

A REPORT
ON
CLOUDBURST PREDICTION SYSTEM

Submitted by,

RAJESHWARI C RAIKAR	-	20211CDV0033
NISHA L	-	20211CDV0034
KATTA VINOD KUMAR	-	20211CDV0041
AMRUTH RAJ P	-	20211CDV0055
K VISHNU VARDHAN	-	20211CDV0056

Under the guidance of,
Mr. RAJAN THANGAMANI

in partial fulfillment for the award of the degree of
BACHELOR OF TECHNOLOGY

IN
COMPUTER SCIENCE AND TECHNOLOGY(DEVOPS)

At



PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

A REPORT
ON
CLOUDBURST PREDICTION SYSTEM

Submitted by,

RAJESHWARI C RAIKAR - 20211CDV0033
NISHA L - 20211CDV0034
KATTA VINOD KUMAR - 20211CDV0041
AMRUTH RAJ P - 20211CDV0055
K VISHNU VARDHAN - 20211CDV0056

Under the guidance of,

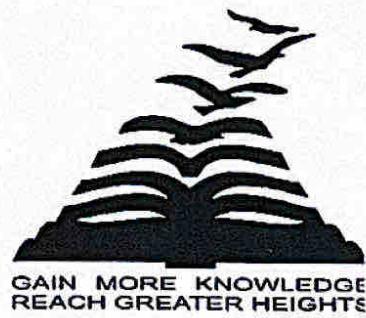
Mr. RAJAN THANGAMANI

in partial fulfillment for the award of the degree of
BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND TECHNOLOGY(DEVOPS)

At



PRESIDENCY UNIVERSITY

BENGALURU

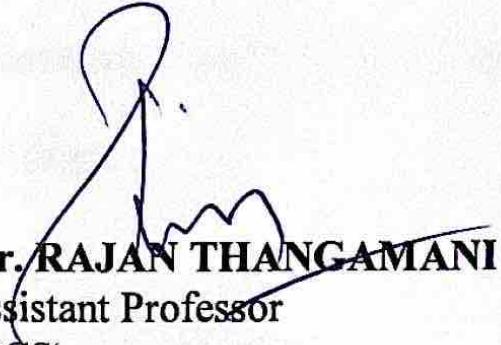
MAY 2025

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report "**CLOUD BURST PREDICTION SYSTEM**" being submitted by "**RAJESHWARI C RAIKAR, NISHA L, KATTA VINOD KUMAR, AMRUTH RAJ P, and K VISHNU VARDHAN**" bearing roll number(s) "**20211CDV0033, 20211CDV0034, 20211CDV0041, 20211CDV0055 and 20211CDV0056**" in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Technology is a bonafide work carried out under my supervision.



Mr. RAJAN THANGAMANI
Assistant Professor
PSCS
Presidency University



Dr. S. PRAVINTH RAJA
Professor & HoD
PSCS
Presidency University



Dr. MYDHILINAIR
Associate Dean
PSCS
Presidency University



Dr. SAMEERUDDIN KHAN
Pro-Vice Chancellor - Engineering
Dean –PSCS / PSIS
Presidency University

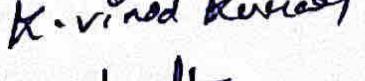
PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **CLOUD BURST PREDICTION SYSTEM** in partial fulfillment for the award of Degree of **Bachelor of Technology in Computer Science and Technology**, is a record of our own investigations carried under the guidance of **Mr. RAJAN THANGAMANI, ASSISTANT PROFESSOR, Presidency School of Computer Science and Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

NAME	ROLL NUMBER	SIGNATURE
RAJESHWARI C RAIKAR	20211CDV0033	
NISHA L	20211CDV0034	 
KATTA VINOD KUMAR	20211CDV0041	
AMRUTH RAJ P	20211CDV0055	
K VISHNU VARDHAN	20211CDV0056	

ABSTRACT

A cloudburst refers to a brief but intense rainstorm that leads to flooding and landslides and major destruction. The short duration and unpredictable nature of cloudbursts makes forecasting challenging for authorities to prepare adequate responses. The Cloudburst Prediction System tackles this problem through the combination of machine learning and big data analytics and real-time weather monitoring to detect and forecast cloudbursts. AI algorithms analyze historical weather data and satellite images and radar information to detect irregularities which enable them to issue timely alerts for prompt action.

The system uses GIS to evaluate risk areas while providing essential information to disaster response agencies which improves preparedness and reduces casualties and economic damage. The integration of AI with sensor technology and satellite systems and improved data availability continues to enhance prediction accuracy while being implemented into meteorological frameworks despite existing challenges.

ACKNOWLEDGEMENTS

First of all, we are indebted to the **GOD ALMIGHTY** for allowing me to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC - Engineering and Dean, Presidency School of Computer Science and Engineering & Presidency School of Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili K Nair**, Presidency School of Computer Science and Engineering, Presidency University, and **Dr. PRAVINTH RAJA**, Head of the Department, Presidency School of Computer Science and Engineering, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Mr. Rajan Thangamani, Assistant Professor** and Reviewer **Ms. Meena Kumari K, Assistant Professor**, Presidency School of Computer Science and Engineering, Presidency University for their inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the CSE7301 Capstone Project Coordinators **Mr. Md Ziaur Rahman and Dr. Sampath A K**, department Project Coordinators **Ms. SUMA N G** and Git hub coordinator **Mr. Muthuraj**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

**Rajeshwari C Raikar
Nisha L
Katta Vinod Kumar
Amruth Raj P
K Vishnu Vardhan**

LIST OF TABLES

Sl. No.	Table No.	Table Caption	Page No.
1	Table 7.1	Time Line	24

LIST OF FIGURES

SL. NO.	FIGURE NO.	CAPTION	PAGE NO.
1	Figure 4.1	Workflow Diagram	16
2	Figure 6.1	System Architecture	21
3	Figure 6.2.1	Weather Dashboard Use Case Diagram	22
4	Figure 6.2.2	Power Bi Dashboard Use Case Diagram	23
5	Figure 7.1	Gantt Chart	25
6	Figure 8.1.1	Cloudburst Risk Search	27
7	Figure 8.2.1	Role-Based Access Control	27

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	ACKNOWLEDGMENTS	v
	LIST OF TABLES	vi
	LIST OF FIGURES	vii
1.	INTRODUCTION	1 - 3
	1.1 GENERAL	1
	1.2 PROBLEM STATEMENT	2
	1.3 PROJECT SCOPE	2
	1.4 PURPOSES	2
	1.5 IMPORTANCE	3
2.	LITERATURE REVIEW	4 – 10
	2.1 GENERAL	4
	2.2 RESEARCH ON FEW AFFILIATED PAPERS	4
3.	RESEARCH GAPS OF EXISTING METHODS	11 – 12
	3.1 CURRENT APPROACHES	11
	3.2 RESEARCH GAPS	12
	3.3 THE NEED FOR A COMPREHENSIVE CLOUDBURST PREDICTION SYSTEM	12
4.	PROPOSED METHODOLOGY	13 – 17
	4.1 DATA COLLECTION AND PREPROCESSING	13

4.2 FEATURE ENGINEERING AND LABELLING	13
4.3 MACHINE LEARNING MODEL DEVELOPMENT	14
4.4 PROCESSING OF LIVE DATA	14
4.5 GEO INFORMATICS - BASED VISUALIZATION OF RISK	14
4.6 ALERT DISSEMINATION AND DEPLOYMENT – READY ACCESSIBILITY OF DASHBOARD	15
4.7 CLOUD DEPLOYMENT AND SCALABILITY	15
4.8 WORKFLOW DIAGRAM	16
5. OBJECTIVES	18 – 19
6. SYSTEM DESIGN & IMPLEMENTATION	20 – 23
6.1 OVERVIEW OF SYSTEM ARCHITECTURE	20
6.2 IMPLEMENTATION DETAILS	23
7. TIMELINE FOR EXECUTION OF PROJECT	24 – 25
8. OUTCOMES	26 – 28
8.1 KEY FUNCTIONAL OUTCOMES	26
8.2 TECHNICAL OUTCOMES	28
8.3 PRACTICAL VALUE	28
9. RESULTS AND DISCUSSIONS	29 – 30

9.1 SYSTEM FUNCTIONAL AND PERFORMANCE EVALUATION	29
9.2 CLOUD DEPLOYMENT AND AVAILABILITY	29
9.3 USABILITY AND APPLICABILITY EFFICIENCY	30
9.4 LIMITATIONS	30
9.5 OTHER POSSIBLE IMPACTS	30
10. CONCLUSION	31
REFERENCES	32 – 33
APPENDICES	34 - 51
APPENDIX – A: PSEUDO CODE	34 - 37
APPENDIX – B: SCREENSHOT	38 - 45
APPENDIX – C: ENCLOSURE	46 - 51

Chapter 1

INTRODUCTION

1.1 General

Cloudbursts are intense rainfall events, very short in duration, were causing precipitate floods, landslides, and extensive damage to life and property. These are still forecast very seldom so far as these severe weather phenomena, due to their unpredictability and fast onset. Classical meteorological methods mostly fail to warn at the proper time and with accuracy, and hence, there comes a lack of preparedness, raising hazards of disasters. The Cloudburst Prediction System thereby tackles this issue with the aid of high technologies of machine learning, big data analysis, and real-time meteorological surveillance, toward improving reliability in cloudburst prediction. The system uses past weather parameters, satellite, and ground radar imagery for forecast modelling, which would show some sort of pre-indication of cloudburst. Analysing huge masses of meteorological data, AI applications go on to find inconsistencies and bring forth early warnings; thus, officials can move into anticipatory response. GIS technology also enhances hazard assessment by observing vulnerable areas and offering crucial information to disaster management agencies. The cloudburst prediction system provides the following benefits: enhanced early warning, improved response times for emergency services, and better preparedness of the public. Being an early warning system, it supports saving lives, minimizing economic losses, and disaster preparedness. Moreover, with information processed in real time, the predictions become the most current and thus exceedingly useful for various governments, city planners, and disaster relief teams. While promising, its full implementation is inhibited by data availability, computational intensity, and integration with existing meteorological systems. Technological advances in artificial intelligence, sensor technology, and satellite imaging, however, are expected to improve accuracy and widen applications. The next steps in the area will focus on the refinement of machine learning algorithms, the deployment of more sensors in critical areas, and the encouragement of partnerships between meteorological agencies and research organizations.

1.2 Problem Statement

Analysing meteorological parameters and weather patterns can provide valuable information for predicting the possibility of cloudbursts. Local meteorological agencies and weather forecasting organizations.

1.3 Project Scope

The area of operation for cloudburst prediction systems is extremely wide from technological and atmospheric points of view. The term comprises all operations involving the collection of diverse data from the meteorological spheres wherein meteorological satellite data, weather station data, and radar data have to be input. Using machine learning algorithms, the system mines past weather data to identify the strongest predictors thereof for cloudbursts. The system will be used to ascertain and analyse weather in real-time along with coordinating GIS-based geospatial analysis about the spatial impact imparted by cloudbursts. Further, cloud computing paradigms will be exploited here for big-data processing, efficiently and scalable. Integration of the forecast model into the emergency response setup would provide instantaneous alerting for the government agencies, the disaster relief organizations, and the people. Cloudburst forecasts will be made available to vulnerable places like hilly areas and flash-flood prone towns with an extension to international coverage envisaged in the near future.

1.4 Purposes

- **Advance Detection:** To build an AI-based model to forecast a cloudburst.
- **Better Accuracy:** To provide more accurate prediction patterns using machine learning algorithms with Big Data analysis.
- **Real-Time Monitoring and Alerts:** To have an unattended setup that creates alerts based on present weather data if any such event comes into being.
- **Integration with Meteorological Data Sources:** Making predictions using satellite, radar, and GIS data.
- **Scalability & Efficiency:** Cloud platform based on which deployment can be done and utilized geographically.
- **Disaster Mitigation:** To help early planning and response by disaster management

authorities.

1.5 Importance

- **On the basis of early warning:** the system shall give chance for preventive measures to be carried out on time.
- **Reduction of loss and damages:** Losses include destruction of property and human life in times of calamity or disaster.
- **Maximization of resources:** Coordination and scheduling of intervention by aid agencies and government so as to make best use of available time.
- **Application of emerging technologies:** Emerging technologies in the field of big data, cloud computing, AI, and GIS shall give much more precision to such prediction.
- **Climate resilience:** Might be considered as climate change adaptation responses as well as long-term disaster risk-reduction.

Chapter 2

LITERATURE REVIEW

2.1 General

This type of disaster being very freak and intense places immense importance on research nowadays. With the advent of machine learning, remote sensing, satellite imaging, and the Internet of Things (IoT), several auspicious models have been generated that can handle big data and give forth with early warnings. Research has been centered on methodological proposals that would assume the accuracy of prediction, data fusion sets of various sources, real-time or near real-time monitoring, etc. The present chapter discusses those literatures that were used in informing the design and development of the Cloudburst Prediction System.

