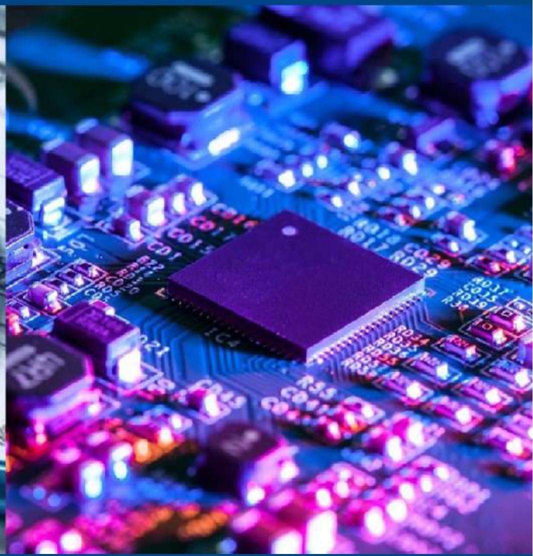
ISSN: 2582-7219

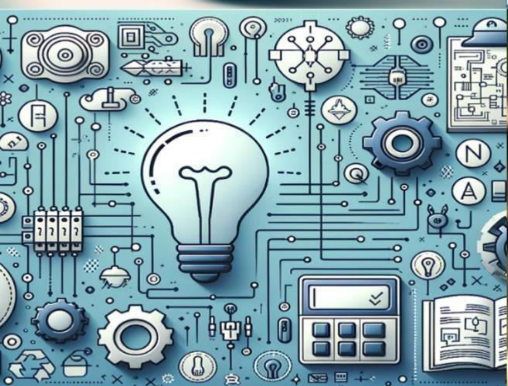




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.206

Volume 9, Issue 1, January 2026

www.ijmrset.com ijmrset@gmail.com +91 6381 907 438 +91 63819 07438

© 2026 IJMRSET | Volume 9, Issue 1, January 2026| DOI:10.15680/IJMRSET.2026.0901022

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Living Weather Dataset (Atmosight – Intelligent Weather Insight System)

Saurabh Kumar Barhmbhatt, Prof. Gunasekaran K

Student, Department of MCA, AMC Engineering College, Bengaluru, India

Assistant Professor, Department of MCA, AMC Engineering College, Bengaluru, India

ABSTRACT: The Living Weather dataset is a web-based weather monitoring system that manages and maintains a continuously updating weather dataset. Weather data gathered from online sources can contain errors or unusual values due to sudden changes or issues with APIs. To address this, the system automatically stores weather data and detects anomalies using statistical methods. When it identifies abnormal temperature values, the system corrects the data and keeps a clear explanation for each correction.

The application also offers short-term weather forecasts for upcoming days using basic data analysis techniques. All data is kept in a local database, making it simple to track historical changes and system activities. The user interface features a modern glassmorphism style, animations, and dynamic visuals of cities or states to enhance user interaction. Auto-refresh and notification features ensure that users always see updated and reliable information. Overall, this project shows an effective way to create a “living dataset” that continuously updates, cleans, and improves data quality for accurate weather analysis.

KEYWORDS: Weather Monitoring System, Living Dataset, Anomaly Detection, Self-Healing Data, Temperature Analysis, Statistical Data Cleaning, Z-Score Method, Weather Prediction, Flask Web Application, SQLite Database, Real-Time Weather Data, API Integration, Data Visualization, Glassmorphism UI, Auto Refresh System, Predictive Analysis, Smart Data Management

I. INTRODUCTION

Weather data is commonly used in areas like forecasting, agriculture planning, travel, and environmental analysis. However, data collected from online sources isn't always accurate. It can include sudden unusual values, missing entries, or inconsistent readings due to sensor errors, network issues, or API limitations. These problems lower the reliability of the datasets and impact decision-making and predictions.

The Living Weather Dataset project aims to tackle these challenges by creating a smart and dynamic weather data management system. The main idea is to treat weather data as a living dataset that updates continuously, monitors itself, and improves data quality over time. The system automatically collects temperature data from online weather APIs and stores it in a local database for analysis and tracking.

To ensure data reliability, the Living Weather Dataset system detects unusual temperature values using statistical methods. It also performs a self-healing process to fix these anomalies. Each correction is saved along with an explanation, which makes the dataset transparent and trustworthy. The system provides short-term weather predictions to help users understand future trends.

The project is built as a web application using Flask. It offers a modern user interface with visual elements, animations, and auto-refresh functionality. Overall, the Living Weather Dataset project shows an effective way to keep weather data accurate, reliable, and continually improving for analysis and monitoring.

