

# A Framework for Air Quality Intelligence and Decision Support in India

Swapnil Patil<sup>1</sup>, Dattaraj Shinde<sup>1</sup>, Aarya Hingane<sup>1</sup>, Payal Kumkale<sup>1</sup>, and Pratik Thorbole<sup>1</sup>

<sup>1</sup>School of Computing, MIT ADT University, Pune, Maharashtra, India

**Abstract:** Urban air quality monitoring in India remains predominantly descriptive—systems report pollution indices but offer little in terms of causative explanation, short-term forecasting, or actionable citizen guidance. This paper presents *Sanjeevan*, a lightweight, mobile-first air intelligence framework designed to bridge this gap through a layered pipeline combining open data aggregation, rule-based reasoning, and a dual-user interface serving both everyday citizens and policymakers. The system integrates real-time AQI data from OpenAQ and meteorological parameters from Open-Meteo, processing them through a heuristic intelligence layer that performs source-contribution estimation, weather-influenced trend forecasting, and context-aware health guidance. The application frontend is implemented as a functional prototype using Expo (React Native), enabling cross-platform deployment without dedicated server infrastructure. Unlike approaches predicated on large-scale trained models, *Sanjeevan* prioritizes explainability and operational feasibility under resource constraints. Qualitative evaluations suggest improvements in user interpretability and decision support clarity, though formal quantitative validation remains a subject of ongoing work.

**Keywords:** air quality index, source attribution, urban air pollution, rule-based forecasting, mobile computing, explainable AI, decision support systems

## I. INTRODUCTION

Air pollution constitutes one of the most pressing public health emergencies in contemporary India. According to the World Health Organization, 21 of the 30 most polluted cities globally are located in India, with particulate matter concentrations in metropolitan areas such as Delhi, Mumbai, and Pune frequently exceeding permissible limits by multiples rather than marginal degrees [1]. The health burden is commensurate: chronic exposure to elevated PM<sub>2.5</sub> and PM<sub>10</sub> concentrations is associated with increased incidence of cardiovascular disease, respiratory impairment, and premature mortality [2]. Economically productive urban populations bear a disproportionate share of this burden, and the demographic most exposed—low-income residents in high-density zones—typically has the least access to timely, intelligible air quality information.

Existing air quality monitoring infrastructure in India,

while expanding, operates largely in a *monitoring-only* paradigm. Platforms such as the Central Pollution Control Board (CPCB) dashboard and SAFAR (System of Air Quality and Weather Forecasting and Research) disseminate numerical AQI values and categorical breakdowns by pollutant [3]. While technically accurate, these outputs are rarely contextualized for individual decision-making: a citizen observing an AQI of 180 receives no intelligible guidance on whether to modify their commute, avoid outdoor exercise, or take preventive health measures. Similarly, policymakers lack integrated tools that map current pollution readings to probable contributing sources in near real-time.

This deficit is not merely one of interface design but reflects a deeper architectural gap. AQI systems are fundamentally descriptive: they aggregate sensor readings and compute indices. They do not, as a default behavior, reason about causal contributors, project short-term trajectories using local meteorological data, or adapt guidance to user-specific parameters such as age, pre-existing conditions, or activity patterns. The consequences of this limitation are measurable: uninformed citizens make suboptimal behavioral decisions, and policymakers without attribution data struggle to design effective targeted interventions.

*Sanjeevan*—a name derived from the Sanskrit word for life-restoring vitality—is conceived as a response to these structural shortcomings. The system does not seek to replace scientific sensor networks or government monitoring infrastructure. Instead, it occupies a complementary position: an intelligent interface layer that transforms publicly available air quality and meteorological data into interpretable, personalized, and actionable outputs for two primary user classes—urban citizens and municipal/policy-level stakeholders. The contribution is not algorithmic novelty but rather *system-level integration*: combining data aggregation, rule-based intelligence, and usable interface design within a lightweight, deployable mobile prototype.

The remainder of this paper is organized as follows. Section II formalizes the problem statement. Section III reviews relevant prior work. Section IV describes the *Sanjeevan* proposed system. Section V details the methodology. Section VI covers the system architecture. Section VII presents the implementation. Section VIII reports observations and qualitative results. Section IX candidly enumerates limitations. Section X outlines future directions, and Section XI concludes.

