# A Traceable, Human-In-The-Loop Framework for Detecting Anomalous Automatic Weather Station Observations from The India Meteorological Department Using Isolation Forest

Mubraali L[1], Ashmitha T.H[2], Harini Bala R[3], Kaviksha B[4], Mathesh N[5], Nithin Johar D[6]
[1,2,3,4,5,6]*Department of Computer Science and Engineering, SNS College of Technology, Coimbatore, Tamil Nadu, India*

***Abstract*—Automatic Weather Stations (AWS) supply continuous records of temperature, atmospheric pressure, and humidity, but sensor faults, transmission errors, and genuine extreme events can all appear as unusual values, and distinguishing among them requires review. This paper presents SkyGuard AI, a software framework that retrieves available observations from the India Meteorological Department (IMD) API, processes and validates them, estimates relative humidity from temperature and dew-point measurements, and applies an Isolation Forest model trained on historical IMD observations to flag records that deviate from learned patterns.**

**Results are shown on an interactive dashboard that supports analyst review.The framework comprises an ingestion layer, a data-validation and feature layer, an unsupervised anomaly-scoring layer, a persistence layer, and a presentation layer.**

**Anomaly scores are treated as reviewable signals, not as confirmed sensor faults or confirmed weather events, and the system is intended to assist qualified personnel rather than replace established meteorological quality-control procedures. Contributions are an integration architecture, a data-processing and provenance model, a risk-oriented treatment of unsupervised model output, and a reproducible evaluation protocol covering detection quality, latency, robustness to missing data, and usability.**

**No measured performance results from operational deployment are reported; the paper defines an evaluation plan rather than claiming effectiveness.**

***Index Terms*—anomaly detection, automatic weather station, Isolation Forest, India Meteorological Department, data quality, human-in-the-loop**

## I. INTRODUCTION

### *A. Meteorological context*

Weather observations underpin forecasting, agriculture advisories, disaster preparedness, and climate records. Such data also inform renewable energy planning, an area where AI-based computational design and optimization are increasingly applied [1]. Automatic Weather Stations (AWS) record parameters such as air temperature, atmospheric pressure, and humidity at regular intervals without a human observer, and the World Meteorological Organization (WMO) provides guidance on instrument performance and observing practice [2]. The India Meteorological Department (IMD) operates a national network of observing systems, including AWS, and publishes observations for use by researchers and the public [3]. Because such data feed downstream decisions, reviewers need to know which readings are unusual, why they were flagged, and whether an authorized person examined them.

### *B. Technical problem*

An unusual reading may result from a failed or drifting sensor, a communication or decoding error, a station reset, or a real but rare weather event. Fixed range checks catch physically impossible values but can miss readings that are plausible individually and inconsistent jointly, for example a temperature, pressure, and humidity combination that has not occurred at that station before. Surveys of anomaly detection describe point, contextual, and collective

anomalies and stress that the appropriate method depends on the data and on the cost of errors [4]. The design problem is therefore not only to detect outliers but to preserve provenance, expose uncertainty, and keep a human reviewer in the decision path.

*C. Positioning*

Isolation Forest is an established unsupervised method that isolates observations using random partitions [5]. This paper does not propose a new anomaly-detection algorithm, a new humidity formula, or a new quality-control standard. It addresses a system-integration problem: how live IMD observations, derived variables, model scores, and analyst review can be combined in one traceable, inspectable application. The work follows a design-oriented approach in which an artifact is specified and its evaluation criteria are defined before empirical validation [6].

*D. Aim and objectives*

The aim is to specify a modular application for detecting and reviewing anomalous AWS observations. Objectives are to: (i) define system layers and data flow; (ii) specify validation and humidity-estimation steps; (iii) describe model training and scoring on historical IMD data; (iv) define how results are presented without overstating certainty; and (v) provide a reproducible evaluation protocol.

*E. Research questions and contributions*

RQ1: How can retrieval, validation, derived-variable estimation, scoring, and review be represented as one traceable workflow? RQ2: Which controls prevent an anomaly score from being mistaken for a confirmed fault or event? RQ3: Which measurements are needed before operational claims can be made? These questions guide the design and are not answered by experimental results. The bounded contributions are: (i) an integration architecture; (ii) a data-processing and provenance model; (iii) a risk-oriented treatment of model output as a reviewable signal; and (iv) a controlled evaluation protocol.