II. OBJECTIVES

Real-Time Data Collection:
The main goal of the Living Weather Dataset is to gather live weather data with minimal delay from online APIs. The system constantly retrieves temperature information and updates the dataset automatically. This ensures that the data always reflects current weather conditions.

Continuous Data Monitoring:
The project continuously monitors incoming weather data to ensure consistency and accuracy. Each new data point is analyzed as it enters the system, making the dataset active and responsive instead of static.

Anomaly Detection:
A key goal is to find unusual or unrealistic temperature values that may arise from API errors, sensor problems, or sudden spikes. The system employs statistical methods to detect these anomalies automatically without requiring manual checks.

Self-Healing Data Mechanism:
The project aims to fix detected anomalies through a self-healing process. When incorrect values are identified, they are replaced with statistically valid values based on nearby data points, which improves overall dataset reliability.

Explanation and Transparency:
Another goal is to keep data correction transparent. Each corrected data point is stored with a clear explanation, helping users understand what was changed and why, which builds trust in the dataset.

Short-Term Weather Prediction:
The Living Weather Dataset seeks to provide short-term temperature predictions based on historical data. These predictions help users analyze upcoming weather trends and improve the practical usefulness of the dataset.

Unified Dashboard:
The project aims to display all information through a single, easy-to-use dashboard. Live data, anomaly status, corrected values, dataset health, and predictions are shown together to simplify the experience for users.

Automation and Auto-Refresh:
Automation is a key goal of the system. The dataset refreshes itself at regular intervals, which reduces user effort and ensures continuous monitoring.

Data Storage and Historical Analysis:
Another aim is to save both live and historical weather data in a database. This enables users to review past records, study trends, and assess how the dataset improves over time.

Reliable and Scalable Architecture:
The project intends to use a lightweight but reliable structure that supports continuous data flow and processing. The system is designed to remain stable during frequent updates and allow for easy future improvements.

User Empowerment:
The final goal is to give users accurate, clean, and continuously improving weather data. By minimizing errors and manual work, the Living Weather Dataset helps users make better, more informed decisions based on reliable information.

III. PROBLEM STATEMENT

Weather data is commonly used for monitoring, analysis, and decision-making in areas like agriculture, travel, environmental studies, and daily planning. However, weather data from online sources and APIs is often unreliable. It can have sudden abnormal values, missing records, network failures, or errors related to sensors. These problems create datasets that are not fully reliable for analysis or short-term predictions.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Most existing weather systems primarily display real-time data but do not actively check data quality. Once wrong or unusual values enter the dataset, they stay unchanged and continue to affect future analysis. Manually cleaning the data takes a lot of time and is inefficient. This method is impractical for datasets that are continuously updated in real-time.

Many systems also lack transparency. If they do make data corrections, these changes are often not clearly explained or logged. This lack of clarity can make it hard for users to understand how the data was changed and can lower their confidence in the dataset. Additionally, many systems do not offer a combined view that includes live data, anomaly detection, correction status, and predictions all in one place.

Therefore, a new system is needed that treats weather data as an active dataset. This system should continuously update itself, automatically detect and correct errors, maintain clarity in data handling, and present accurate information through one easy-to-use interface. The Living Weather Dataset project aims to tackle these problems by offering a reliable, self-healing, and ever-improving weather data management system.

IV. METHODOLOGY

Data Collection:
The Living Weather Dataset system starts by gathering real-time weather data from online weather APIs. It fetches temperature data at regular intervals with minimal delay to keep the dataset updated with current weather conditions. If live data is unavailable, the system can use sample data to ensure continuity.

Data Storage:
All collected weather data is stored in a structured local database. Each record includes the date, temperature value, data source, update time, and status information. Using a database allows for efficient retrieval, tracking historical values, and further analysis.

Continuous Monitoring:
As new data comes into the system, it is monitored continuously for consistency. The system processes data in order and prepares it for validation, ensuring that each data point is checked before being considered reliable.

Anomaly Detection:
Statistical techniques help detect abnormal temperature values. The system compares each data point with overall data patterns and identifies sudden spikes or unrealistic readings. These abnormal entries are marked as anomalies for further processing.

Self-Healing Process:
When an anomaly is found, the system automatically applies a self-healing process. It replaces the incorrect value with nearby valid data points or calculated averages. This correction improves data accuracy without any manual work.

Explanation and Logging:
Every detected anomaly and its corresponding correction is logged in the system. An explanation is included with the corrected value, providing transparency and allowing users to understand how and why the data was changed.

Prediction:
After cleaning and validating the dataset, the system performs short-term weather prediction using historical temperature data. It generates temperature estimates for the upcoming days to support analysis and trend observation.