## II. PROBLEM STATEMENT

The central problem motivating Sanjeevan is that current AQI dissemination systems in India are *descriptive without being prescriptive*. This distinction is operationally significant: a system that tells a user "PM<sub>2.5</sub> is 95  $\mu\text{g}/\text{m}^3$ , category: Unhealthy" has conveyed a measurement, not a recommendation. Three structural deficiencies underlie this gap.

**Absence of source-level attribution in real-time interfaces.** While academic source apportionment studies exist and some government bodies publish annual source contributions, this information is not surfaced in citizen-accessible platforms in a timely or location-specific manner. A user cannot determine, from standard interfaces, whether current pollution is driven predominantly by vehicular emissions, industrial activity, crop burning, or meteorological inversion-information that would materially affect both individual decisions and policy priorities.

**Lack of integrated short-term forecasting.** Meteorological conditions-specifically wind speed, wind direction, temperature inversions, and relative humidity-have well-established relationships with pollutant dispersion and concentration [4]. Despite this, most citizen-facing platforms present only current-state data without forward projections, even when adjacent meteorological APIs provide the necessary inputs.

**Absence of personalized, context-sensitive guidance.** Health recommendations in AQI systems, when present, are typically static and population-generic. A system that advises "avoid outdoor activities" uniformly across all users ignores significant variance in individual risk profiles: an asthmatic elderly person, a healthy adult cyclist, and a schoolchild require substantively different advisories under identical pollution conditions.

The consequence of these three deficiencies is a class of platforms that are technically sound but behaviorally ineffective: they produce outputs that informed users can interpret but that fail to meaningfully guide the broad population they nominally serve.

## III. RELATED WORK

### A. Air Quality Forecasting Models

Computational forecasting of air quality has been an active research domain for over two decades, with approaches spanning physics-based chemical transport models (CTMs), statistical regression, and data-driven machine learning. CTMs such as WRF-Chem and CMAQ provide high-fidelity predictions by simulating atmospheric chemistry, but their computational demands and domain expertise requirements render them unsuitable for lightweight mobile deployment [5].

Statistical and machine learning approaches offer greater operational tractability. Zheng et al. demonstrated that urban air quality inference from heterogeneous data sources-including meteorological readings and traffic patterns-could support spatially rich predictions not achievable from sparse sensor networks alone [6]. Hybrid models combining sequential neural architectures with attention mechanisms have shown robustness to missing sensor data and irregular

sampling-conditions common in developing-country monitoring networks [7]. However, these approaches generally require historical training datasets of significant length, periodic retraining infrastructure, and non-trivial inference latency on mobile hardware.

Simpler ML approaches-gradient boosted trees, random forests, and support vector regressors-have demonstrated competitive performance for short-horizon forecasting tasks while remaining computationally tractable [7]. The systematic review by Bai et al. notes that model selection for AQI prediction is strongly domain-dependent: the relative performance of architectures varies substantially across cities, seasons, and sensor configurations, complicating generalizable deployment [8].

Sanjeevan does not employ any trained ML model in its current implementation. This design choice is deliberate and is discussed further in Section V. The system instead uses meteorological heuristics for forward projection, sacrificing quantitative forecasting accuracy for interpretability and zero-dependency deployability.

### B. Source Apportionment Methods

Source apportionment-the attribution of ambient pollutant concentrations to contributing emission categories-is methodologically mature but computationally intensive. Receptor modeling techniques such as Positive Matrix Factorization (PMF) and Chemical Mass Balance (CMB) are widely applied in India and have produced sector-specific contribution estimates for major cities [9]. Studies on Delhi's pollution sources have identified vehicular emissions, road dust, and industrial contributions as primary PM<sub>2.5</sub> drivers, with seasonal variation driven significantly by crop residue burning in upwind agricultural regions [10].

These methods, while scientifically rigorous, operate on time scales of months to years and require speciated chemical measurements from filter-based sampling-data streams unavailable from the real-time public APIs that Sanjeevan consumes. The system's source attribution module consequently operates as a *heuristic estimator* rather than a scientific apportionment tool. This distinction is explicitly preserved in the interface design and is a recognized limitation (Section IX).

### C. Existing Citizen-Facing Platforms

SAFAR, developed jointly by the Ministry of Earth Sciences and IITM Pune, represents the most scientifically sophisticated citizen-facing air quality platform in India [3]. It integrates high-resolution WRF-based meteorological modeling with chemical transport simulations to generate 72-hour forecasts for major Indian cities. SAFAR's scientific credibility is high; its citizen accessibility, however, is constrained by interface complexity and limited coverage outside major metropolitan centers.