## II. RELATED WORK AND TECHNICAL CONTEXT

*A. Quality control of automatic station data*

Operational meteorology applies quality-control checks such as range, step, persistence, and internal-consistency tests to station data, and WMO guidance describes these as part of managing observation quality [2], with dedicated guidelines for automatic station data [7], [8]. Such rule-based checks are transparent but require thresholds that differ by station, season, and parameter. Learned models can complement rules by capturing multivariate patterns, but they do not establish the physical cause of a deviation.

*B. Isolation Forest*

Isolation Forest builds an ensemble of random trees that recursively split data on randomly chosen features and split values; points that are separated after few splits receive higher anomaly scores [5]. For a sample of size n, the score of an observation x is

$$s(x, n) = 2^{-E[h(x)] / c(n)} \quad (1)$$

where h(x) is the path length of x in a tree, E[h(x)] is its average over the ensemble, and $c(n) = 2H(n-1) - 2(n-1)/n$ is the average path length of an unsuccessful search, with H(i) the harmonic number [5]. Scores near 1 indicate likely anomalies.

Extensions that replace axis-parallel splits with random hyperplanes reduce score bias along coordinate axes [9]. The method needs no labelled anomalies, which suits AWS archives where confirmed faults are rarely annotated, and an implementation is available in scikit-learn [10]. Extended analyses of the method appear in [11], and density-based alternatives such as the local outlier factor [12] provide comparison baselines. Broader surveys of outlier detection are given in [13]. The contamination parameter and decision threshold, however, directly set how many records are flagged and must be justified rather than assumed.

*C. Humidity estimation*

Where relative humidity is not reported directly, it can be estimated from air temperature T and dew-point temperature Td (both in °C) using a Magnus-type approximation, whose coefficients have been evaluated against reference vapour-pressure data [14]:

$$RH = 100 \cdot \exp(a \cdot Td / (b+Td)) / \exp(a \cdot T / (b+T)) \quad (2)$$

with a = 17.625 and b = 243.04 °C as one published coefficient set [14]. The estimate inherits sensor error from both inputs, and values above 100% or with Td greater than T indicate a data problem that should be flagged, not silently clipped.

*D. Design implications*

The literature supports cautious use of each component, not a presumption of end-to-end effectiveness. Transferable principles are to validate inputs before modelling, to record model and parameter versions, to expose uncertain results, and to evaluate the complete workflow in addition to the individual algorithm.

*E. Dashboards and decision support*

Interactive dashboards allow analysts to filter by station and period, inspect context around a flagged record, and compare parameters side by side. Evidence from decision-support research suggests that presenting a score without context can encourage over-reliance [15], so the interface should show the underlying values, the neighbouring observations, and the model version next to each flag. Alert volume is also a practical concern: if too many records are flagged, reviewers may stop examining them, so the review burden is treated as a measured quantity in the evaluation protocol rather than as an afterthought.

*F. Rule-based versus learned detection*

Rule-based checks and learned models fail in different ways, which motivates combining them (Table I). Rules are auditable and cheap but rigid; Isolation Forest adapts to multivariate patterns but cannot explain a flag in physical terms and can be affected by masking, where clusters of similar anomalies look normal, and by swamping, where normal points are wrongly flagged near dense anomalous regions [4], [5]. SkyGuard AI therefore applies validation rules first and model scoring second.

TABLE I Qualitative comparison of detection approaches

| Approach | Strengths | Limitations |
|---|---|---|
| Range / step rules | Transparent; catches impossible values | Fixed thresholds; misses joint anomalies |
| Statistical (z-score) | Simple; fast | Assumes stable distribution; univariate |
| Isolation Forest | Unsupervised; multivariate; scalable | Hard to explain; sensitive to threshold |
| Hybrid (proposed) | Rules first, model second, human last | Needs tuning and review capacity |

## III. SYSTEM ARCHITECTURE AND WORKFLOW

*A. Five-layer architecture*

Layer 1 (presentation) is an interactive dashboard that shows stations, time series, flagged observations, and analyst status. Layer 2 (ingestion and orchestration) retrieves available observations from the IMD API on a schedule and coordinates processing. Layer 3 (analytics services) performs cleaning, humidity estimation, feature assembly, and Isolation Forest scoring; each service returns a typed result with a reason code and version. Layer 4 (persistence) stores raw and processed observations, scores, and review records. Layer 5 (observability and governance) provides logs, metrics, retention rules, and access review. Fig. 1 summarizes the layers.