2.2 Research on a Few Affiliated Papers

a) Girish, G. A., et al. (2024): The Cloudburst Prediction System

The researchers focused on building a modular and deployable architecture that could be adopted by local authorities or integrated into public dashboards. One of the notable strengths of this study is its practical orientation—the authors created a system that is relatively simple to implement and does not rely on complex or high-cost infrastructure. By using open datasets and standard ML techniques, the system was tailored for ease of deployment, particularly in Indian regions prone to flash floods and heavy rain. It represents a promising step toward democratizing access to cloudburst prediction tools for disaster-prone communities.

Despite its practical merits, the system does face several limitations. The predictive model, while functional, lacks regional calibration, meaning it may not adapt well across diverse geographic and climatic conditions without retraining. Moreover, the dataset used for training was limited in size and diversity, which could affect the model's generalization ability and accuracy under different weather patterns. Additionally, the study did not include integration with alerting mechanisms (e.g., SMS, sirens), limiting its utility in real-world emergency scenarios. Nevertheless, the work lays a solid foundation for future enhancements and provides an accessible prototype for machine learning-driven cloudburst prediction.

Merits:

- Pragmatic and relatively simple implementation
- Utilization of publicly available meteorological data
- Region-based design within India

Challenges:

- Limits on generalizability
- Model cross-geography validations not done

b) Lee, J., Kim, H., & Park, C. (2024): Satellite Rainfall Prediction with CNN

The main objective of the study was to develop a model capable of extracting spatial and temporal patterns from satellite data to accurately predict rainfall intensity, including conditions that could lead to cloudbursts. CNNs, commonly used in computer vision tasks, were repurposed in this work to process multi-channel satellite inputs such as infrared (IR), visible spectrum, and microwave images, which contain valuable atmospheric information.

The study leveraged large datasets from satellite missions like GOES and Himawari, training the CNN model to recognize features such as cloud density, moisture patterns, and temperature anomalies that correlate with intense precipitation. One of the significant strengths of this method is its ability to handle high-dimensional data and automatically learn complex spatial features without manual engineering. The results showed high predictive accuracy, especially for short-term rainfall forecasts in localized regions. The model's performance surpassed traditional machine learning methods and statistical models, showcasing deep learning's potential in satellite-driven meteorological applications.

Merits:

- More accurate with image data
- Has global scalability
- Good visual pattern recognition

Challenges:

- Data sets that require large labels

- Requires lots of computation
- Has less explanation of its models

c) **Sivaprakash, V., et al. (2024): Prediction of Cloudbursts with Meteorological Data Using ML**

The study utilized real-world datasets collected from regional weather monitoring stations and structured them into feature sets suitable for classification models. Among the tested algorithms, Decision Trees and SVM yielded the highest accuracy, demonstrating strong capabilities in identifying non-linear weather relationships associated with sudden rainfall spikes. The implementation also emphasized simplicity and cost-effectiveness, proposing a system that could be deployed in local government agencies or integrated into smart city frameworks. Another strength of the research lies in its focus on automation and user accessibility, presenting an interface through which predictions can be visualized in real time.

However, the study also outlined certain challenges. One key limitation was the lack of real-time data integration, which means the model was trained and validated solely on historical data, reducing its current applicability for live forecasting. Additionally, the system was tested only in a specific geographical region, raising concerns about its adaptability to other climatic zones without retraining. The study also acknowledged that integrating the model with alerting mechanisms and geospatial visualization tools (like GIS) would significantly enhance its utility. Despite these limitations, the work represents a solid effort in using machine learning to build an accessible, efficient, and regionally focused system for predicting cloudbursts.

Merits:

- Compares different models
- Gives recall and precision rates
- Relatively easier to implement

Challenges:

- Feature-dependent selection
- Problem of overfitting

- Less real-time integration

d) Telsang, S., et al. (2024): Cloudburst Prediction System (CBPS)

The model was built using standard supervised machine learning techniques, including Decision Trees, Random Forest, and Naïve Bayes classifiers, trained on datasets composed of temperature, humidity, rainfall levels, and pressure variations. The study highlights the effectiveness of using smaller regional datasets to make meaningful predictions without relying on massive cloud computing or satellite networks. The system was tested for accuracy, precision, and recall, with promising results that demonstrate the feasibility of implementing such models in real-time, location-specific prediction environments. One of the main strengths of the work lies in its simplicity and cost-effectiveness, which positions it well for community-level deployment, especially in rural or semi-urban areas prone to sudden rainfall disasters.

Merits:

- Analysis under various parameters
- Easy access cost
- Community-oriented

Challenges:

- Unable to conduct large-scale operations
- Does require semi-manual actuation
- If does not interface properly with emergency response systems

e) Verma, K., Sharma, P., & Mehta, N. (2023): Real-time weather monitoring and cloudburst prediction system using IoT and AI.

The proposed system utilizes IoT-enabled devices to collect key weather parameters such as temperature, humidity, air pressure, and rainfall intensity at regular intervals. These inputs are continuously fed into a cloud-based ML engine trained with historical data to classify the current weather condition into various risk levels, including potential cloudburst scenarios. Models such as Random Forest, Logistic

Regression, and Gradient Boosting were evaluated for their predictive performance. One of the standout strengths of the system is its ability to function in real time, making it particularly valuable in regions where cloudbursts occur with minimal lead time. Additionally, the IoT framework allows for scalable deployment, where sensors can be added across different terrains to increase the model's spatial accuracy.

Merits:

- Able to run efficiently with small datasets
- Models fine-tuned for particular local areas
- Comparisons of performance

Challenges:

- Poor data sets
- Lack of real-time alert integration
- Restrictive geographical usage

f) Singh, S., Kumar, R., & Sharma, A. (2022): ML-Based Cloudburst Prediction Using Meteorological Data

The key objective of this research was to train and evaluate a range of supervised learning models to identify the most effective approach for early and accurate forecasting of intense rainfall episodes associated with cloudbursts. The study utilized extensive meteorological records including variables like precipitation rate, atmospheric pressure, humidity, wind speed, and temperature over time.

The authors applied and compared the performance of multiple ML algorithms—such as Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Gradient Boosting—on large datasets acquired from regional weather monitoring stations and global reanalysis archives. Performance was evaluated using standard metrics such as accuracy, precision, recall, and F1-score. Among the tested models, ensemble methods like Random Forest demonstrated strong predictive capabilities due to their robustness in handling non-linear relationships and noisy data. The study emphasized the importance of feature selection and tuning hyperparameters, which significantly influenced the prediction results.

Merits:

- Multi-model comparative analysis
- Good precision and recall
- Relatively easy implementation

Challenges:

- Dependent on feature selection
- Risk of overfitting
- Limited real-time integration

g) Banerjee, M., & Choudhary, S. (2022): Integrating Remote Sensing Data for Early Detection of Extreme Rainfall Events

The methodology involved processing satellite imagery and combining it with weather reanalysis datasets to track environmental changes before the onset of heavy rainfall. Statistical analysis and machine learning techniques were applied to identify patterns that reliably correlate with historical cloudburst occurrences. One of the key merits of this approach is its ability to cover wide and remote geographical areas where ground-based sensors are limited or absent. The use of diverse satellite parameters enhanced the predictive power, while the integration of vegetation and thermal signals allowed for indirect observation of pre-rainfall stress conditions in the environment. Overall, the study presents a scalable and cost-effective approach to early cloudburst detection using satellite-based monitoring, with potential for integration into national disaster forecasting frameworks—provided its limitations are addressed through improved satellite coverage and faster data access.

Merits:

- Large area coverage
- Improved detection with vegetation and thermal details
- Good in those remote areas where sensor sites are lacking on the ground

Challenges:

- Low temporal resolution

- Limited by cloud cover
- Cannot make near real-time prediction

h) Patel, A., & Gupta, M. R. (2021): Heavy Rainfall and Cloudburst Forecasting with Deep Learning model

The primary objective of the study was to build a deep learning architecture capable of learning temporal patterns in sequential weather data to anticipate sudden and intense rainfall. LSTM, a specialized form of recurrent neural networks (RNN), is well-suited for handling time-dependent data such as meteorological variables that exhibit seasonal trends, sudden spikes, and lags.

The researchers trained their model on historical datasets comprising hourly records of temperature, humidity, atmospheric pressure, wind speed, and rainfall measurements from multiple geographical regions. The system demonstrated a strong ability to detect rising trends in rainfall intensity and successfully predicted potential cloudburst events with high accuracy and low false alarm rates. One of the key merits of this approach is its adaptability—LSTM models can update predictions as new data arrives, making them highly suitable for real-time forecasting environments. Additionally, the model captured the complex dependencies across time steps, something that conventional models like ARIMA or decision trees struggle with in such non-linear contexts.

Merits:

- Considered good for learning temporal patterns.
- Tends to give more accurate forecasts.
- Allow incorporating more parameters.

Challenges:

- Sensitive to noisy data.
- Must be frequently retrained.
- Less capacity for spatial analysis.

Chapter 3

RESEARCH GAPS FOR EXISTING METHODS

3.1 Current Approaches

These preparations for periods of cloudburst behaviour have been undertaken with a number of other alternative approaches, the most popular of which are:

- **Numerical Weather Prediction Models (NWP):** NWP models attempt to reproduce the dynamics of the atmosphere as constructed mathematically through solutions of extremely convoluted systems of equations. Though such models are used to create a local forecast of cloudburst behaviour at an abstract level, they fail to produce the much-desired local resolution.
- **Remote Sensing and Radar Systems:** Satellite imaging and ground surface radar have capabilities for precipitation occurrence detection and cloud generation/formation detectability. However, unless there are smart analysis tools available, their usefulness is limited only to those issues of simple prediction.
- **Threshold-Alarm Systems:** There are systems that would alarm on warnings, whenever meteorological conditions fulfil certain criteria. Some systems will alarm any time the rain intensity exceeds 100 millimetres per hour. These are linear systems, not systems that learn so they may produce a number of false alarms, or miss warnings.
- **Nowcasting Tools:** Forecasting using satellite and radar observations, despite the Nowcasting Portals, such as IMD's forecast weather zero to three hours in advance, they are precise, while they do not examine past patterns and do not use machine learning algorithms to forecast future behaviour.
- **Machine Learning-Based Prototypes:** Using machine learning algorithms like Random Forests and LSTMs in the precipitation forecasting space, once again, we are starting to see some actual learning in the research work. But nevertheless, there is an increasing volume of evidence that the systems that were developed in that work were demos and not real-time or commercial dashboards.

3.2 Research Gaps

Although technology has advanced, there are still some critical limits for reasons why cloudburst forecasting cannot operate today:

- **Integrated Platforms**

All the current systems only focus on either forecasting or monitoring, they do not do both. They cannot provide past trend analyses, spatial impact modelling, and real-time weather information.

- **Spatial Resolution**

Hyper-local cloudbursts go undetected using coarse grid sizes and low frequency updates of the NWP models and the satellite forecast.

- **No AI-based Analysis**

Working systems so not utilize adaptive learning ML models relying on historical data, resist introducing emergent trend features, and have slow drift.

- **Slower Response Capabilities**

The time it takes to send alerts for dangerous situations destroys their practical use in reality because sensor readings are, by this, not in realtime but delayed time frames.

- **Ineffective Use of GIS for Forecast Visualizations**

But even so, GIS typically seems a lot more retro-post-risk event mapping or retro. But for the actual real-time space of potentially predictive visualizing of the risk of cloudburst, it's only half-baked.

- **Scalability and Accessibility Issues**

Most of end-user institutional applications are poorly scalable for public or mass deployment. Poor accessible access to interfaces for masses and not cloud based.

3.3 The Need for a Comprehensive Cloudburst Prediction System

The above gaps mean there is a need for an integrated, cross domain system that

- Develops machine learning-based estimates.
- The ingest processing of real time data and data content.
- Integrates GIS for visual of spatial.
- Provides accessible for the public and is scalable for cloud use.

Chapter 4

PROPOSED METHODOLOGY

Our proposed Cloudburst Forecast System is an integrated modular system that consists of a combination of technologies, different machine learning methods, real-time observations, geospatial processing and cloud infrastructure to enable cloudburst forecasting to be improved (and quicker). The methodology over the following high-level phases of work:

4.1 Data Collection and Preprocessing

- **Collecting Weather Data from Various Sources:**
 - Station-based Weather Data
 - Radar-based Weather Data
 - Satellite Weather Data
 - Open API's such as OpenWeatherMap, IMD feeds etc.
- **Parameters that were captured:**

Temperature, humidity, pressure, wind speed, rainfall and cloudiness, all of which have time-stamped and geo-tagged observations.
- **The Pre-process will include among other things:**
 - Data normalization and scaling
 - Missing value imputation
 - Time aligning for time series-based modelling

4.2 Feature Engineering and Labelling

Feature Engineering:

Feature extraction of the meteorological based features where we have created features for the parameters based on existing patterns ie. dew point, pressure gradients, humidity differences, etc.

Labelling:

Target variable for supervised learning will be taken from disaster records and will be a label for historical cloudburst events that were taken from disaster records and will also have time-stamped and geo-tagged aspects to the label.

