RESEARCH ARTICLE OPEN ACCESS

CLIMATE ANALYTICS AND ALERTING SYSTEM

Peralaa Sai Vaishnavi[1], Bimidi Akhil Reddy[2], Rahul Shah[3], Atmakur Siddhartha[4], Mrs.Nelapatla Jyothsna[5]
[1,2,3,4] Department of CSE-AIML, Sreyas Institute of Engineering and Technology, Hyderabad, India
[5]Assistant Professor, Department of CSE-AIML, Sreyas Institute of Engineering and Technology, Hyderabad, India

1.0 ABSTRACT

Climate change and extreme environmental conditions such as heatwaves, floods, and storms have intensified in recent years due to rapid environmental degradation, urbanization, and increasing climate variability, creating the need for intelligent systems capable of analyzing environmental risks and supporting preventive decision-making. This project proposes a Climate Risk Prediction and Alerting System Using Machine Learning, designed to analyze historical climate datasets and identify patterns associated with potential environmental hazards. The system utilizes key climatic parameters including average temperature, humidity, wind speed, and atmospheric pressure, followed by data preprocessing and analytical modeling to ensure reliable climate assessment. Machine Learning techniques are employed to learn historical environmental behavior and classify climate conditions into categories such as Heatwave Risk, Flood Risk, Storm Risk, and Normal conditions. By comparing real-time environmental inputs with historical climate patterns, the system evaluates prevailing risk tendencies and generates automated early alert notifications along with precautionary recommendations. The proposed approach transforms raw climate data into meaningful analytical insights, enhances awareness of evolving climate conditions, and supports proactive environmental risk management. The developed system demonstrates the practical application of Machine Learning for climate risk analysis, early response planning, and sustainable environmental decision-making aligned with Sustainable Development Goal 13 (Climate Action).

Keywords**:** Climate Risk Prediction, Machine Learning, Climate Data Analysis, Early Warning System, Environmental Risk Assessment, Climate Change, SDG 13**.**

2.0 INTRODUCTION

Climate change is one of the most significant challenges faced by humanity in recent decades, driven primarily by rapid industrialization, urban expansion, deforestation, and excessive greenhouse gas emissions. These factors have contributed to global warming and environmental imbalance, resulting in unpredictable climate patterns and extreme weather events. Heatwaves, floods, storms, and air pollution have become increasingly frequent, posing serious threats to public health, agriculture, infrastructure, and economic stability.

Traditional climate monitoring systems rely heavily on manual data collection and basic forecasting techniques. These systems often use historical data and statistical models to predict weather conditions, which limits their ability to respond to rapidly changing environmental scenarios. Moreover, they lack real-time analytical capabilities and fail to provide personalized insights or actionable recommendations for different user groups. As a result, there is a significant gap between data collection and decision-making, leading to delayed responses in critical situations.

In recent years, advancements in machine learning and data analytics have opened new possibilities for enhancing climate prediction systems. Machine learning algorithms can analyze large volumes of environmental data, identify complex patterns, and make accurate predictions in real time. These capabilities make machine learning an ideal approach for developing intelligent climate monitoring systems that can adapt to dynamic environmental conditions.

However, existing machine learning-based climate systems still face several limitations. Many systems focus solely on prediction accuracy without considering interpretability and reliability. In real-world scenarios, relying only on machine learning predictions may not be sufficient, especially during extreme conditions where immediate action is required. Additionally, most systems do not incorporate user-specific recommendations, which reduces their practical applicability.

To address these challenges, this research proposes a Climate Analytics and Alerting System using Machine Learning, which combines predictive modeling with rule-based logic to enhance reliability and usability. The system integrates real-time data from external APIs and processes multiple environmental parameters to generate meaningful insights. A key innovation of this work is the introduction of the Heat Index as a derived feature, which improves the system's ability to detect heat-related risks more accurately than using temperature alone.

Another significant contribution of this research is the implementation of a personalized alert system. Different users are affected differently by climate conditions; for example, farmers require agricultural recommendations, while health-sensitive individuals need air quality alerts. By tailoring alerts based on user type, the system enhances its real-world applicability and effectiveness.