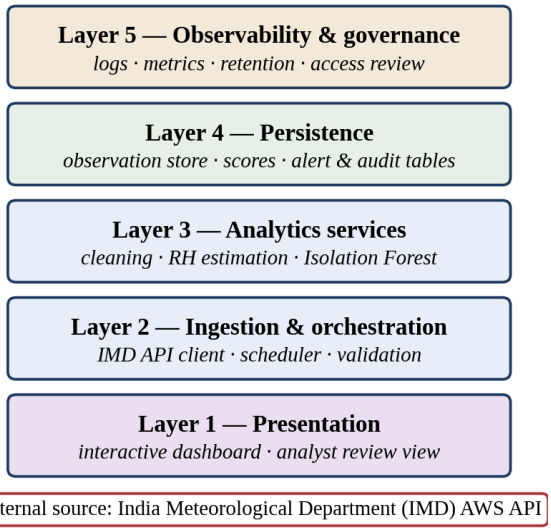


Fig. 1. Five-layer architecture of SkyGuard AI with the external IMD data source.

*B. Data-processing flow*

The flow is: IMD API → parsing and validation → humidity estimation → feature assembly → Isolation Forest scoring → thresholding and reason annotation → dashboard and analyst review (Fig. 2). Raw observations are retained unchanged alongside derived values so that any flagged record can be traced to its source response, retrieval time, and processing version. Records that fail schema or range validation are labelled with a reason code and kept separate from model-scored records, and missing parameters are recorded as missing instead of imputed without notice.

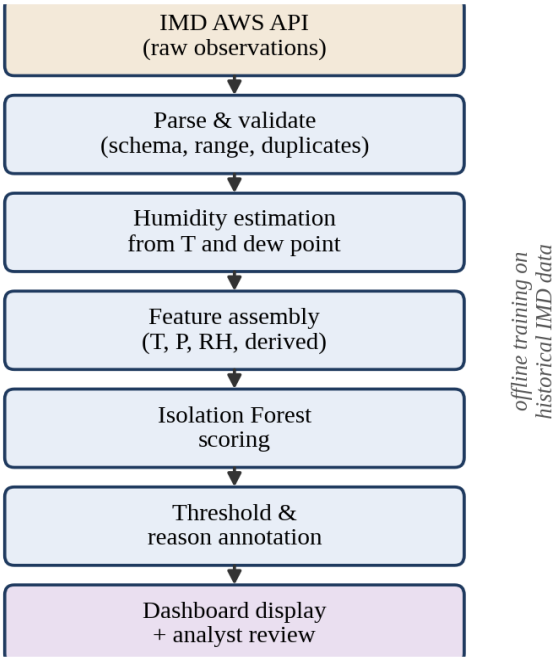


Fig. 2. Data-processing pipeline from IMD API retrieval to analyst review. The model is trained offline on historical IMD data.

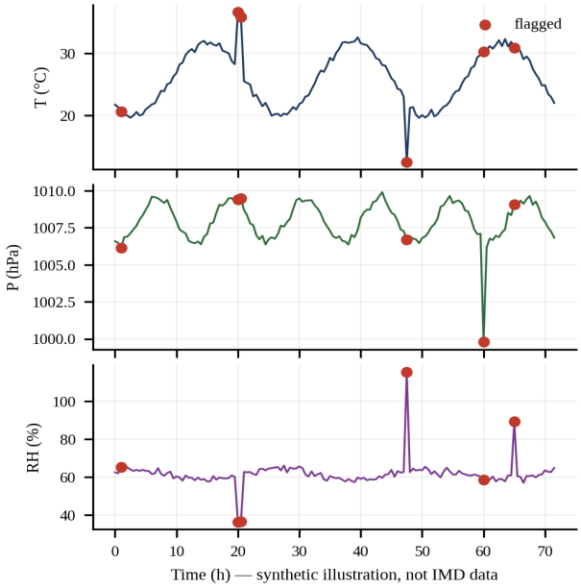


Fig. 3. Illustrative output on synthetic data (not IMD data): a multivariate Isolation Forest flags records across temperature, pressure, and humidity. Flagged points are candidates for review, not confirmed faults.

