. Lambiotte, and C. Mascolo, "Socio-Spatial Properties of Online Location-Based Social Networks," in *Proc. ICWSM*, 2011.
- [4] Z. Wang et al., "Discovering and Profiling Overlapping Communities in Location-Based Social Networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 44, no. 4, pp. 499–509, 2014.
- [5] M. J. Chorley, R. M. Whitaker, and S. M. Allen, "Personality and location-based social networks," *Computers in Human Behavior*, vol. 46, pp. 45–56, 2015.
- [6] Y. Ruan and A. Duresi, "A survey of trust management systems for online social communities: Trust modeling, trust inference and attacks," *Knowledge-Based Systems*, vol. 106, pp. 150–163, 2016.
- [7] M. Ahmadian, M. Ahmadi, and S. Ahmadian, "A reliable deep representation learning to improve trust-aware recommendation systems," *Expert Systems with Applications*, 2022.
- [8] A. Zarghami, S. Fazeli, N. Dokoohaki, and M. Matskin, "Social Trust-Aware Recommendation System: A T-Index Approach," in *Proc. WI-IAT Workshops*, 2009.
- [9] H. Allam et al., "Text Classification: How Machine Learning Is Revolutionizing Text Categorization," *Information*, 2025.
- [10] L. Dwarakanath et al., "Automated Machine Learning Approaches for Emergency Response and Coordination via Social Media in the Aftermath of a Disaster," *IEEE Access*, vol. 9, pp. 68917–68931, 2021.
- [11] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification," in *Proc. ACL*, 2018.
- [12] L. Zhang and J. Yi, "Management methods of spatial data based on PostGIS," in *Proc. PACCS*, 2010.
- [13] PostgreSQL Global Development Group, "PostgreSQL: The World's Most Advanced Open Source Relational Database," 2023. [Online]. Available: <https://postgresql.org>
- [14] R. Shokri, J. Freudiger, and J.-P. Hubaux, "A Unified Framework for Location Privacy," in *HotPETs Workshop*, 2010.
- [15] G. Sun et al., "Location Privacy Preservation for Mobile Users in Location-Based Services," *IEEE Access*, vol. 7, pp. 87425–87438, 2019.
- [16] B. Jia et al., "A Blockchain-Based Location Privacy Protection Incentive Mechanism in Crowd Sensing Networks," *Sensors*, vol. 18, no. 11, p. 3894, 2018.
- [17] A. Cullen and S. Kairam, "Practicing Moderation: Community Moderation as Reflective Practice," in *Proc. ACM HCI*, 2022.
- [18] Meta Platforms Inc., "React Native: Learn once, write anywhere," 2024. [Online]. Available: <https://reactnative.dev>
- [19] Pallets Projects, "Flask: A lightweight WSGI backend framework," 2024. [Online]. Available: <https://flask.palletsprojects.com>
- [20] Google LLC, "Google Maps Platform – Geocoding API," 2023. [Online]. Available: <https://developers.google.com/maps>