International Journal of Science and Research Archive
eISSN: 2582-8185
Cross Ref DOI: 10.30574/ijsra
Journal homepage: https://ijsra.net/

(RESEARCH ARTICLE)

Check for updates

Serverless data processing and forecasting system using cloud computing

Dhananjay Adikane [1,*] and Dr. Odelu Ojjela [2]

[1] *Department of Applied Mathematics (Data Science) Defence Institute of Advanced Technology (DIAT), Pune, India.*
[2] *Department of Applied Mathematics, Defence Institute of Advanced Technology (DIAT), Pune, India.*

International Journal of Science and Research Archive, 2025, 15(02), 706-711

Publication history: Received on 01 April 2025; revised on 11 May 2025; accepted on 14 May 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.15.2.1394

Abstract

This paper presents a serverless data processing architecture designed for real-time weather forecasting using Google Cloud Functions, BigQuery, and integrated machine learning models. The system enables scalable ingestion of weather data, efficient storage, and dynamic visualization through an interactive dashboard. In addition to predictive analytics for temperature, AQI, and humidity, the system includes cost opti- mization techniques and real-time alerts. This work highlights the benefits of serverless computing for data-intensive applications and offers a practical, low-latency solution suitable for smart city and environmental monitoring use cases.

Keywords*:* Serverless Computing; Google Cloud Platform; Cloud Functions; BigQuery; Machine Learning; Weather Fore- casting; Real-time Dashboard; Cost Optimization

1. Introduction

The exponential growth of data, coupled with the rising need for instant insights, has pushed traditional data processing systems to their limits. In domains like weather forecasting and environmental monitoring, delays in data processing or system failures can lead to significant consequences. To address these challenges, serverless computing has emerged as a powerful paradigm that abstracts away infrastructure management, al- lowing developers to deploy scalable, event-driven solutions effortlessly.

This research introduces a serverless weather forecasting framework built entirely on cloud-native technologies. The proposed system harnesses the capabilities of Google Cloud Functions for real-time data ingestion, BigQuery for efficient storage and analytical querying, and machine learning models to predict key environmental parameters such as temperature, air quality index (AQI), and humidity. A dynamic, user- friendly Streamlit dashboard further brings the system to life by enabling real-time visualization and decision-making.

Unlike conventional architectures, our serverless design is highly scalable, cost-effective, and requires zero manual server provisioning. It supports seamless data flow from ingestion to prediction and visualization with minimal latency. The system is further enhanced by a cost optimization module and alerting mechanism, making it particularly well-suited for smart city infrastructure, disaster readiness, and urban environmental management.

The structure of this paper is as follows: Section 4 presents relevant literature and prior works; Section 5 details the proposed methodology and system architecture; Section 6 explains the implementation of the system; Section 7 analyzes the results; and Sections 8 and 9 conclude the paper and discuss future directions.

* Corresponding author: Dhananjay Adikane

2. Literature review

The convergence of cloud computing and machine learn- ing has opened new avenues in the design of intelligent, scalable systems, particularly in domains requiring real-time responsiveness such as weather forecasting. Several studies have explored the role of serverless architectures, predictive analytics, and cloud-based dashboards in environmental data processing.

Google Cloud Functions, as discussed in [1], offer an event- driven serverless model that can scale automatically, making them suitable for real-time data ingestion. Prior work in [2] emphasizes the power of lightweight Python frameworks like Streamlit in developing interactive dashboards that allow rapid prototyping and deployment.

Researchers in [3] developed a machine learning-based weather prediction model using historical temperature and humidity data, demonstrating that simple regression techniques can achieve high accuracy. Similarly, [4] integrated AQI pre- diction models with alert systems, though their architecture relied on traditional server-based infrastructure.

A study by Sharma et al. [5] evaluated the cost-benefit analysis of serverless versus containerized architectures for scalable applications. Their findings showed that serverless models reduce idle costs significantly while maintaining com- parable performance.

Although substantial work exists in cloud-based weather forecasting and real-time dashboards, few solutions integrate end-to-end serverless data pipelines with machine learning, visualization, and cost optimization. This paper aims to fill that gap by combining all these components into a single cohesive system optimized for performance, scalability, and usability.

3. Methodology

The proposed system follows a modular, event-driven ar- chitecture designed using serverless cloud components. It integrates real-time data ingestion, scalable storage, predictive analytics, and a lightweight visualization layer. The method- ology is divided into six core stages:

Weather Data Simulation: Simulated real-time weather data, including temperature, AQI, humidity, and pres- sure, is generated using a Python script to mimic sensor output across multiple cities.

Serverless Ingestion with Cloud Functions: Google Cloud Functions are used to ingest this simulated data in real-time. The function is triggered periodically and forwards the data to BigQuery using service account authentication.