*C. Model training and scoring*

The model is fitted offline on historical IMD observations using temperature, pressure, and estimated or reported humidity as features, after standardization. Training data should be split chronologically, and station, season, and time-of-day effects should be examined, since a single global model may flag normal regional differences. The fitted model, scaler, parameters (for example, the number of trees, sub-sample size, and contamination value), and training-data window are stored with a version identifier. At run time each new record receives a score and a binary flag against the configured threshold, and both are saved with the model version.

*D. Human-in-the-loop review*

A flagged record is displayed as a candidate anomaly, together with its score, the contributing parameter values, and neighbouring observations (Fig. 3 and Fig. 4). The interface uses wording such as “flagged for review”, not “faulty” or “error”. An analyst may confirm, dismiss, or annotate a flag, and each decision is stored with the analyst identity and time. These decisions can later serve as labelled data for evaluation, but should not automatically retrain the production model without governance approval.

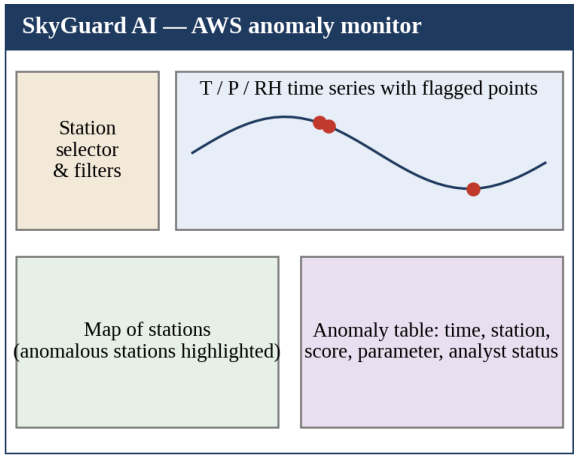


Fig. 4. Wireframe of the interactive dashboard: station filters, time series with flagged points, station map, and anomaly review table.

*E. Input validation rules*

Before modelling, each record passes configurable checks (Table II). The limits shown are illustrative defaults that must be set per station group with domain experts; they are not claims about the physical limits of any specific site. Records failing a rule are stored with a reason code and excluded from model scoring, but remain visible to reviewers so that data problems are not hidden.

TABLE II Illustrative validation rules applied before scoring

| Parameter | Rule (configurable) | Action on failure |
|---|---|---|
| Timestamp | Parsable, not in future, not duplicate | Reject; log reason code |
| Temperature (T) | Within station-group range; step limit between reports | Flag as invalid; exclude from scoring |
| Pressure (P) | Within station-level range; step limit | Flag as invalid; exclude from scoring |
| Dew point (Td) | Td ≤ T; both present | Skip RH estimate; mark missing |
| Relative humidity | 0–100% after estimation | Flag; do not clip silently |
| Completeness | Required fields present | Store as incomplete |

*F. Threshold selection*

The anomaly threshold determines the trade-off between missed anomalies and review workload. Rather than fixing it by convention, the threshold is chosen on a validation period using injected faults and expert-labelled samples, and then frozen and versioned. Fig. 5 illustrates, on synthetic data, how scores of normal and injected records separate and where a threshold may fall; overlap between the distributions shows why some flags are expected to be false alarms and some anomalies missed.

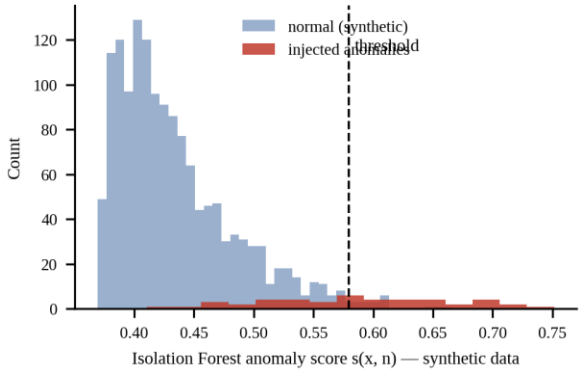


Fig. 5. Illustrative Isolation Forest score distribution on synthetic data (not IMD data) with a candidate threshold. Real distributions must be estimated from IMD observations.