The CPCB's National Air Quality Index dashboard provides near-real-time data from the National Ambient Air Quality Monitoring (NAAQM) network but presents raw station-level data without forecasting, source attribution, or personalized guidance layers [11].

International platforms such as IQAir, AirVisual, and Plume Labs have addressed some of these gaps through mobile-first design, but their primary business context is global commercial deployment rather than India-specific policy utility.

#### D. Identified Gap and Positioning of This Work

A review of existing systems reveals a consistent structural gap: no single platform-scientific or commercial-integrates *all three* of the following capabilities within a citizen-accessible mobile interface: (i) short-term qualitative forecasting informed by local meteorological conditions, (ii) heuristic source attribution contextualized by time of day and season, and (iii) personalized health guidance conditioned on individual risk profiles. SAFAR addresses forecasting but not personalization or source transparency. CPCB addresses monitoring but not forecasting or guidance. Commercial platforms address usability but not source attribution or policy-relevant outputs.

Sanjeevan's contribution is positioned explicitly as *system-level integration* rather than algorithmic novelty. The value proposition is not a superior forecasting model or a more accurate source apportionment technique, but a unified pipeline that brings these three functional capabilities together within a single, lightweight, openly deployable mobile prototype. This framing is important: evaluating the system against state-of-the-art forecasting benchmarks would be a category error. Its appropriate evaluation criterion is whether the integrated system meaningfully improves citizen decision support relative to the monitoring-only baseline.

### IV. PROPOSED SYSTEM: SANJEEVAN

Sanjeevan is architected as a four-stage pipeline: **Data** → **Processing** → **Intelligence** → **Decision**. Each stage is described below.

#### A. Data Sources

Two public APIs serve as primary data ingestion points:

**OpenAQ** [12]: An open-access platform aggregating real-time air quality measurements from government monitoring networks worldwide. The API exposes station-level readings for pollutants including PM<sub>2.5</sub>, PM<sub>10</sub>, NO<sub>2</sub>, O<sub>3</sub>, SO<sub>2</sub>, and CO, with geographic query capabilities. For Indian cities, OpenAQ aggregates data from CPCB-affiliated monitoring stations.

**Open-Meteo** [13]: A free, open-source meteorological API providing forecast and historical weather data including wind speed, wind direction, relative humidity, surface temperature, and atmospheric pressure. Open-Meteo uses ECMWF and other NWP model outputs as underlying data sources.

No proprietary or authenticated data sources are required, ensuring that the system can be deployed and evaluated without institutional data agreements.

#### B. Processing Layer

Raw API responses undergo normalization and alignment before entering the intelligence layer. This includes:

- Unit standardization: converting OpenAQ pollutant values to  $\mu\text{g}/\text{m}^3$  where applicable and performing approximate AQI categorization based on dominant pollutant values, consistent with published national index definitions [11].
- Temporal alignment: matching weather forecast timestamps with available AQI data points to construct a synchronized input vector for heuristic analysis.
- Spatial fallback: when station-level readings are sparse or absent, the nearest available monitoring station's data is substituted with appropriate uncertainty flagging in the interface.

#### C. Intelligence Layer

The intelligence layer is the distinguishing architectural component of Sanjeevan. It comprises three sub-modules:

**Rule-Based Forecasting Engine.** Rather than deploying a trained forecasting model, the system applies a set of meteorological influence rules to project qualitative AQI trend directions for the subsequent 24–48 hours. The rule set encodes relationships between meteorological conditions and pollutant behavior that are consistent with general atmospheric dispersion principles [4]:

- Higher wind speeds from directions away from known emission zones tend to improve pollutant dispersion, projecting a decreasing AQI trend.
- Elevated relative humidity combined with low wind speeds tends to inhibit dispersion and promote hygroscopic growth of particulates, projecting a worsening trend.
- Temperature inversion indicators (calm, cool overnight conditions followed by a slow morning warm-up) suggest near-surface pollutant accumulation, projecting a stable or worsening trend.
- Forecast precipitation is associated with wet scavenging of particulates, projecting an improving trend.

These rules produce qualitative projections (improving / stable / worsening) rather than numerical point estimates. This is an intentional design choice that reflects the system's epistemic position: heuristic rules cannot responsibly generate precise numerical forecasts, and presenting them as such would misrepresent system capability to users.