Storage and Querying in BigQuery: BigQuery acts as the central data warehouse. It stores structured weather data and supports high-speed SQL queries for analysis and model training.

Machine Learning for Prediction: Using regression models such as Linear Regression and Random Forest, temperature, AQI, and humidity forecasts are generated. The models are trained on historical data fetched from BigQuery.

Visualization with Streamlit: A responsive Streamlit dashboard fetches real-time data from BigQuery and displays KPIs, graphs, prediction results, and weather alerts. The dashboard supports city-wise filtering, trend analysis, and forecast visualizations.

Cost Optimization and Alerts: A secondary module forecasts cloud costs using synthetic billing data and ML, and the system sends email/SMS alerts if temper- ature, AQI, or humidity cross predefined thresholds.

3.1. System Architecture

Figure 1 illustrates the end-to-end flow of the system. The architecture emphasizes modularity, scalability, and low operational overhead.

Each component is stateless and interacts through APIs or scheduled triggers, ensuring minimal resource wastage and maximum scalability. The use of BigQuery allows near real- time analytics, and the Streamlit interface ensures that both technical and non-technical users can interact with the system intuitively.

4. Implementation

The system was implemented using a combination of Google Cloud services, Python-based machine learning li- braries, and Streamlit for interactive visualization. The entire pipeline was developed with modularity in mind, enabling independent deployment, scalability, and rapid updates.

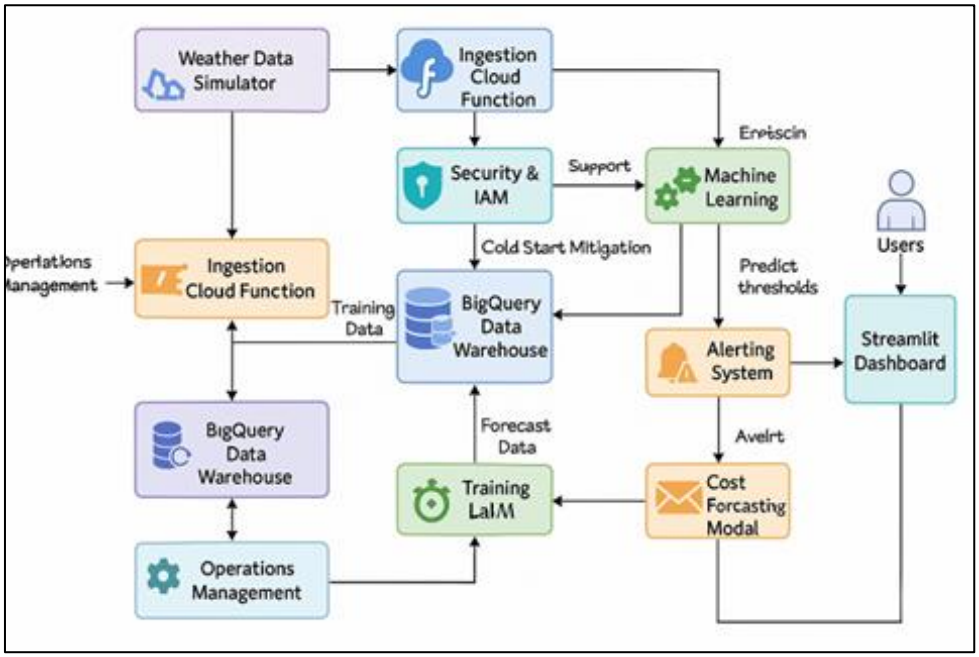

Figure 1 Proposed Serverless Data Processing and Forecasting Architecture

4.1. Simulated Data Generation

A Python script was developed to simulate weather data such as temperature (°C), AQI, humidity (%), pressure (hPa), and timestamps across selected Indian cities including Delhi, Mumbai, Pune, Bangalore, and others. The script generates JSON records and pushes them at regular intervals to a Google Cloud Function via an HTTP trigger.

4.2. Real-time Ingestion using Google Cloud Functions

Google Cloud Functions serve as the backbone of real-time ingestion. The deployed function receives the simulated data, authenticates using a service account, and inserts the records directly into a BigQuery table. This ensures continuous, near real-time data flow without the need for any server infrastruc- ture.

4.3. BigQuery for Data Storage and Access

The ingested data is stored in a BigQuery table with a defined schema. BigQuery supports analytical queries across large datasets with minimal latency, and also serves as the data source for both model training and the Streamlit dash- board. The dataset schema includes: city, temperature, humidity, AQI, pressure, and timestamp.