*G. Provenance data model*

Each processed record carries: station identifier; observation time reported by the source; retrieval time recorded by the server; raw payload reference; validation outcome and reason code; derived humidity value and formula version; feature vector; model identifier and training window; score and threshold; flag status; and analyst decision with identity, time, and note. Separating observation time from retrieval time supports detection of late or repeated transmissions, and storing raw payloads allows a flag to be re-examined if a parsing rule changes. Corrections are appended as new entries so that earlier states remain available for audit.

*H. Deployment considerations*

Retrieval should respect API availability and any usage limits, apply bounded retries with back-off, and treat an empty or malformed response as an event to log and not as a normal observation. Credentials and configuration are kept outside source code, and access to the review interface is limited to authorized users. Because the IMD interface may change, parsing is isolated in a single adapter with versioned tests, so that format changes do not silently alter the data that reach the model. Periodic re-evaluation is needed to detect model drift, and retraining is a controlled, versioned release.

*I. Feature design*

The baseline feature set is temperature, pressure, and humidity. Optional derived features can be evaluated one at a time: dew-point depression (T − Td), the change since the previous report, hour of day encoded as sine and cosine to respect its cyclic nature, and day of year. Each added feature must show a measured benefit on validation data, because extra features can increase false alarms and make flags harder to interpret. All features are standardized using statistics from the training window only, and the same transformation is applied at run time to avoid leakage between training and evaluation.

*J. Example of a stored review record*

Table III shows the fields shown to an analyst for one flagged record. The values are fictitious and serve only to illustrate the structure.

TABLE III Fictitious example of a flagged record

| Field | Example value |
|---|---|
| Station ID | STN-0001 (placeholder) |
| Observation time | 2026-01-01 09:00 (source) |

| Field | Example value |
|---|---|
| Retrieved at | 2026-01-01 09:05 (server) |
| T / P / RH (est.) | 36.2 °C / 998.1 hPa / 22.4 % |
| Score / threshold | 0.68 / 0.60 |
| Model version | iforest-v1 (training window recorded) |
| Flag status | Flagged for review |
| Analyst decision | Pending (confirm / dismiss / note) |

*K. Worked example of humidity estimation*

Consider a record with T = 30 °C and Td = 20 °C. Substituting into (2) gives a·Td/(b+Td) = 1.340 and a·T/(b+T) = 1.937, so RH = 100·exp(1.340 − 1.937) ≈ 55.1%. If a faulty record instead reports Td = 31 °C with T = 30 °C, the same formula returns about 105.9%, which exceeds 100% and also violates the rule Td ≤ T. Such a record is rejected by validation and reported with a reason code, so it never reaches the model as a normal-looking value. A record that passes validation but whose combination of T, P, and RH is rare for the station receives a high score and is flagged for review.

*L. Intended users and scenarios*

Three scenarios guide the design. In routine monitoring, an analyst opens the dashboard, selects a station group, and reviews flagged records from the latest retrieval cycle. In fault triage, a maintenance engineer examines a station with repeated flags, compares its recent records with neighbouring stations, and decides whether an inspection is warranted. In event review, a forecaster checks whether a cluster of flags across nearby stations coincides with a real weather event rather than a sensor problem. In each case the application supplies evidence for a human decision; it does not issue maintenance orders or declare an event.

## IV. IMPLEMENTATION MAPPING AND STATUS

Table IV is a provisional mapping based on the project description, not a code audit. “Implemented” means an artifact is reported to exist and does not mean it has passed systematic testing; “Planned” means the capability is specified but not verified. Implementation status must not be confused with validated operational effectiveness.