**Source Estimation Module.** A heuristic source contribution estimator distributes current AQI into approximate percentage contributions from five source categories: vehicular emissions, industrial activity, dust/construction, biomass/waste burning, and meteorological concentration (i.e., dispersion-limited accumulation rather than direct emission). The distribution is computed based on:

- Time of day (peak vehicular hours vs. off-peak).

- Season (post-harvest crop burning periods in October–November; dust-heavy summer months).
- Dominant pollutant species in the current reading (elevated NO<sub>2</sub> is indicative of combustion sources including vehicular; elevated SO<sub>2</sub> is more consistent with industrial activity).
- Wind direction relative to known industrial zones (where available from static geo-metadata).

This module explicitly does not perform chemical mass balance or receptor modeling. Its output should be interpreted as a reasoned estimate for contextual awareness, not as scientifically validated attribution.

**Guidance Engine.** A rule-based personalization engine generates health and behavioral recommendations conditioned on three inputs: the current AQI category, the user's stated risk profile (gathered at onboarding: age group, respiratory conditions, activity level), and the time of day. The rule set spans outdoor activity advisories, mask recommendation thresholds, ventilation guidance, and escalation cues to seek medical attention. At the policy-user interface level, guidance shifts to aggregate trend summaries, source contribution narratives, and threshold exceedance alerts designed to inform intervention decisions.

## D. Application Layer

The application is implemented as a cross-platform mobile interface using Expo (React Native). The UI is organized into four primary screens: an AQI Dashboard (real-time summary and station data), a Forecast Screen (48-hour qualitative trend projection), a Source Breakdown Screen (heuristic contribution visualization), and a Personalized Guidance Screen (adaptive health recommendations). A settings module handles user profile configuration and stored preferences.

Table-1 provides a concise overview of each implemented module, its functional role, and the type of implementation employed.

## V. METHODOLOGY

### A. Design Philosophy: Rule-Based Intelligence Over Trained Models

The deliberate exclusion of trained machine learning models from Sanjeevan's intelligence layer merits explicit justification, as it represents a significant departure from the direction of recent research in computational AQI analysis.

The primary justification is *explainability*. An opaque model predicting that tomorrow's AQI will be 160 cannot, without additional interpretability scaffolding, explain *why*. In a system designed to support informed citizen decision-making, opacity in the reasoning chain—even in the presence of statistical accuracy—undermines the core design objective. Rule-based systems, by contrast, produce directly inspectable reasoning: the interface can convey that AQI is projected to worsen because atmospheric conditions currently favor pollutant accumulation, with causal transparency that a black-box prediction cannot match.

Table 1: Table-1: Feature Overview of Sanjeevan Modules

| Module            | Function                                                                    | Implementation Type                     |
|-------------------|-----------------------------------------------------------------------------|-----------------------------------------|
| AQI Dashboard     | Displays real-time pollutant readings and AQI category from nearest station | API (OpenAQ) + UI                       |
| Forecast Engine   | Projects qualitative 48-hour AQI trend with meteorological rationale        | Rule-based + API (Open-Meteo)           |
| Source Estimation | Estimates percentage contribution of emission source categories             | Rule-based (heuristic)                  |
| Guidance Engine   | Generates personalized health and behavioral recommendations                | Rule-based + UI                         |
| Policy View       | Aggregate trend summaries and multi-station comparison                      | UI (conceptual / partially implemented) |

The secondary justification is *feasibility within constraints*. Training a geographically valid AQI forecasting model requires historical sensor data at appropriate temporal and spatial resolution, which is not uniformly available across Indian cities via public APIs. Deploying a pre-trained model within a mobile application introduces size and latency constraints that conflict with the lightweight design objective. Maintaining model validity over time requires periodic retraining infrastructure that exceeds the resource envelope of a prototype research system.

This is not a claim that rule-based systems are superior to ML-based forecasting—they are not, in terms of quantitative predictive accuracy. It is a claim that, under the specific constraints of this system (public data only, mobile deployment, prototype scope, explainability priority), rule-based intelligence represents a more appropriate architectural choice.

### B. Forecasting Methodology

The forecasting module operates on a 48-hour meteorological forecast retrieved from Open-Meteo at six-hour resolution. For each forecast interval, the rule engine evaluates a ranked set of meteorological conditions and produces a qualitative AQI tendency (improving / stable / worsening) with a natural-language rationale string. The rationale string is displayed directly in the UI, ensuring transparency. No numerical AQI prediction is generated; the system presents directional trend information only.