4.4. Machine Learning Model Development

Multiple machine learning models were developed for fore- casting temperature, AQI, and humidity. Linear Regression and Random Forest were chosen based on performance and interpretability. The models were trained on historical data with lag features and rolling windows for improved accuracy. Evaluation metrics such as R^2 score and MAE were used to assess model performance. Hyperparameter tuning was applied using RandomizedSearchCV for optimal results.

4.5. Interactive Dashboard using Streamlit

A responsive dashboard was built using Streamlit, providing city-wise filtering, historical trend plots, real vs predicted graphs, and forecast metrics. The dashboard also supports dark/light theme switching, rolling forecast controls, and CSV export functionality. Plotly and seaborn libraries were used for visually rich charts.

4.6. Cost Prediction and Alerting Module

A separate module simulates cloud cost data (for BigQuery, Cloud Functions, and Cloud Storage) and predicts future usage costs using regression techniques. Additionally, a real-time alert system is embedded in the dashboard using email and SMS notifications powered by Sendinblue and Twilio APIs. Thresholds for AQI, temperature, and humidity are defined, and alerts are triggered accordingly.

5. Results and discussion

The developed system was tested using simulated real-time weather data for ten major Indian cities over a span of 20 days. The results demonstrate the effectiveness of the serverless pipeline in terms of both data flow reliability and machine learning prediction accuracy.

5.1. Prediction Accuracy

Three machine learning models were evaluated for each parameter: temperature, AQI, and humidity. Among the tested models, Random Forest consistently delivered the lowest Mean Absolute Error (MAE), while Linear Regression showed strong baseline performance.

Table 1 Model Performance Summary (Best Models)

Parameter	Model	R^2 Score	MAE
Temperature	Random Forest	0.89	1.67
AQI	Random Forest	0.86	12.3
Humidity	Linear Regression	0.84	4.2

These results confirm the system's ability to provide reason- ably accurate short-term forecasts in a lightweight, serverless setting.

5.2. Dashboard Visualizations

Figure 2 shows the live dashboard interface, featuring a dropdown to select cities, plots comparing real vs predicted values, rolling forecast sliders, and visual alert banners for threshold breaches.

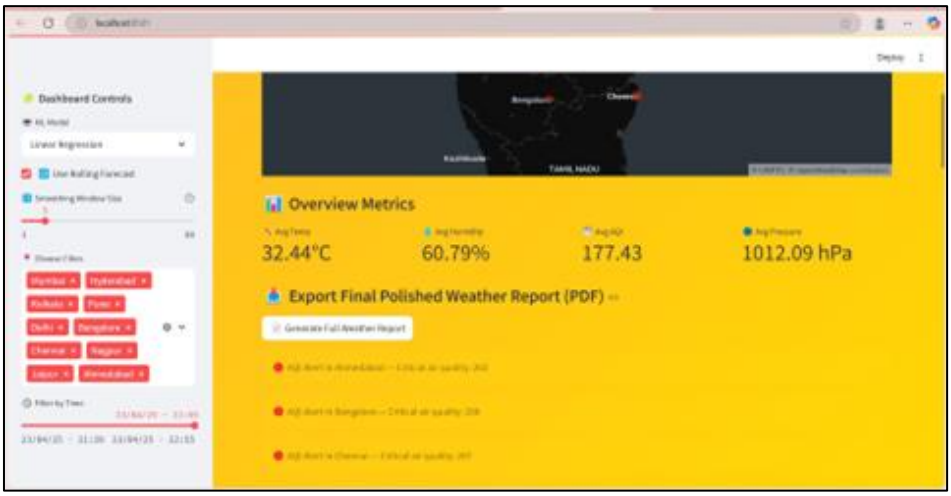

Figure 2 Real-Time Weather Dashboard Interface

5.3. Real-time Alerts and Notifications

The visual alert system accurately flagged extreme AQI and temperature values based on preset thresholds. Mobile SMS alerts were successfully triggered through Twilio, and email notifications were delivered using Sendinblue.

5.4. Cost Prediction Insights

The cost prediction dashboard demonstrated how future resource usage could be forecasted. Figure 3 illustrates the expected 7-day cost trend for Cloud Functions and BigQuery operations.