TABLE IV Provisional implementation mapping and evidence needed

| Module | Artifact to verify | Provisional status / evidence |
|---|---|---|
| IMD retrieval | API client, request logs | Implemented; failure and empty-response tests |
| Validation | Schema and range rules | Partially implemented; rule test cases |
| Humidity estimation | RH from T and Td | Implemented; comparison with reported RH |
| Isolation Forest | Trained model, scaler, version | Implemented; held-out and injected-fault tests |
| Dashboard | Interactive views | Implemented; usability test with reviewers |
| Review log | Analyst decisions, audit | Planned; unauthorized-edit tests |
| Scheduling | Periodic retrieval job | Planned; latency and retry tests |

## V. EVALUATION PROTOCOL

*A. Data and labelling*

Evaluation should use a chronologically held-out period of IMD observations not seen in training. Because confirmed anomalies are scarce, two label sources are proposed: (i) synthetic faults injected at known positions (spikes, stuck values, drift, and dropouts) with documented magnitudes, and (ii) expert annotations of a sampled subset of flagged and unflagged records by qualified meteorological staff, with inter-rater agreement reported.

*B. Measures and baselines*

Detection quality is reported as precision, recall, F1, and area under the precision–recall curve, per parameter and per station group, with confidence intervals. Baselines should include fixed range and step checks and a z-score rule so that the benefit of the learned model is measured and not assumed. System measures include retrieval-to-display latency, processing throughput, behaviour under missing or malformed API responses, and reproducibility of results from stored model versions. Usability is assessed with task-based sessions and a standard questionnaire such as the System Usability Scale [16],

and review effort is recorded as the number of flags per station per day.

TABLE V Planned evaluation measures (no results reported)

| Dimension | Measure | Reporting field |
|---|---|---|
| Detection | Precision, recall, F1, PR-AUC | Per parameter, station, season |
| Robustness | Missing/malformed input handling | Failure modes, recovery |
| Performance | End-to-end latency, throughput | Median, 95th percentile |
| Usability | Task success, questionnaire | Participants, tasks, scores |
| Reproducibility | Re-run with stored version | Model ID, data window, seed |

*C. Threats to validity*

Injected faults may be easier or harder to detect than real ones, so results on them are reported separately from results on expert-labelled real data. Expert labels are themselves uncertain, and annotators should be blind to model scores where feasible. Station networks differ in climate, instrumentation, and reporting frequency, so findings from one region or period should not be generalized without testing on others. Random seeds and sampling choices in Isolation Forest can change borderline scores, so repeated runs and their variation are reported. Table V lists the planned reporting fields.

*D. Reporting requirements*

Any future report should state the data period, station list, number of records before and after validation, handling of missing values, model parameters and version, threshold and how it was chosen, baseline methods, label sources, and all evaluation measures with uncertainty. Negative and inconclusive results should be reported alongside positive ones, and operational claims should be limited to the conditions that were actually tested.

*E. Staged pilot plan*

Evaluation proceeds in three stages. Stage 1 is an offline replay, in which archived IMD observations are streamed through the pipeline to test correctness, reproducibility, and latency without touching live operations. Stage 2 is a shadow run on live retrieval, where flags are generated and stored but reviewed only by the project team, to estimate alert volume and stability. Stage 3 is a supervised pilot in which authorized reviewers use the dashboard and record decisions, with agreed stopping criteria if flag volume, error rates, or usability problems exceed pre-set limits. Progress from one stage to the next requires that the previous stage's reporting fields are complete.

## VI. GOVERNANCE, LIMITATIONS, AND RESPONSIBLE USE

*A. Interpretation risks*

An anomaly score is not a diagnosis. A high-scoring record may be a rare but genuine weather event, and suppressing it as an error would remove valuable information; a low-scoring record may still be wrong if the fault resembles typical data. For this reason, the interface uses neutral wording, shows the supporting values, and requires an analyst decision before any record is labelled as a fault or an event.

*B. Model drift and generalization*

The model reflects the historical period, stations, and conditions on which it was trained and may degrade under climate variability, new station hardware, seasonal shifts, or changes in the API format. Adapting to such changes in the data distribution is a well-studied problem known as concept drift [17]. A global model may flag normal regional differences, whereas station-specific models need enough history at each site.

Monitoring of flag rates over time, with alerts for sudden changes, helps distinguish drift from genuine events. Retraining is a controlled, versioned release with a documented evaluation, not an automatic background process.