### C. Source Estimation Methodology

Source estimation is implemented as a priority-weighted heuristic classification. The system first identifies the dominant pollutant species in the current AQI reading (PM<sub>2.5</sub>, PM<sub>10</sub>, NO<sub>2</sub>, etc.) and maps it to a primary source category using a lookup table informed by published Indian source apportionment literature [9]. Secondary source weights are

then adjusted based on time-of-day and seasonal signals. The resulting percentage distribution is normalized to sum to 100% and is presented with an explicit disclaimer noting its heuristic nature.

## D. Guidance Generation Methodology

The guidance engine maps the Cartesian product of AQI category  $\times$  user risk tier  $\times$  time context to a guidance template set. AQI categories follow the CPCB six-tier classification (Good, Satisfactory, Moderate, Poor, Very Poor, Severe) [11]. User risk tier is a three-level classification (standard, sensitive, high-risk) derived from onboarding inputs. Time context distinguishes morning commute, daytime activity, and evening hours. The resulting matrix spans 54 guidance states, each mapped to a distinct advisory template with parameterized text. This approach ensures coverage, consistency, and auditability of guidance outputs.

## VI. SYSTEM ARCHITECTURE

The Sanjeevan system is structured across three architectural layers, each with distinct responsibilities. Figure-1 illustrates the overall system architecture and the directional flow of data between layers.

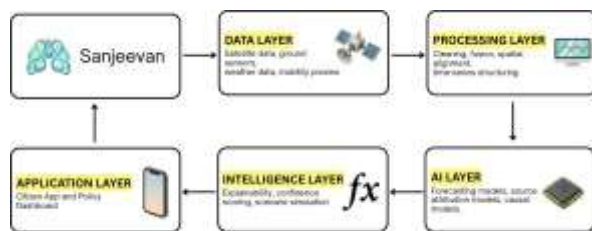


Figure 1: Fig-1: System Architecture of Sanjeevan Framework

### A. UI Layer

The mobile application, built with Expo (React Native), constitutes the UI layer. React Native's component model enables declarative UI construction with a state management approach implemented via React Context and local component state. Navigation between the four primary screens is managed through React Navigation's stack navigator. The UI layer is responsible solely for rendering and user input collection; it contains no direct API calls and no business logic.

### B. Backend Logic Layer

All intelligence and processing logic executes client-side on the mobile device. A service module hierarchy handles API calls, response parsing, and data normalization. Separate service modules implement the forecasting engine, source estimator, and guidance engine. This client-side processing model eliminates the need for a dedicated application server,

reducing deployment complexity and operational costs at the cost of exposing implementation logic in the distributed application bundle. For a prototype system without competitive intellectual property concerns, this trade-off is acceptable.

State management uses React Context to propagate normalized data and computed outputs across the component tree. AsyncStorage (React Native's key-value persistence layer) is used to persist user profile settings and cached API responses across sessions, reducing redundant network requests.

## C. Data Layer

The data layer consists of two external APIs: OpenAQ and Open-Meteo. API calls are made from the service modules in the logic layer using the Fetch API. Responses are JSON-formatted and parsed directly. No intermediate back-end, caching server, or database is employed. Error handling accounts for API rate limits, network unavailability, and missing station data, with appropriate fallback messaging surfaced to the user interface.

The overall data flow can be summarized as:

API Response  $\rightarrow$  Normalization  $\rightarrow$  Rule Engine  $\rightarrow$  State Update  $\rightarrow$  UI (1)

This linear pipeline ensures that data transformations are traceable and debuggable at each stage.

## VII. IMPLEMENTATION

### A. Technology Stack

The frontend application is implemented in JavaScript using Expo (SDK 49), which provides a managed React Native environment with native module access without requiring native build toolchains. The choice of Expo reflects the prototyping context: the managed workflow accelerates development and simplifies cross-platform testing on iOS and Android without separate native configurations.

Key libraries employed include:

- @react-navigation/native and @react-navigation/stack for screen navigation.
- @react-native-async-storage/async-storage for local persistence.
- react-native-svg and victory-native for data visualization components (source breakdown charts, trend indicators).
- expo-location for device GPS integration, enabling automatic nearest-station selection.