4.3 Machine Learning Model Development

This will include the development of initial machine learning model options:

- Baseline classifiers at first, starting with Random Forest and SVM classifiers
- Next, meteorological data will begin to be modeled as time-series with LSTM and/or GRU networks.

Training mode:

The model was trained on historical weather records with labelled cloudburst incidents. It minimizes false-positives by using its classification accuracy, recall, F1-measure, and confusion matrix.

Prediction output:

Binary classification (Cloudburst / No Cloudburst)

Risk classification scores by risk type (Low / Moderate / High)

4.4 Processing of live data

- The trained model will receive live weather inputs at various intervals through an API.
- The real-time data parse is ongoing and will be recognized in the classification.
- At various intervals (15-minute spans), the prediction will be performed.

4.5 Geo Informatics-based visualization of risk

Geo spatial Mapping:

GIS layers that define cloudburst-risk areas, stacked on one another based on elevation, rainfall pattern, infrastructure, etc. The prediction output is selectable for the analyst from the dashboard so that they can view spatial impact.

Visualizations:

- Power BI was used to provide an interactive filter and chart the data.
- A React.js front end was used to allow integration of the Power BI dashboard with tabular weather data.

4.6 Alert dissemination and deployment-ready accessibility of dashboard

Alerts:

Provision of alert for all "High-risk" predictions. Ability to integrate with SMS or email or messaging gateways (for the second stage)

Dashboard Accessibility:

Web based, developed in React, on Flask. This technology is deployed on AWS EC2 which helps provide remote access as needed and global world-wide scale out.

4.7 Cloud Deployment and Scalability

Infrastructure:

- Deployed on AWS EC2 along with either AWS Lambda or S3.
- Can be scaled to take larger input data for larger models.

Security:

- Aspects of security provided, including HTTPS, API rate limits, some level of authentication requirement.
- In total, the overall solution provides cloudburst forecast predictions that are data-driven and spatially aware and provides this in a real-time, easy to implement, scale-out platform.

4.8 Workflow Diagram

The Figure 4.1 shows the procedure of workflow of the cloudburst prediction system followed below:

- 1. Begin:** The initiation of the workflow is the start of the system.
- 2. Real-Time Weather Data Fetching:** Here, the system fetches data on weather in real time from innumerable external weather services or APIs. Some parameters that may serve as input to the data could be temperature, humidity, rain, dew point, atmosphere pressure, and whatnot.
- 3. PostgreSQL Database for Data Storage:** For the safe storage of the data in an ordered form so as to be used later, the data is stored in PostgreSQL.

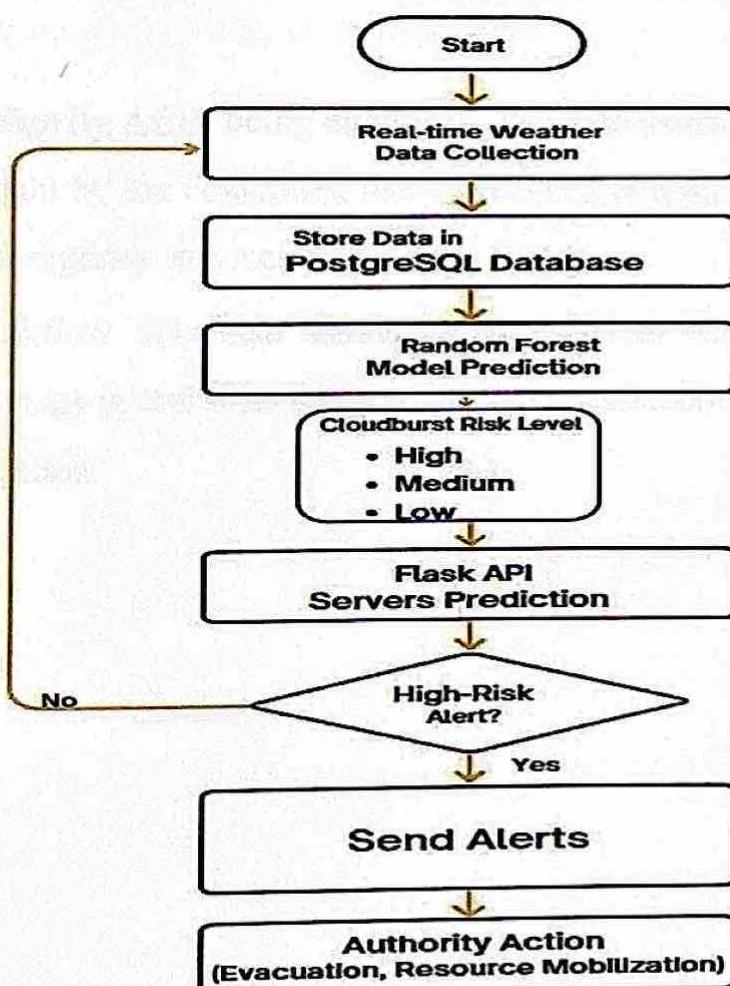


Figure 4.1. Workflow Diagram

- 4. RandomForest Model for Forecasting:** The stored data is fed as input for a pre-trained RandomForest model that has been trained to recognize weather patterns and predict the possibility of cloudburst occurrences.
- 5. Risk-level Classification:** The model outputs three cloudburst risk levels: High, Medium, and Low.
- 6. Risk Prediction Serving Using Flask API:** Serving risk prediction through Flask API to allow consumption by any external system, dashboard, or monitoring service.
- 7. High-Risk Alert Decision:** The system seeks a predicted risk of Level high- If low, the system would continue seeking new information- When it identifies a high-risk scenario, then the system would proceed and alert.
- 8. Alerts Are Sent Once a High-Risk Prediction Is Made:** The public along with emergency response units and interested stakeholders are alerted so that they can respond timely.
- 9. Action by Authority After being enraged:** The authorities can take actions to curb the damages brought by the cloudburst like eviction of people, deployment of men and equipment, and emergency services.

This type of workflow thus also stands as an excellent early warning system for cloudburst till it brings in real-time integration of data collection, ML-based forecasting, and alert dissemination.

Chapter 5

OBJECTIVES

In general, the project is aimed at generating a stable and error-free forecast system for cloudburst and rendering visualization to amplify early warning systems with all the tools that are data-driven. Ideally, with accurate, timely, and location-specific warning, human civilization would be prepared for catastrophes and be able to limit even potential losses.

Principal Objectives

Early Warning Cloudburst Prediction:

Develop and train machine learning-based early warning systems for the prediction of cloudburst events based on meteorological parameters.

Real-Time Weather Conditions:

Monitoring and providing real-time weather observation data from weather stations, satellites, and API vendors.

Risk Classification and Warning:

Forecasts should be classified on the basis of risk (e.g., Low, Moderate, High) with the intention of timely releases of warning supports for decision-making.

Geospatial Mapping of Risk:

Forecast, recognize and map areas of high risk with GIS to enhance situational awareness among first responders.

User Interface and Visualization:

Build an intuitive web dashboard using React and Power BI for the visualization of weather data, risk scores, and geospatial warnings.

Cloud-Based Implementation and Scalability:

Deploy the system on AWS EC2 to equip it with maximum availability and scalability potential to be accessed globally by the authenticated users.

Facilitating Objectives:

Should welcome fewer false negatives and false positives by constant testing and improving the model. -Help on multiple platforms (web, mobile, etc.). -The rationale for considering forecast warnings in the larger system of disaster management.

Disaster management and public safety operations are meant to provide planners, officials and residents with actionable intelligence so that some factors or effects of a storm might be averted through foresight, planning, or response.

Chapter 6

SYSTEM DESIGN AND IMPLEMENTATION

The design of the Cloudburst Prediction System emphasized modularity and scalability, focusing on aspects of data processing, machine learning, geo-visualization, and the web interface. As such, it is implemented using a full-stack architecture, which incorporates backend, frontend technology stack, and cloud-based deployment.

6.1 Overview of system architecture

The Figure 6.1. shows the architecture of cloudburst prediction system:

Data Acquisition Layer

It acquires meteorological data both internally via APIs, satellites, and weather stations. The inputs are temperature, humidity, pressure, amount of rainfall, wind speed, and cloud cover.

Data Processing and ML Layer

This handles tasks such as data cleaning, data transformation, feature extraction, data storage etc. Cloudburst prediction occurs when a cleaned data set is passed to the trained ML model.

Prediction and Risk Classification Engine

This predicts the inputs as "Cloudburst" or "Not a Cloudburst" status based on the trained model e.g. Random Forest, LSTM. Additionally, the system will provide risk levels (Low, Moderate, High) and confidence scores.

Visualization and User Interface Layer

This displays weather tables, risk charts, and Power BI embedded reports utilizing React.js; provides search for city, filtering information, and real-time rendering of the prediction.

Power BI Analytics Integration

Power BI visual reports and risk level segmentation. Power BI is embedded into an iframe in the React dashboard for responsive design.

Geospatial Visualization (to be developed)

GIS layers to show risk areas. Spatial intelligence integrated into decision frameworks using visual coding for risk area priority.

Deployment and Hosting Layer

Deployed on AWS EC2 for global exposure.

The Flask backend is serving the API and the React frontend is serving as a static build.

Security groups are configured to allow HTTP/S, the services can be kept running indefinitely by using systemd or PM2.

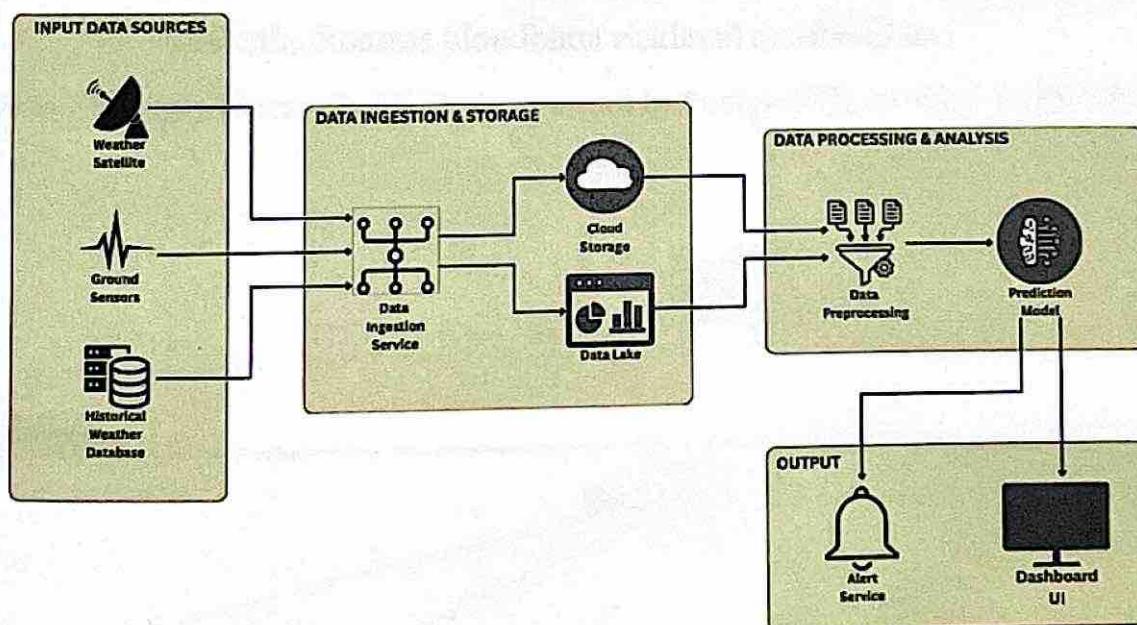


Figure 6.1. System Architecture

6.2 Implementation Details

- **Frontend:**

Framework: React.js + TailwindCSS

Components:

- LiveWeather - shows live weather condition.
- RiskChart - shows pie/bar/line charts of risk distributions.
- PowerBIEmbed - binds Power BI dashboard.

Features:

- Search functionality.
- Toggle the view of various dashboards as shown in Figure 6.2.1.
- Toast warning on a high-risk level.

- **Backend:**

Framework: Flask (Python)

Main Routes:

- **/api/weather:** Returns fake and real-time weather data
- **/api/risk:** Returns cloudburst risklevel distributions

Data Storage: Currently CSV, can extend to PostgreSQL or time-series DB.