Visualization and Dashboard:
The processed data is displayed on a single interactive dashboard. The dashboard shows live weather values, anomaly indicators, corrected data, dataset health percentage, and future predictions in a clear, user-friendly format.

Automation and Auto-Refresh:
The entire workflow is automated. Data collection, anomaly detection, correction, and display happen at regular intervals using auto-refresh functionality, ensuring continuous operation without user involvement.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

System Evaluation:

The system regularly checks the health of the dataset based on the number of corrected and valid records. This helps measure how effective the self-healing process is and the overall reliability of the data.

V. PROPOSED MODEL

The proposed model of the Living Weather Dataset aims to create a dynamic and smart system for managing real-time weather data. Unlike traditional weather systems that store raw data without checking it, this model treats weather information as a living dataset that continuously updates, monitors itself, and improves data quality over time. The system collects live weather data from online APIs and processes it automatically with minimal delay, ensuring up-to-date information at all times.

Once the data is collected, it is analyzed right away to check its accuracy and consistency. Statistical methods are used to find abnormal temperature values caused by sensor errors, network issues, or sudden changes. When such problems are detected, the system automatically activates a self-healing mechanism that corrects the incorrect values using nearby valid data points or calculated averages. This process happens without manual input and ensures that only reliable data stays in the dataset.

The corrected and verified data is then stored in a structured database that keeps both live and historical weather records. Each correction is logged with a clear explanation, providing transparency and allowing users to see how the dataset changes over time. In addition to cleaning data, the model makes short-term weather predictions using historical data, helping users analyze upcoming weather trends.

All processed information is shown through a web-based dashboard that displays live weather data, dataset health status, corrected values, and future predictions in a simple and user-friendly way. The entire model operates in an automated environment with continuous data refresh, ensuring reliable performance and minimal user effort. Overall, the proposed model of the Living Weather Dataset offers an efficient, self-healing, and trustworthy approach to managing real-time weather data.

VI. PROTOTYPE MODEL

The prototype model of the Living Weather Dataset showcases a working version of the proposed system. It illustrates how real-time weather data can be collected, processed, and maintained as a constantly improving dataset. The prototype is a web-based application that combines live weather APIs with backend processing and database storage. It aims to validate the main ideas of real-time data handling, anomaly detection, and self-healing before a full-scale launch.

In the prototype, weather data is collected regularly from online APIs and promptly stored in a local database. When new data arrives, the system analyzes temperature values to spot abnormal or inconsistent readings. If it finds any issues, the prototype implements a self-healing mechanism that corrects the data using nearby valid values. Each corrected record is saved with an explanation, helping users understand how data quality is maintained.

The prototype also features a basic prediction tool that uses historical temperature data to create short-term weather forecasts. All system outputs appear on a straightforward but interactive web dashboard. This dashboard displays live weather values, the status of corrected data, dataset health, and predictions. An auto-refresh function keeps the information current without needing manual updates.

Overall, the prototype model effectively demonstrates the possibility of the Living Weather Dataset concept. It shows that real-time data can be automatically monitored, corrected, stored, and visualized in an efficient and user-friendly way. The prototype lays the groundwork for future improvements, scalability, and the integration of more weather parameters in upcoming versions of the system.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VII. DETAILED OVERVIEW OF PAGES IN LIVING WEATHER DATASET

Integrated Single-Page Dashboard:
The Living Weather Dataset is built as a single-page web application that keeps all system functions in one dashboard. This setup eliminates the need to switch between multiple pages. Users can view live weather data, data quality indicators, and analytical outputs all in one place. The dashboard refreshes continuously, showcasing the "living" aspect of the dataset. By centralizing everything, the system simplifies the process and helps users focus on monitoring and understanding the weather data.

Real-Time Data Flow and Monitoring:
The system continuously gathers live temperature data from online weather APIs and processes it for display right away. Each new data point is monitored in real time to ensure consistency and accuracy. Sudden changes, missing values, or unrealistic readings are flagged as soon as they appear. This ongoing monitoring means data quality issues are fixed immediately rather than going unnoticed.

Anomaly Detection and Self-Healing Process:
A key part of the single-page system is its automated anomaly detection and self-healing feature. Statistical methods are used to find abnormal temperature values that stray significantly from normal patterns. When these values are identified, the system automatically corrects them using nearby valid data or calculated averages. The corrections replace the incorrect values in real time, and the system logs detailed explanations for each fix. This process enhances reliability while keeping data handling transparent.