### B. Data Integration

OpenAQ API calls are parameterized by geographic coordinates derived from the device's GPS or a manually entered

city selection. The API returns the most recent measurements from all monitoring stations within a configurable radius (default: 25 km). The system selects the station with the most recent timestamp and freshest data across the highest number of pollutant parameters. PM<sub>2.5</sub> is treated as the primary AQI determinant in the absence of complete pollutant panels, reflecting standard practice in Indian air quality monitoring.

Open-Meteo API calls retrieve a 48-hour hourly forecast for temperature, relative humidity, wind speed at 10m, wind direction at 10m, and precipitation probability. These parameters are indexed at six-hour intervals for the forecasting engine.

## C. Implemented Features

The current prototype implements the following features in functional state:

- **AQI Dashboard:** Real-time AQI display with CPCB category color coding, pollutant breakdown table, and station metadata.
- **Forecast Screen:** 48-hour qualitative trend projection with natural-language meteorological rationale per interval.
- **Source Breakdown Screen:** Heuristic pie-chart visualization of estimated source contributions with disclaimer messaging.
- **Personalized Guidance Screen:** Adaptive health and behavioral recommendations conditioned on user risk profile and current AQI.
- **User Profile Configuration:** Onboarding flow capturing age group, health conditions, and typical daily activity patterns.

A policy-dashboard view providing aggregate trend summaries and multi-station comparison is *currently a conceptual and partially implemented component in the prototype*: the interface shell is present but connects to placeholder data in the current build, and full integration with the intelligence layer for this view is deferred to a subsequent development phase.

This system is a *functional prototype*. It has not undergone formal usability testing with representative user populations, security auditing, or production-hardening processes. It should not be interpreted as a deployment-ready system.

## VIII. RESULTS AND OBSERVATIONS

Given the prototype nature of Sanjeevan and the absence of a controlled user study or quantitative forecasting benchmark, this section reports qualitative observations from internal testing and structured scenario evaluation. Fabricating accuracy metrics or engagement statistics would misrepresent the system's current state of validation; the observations below reflect honest characterization of demonstrated prototype behavior.

## A. Interpretability Improvement

Across internal testing scenarios using real OpenAQ data from Pune and Delhi monitoring stations, the guidance engine consistently produced contextually appropriate advisory outputs across AQI categories. A user classified as high-risk (respiratory condition, elderly tier) received escalated guidance-recommending N95 mask usage and indoor activity-at AQI values where a standard-risk user received only a moderate caution. This differentiation, absent from all baseline platforms examined, represents a qualitatively meaningful improvement in guidance specificity.

The meteorological rationale strings produced by the forecasting engine were evaluated against general atmospheric dispersion principles [4]. In scenarios where elevated humidity and low wind speed co-occurred, the system correctly projected worsening trends; under active wind conditions with forecast precipitation, improving trends were correctly indicated. No formal skill score (e.g., Heidke Skill Score) has been computed; such validation requires ground-truth AQI time series paired with the meteorological inputs, which is a target for future work.

Table-2 illustrates representative system behavior across a selection of scenario combinations, demonstrating the consistency of guidance outputs with input conditions.

Table 2: Table-2: Example System Behavior Across Scenario Combinations

| AQI Category | Weather Condition              | User Type | Output Guidance                                                  |
|--------------|--------------------------------|-----------|------------------------------------------------------------------|
| Good         | High wind, clear sky           | Standard  | No restrictions; outdoor activity advised as normal.             |
| Moderate     | High humidity, calm wind       | Sensitive | Reduce prolonged outdoor exertion; monitor for symptoms.         |
| Poor         | Calm, fog or haze likely       | High-risk | Avoid outdoor activity; use N95 mask if travel is necessary.     |
| Very Poor    | Low wind, post-rain humidity   | Standard  | Limit outdoor time; keep windows closed during peak hours.       |
| Severe       | Stagnant air, inversion likely | High-risk | Stay indoors; seek medical advice if experiencing symptoms.      |
| Moderate     | Windy, precipitation forecast  | Standard  | Conditions likely to improve; brief outdoor activity acceptable. |

## B. Source Attribution Consistency

The heuristic source estimator produced internally consistent outputs under controlled test conditions. During simulated morning peak-hour scenarios with elevated NO<sub>2</sub> as the dominant pollutant, vehicular emissions received the highest estimated contribution weight, consistent with established emissions patterns for Indian urban environments [9]. During post-monsoon autumn scenarios (October simulation), the biomass burning category received elevated weighting, reflecting the seasonal crop residue burning pattern documented for northern India [10].