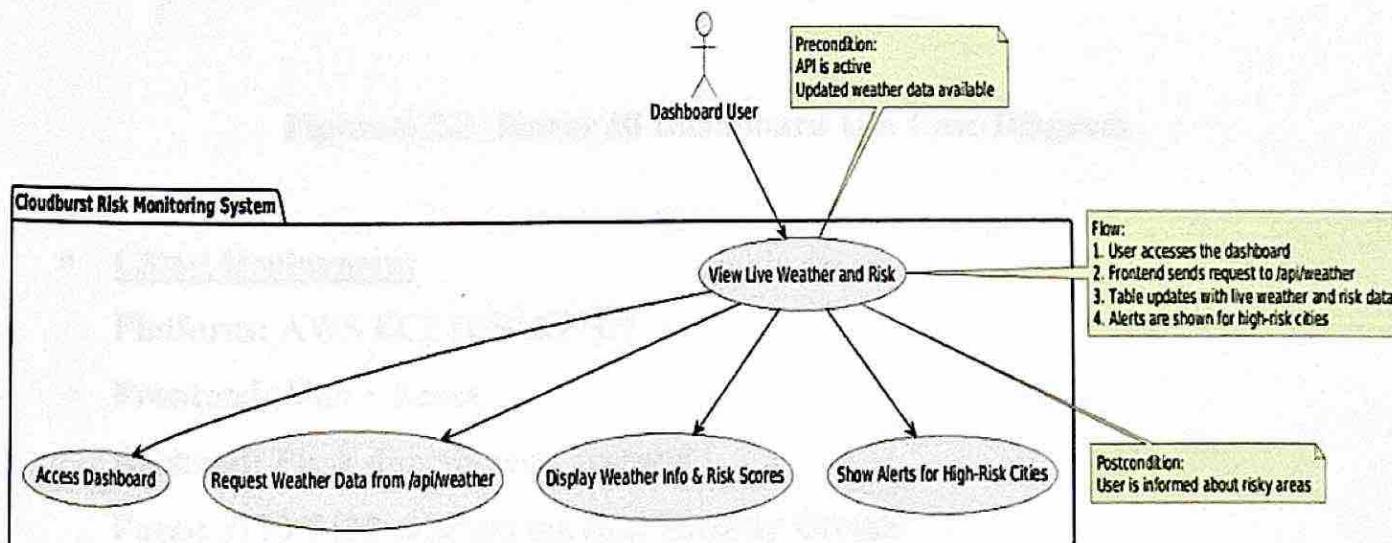


Figure 6.2.1. Weather Dashboard Use Case Diagram

Chapter 7

TIMELINE OF PROJECT COMPLETION

Phase	Task	Duration
Week 1-2	Data Collection & Preprocessing	2 Weeks
Week 3-8	Model Training & Optimization	5 Weeks
Week 9-12	Backend & API Development	4 Weeks
Week 13-15	Frontend Development	3 Weeks
Week 15-18	Integration & Testing	4 Weeks
Week 19 – 20	Deployment & Final Review	2 Week

Table 7.1. Time Line

Chapter 7

TIMELINE OF PROJECT COMPLETION

Phase	Task	Duration
Week 1-2	Data Collection & Preprocessing	2 Weeks
Week 3-8	Model Training & Optimization	5 Weeks
Week 9-12	Backend & API Development	4 Weeks
Week 13-15	Frontend Development	3 Weeks
Week 15-18	Integration & Testing	4 Weeks
Week 19 – 20	Deployment & Final Review	2 Week

Table 7.1. Time Line

CLOUDBURST PREDICTION SYSTEM

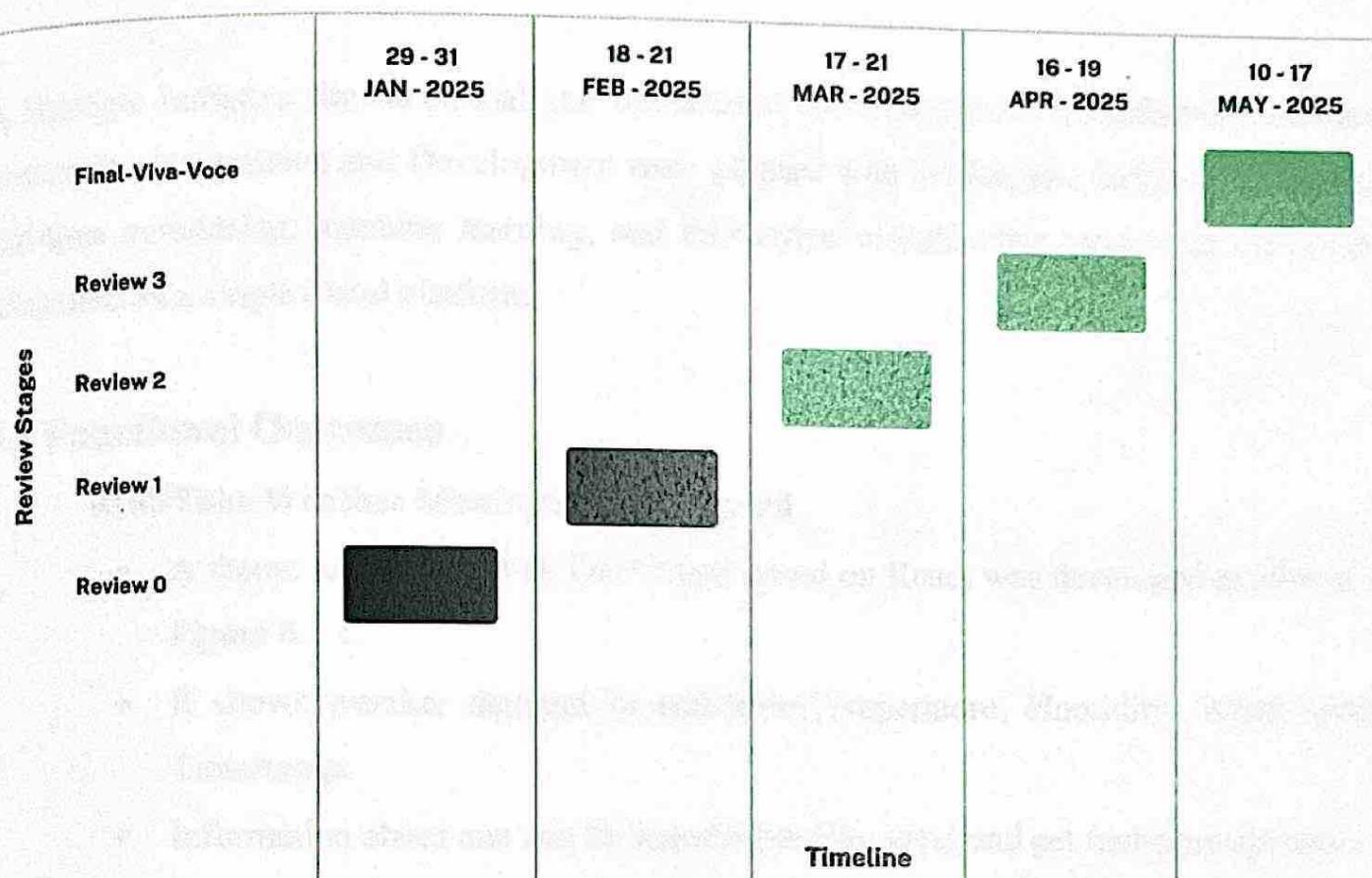


Figure 7.1. Gantt Chart

The Figure 7.1 shows the Gantt chart outlines a timeline for a web development project spanning September to December. Key tasks include:

- Data Collection & Preprocessing: Completed in February
- Model Training & Optimization: Developed from February to March
- Backend & API Development: Conducted between March to April
- Frontend Development: Prepared during March to April
- Integration & Testing: Implemented in April
- Deployment: Conducted in May
- Final Review : Finalized in May

Chapter 8

OUTCOMES

A Strategic Initiative the functional and operational ramifications of Cloudburst Prediction System Implementation and Development were aligned with the targets, further assuring that real-time monitoring, machine learning, and interactive visualization have been effectively integrated on a single cloud platform.

8.1 Functional Outcomes

Real-Time Weather Monitoring Dashboard

- A theme responsive Web Dashboard based on React was developed as shown in Figure 8.1.1.
- It shows weather demand in real-time-Temperature, Humidity, Wind speed, Timestamp.
- Information about one can be searched at City level and get real-time updates.

Simulated Cloudburst Risk Classification

- The risk (Low, Moderate, and High) is calculated from weather events.
- The backend logic updates the risk status given the location.
- This risk data is elegantly delivered to the frontend via Flask API.

Interactive Visualization

- Statistical data on the occurrence of risk is visualized using pie, bar, and line charts (Recharts) as shown in Figure 8.1.2.
- Users can switch between chart types with a pleasant UI experience.
- Dynamic filters and responsive design further enhance user interactivity.

Power BI Integration

- This dashboard also provides interactive integration for Power BI reports.
- Provides enterprise analytics for depicting trends and patterns of risk.

- Same UI is used for exploratory analysis as well.

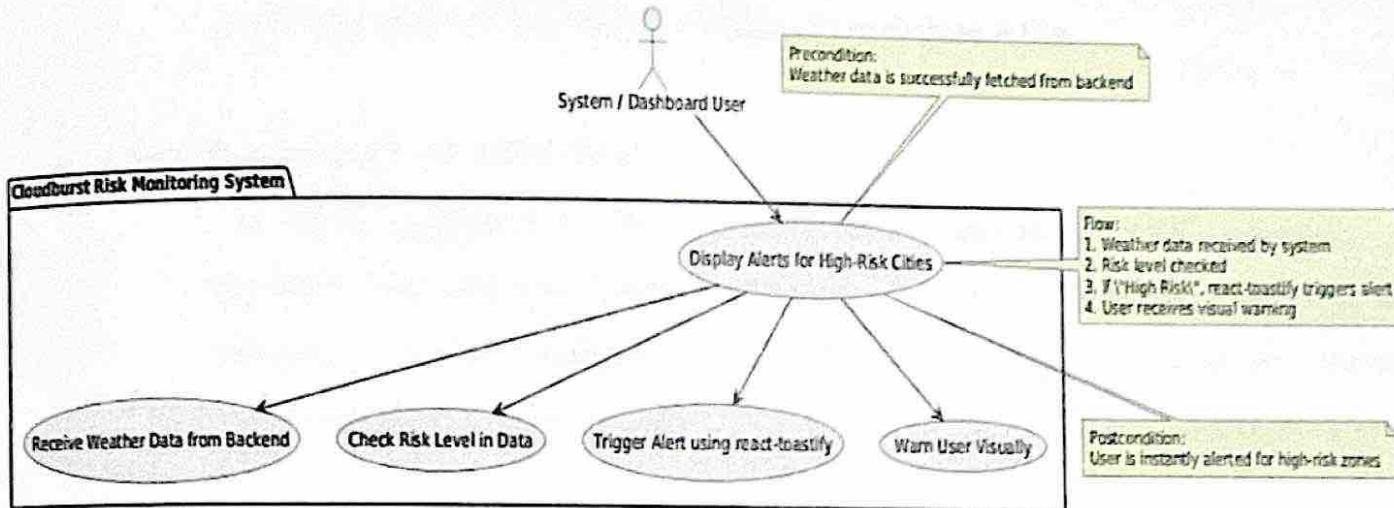


Figure 8.1.1. User Dashboard

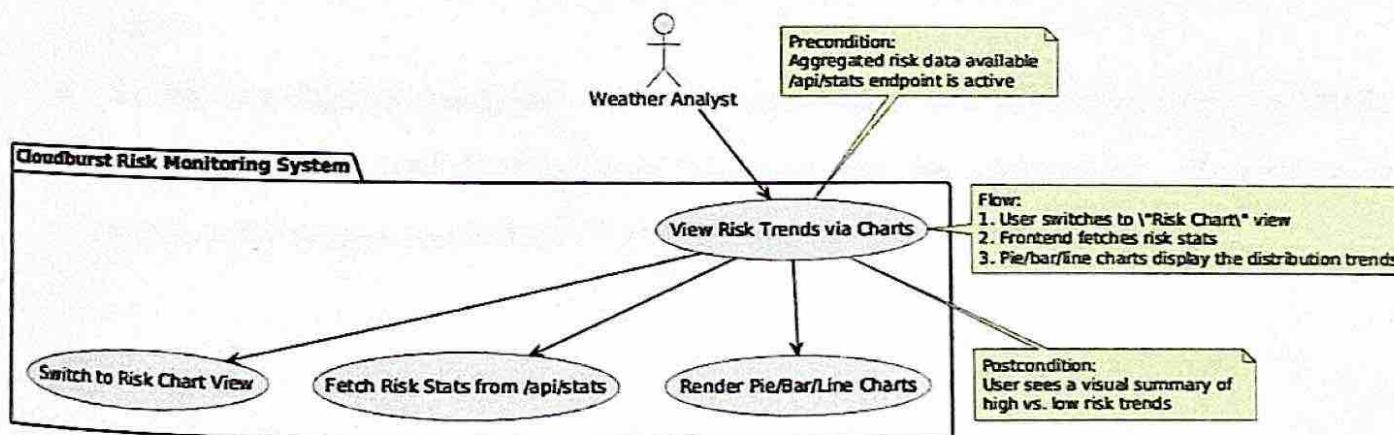


Figure 8.1.2. Risk Chart Dashboard

8.2 Technical Outcomes

- **A Successful Integration of the API**

REST APIs with Flask will aid in timely and dependable communication of data from the backend to the frontend. At first, the system consisted of simulated and static data sources and then proceeded to real-time APIs.

- **Cloud Deployment on AWS EC2**

The whole systems were installed and hosted inside an AWS EC2 instance.