*C. Data use and access*

Public data should be used in line with IMD terms, and the application should not republish data in ways the source does not permit. Review actions are attributed to named users, access is limited to authorized personnel, and the model version, training window, and threshold are visible to dashboard users. Any operational use should be agreed with the responsible meteorological authority. The framework does not replace formal quality-control procedures, and the humidity values it estimates are approximations, not instrument measurements.

*D. Limitations of this work*

This paper specifies a design and evaluation plan. It reports no performance measurements on IMD data, no comparison with operational quality-control systems, and no user study. Figures labelled synthetic are illustrations produced to explain the method and must not be read as results. The implementation status in Table IV is provisional and should be verified against the source repository before submission or citation.

## VII. FUTURE WORK

Five directions follow from the design. First, execute the evaluation protocol on held-out IMD observations and report all measures with uncertainty. Second, compare station-specific, regional, and global models, and test additional detectors such as local outlier factor and autoencoder-based methods against the Isolation Forest baseline.

Third, add contextual features such as time of day and season, and spatial comparison with neighbouring stations, because a reading that is unusual for one station but consistent with its neighbours is more likely to be a real weather event than a sensor fault.

Fourth, add explanation support, for example by showing which parameter contributed most to a flag using feature-attribution methods [18], so that reviewers can judge it faster. Fifth, conduct a supervised pilot with meteorological reviewers and use their confirmed decisions to build a labelled evaluation set.

## VIII. CONCLUSION

SkyGuard AI specifies a traceable workflow that connects IMD AWS observations, validation, derived humidity estimation, unsupervised anomaly scoring, and analyst review in one interactive application. Its main design principle is to treat model output as a reviewable signal with preserved provenance, so that flagged records support human judgment instead of replacing it.

No operational effectiveness is claimed; the evaluation protocol, staged pilot plan, and reporting requirements define what must be demonstrated before such claims can be made.

## DECLARATION

A generative AI tool was used solely for grammar checking of the manuscript text. The authors reviewed all content and take full responsibility for the manuscript.

## REFERENCES

[1] R. Rasappan Senthil Kumar, U. Arun Kumar, A. Flah, and H. Kraiem, "Computational design and AI optimization of a symmetry-engineered ultra-broadband metasurface solar absorber for renewable energy systems," *Energy Reports*, vol. 16, Art. no. 109524, 2026, doi: 10.1016/j.egyr.2026.109524.

[2] World Meteorological Organization, *Guide to Instruments and Methods of Observation*, WMO-No. 8. Geneva, Switzerland: World Meteorological Organization.

[3] India Meteorological Department, Ministry of Earth Sciences, Government of India, "Automatic weather stations and observational network." IMD, New Delhi, India.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58, 2009.

[5] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, 2008, pp. 413–422.

[6] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[7] M. Zahumenský, *Guidelines on Quality Control Procedures for Data from Automatic Weather Stations*. Geneva, Switzerland: World Meteorological Organization, 2004.

[8] J. Estévez, P. Gavilán, and J. V. Giráldez, "Guidelines on validation procedures for meteorological data from automatic weather stations," *Journal of Hydrology*, vol. 402, nos. 1–2, pp. 144–154, 2011, doi: 10.1016/j.jhydrol.2011.02.031.

[9] S. Hariri, M. Carrasco Kind, and R. J. Brunner, "Extended isolation forest," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1479–1489, 2021, doi: 10.1109/TKDE.2019.2947676.

[10] F. Pedregosa *et al*., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[11] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 1, Art. no. 3, 2012.

[12] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, 2000, pp. 93–104.

[13] V. J. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial Intelligence Review*, vol. 22, no. 2, pp. 85–126, 2004.

[14] O. A. Alduchov and R. E. Eskridge, "Improved Magnus form approximation of saturation vapor pressure," *Journal of Applied Meteorology*, vol. 35, no. 4, pp. 601–609, 1996.

[15] R. Parasuraman and D. H. Manzey, "Complacency and bias in human use of automation: An attentional integration," *Human Factors*, vol. 52, no. 3, pp. 381–410, 2010.

[16] J. Brooke, "SUS: A quick and dirty usability scale," in *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, Eds. London, U.K.: Taylor & Francis, 1996, pp. 189–194.

[17] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, Art. no. 44, 2014.

[18] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017, pp. 4765–4774.