Importantly, the consistency of these outputs is a property of the rule set rather than an empirical validation of their accuracy. The estimates have not been compared against contemporaneous scientifically validated source apportionment data; any such comparison would require access to speciated chemical measurement data not available through OpenAQ.

## C. System Behavior Under API Constraints

Error handling and fallback behaviors were tested under conditions of API unavailability and missing station data. In the absence of OpenAQ data for a queried location (no stations within the default 25 km radius), the system surfaces an appropriate messaging state rather than displaying incorrect or stale data. Similarly, OpenAQ rate limit responses are caught and communicated to the user without crashing the application state. These behaviors confirm basic robustness of the implemented features appropriate to prototype operation.

## IX. LIMITATIONS

The following limitations are explicitly acknowledged and should be considered by any evaluator of this work:

**Absence of real-time sensor network.** Sanjeevan is entirely dependent on government-operated monitoring stations aggregated through OpenAQ. The spatial density of these stations is low in many Indian cities, and data latency can reach several hours. The system does not address the fundamental sensor coverage gap.

**No validated forecasting model.** The rule-based forecasting engine produces qualitative projections that have not been benchmarked against observed outcomes. The system makes no claim to quantitative forecasting accuracy; its projections should be treated as heuristic guidance rather than meteorological predictions.

**Heuristic source attribution.** The source estimation module does not perform receptor modeling, chemical mass balance analysis, or any form of scientific source apportionment. Its outputs are informed estimates derived from published emissions patterns and temporal/seasonal proxies. They carry significant uncertainty and may be systematically inaccurate in cities with atypical emissions profiles.

**Limited geographical accuracy.** The heuristic rules and seasonal signals embedded in the system are calibrated primarily against northern Indian urban contexts. Their applicability to southern Indian cities, coastal environments, or high-altitude regions has not been assessed.

## Dependency on public API availability and quality.

The system's data quality is a direct function of OpenAQ's data pipeline and CPCB's monitoring network reliability. Station outages, calibration drift, and reporting gaps in the underlying network are propagated to the user without the ability for correction.

**Policy dashboard incompleteness.** The policy-facing view is currently a conceptual and partially implemented component. Claims about its utility for policymakers are forward-looking and cannot be evaluated on the basis of the current prototype.

**Absence of formal user study.** Claims regarding improved interpretability and user decision support are based on internal qualitative assessment, not on a controlled study with representative participants. A formal usability evaluation is required before any conclusions about user benefit can be drawn with confidence.

## X. FUTURE WORK

Several development directions are identified as high-priority for advancing Sanjeevan beyond prototype status:

**ML-based forecasting integration.** Replacing the rule-based projections with a trained forecasting model—potentially a gradient-boosted regressor or a lightweight sequential architecture—trained on historical AQI and meteorological data from Indian cities would substantially improve quantitative forecasting fidelity. Interpretability tools such as SHAP (SHapley Additive exPlanations) could be applied to preserve the explainability property [14].

**Scientifically validated source attribution.** Integration with published PMF and CMB model outputs from IIT and government institutions, structured as a static knowledge base for major cities, would replace the current heuristic estimator with evidence-based attribution data. Periodic updates aligned with published annual apportionment studies would maintain relevance.

**Policy dashboard completion.** The multi-station aggregate view for policymakers, currently a conceptual and partially implemented component in the prototype, requires backend integration and potentially a server-side component for managing multi-city data aggregation and trend analysis at the scale required by municipal decision-making.

**Real-time sensor integration.** Partnerships with low-cost sensor networks—such as those deployed by citizen science initiatives or municipal smart city programs—could supplement the sparse government station coverage and provide hyperlocal AQI estimates.

**Formal evaluation and usability study.** A structured user study with participants across risk tiers, literacy levels, and urban contexts would provide the empirical basis for assessing Sanjeevan's decision-support value.

**Scalability to national deployment.** Architectural refactoring to introduce a backend API service layer would enable multi-city scalability, server-side caching, and push notification delivery for threshold exceedance alerts—features that are not feasible within the current fully client-side architecture.