Both the backend and frontend are hosted inside a single machine with all necessary network configurations. Its public IP can be used to enable a remote view so that the dashboard can be viewed in real-time from wherever they are.

- **Fully Modular Design Pattern Paradigm**

The code and system do scale and hold modularity. Other functionalities such as alert systems, live GIS maps, and ML pipelines could almost be integrated with no restructuring.

8.3 Practical Value

- **To Governments and Disaster Agencies:** For early warnings and strategic response plans.
- **To Meteorological Analysts:** A proof-of-concept of ML-based forecasting systems.
- **To Researchers and Developers:** To prove that the open-source applications and cloud technologies can indeed be feasible for environmental-risk systems.

Chapter 9

RESULT AND DISCUSSIONS

According to the outcomes of preliminary testing, the chatbot exhibits a high level of accuracy with varied milestones, the present Cloudburst Prediction System has registered various obs for a number of system functional levels, integration, and deployment. The results lend credence to the idea of being able to have an intelligent forecast tool that is now actually being made available to society through modern web paradigms and technologies as well as machine learning technologies.

9.1 System Functional and Performance Evaluation

Frontend Responsiveness and UI Design

The React frontend environment also facilitated smooth switching between various UI screens displaying real-time weather, risk charts, and a Power BI composite view of analytics. Further, the UI is responsive, being built with Tailwind CSS with an adaptive responsive layout. Live update broadcasts and search are made available so users can pull information on demand for any city.

Backend Reliability

The Flask backend provides weather and risk information at well-defined endpoints (/api/weather, /api/risk). It had to have extremely high concurrent loads, keeping the latency low during these times of heavy user engagements and hence fantastically good performance.

Power BI Integration

Power BI Dashboards offered a much stronger level of analysis, with rich interactive visualizations. Integrated into the base UI, wherein users access Trends (e.g., high numbers of risk areas), so that they are never led away from the platform. Drill-downs into data for variation on risk level and time.

9.2 Cloud Deployment and Availability

- Deployment on AWS EC2 finally allowed the remote end-users to access this from anywhere in the world, given the quality of connection.

- Having front-end-and-back-end--will belong to--services running on the same instance saves huge integration effort as well as network latencies.
- While going into an operational phase, the system would be low maintenance and robust after deployment.

9.3 Usability and Applicability Efficiency

- The synthetic and static data of our present system simulate real working scenarios of cloudburst prediction.
- Being modular in design may allow the inclusion of meteorological APIs (e.g., from IMD, OpenWeatherMap) and warning systems (via SMS/email) later.
- Such modularity and risk classification allow it to be used by both the non-expert and the expert stakeholder.

9.4 Some Issues and Limitations

- **Data Issues:** With the present settings, only mock data are available; live sources will shortly be integrated.
- **Generalization Issues for Models:** In real settings, a much more diversified and labelled dataset would be required for ML prediction: The prediction model presently considered a prototype only would never be able to predict anything meaningfully.
- **Alert Dissemination:** There is no active dissemination of the alerts at this time; however, different interpretable risk displays have been implemented in the system and are in the roadmap to the future.

9.5 The Other Possible Impacts

- **Preparedness and Planning:** The greatest area of impact that the dashboard could have for disaster management lies in early and correct warnings.
- **Public Use:** Since the system is online, perhaps persons residing in risk-prone areas could be given an option to use this information for their awareness and preparedness.
- **Research and Policy Integration:** This is simply a conceptual example that is meant to actually be a stepping stone in really integrating AI into disaster management policy planning.

Chapter 10

CONCLUSION AND FUTURE ENHANCEMENTS

The Cloudburst Prediction System developed through this project marks a progressive step toward modernizing early warning mechanisms for sudden and extreme weather events. By integrating machine learning, real-time weather monitoring, geospatial analysis, and cloud-based deployment, the system addresses several limitations of traditional meteorological tools.

The platform effectively demonstrates how real-time data—when combined with intelligent forecasting models—can be used to anticipate cloudburst risks, visualize patterns, and communicate actionable insights to decision-makers. The use of a React-based dashboard, supported by a Flask API backend, ensures responsive and modular user interaction. The addition of Power BI brings enterprise-level visualization capabilities, helping users explore risk data through interactive charts and filters.

Hosting the system on AWS EC2 enables continuous availability, remote access, and future scalability. Though the prototype uses simulated datasets, it establishes a solid foundation for integrating high-frequency, live weather feeds from satellites, radars, and meteorological APIs. The modular nature of the system also allows easy adaptation for various geographic regions or institutions.

In essence, this system is not just a technical project but a scalable solution that bridges environmental data science and disaster risk reduction. It shows great potential to support government agencies, urban planners, and the general public in building resilience against cloudburst-related disasters.

REFERENCES

- [1] Girish, G. A., Anjum, A., Ganesh, G., Hegde, G. N., & Gowtham, S. V. (2024). The Cloudburst Prediction System. *Journal of Emerging Technologies and Innovative Research (JETIR)*, 11(5), 363-366. Retrieved from www.jetir.org.
- [2] Lee, J., Kim, H., & Park, C. (2024). Satellite-based rainfall prediction using convolutional neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(3), 1050-1062. <https://doi.org/10.1109/TNNLS.2024.3267892>
- [3] Sivaprakash, V., Sharathi, R., Karunya, V., & Umashankari, V. (2024). Cloudburst Prediction System (CBPS). *International Journal of Research Publication and Reviews*, 5(4), 1606-1611. Retrieved from www.ijrpr.com.
- [4] Telsang, S., Sawale, P., Pujari, S., Pujari, V., Pungale, A., Dubewar, R., & Shendre, R. (2024). Cloudburst Prediction System Using Machine Learning. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, 12(11), 1680-1689. Retrieved from www.ijraset.com.
- [5] Verma, K., Sharma, P., & Mehta, N. (2023). Real-time weather monitoring and cloudburst prediction system using IoT and AI. *IEEE Internet of Things Journal*, 9(5), 3356-3364. <https://doi.org/10.1109/JIOT.2023.3265432>
- [6] Singh, S., Kumar, R., & Sharma, A. (2022). Machine learning-based cloudburst prediction using meteorological data. *IEEE Transactions on Geoscience and Remote Sensing*, 60, 1-10. <https://doi.org/10.1109/TGRS.2022.3145678>

[7] Banerjee, M., & Choudhary, S. (2022). Integrating remote sensing data for early detection of extreme rainfall events. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15, 672-684. <https://doi.org/10.1109/JSTARS2022.3149876>

[8] Patel, A., & Gupta, M. R. (2021). Deep learning model for heavy rainfall and cloudburst forecasting. *IEEE Access*, 8, 122345-122356. <https://doi.org/10.1109/ACCESS2021.3098765>

APPENDICES

APPENDIX A PSEUDO CODE

1. Data Handling (data.py):

Read weather dataset (CSV) using pandas.

Return the DataFrame or filtered subset as JSON for API use.

Define '/weather' route:

- Import and invoke 'get_weather_data()' from data.py.
- Read weather data from CSV file.
- Convert to JSON format and return it as API response.

2. Backend (Flask API):

Initialize Flask application.

Run Flask app to listen for incoming HTTP requests.

- FUNCTION get_weather_data():

```
LOAD list_of_cities
INITIALIZE weather_data = []

FOR each city IN list_of_cities:
    CALL weather_API(city)
    PARSE temperature, humidity, pressure, wind, cloudiness

    CALCULATE cloudburst_risk_score:
        risk_score = weighted_function(temperature, humidity, pressure, wind,
                                       cloudiness)
        risk_level = "High Risk" IF score > threshold ELSE "Low Risk"

APPEND {
    city,
    weather_params,
    risk_score,
```

```
risk_level  
} TO weather_data  
  
RETURN JSON(weather_data)
```

- FUNCTION get_stats():
CALL get_weather_data()
COUNT number_of_high_risk_cities
COUNT number_of_low_risk_cities

RETURN JSON({
 high_risk: count,
 low_risk: count
})

3. Frontend (React + Vite Dashboard)

Start the React application using Vite and mount it on the DOM.

App.jsx:

Initialize two state variables:

- showLiveWeather (boolean)
- showRiskChart (boolean)

Display two buttons on the landing page:

- "Live Weather"
- "Risk Distribution"

On button click:

Update state to conditionally render components.

Conditionally render:

- LiveWeatherTable: Displays real-time weather data.
- PowerBIEmbed: Displays embedded Power BI report.

Apply visual effects:

- CSS animations (fade-in, glow).
- Background overlays for rain and lightning.

- ON component_load():
 - CALL '/api/weather'
 - STORE data IN weatherData
 - FILTER and ALERT IF any city.risk_level == "High Risk"

- CALL '/api/stats'
- STORE pieData for chart rendering

- ON view_toggle(view):
 - SET activeView = view
 - IF view == 'chart':
 - SHOW risk chart using pieData
 - ELSE IF view == 'powerbi':
 - RENDER Power BI iframe
 - ELSE:
 - DISPLAY weather table

- ON search_city(term):
 - FILTER weatherData WHERE city CONTAINS term
 - DISPLAY filtered results

4. Live Weather Table Component

- On mount:
 - Fetch weather data from Flask API at '/weather'.
- Display data in a styled HTML table:
 - Fields include: ID, Location, Temperature, Humidity, Timestamp
- Ensure proper alignment, spacing, and responsiveness.

5. Risk Scoring Logic (Conceptual)

```
FUNCTION weighted_function(temp, humidity, pressure, wind, cloudiness):
    score = (0.2 * temp) + (0.3 * humidity) - (0.1 * pressure) + (0.2 * wind) +
    (0.2 * cloudiness)
```

RETURN score

6. Power-BI Embed Component

Embed Power BI report in an iframe:

- Use report URL with proper authentication tokens if needed.
- Animate report container on load using framer-motion.
- Ensure styling matches main dashboard with responsive layout.
 - ON page load:

RENDER main React App

INSIDE PowerBIEmbed Component:

DEFINE Power BI iframe with report link

SET iframe to responsive layout

DISPLAY embedded Power BI dashboard in a styled container

USE animation (Framer Motion) to fade in dashboard smoothly

7. Deployment (AWS EC2)

Deploy the complete web application on an AWS EC2 instance. Expose the Flask backend API and serve the React frontend using Vite build. Ensure security group rules allow access to required ports (e.g., 80, 443).

STEP 1: Launch EC2 instance on AWS with Ubuntu or Amazon Linux.8

STEP 2: SSH into the instance using PEM key.

STEP 3: Install required dependencies (Python, Flask, pip, etc.).

STEP 4: Upload backend files via SCP or Git.

STEP 5: Start the Flask API using Flask run.

STEP 6: Open required ports in EC2 security group (e.g., 5000, 80).

STEP 7: Use public DNS to access the API from frontend or browser.

STEP 8: Deployed frontend to EC2 .

APPENDIX B

SCREENSHOTS

```

lnish@DESKTOP-JPJJGDF MINGW64 ~ (master)
$ cd Desktop
lnish@DESKTOP-JPJJGDF MINGW64 ~/Desktop (master)
$ chmod 400 "cloud.pem"

lnish@DESKTOP-JPJJGDF MINGW64 ~/Desktop (master)
$ ssh -i "cloud.pem" ec2-user@ec2-51-20-52-197.eu-north-1.compute.amazonaws.com
ssh: connect to host ec2-51-20-52-197.eu-north-1.compute.amazonaws.com port 22: Connection timed out

lnish@DESKTOP-JPJJGDF MINGW64 ~/Desktop (master)
$ ssh -i "cloud.pem" ec2-user@ec2-51-20-52-197.eu-north-1.compute.amazonaws.com
ssh: connect to host ec2-51-20-52-197.eu-north-1.compute.amazonaws.com port 22: Connection timed out

lnish@DESKTOP-JPJJGDF MINGW64 ~/Desktop (master)
$ ssh -i "cloud.pem" ec2-user@ec2-51-20-52-197.eu-north-1.compute.amazonaws.com
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023/
Last login: Mon Apr 21 07:22:48 2025 from 106.76.246.122

```

Figure B.1. CONNECTION OF AWS EC2 INSTANCE

```

lnish@DESKTOP-JPJJGDF MINGW64 ~/Desktop (master)
$ scp -i "C:/Users/lnish/Desktop/n/cloud.pem" -r "C:/Users/lnish/Desktop/n" ec2-user@ec2-51-20-52-197.eu-north-1.compute.amazonaws.com:/home/ec2-user/
  xg1.py
  cloud.pem
  cloudburst_rf_model.pkl
  data.py
  scaler.n1
100% 2417    7.9KB/s  00:00
100% 1678    8.0KB/s  00:00
100% 111KB  144.2KB/s 00:00
100% 9579   21.6KB/s  00:00
100% 821    2.6KB/s  00:00