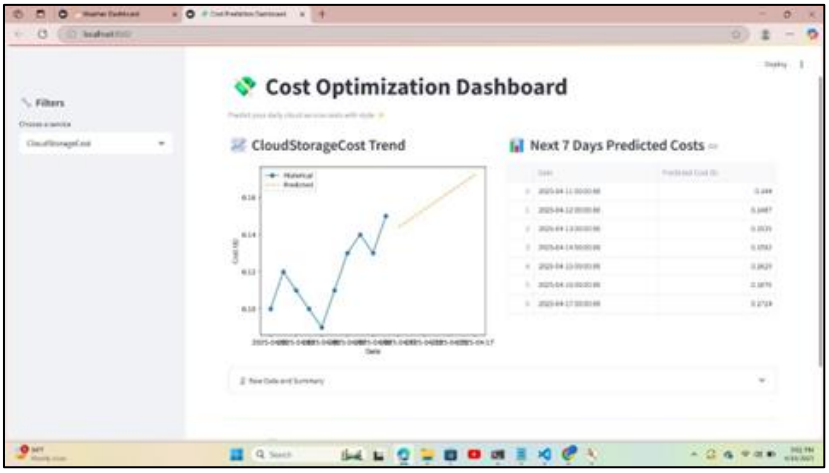

Figure 3 Predicted Cost Forecast for GCP Services

5.5. System Responsiveness and Efficiency

The system maintained consistent performance across test cycles, with average response times under 500ms for data in- gestion and under 2 seconds for dashboard rendering. The use of serverless tools minimized idle resource usage and allowed the architecture to auto-scale without manual intervention.

6. Conclusion

This paper presented a complete, serverless architecture for real-time weather data processing and forecasting us- ing Google Cloud Platform. The proposed system leverages Google Cloud Functions for event-driven data ingestion, Big- Query for scalable data storage and querying, and Streamlit for dynamic, user-friendly visualization.

Machine learning models were developed to forecast tem- perature, AQI, and humidity using historical weather patterns, with Random Forest demonstrating superior accuracy. A cost prediction module was also implemented to estimate future resource usage on cloud services. Visual alerts and SMS/email notifications were integrated to ensure timely updates about extreme weather conditions.

The key contributions of this work include:

- End-to-end serverless pipeline from ingestion to visual- ization.
- Real-time prediction and alerting system using ML.
- Forecasting of both weather parameters and cloud re- source costs.
- Fully interactive dashboard accessible to non-technical users.

The system was evaluated using real-time simulated data across multiple Indian cities and showed high accuracy in predictions and excellent responsiveness in dashboard perfor- mance. Overall, the project demonstrates the effectiveness of serverless computing in building intelligent, scalable, and cost- efficient data-driven applications.

Future scope

While the current system successfully demonstrates real- time weather forecasting using a serverless cloud architecture, several enhancements can be explored to extend its function- ality, accuracy, and scalability.

- Integration of Public APIs: Incorporating real-world weather APIs (such as OpenWeatherMap or IMD data) can replace the simulated dataset with live, authentic environmental data, improving the system's reliability and real-world applicability.
- Advanced Machine Learning Models: Current models use traditional regression techniques. Incorporating deep

learning models such as LSTM or GRU can improve accuracy for time series forecasting, especially in the presence of seasonal variations.

- Edge Computing and IoT: Deploying the data collection layer on edge devices like Raspberry Pi or IoT sensors can make the system suitable for remote or resource- constrained environments.
- Multi-cloud Deployment: Expanding the architecture to support AWS Lambda or Azure Functions can enable a hybrid or multi-cloud strategy, ensuring redundancy, resilience, and broader platform compatibility.
- Support for Additional Domains: The same architecture can be extended to other applications such as traffic monitoring, crop health prediction, flood alerts, and en- ergy usage forecasting, making it a versatile smart city component.
- User Customization and Historical Analytics: Future versions of the dashboard can allow users to define custom alert thresholds, export detailed reports, and view long-term trend analytics across seasons or years.

By continuing to enhance this serverless platform with emerging technologies, the system can evolve into a comprehensive environmental intelligence tool that contributes to both sustainable development and disaster preparedness. .

Compliance with ethical standards

Acknowledgments

The author expresses sincere gratitude to Dr. Odelu Ojjela, Associate Professor and Head of the Department of Ap- plied Mathematics, Defence Institute of Advanced Technology (DIAT), Pune, for his valuable guidance, mentorship, and continuous support throughout the development of this project.

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Google Cloud Platform, "Google Cloud Functions Documentation," 2020. [Online]. Available: https://cloud.google.com/functions

[2] Streamlit Inc., "Streamlit: The fastest way to build and share data apps," 2021. [Online]. Available: https://streamlit.io

[3] A. Patel and S. Mehta, "Weather forecasting using machine learning algorithms," *International Journal of Computer Applications*, vol. 182, no. 44, pp. 25–30, 2019.

[4] R. Banerjee, K. Roy, and A. Chatterjee, "Air quality prediction using machine learning with real-time alerts," *Environmental Monitoring and Assessment*, vol. 192, no. 5, pp. 1–12, 2020.

[5] P. Sharma and M. Gupta, "Evaluating serverless computing for scalable cloud applications: Performance vs cost," *Journal of Cloud Computing*, vol. 10, no. 1, pp. 1–13, 2021.