Data Storage, History, and Prediction:
All validated and corrected weather data is stored in a structured database that keeps both live and historical records. Users can look at past temperature trends and see how the dataset has changed over time. Additionally, the system uses historical data to produce short-term weather predictions, displayed on the same page. These predictions improve the dataset's analytical value and help users anticipate the weather.

Visualization, Automation, and User Experience:
The single-page interface organizes information in a clean and user-friendly way. Dataset health indicators, corrected values, and prediction results are visible without overwhelming users. The auto-refresh feature ensures updates happen continuously without manual reloading, while background processes handle data fetching, validation, and correction smoothly. Overall, the Living Weather Dataset offers an efficient and reliable solution by combining automation, smart data processing, and clear visualization in one cohesive page.

System Reliability and Error Handling:
The system is built to remain stable even when external data sources encounter problems like API downtime or network delays. In these cases, the application continues to function using previously stored data and clearly shows the data status to users. This ensures ongoing monitoring and stops incorrect or incomplete information from affecting the dataset.

Scalability and Future Enhancement Support:
Though it's a single-page application, the system architecture allows for future growth. More weather parameters, better prediction models, or external integrations can be added without altering the core structure. This scalability guarantees that the Living Weather Dataset can grow over time while keeping its original design principles intact.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VIII. FUTURE UPDATES

Improving Weather Parameters Integration:
In future updates, the Living Weather Dataset can go beyond temperature data and include more weather details like humidity, rainfall, wind speed, air pressure, and air quality index. Adding these parameters will make the dataset more useful for analysis and environmental monitoring.

Better Prediction Models:
The current system offers basic short-term predictions based on past data. Future versions can use machine learning models to boost forecast accuracy. Methods like regression models, time-series analysis, or neural networks can create more reliable and longer-term weather forecasts.

User Alerts and Notifications:
Future updates may include systems to alert users about major weather changes, anomalies, or extreme conditions. Notifications can come through email, mobile alerts, or messaging apps, helping users stay informed even when they are not using the application.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Geographic Expansion and Multi-Location Support:
Right now, the system can be improved to support multiple cities or regions at once. Future updates can let users choose, compare, and monitor weather data from different locations on the same single-page interface.

Improved Data Visualization:
Future changes might include interactive charts, graphs, and trend visualizations to make weather analysis easier. Tools like line graphs, heat maps, and comparison views can help users better understand long-term patterns and anomalies.

Cloud Deployment and Scalability:
The application can move to cloud-based infrastructure in future updates to enhance scalability, availability, and performance. Cloud deployment will enable the system to manage larger datasets, handle more requests, and process data in real-time on a larger scale.

IX. CONCLUSION

The Living Weather Dataset project shows a smart and automatic way to handle real-time weather data in a single-page web application. By treating weather information as a constantly changing dataset, the system addresses the problems of traditional static weather data systems, which do not have ways to validate and correct data. The combination of collecting real-time data, ongoing monitoring, and automatic processing keeps the dataset accurate and current.

One of the key strengths of the project is its ability to identify unusual temperature readings and fix them through a self-healing mechanism. This method greatly improves data reliability and lessens the need for manual cleaning. Providing logs that explain each correction adds clarity and builds user trust in the dataset. Storing both live and historical data further increases the system's value for analysis.

The single-page dashboard is user-friendly, clearly displaying live data, dataset health indicators, corrected values, and short-term predictions in one place. Features like auto-refresh and background processing ensure smooth and continuous operation. Overall, the Living Weather Dataset presents a practical, scalable, and dependable solution for managing and analyzing real-time weather data, laying a solid groundwork for future improvements and applications.

REFERENCES

1. OpenWeatherMap-Official Weather API
Website: https://openweathermap.org/
Used for fetching real-time weather data such as temperature and weather conditions through APIs.
2. Flask Official Documentation
Website https://flask.palletsprojects.com/
Refer to this for building the backend web application, routing, and server-side processing.
3. SQLite Database Documentation
Website https://www.sqlite.org/docs.html
Used for understanding lightweight database management, data storage, and query handling for live and historical weather data.
4. Python Official Documentation
Website: https://docs.python.org/3/
Used for implementing backend logic, data processing, automation, and integration with APIs.
5. NumPy and Pandas Documentation
Websites:
https://numpy.org/doc/
https://pandas.pydata.org/docs/
Used for statistical analysis, anomaly detection, data manipulation, and prediction logic.
6. Research Papers on Anomaly Detection in Time-Series Data
Referred to for understanding statistical techniques, such as z-score methods, used for detecting abnormal values in continuous datasets.
7. Web Resources on Data Cleaning and Self-Healing Systems
Used to study concepts related to automated data validation, correction mechanisms, and maintaining reliable datasets.

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com