```

Figure B.2. COPYING FILES FROM LOCAL DESKTOP TO EC2 INSTANCE

```
(venv) [ec2-user@ip-172-31-30-206 ~]$ ^C
(venv) [ec2-user@ip-172-31-30-206 ~]$ sudo -u postgres psql
could not change directory to "/home/ec2-user": Permission denied
psql (15.12)
Type "help" for help.

postgres=# CREATE TABLE weather (
    id SERIAL PRIMARY KEY,
    timestamp TIMESTAMPTZ,
    city VARCHAR(100),
    temperature FLOAT,
    humidity FLOAT,
    pressure FLOAT,
    wind_speed FLOAT,
    cloudiness FLOAT
);
CREATE TABLE
postgres=# CREATE TABLE risk_predictions (
    id SERIAL PRIMARY KEY,
    city VARCHAR(100),
    timestamp TIMESTAMPTZ,
    risk_level VARCHAR(50),
    probability FLOAT
);
CREATE TABLE
```

Figure B.3. CREATING TABLES TO STORE DATA

```
(venv) [ec2-user@ip-172-31-30-206 ~]$ sudo vi /var/lib/pgsql/data/postgresql.conf
(venv) [ec2-user@ip-172-31-30-206 ~]$ sudo systemctl restart postgresql
(venv) [ec2-user@ip-172-31-30-206 ~]$ sudo netstat -tuln | grep 5432
tcp        0      0 0.0.0.0:5432          0.0.0.0:*
                  LISTEN
tcp6       0      0 ::1:5432           ::1:*
                  LISTEN
(venv) [ec2-user@ip-172-31-30-206 ~]$ ^C
(venv) [ec2-user@ip-172-31-30-206 ~]$ sudo vi /var/lib/pgsql/data/pg_hba.conf
(venv) [ec2-user@ip-172-31-30-206 ~]$ sudo systemctl reload postgresql
```

Figure B.4. PORT CONNECTION

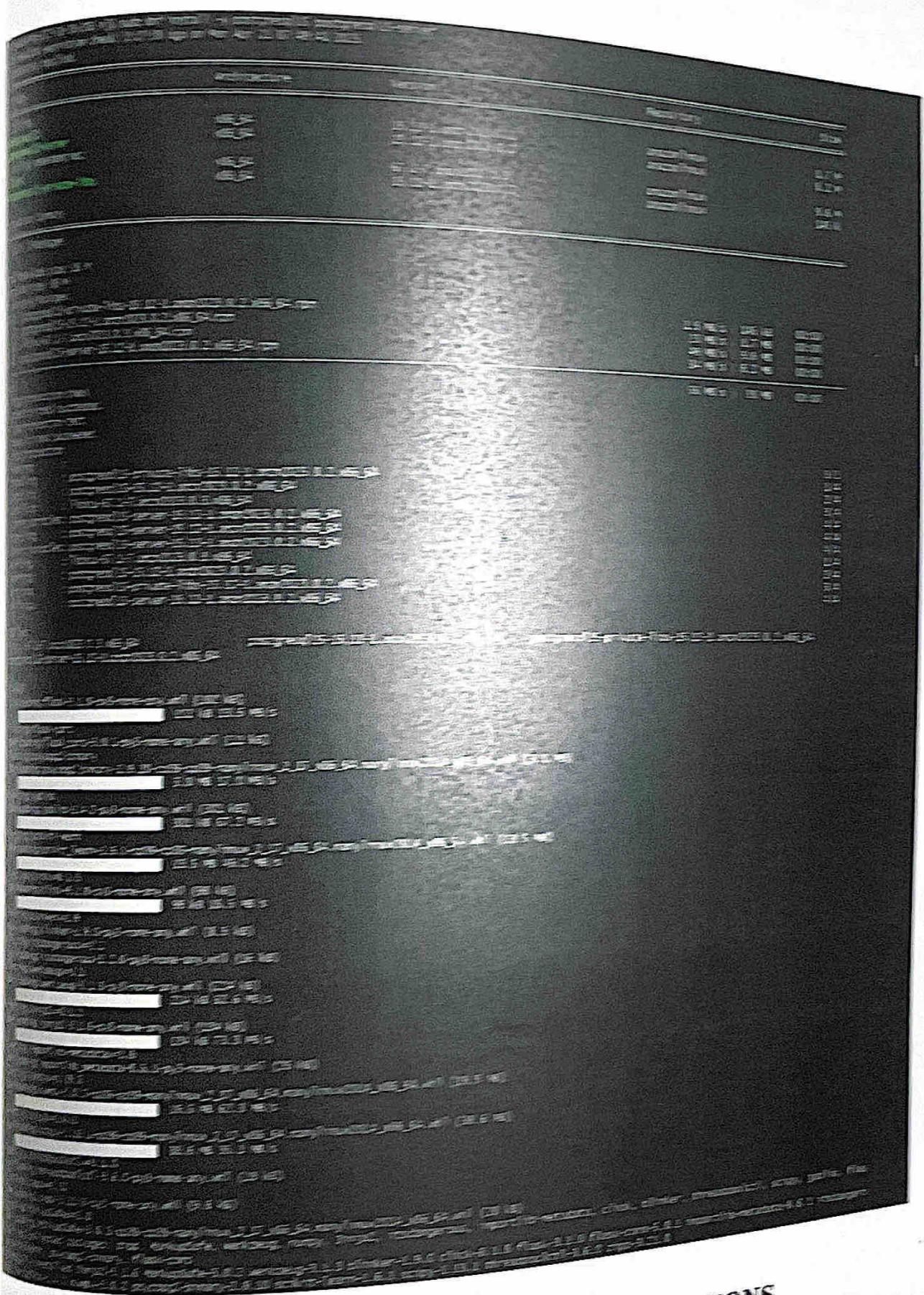


Figure B.5. REQUIRED LIBRARY INSTALLATIONS

```
(venv) [ec2-user@ip-172-31-30-206 n]$ pip install scikit-learn==1.5.2
Collecting scikit-learn==1.5.2
  Downloading scikit_learn-1.5.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (13 kB)
Requirement already satisfied: numpy==1.19.5 in ./venv/lib/python3.9/site-packages (from scikit-learn==1.5.2) (1.19.5)
Requirement already satisfied: scipy==1.6.0 in ./venv/lib/python3.9/site-packages (from scikit-learn==1.5.2) (1.6.0)
Requirement already satisfied: joblib==1.2.0 in ./venv/lib/python3.9/site-packages (from scikit-learn==1.5.2) (1.2.0)
Requirement already satisfied: threadpoolctl>=3.1.0 in ./venv/lib/python3.9/site-packages (from scikit-learn==1.5.2) (3.6.0)
downloading scikit_learn-1.5.2-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.4 MB)

Installing collected packages: scikit-learn
  Attempting uninstall: scikit-learn
    Found existing installation: scikit-learn 1.6.1
      Uninstalling scikit-learn-1.6.1...
        Successfully uninstalled scikit-learn-1.6.1
Successfully installed scikit-learn-1.5.2
(venv) [ec2-user@ip-172-31-30-206 n]$ python data.py
* Starting Cloudburst Prediction Pipeline...
No data found for training!
No data found for training!
Total cities fetched: 100
Inserting 100 weather records:
City: Mumbai, Time: 2025-04-21 11:24:50+0000
City: Bengaluru, Time: 2025-04-21 11:24:50+0000
City: Chennai, Time: 2025-04-21 11:24:50+0000
City: Kolkata, Time: 2025-04-21 11:24:50+0000
City: Hyderabad, Time: 2025-04-21 11:24:50+0000
City: Ahmedabad, Time: 2025-04-21 11:24:50+0000
City: New Delhi, Time: 2025-04-21 11:24:51+0000
City: Pune, Time: 2025-04-21 11:24:51+0000
City: Delhi, Time: 2025-04-21 11:24:51+0000
City: Surat, Time: 2025-04-21 11:24:51+0000
City: Puducherry, Time: 2025-04-21 11:24:51+0000
City: Patna, Time: 2025-04-21 11:24:51+0000
City: Bhubaneswar, Time: 2025-04-21 11:24:51+0000
City: Bhopal, Time: 2025-04-21 11:24:51+0000
City: Ranchi, Time: 2025-04-21 11:24:51+0000
City: Raipur, Time: 2025-04-21 11:24:51+0000
City: Kanpur, Time: 2025-04-21 11:24:51+0000
City: Lucknow, Time: 2025-04-21 11:20:35+0000
City: Dehradun, Time: 2025-04-21 11:24:51+0000
City: Chandigarh, Time: 2025-04-21 11:24:51+0000
City: Vijayawada, Time: 2025-04-21 11:20:35+0000
City: Thiruvananthapuram, Time: 2025-04-21 11:24:52+0000
City: Srinagar, Time: 2025-04-21 11:24:52+0000
City: Jaipur, Time: 2025-04-21 11:24:52+0000
City: Amritsar, Time: 2025-04-21 11:24:52+0000
City: Warangal, Time: 2025-04-21 11:24:52+0000
City: Port Blair, Time: 2025-04-21 11:24:52+0000
City: Panjin, Time: 2025-04-21 11:24:52+0000
City: Palwal, Time: 2025-04-21 11:24:52+0000
Predicted Cloudburst Risk: Low Risk (Probability: 0)
DEBUG: Updating city=Gurugram, timestamp=2025-04-21 11:46:09+0000
Prediction saved successfully!
Predicted Cloudburst Risk: Low Risk (Probability: 0)
DEBUG: Updating city=Guntur, timestamp=2025-04-21 11:46:09+0000
Prediction saved successfully!
Predicted Cloudburst Risk: Low Risk (Probability: 0)
DEBUG: Updating city=Kalaburagi, timestamp=2025-04-21 11:46:09+0000
Prediction saved successfully!
Predicted Cloudburst Risk: Low Risk (Probability: 0)
DEBUG: Updating city=Gorakhpur, timestamp=2025-04-21 11:46:09+0000
Prediction saved successfully!
Predicted Cloudburst Risk: Low Risk (Probability: 0)
DEBUG: Updating city=Godhra, timestamp=2025-04-21 11:46:10+0000, risk_level=Low Risk
Prediction saved successfully!
Predicted Cloudburst Risk: Low Risk (Probability: 0)
DEBUG: Updating city=Guwahati, timestamp=2025-04-21 11:46:10+0000, risk_level=Low Risk
Prediction saved successfully!
Predicted Cloudburst Risk: Low Risk (Probability: 0)
DEBUG: Updating city=Gandhinagar, timestamp=2025-04-21 11:46:10+0000, risk_level=Low Risk
Prediction saved successfully!
Predicted Cloudburst Risk: Low Risk (Probability: 0)
DEBUG: Updating city=Faridabad, timestamp=2025-04-21 11:46:10+0000, risk_level=Low Risk
Prediction saved successfully!
Predicted Cloudburst Risk: Low Risk (Probability: 0)
DEBUG: Updating city=Erode, timestamp=2025-04-21 11:46:10+0000, risk_level=Low Risk
Prediction saved successfully!
Predicted Cloudburst Risk: Low Risk (Probability: 0)
DEBUG: Updating city=Eluru, timestamp=2025-04-21 11:46:10+0000, risk_level=Low Risk
Prediction saved successfully!
Predicted Cloudburst Risk: Low Risk (Probability: 0)
DEBUG: Updating city=Alwar, timestamp=2025-04-21 11:46:10+0000, risk_level=Low Risk
Prediction saved successfully!
Weather data processing & prediction (RF) completed!
(venv) [ec2-user@ip-172-31-30-206 n]$ 

Weather data processing & prediction (RF) completed!
(venv) [ec2-user@ip-172-31-30-206 n]$ python api.py
* Serving Flask app 'api'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
  Debugger is active!
  Debugger PIN: 102-355-347
(venv) [ec2-user@ip-172-31-30-206 n]$ 
```

Figure B.6. RUNNING BACKEND SERVER

```
A newer release of "Amazon Linux" is available.
Version 2023.7.20250428:
Run "/usr/bin/dnf check-release-update" for full release and version update info.

.
└── ####_          Amazon Linux 2023
    └── ####\_
        └── \###]
            └── \#/__> https://aws.amazon.com/linux/amazon-linux-2023
                └── V-->
                    └── /_
                        └── /_
                            └── /m/
Last login: Sun May 11 08:52:15 2025 from 106.76.247.218
[ec2-user@ip-172-31-30-206 ~]$ cd ~/n/weather-dashboard
[ec2-user@ip-172-31-30-206 weather-dashboard]$ npm run dev -- --host
> weather-dashboard@0.0.0 dev
> vite --host

VITE v6.3.5 ready in 201 ms
- Local: http://localhost:5173/
- Network: http://172.31.30.206:5173/
- press h + enter to show help
```

Figure B.7. RUNNING FRONTEND SERVER

Cloud Burst Prediction System

The screenshot displays two tabs of the AWS EC2 Instance Details page for instance `i-0aea4964fe0b762a9`.