## XI. CONCLUSION

Sanjeevan represents a deliberately modest but practically motivated contribution to the problem of air quality decision support in urban India. Its central argument is that the gap between AQI data availability and actionable citizen guidance is not primarily a sensing problem—it is an intelligence and interface design problem. Existing platforms produce accurate measurements but do not reason about them; Sanjeevan demonstrates that even lightweight rule-based reasoning, applied systematically to publicly available data, can meaningfully enrich the interpretive layer between raw AQI values and real-world decisions.

The system's strengths lie in its explainability-first design, its zero-dependency deployability on commodity mobile hardware, and its dual-user architecture that serves both individual citizens and institutional stakeholders within a unified data pipeline. Its limitations—particularly the heuristic nature of its source attribution, the incompleteness of its policy dashboard, and the absence of formal validation—are acknowledged without reservation. These limitations define the boundary between prototype and production system; crossing that boundary is the work of the research trajectory that this paper initiates.

Air pollution in India will not be resolved by a mobile application. But the translation of scientific monitoring data into intelligible, actionable, and personalized guidance is a necessary condition for informed public and policy response—and that translation is precisely the function that systems like Sanjeevan are designed to perform.

## ACKNOWLEDGEMENT

This work was supported by the infrastructure and computational resources provided by MIT ADT University. We acknowledge the OpenAQ and Open-Meteo projects for providing freely accessible APIs that enabled this research. We are grateful to the Ministry of Earth Sciences and the Central Pollution Control Board for their commitment to open air quality data dissemination. The authors thank the internal reviewers whose feedback improved the clarity and completeness of this work.

## References

- [1] World Health Organization, “WHO Global Air Quality Guidelines: Particulate Matter (PM<sub>2.5</sub> and PM<sub>10</sub>), Ozone, Nitrogen Dioxide, Sulfur Dioxide and Carbon Monoxide,” WHO Press, Geneva, 2021.
- [2] J. Lelieveld, J. S. Evans, M. Fnais, D. Giannadaki, and A. Pozzer, “The contribution of outdoor air pollution sources to premature mortality on a global scale,” *Nature*, vol. 525, no. 7569, pp. 367–371, Sep. 2015.
- [3] SAFAR (System of Air Quality and Weather Forecasting and Research), Ministry of Earth Sciences, Government of India. [Online]. Available: <https://safar.tropmet.res.in>
- [4] J. H. Seinfeld and S. N. Pandis, *Atmospheric Chemistry and Physics: From Air Pollution to Climate Change*, 3rd ed. Hoboken, NJ: Wiley, 2016.
- [5] D. Byun and K. L. Schere, “Review of the governing equations, computational algorithms, and other components of the Models-3 Community Multiscale Air Quality (CMAQ) modeling system,” *Applied Mechanics Reviews*, vol. 59, no. 2, pp. 51–77, Mar. 2006.
- [6] Y. Zheng, F. Liu, and H.-P. Hsieh, “U-Air: When urban air quality inference meets big data,” in *Proc. 19th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD)*, Chicago, IL, USA, pp. 1436–1444, Aug. 2013.
- [7] B. S. Freeman, G. Taylor, B. Gharabaghi, and J. The’, “Forecasting air quality time series using deep learning,” *Journal of the Air & Waste Management Association*, vol. 68, no. 8, pp. 866–886, 2018.
- [8] L. Bai, J. Wang, X. Ma, and H. Lu, “Air pollution forecasts: An overview,” *International Journal of Environmental Research and Public Health*, vol. 16, no. 19, p. 3729, Oct. 2019.
- [9] S. Gupta, P. Choudhary, A. Kumar, J. George, T. Majumdar, and S. Hazarika, “Source apportionment of PM<sub>2.5</sub> in Delhi: An overview,” *Aerosol and Air Quality Research*, vol. 17, no. 10, pp. 2590–2607, 2017.
- [10] S. K. Guttikunda and P. Gurjar, “Role of meteorology, emission sources, and urban activity patterns on air quality of a polluted Indian megacity,” *Environmental Monitoring and Assessment*, vol. 184, no. 4, pp. 2277–2295, 2012.
- [11] Central Pollution Control Board (CPCB), Government of India, “National Air Quality Index,” CPCB Publication, New Delhi, 2014. [Online]. Available: <https://cpcb.nic.in>
- [12] OpenAQ, “OpenAQ: Open Air Quality Data Platform.” [Online]. Available: <https://openaq.org>
- [13] Open-Meteo, “Open-Meteo Free Weather API.” [Online]. Available: <https://open-meteo.com>
- [14] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 4765–4774, 2017.