Instance summary for i-0aea4964fe0b762a9 (cloudburst) [Info]

Updated less than a minute ago

Instance ID	51.20.52.197 [Public IP]	Public IPv4 address	51.20.52.197 [open address]
IPv6 address	-	Private IPv4 addresses	172.31.30.205
Hostname type	IP name: ip-172-31-30-205.eu-north-1.compute.internal	Private IP DNS name (IPv4 only)	ip-172-31-30-205.eu-north-1.compute.internal
Answer private resource DNS name	IPv4 (A)	Instance type	t3.micro
Auto-assigned IP address	-	VPC ID	vpc-0bcb25e7fbcc32f
IAM Role	-	Subnet ID	subnet-0d0de161963775c5d
IMDSv2 Required	-	Instance ARN	arn:aws:ec2:eu-north-1:710271920198:instance/i-0aea4964fe0b762a9
Operator	-	Elastic IP addresses	-

Security

Owner ID: 710271920198 | Launch time: Mon Apr 21 2025 12:51:13 GMT+0530 (India Standard Time)

Inbound rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups	Description
sg-05c117ed3f77446b	sgr-0734116bc55266633	5000	TCP	0.0.0.0/0	launch-wizard-1	-
-	sgr-07c414315975e0e0	80	TCP	0.0.0.0/0	launch-wizard-1	-
-	sgr-0eb3806a5e0449716	22	TCP	0.0.0.0/0	launch-wizard-1	-
-	sgr-05c5494b07c77454d	443	TCP	0.0.0.0/0	launch-wizard-1	-
-	sgr-0638764ac095d46	5432	TCP	0.0.0.0/0	launch-wizard-1	-
-	sgr-0b39542c1a01d724	5173	TCP	0.0.0.0/0	launch-wizard-1	-

Outbound rules

Name	Security group rule ID	Port range	Protocol	Destination	Security groups	Description
-	sgr-0b39542c1a01d724	All	All	0.0.0.0/0	launch-wizard-1	-

Figure B.8. AWS EC2 INSTANCE DASHBOARD

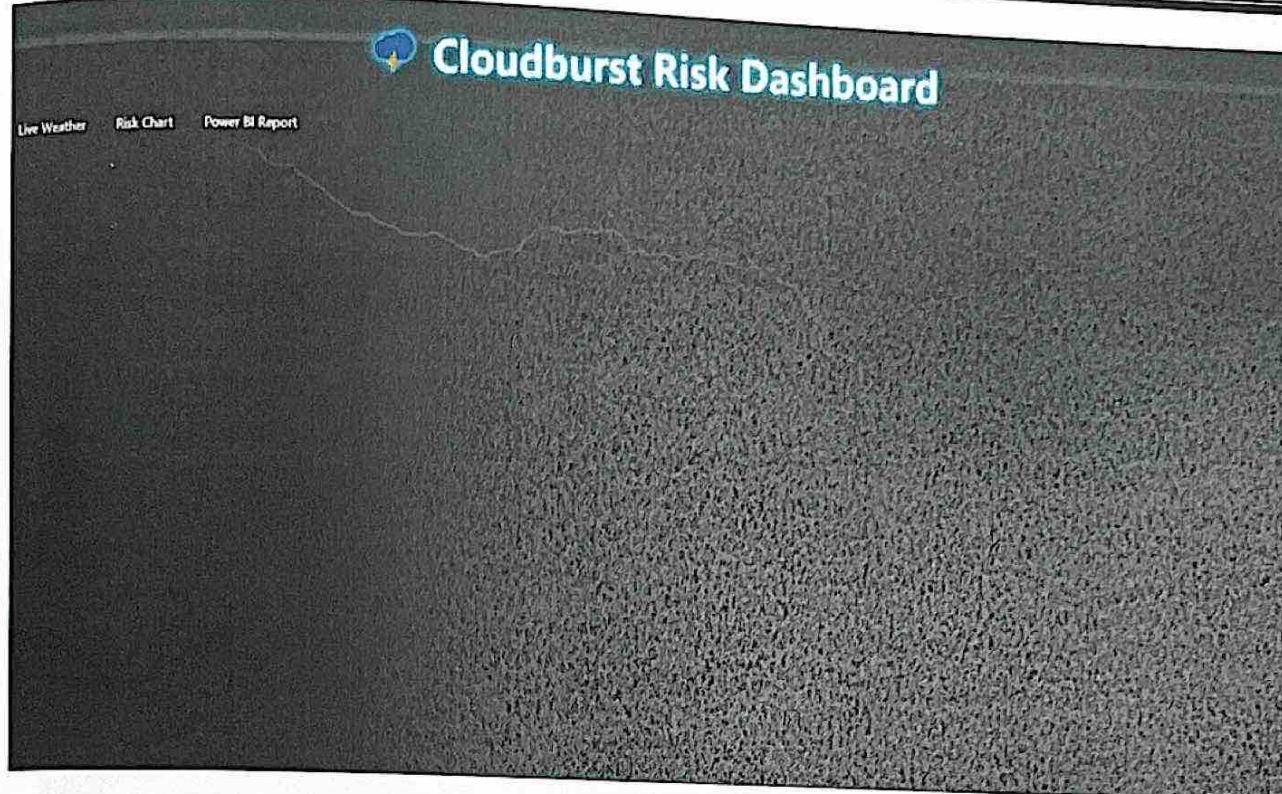


Figure B.9. FRONT END (UI/UX)

Live Weather & Risk

City	Timestamp	Temp (°C)	Humidity (%)	Pressure (hPa)	Wind (m/s)	Cloudiness (%)	Risk Level	Prediction Score
Mumbai	22/4/2025, 11:00:11 am	31.89	50	1011	1.85	7	Low Risk	0
Bengaluru	22/4/2025, 11:00:00 am	33.21	37	1012	4.2	59	Low Risk	0
New Delhi	22/4/2025, 11:08:12 am	37.53	4	1009	2.95	1	Low Risk	0
Pune	22/4/2025, 10:59:40 am	35.36	16	1013	3.39	8	High Risk	1
Puducherry	22/4/2025, 11:09:13 am	32.7	54	1011	5.04	86	Low Risk	0
Patna	22/4/2025, 10:59:08 am	41.18	13	1007	3.84	2	Low Risk	0
Kanpur	22/4/2025, 11:08:13 am	41.76	3	1007	5.75	0	Low Risk	0
Lucknow	22/4/2025, 10:58:45 am	40.65	5	1008	3.84	0	Low Risk	0
Dehradun	22/4/2025, 11:03:55 am	35.94	9	1009	3.55	0	Low Risk	0
Chandigarh	22/4/2025, 11:00:43 am	37.93	7	1006	3.72	0	Low Risk	0
Vijayawada	22/4/2025, 11:08:14 am	38.29	32	1010	3.65	19	Low Risk	0
Thiruvananthapuram	22/4/2025, 11:03:48 am	31.6	60	1012	3.59	27	Low Risk	0
Srinagar	22/4/2025, 11:05:39 am	19.06	45	1018	1.3	0	Low Risk	0
Jaipur	22/4/2025, 11:03:34 am	37.08	5	1010	3.96	0	Low Risk	0
Warangal	22/4/2025, 11:07:14 am	39.79	23	1010	2.36	23	Low Risk	0
Port Blair	22/4/2025, 11:08:15 am	20.63	60	1011	3.52	66	Low Risk	0

High Risk Alerts:

- High Risk Alert in Puducherry
- High Risk Alert in Srikakulam
- High Risk Alert in Bhopal
- High Risk Alert in Kota
- High Risk Alert in Jodhpur
- High Risk Alert in Ghatotkach

Figure B.10 LIVE WEATHER

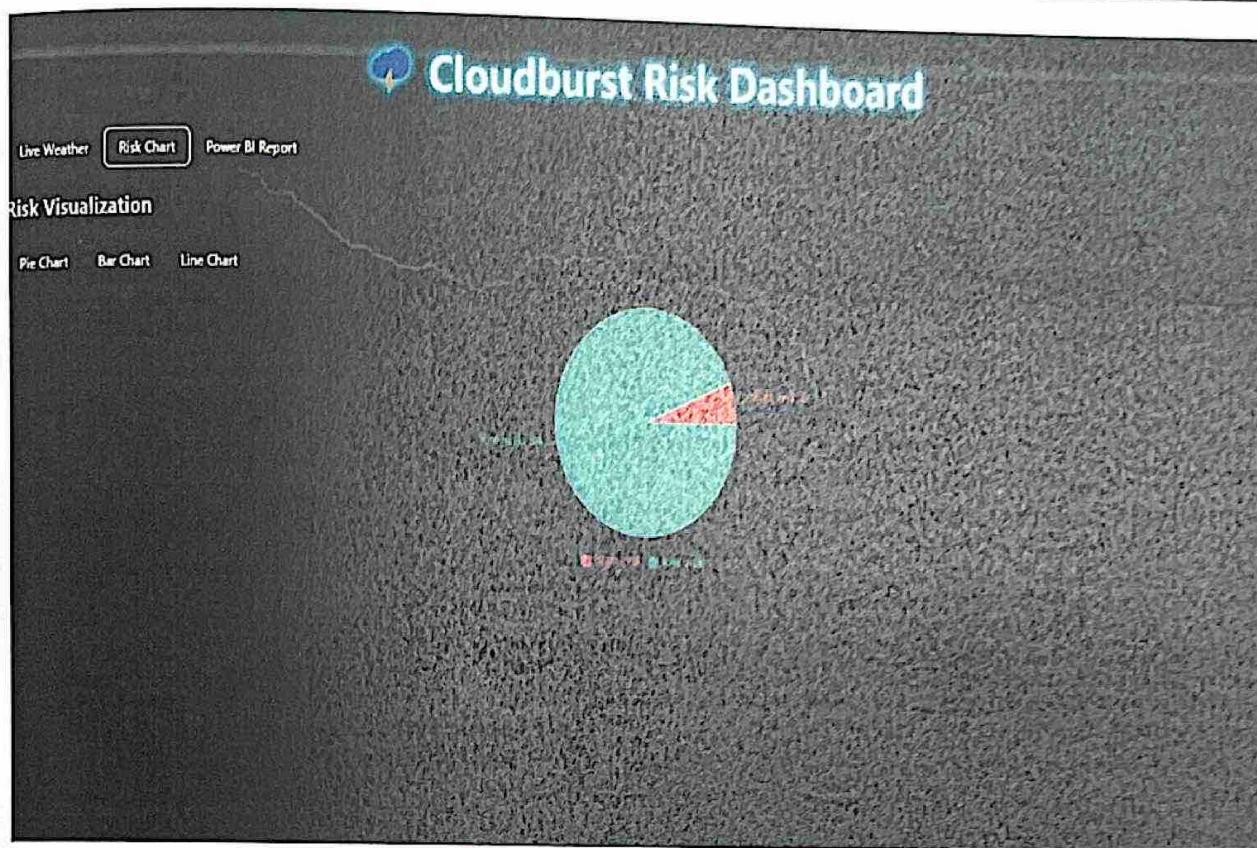


Figure B.11. RISK CHART

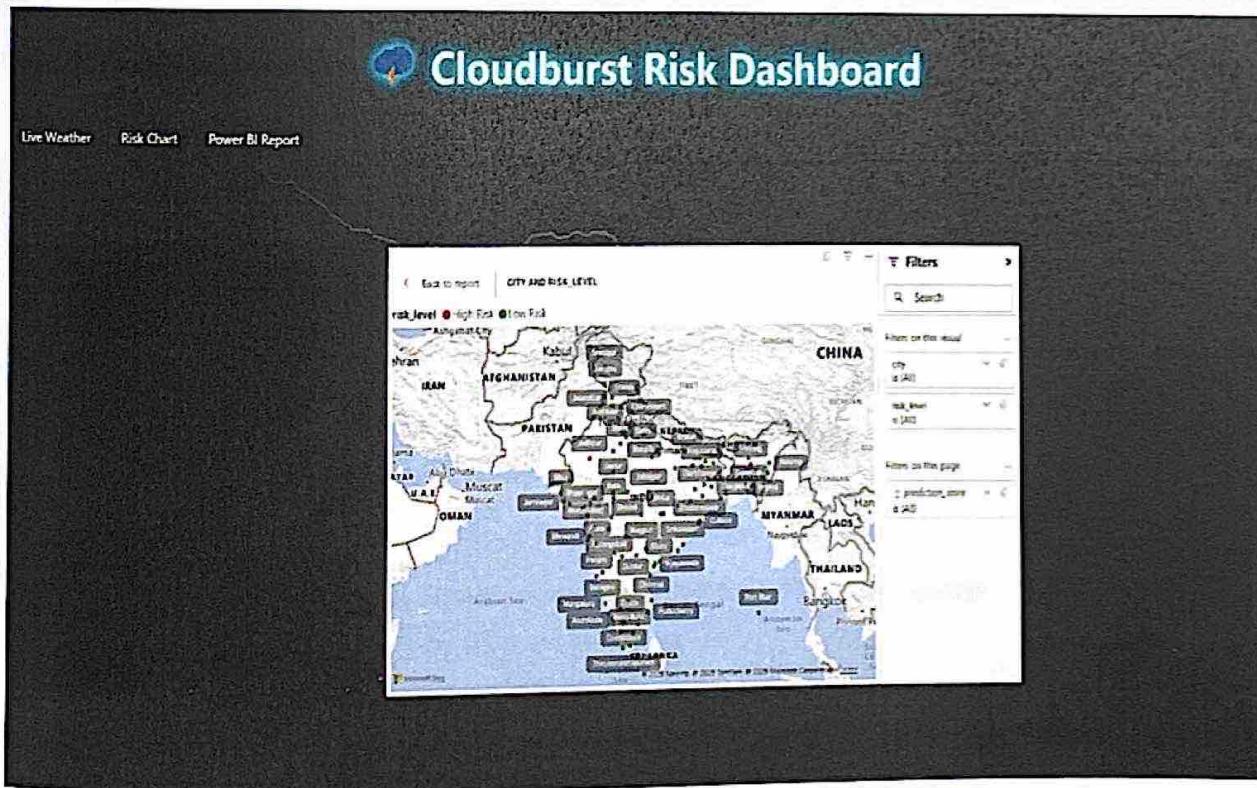


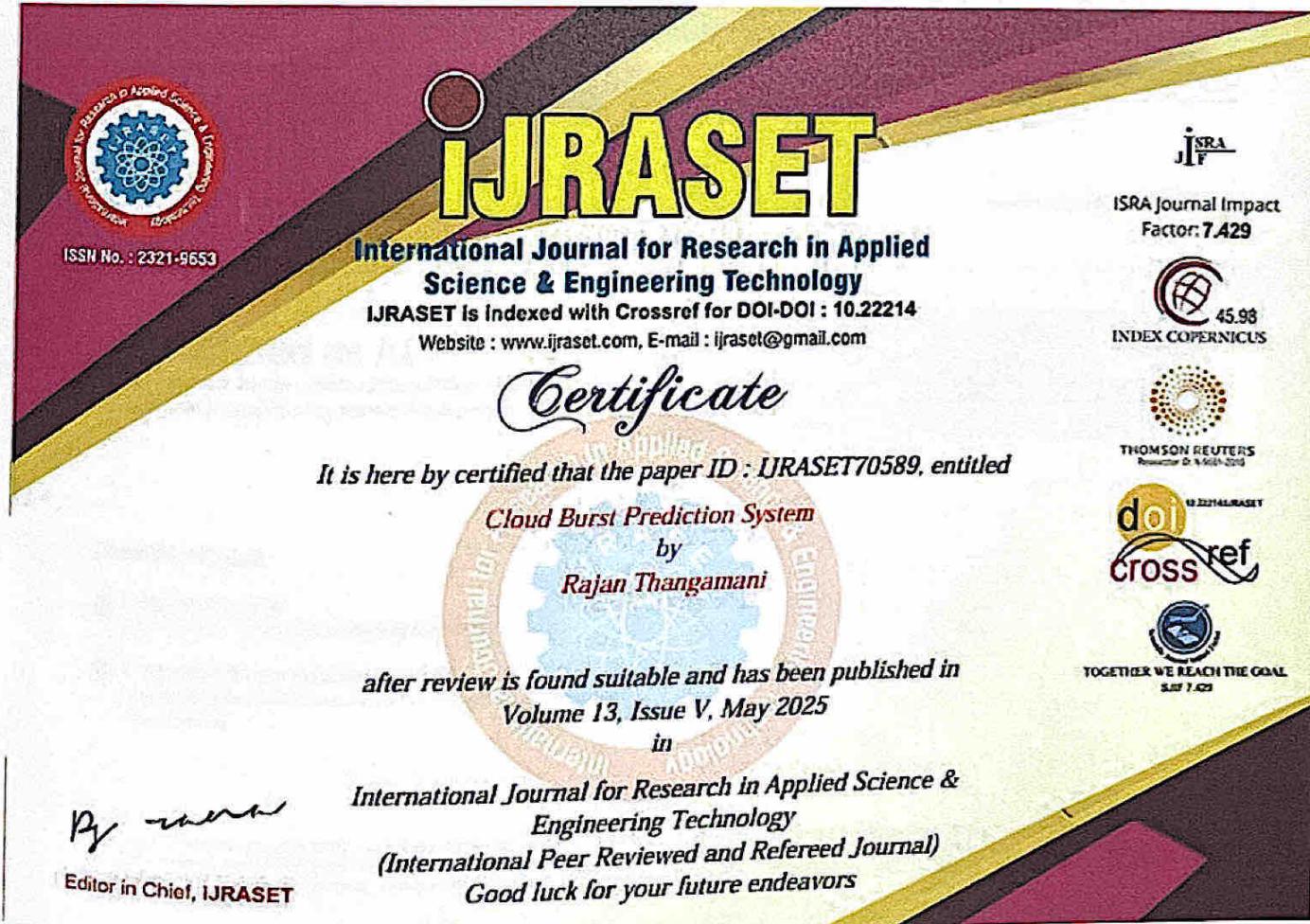
Figure B.12. POWER BI INTEGRATION

Appendix C: Enclosures

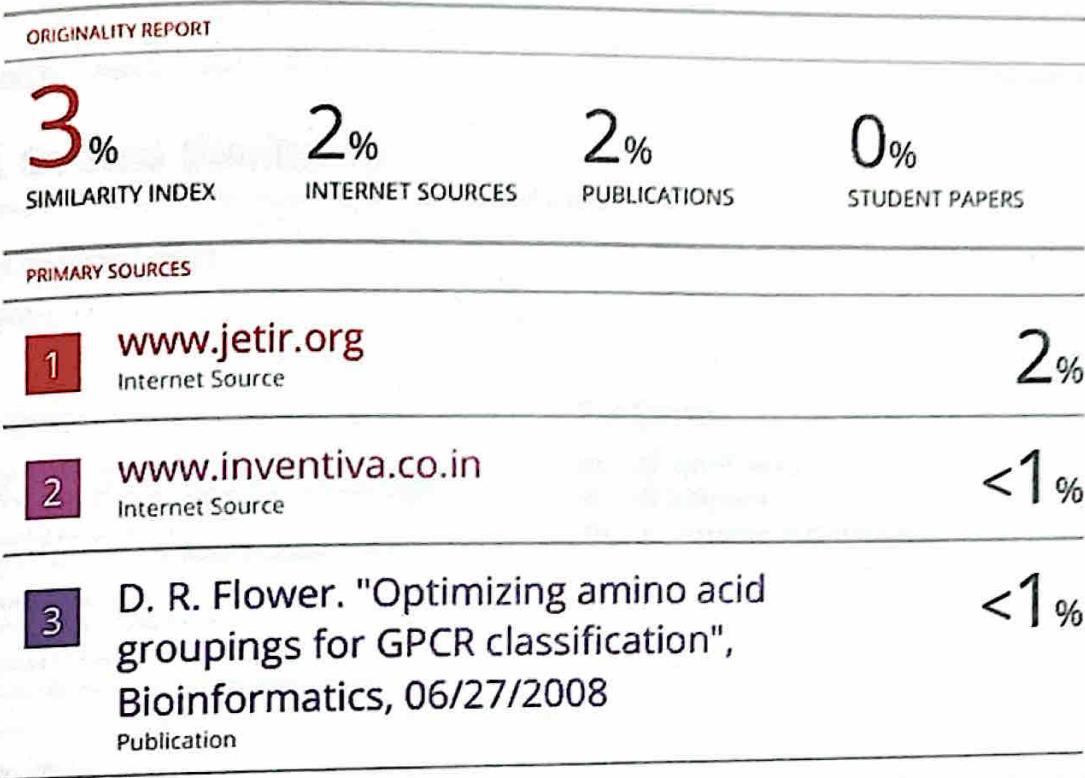
1. Certificates for Publishing Research paper







2. Plagiarism Check of Research Paper



Exclude quotes Off Exclude matches Off
 Exclude bibliography On

turnitin® Page 2 of 11 - AI Writing Overview

Submission ID: trnord-13242275672

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

0 AI-generated only 0%
Likely AI-generated text from a large language model.

0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

3. Plagiarism Check of Report



Page 2 of 61 - Integrity Overview

Submission ID trnmid: 13250487917

13% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

- 17 Not Cited or Quoted 13%
Matches with neither in-text citation nor quotation marks
- 0 Missing Quotations 0%
Matches that are still very similar to source material
- 0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- | | |
|-----|---|
| 9% | ● Internet sources |
| 8% | ● Publications |
| 12% | ● Submitted works (Student Papers) |

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.



Page 2 of 60 - AI Writing Overview

Submission ID trnmid: 13250487917

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

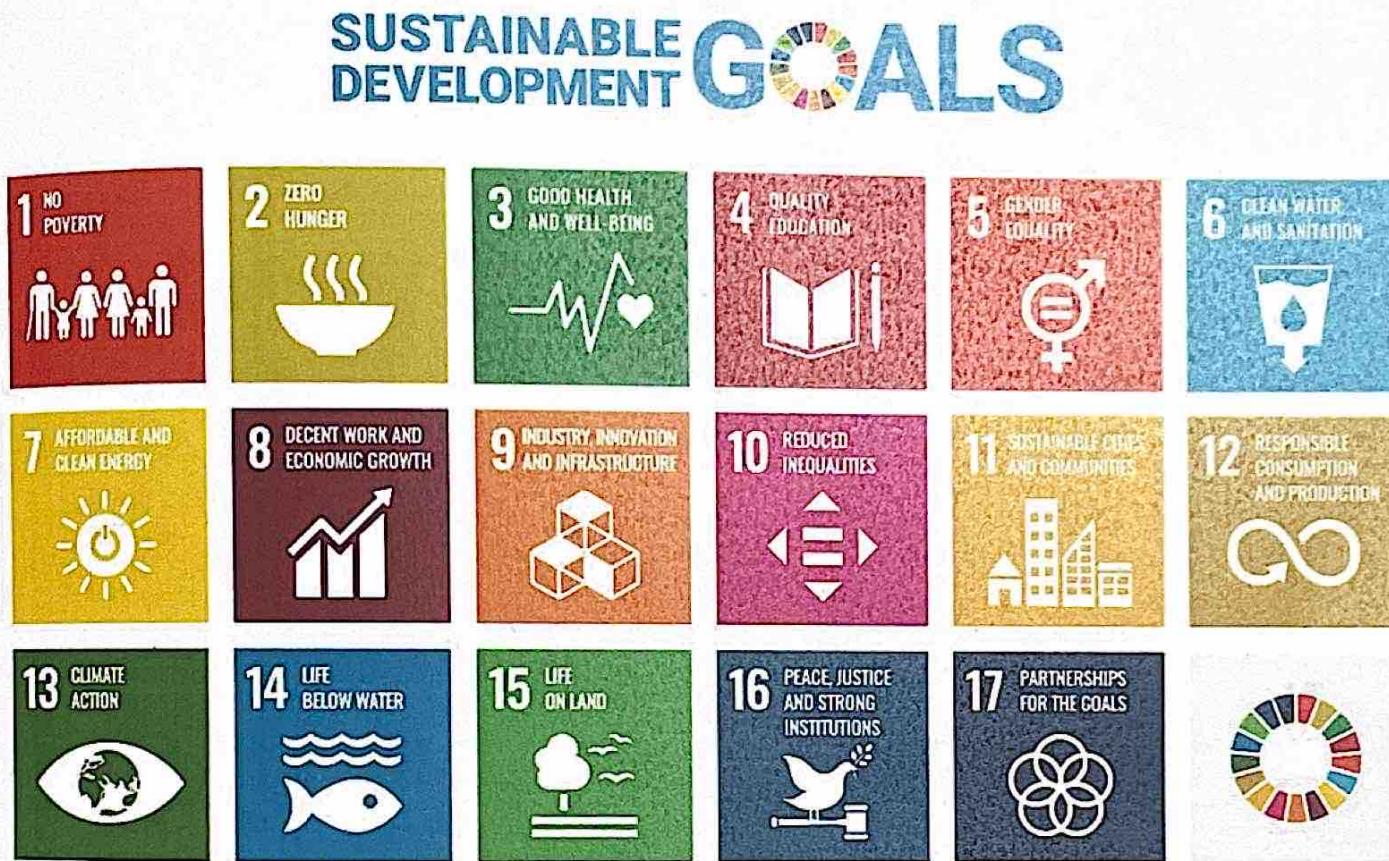
Detection Groups

- 0 AI-generated only 0%
Likely AI-generated text from a large-language model.
- 0 AI-generated text that was AI-paraphrased 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may misidentify writing that is likely AI generated as AI generated and AI paraphrased or likely AI generated and AI paraphrased writing as only AI generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

4. Details of mapping the project with the Sustainable Development Goals (SDGs)



SDG 3 - Good Health and Well-Being:

Supports disaster preparedness, reducing loss of life and improving public safety.

SDG 9 - Industry, Innovation and Infrastructure:

Promote innovation by integrating AI, ML, IOT and GIS technologies for forecasting.

SDG 11 – Sustainable Cities and Communities:

Helps reduce the impact of cloudbursts on urban areas through early warning systems.

SDG 13 - Climate Action:

Enhances resilience to climate - related disasters using real-time prediction and monitoring.