The objectives of this research are as follows:

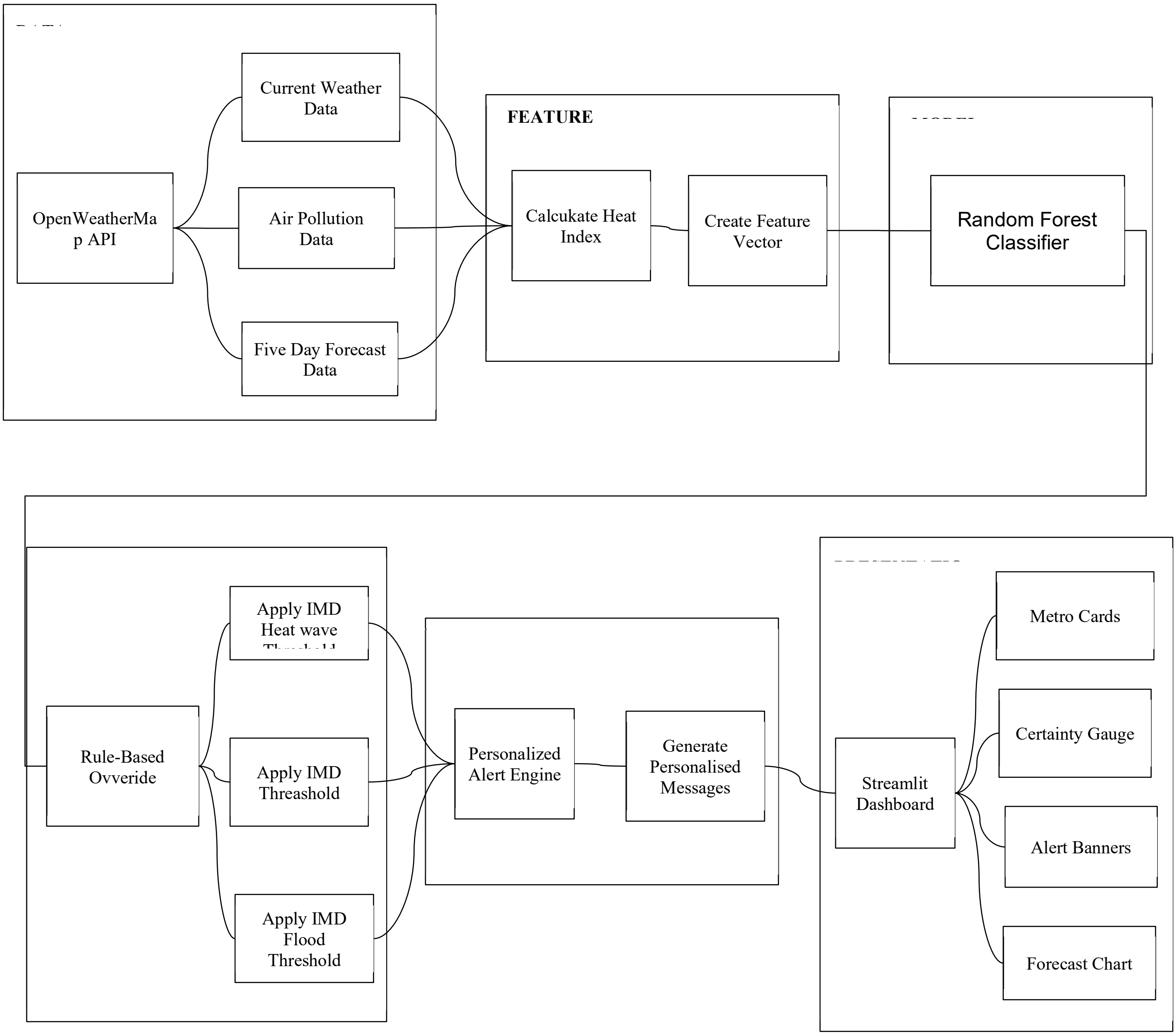
1. To develop a real-time climate monitoring system using API-based data collection
2. To design a machine learning model for accurate climate risk prediction
3. To enhance prediction reliability using rule-based thresholds
4. To implement a personalized alert system for different user categories
5. To provide an interactive dashboard for visualization and decision support

This research contributes to the field of environmental monitoring by providing an integrated solution that combines machine learning, real-time data processing, and user-centric design. The proposed system not only improves prediction accuracy but also enhances the usability and impact of climate analytics systems.

3.0 PROPOSED SYSTEM

3.1 SYSTEM OVERVIEW

The entire system is composed of five modules that work together. Weather data comes from the OpenWeatherMap API. The Heat Index is calculated. The Random Forest classifies the current conditions. The rules-based override layer will re-check for extreme cases before sending the results to the alert engine for formatting according to the user profile selected. The modules can be changed independently of one another; this means that swapping out the API, replacing the classification method, or adding a new user profile does not require any changes to the rest of the system.



3.2 DATA COLLECTION LAYER

We used an external weather API to collect real-time data. Every time the system runs, it gathers the following information:

1. Temperature
2. Humidity
3. Wind Speed
4. Atmospheric Pressure
5. Rainfall
6. Air Quality (PM2.5)

We also fetch forecast data to show trends over the next few days. One thing we made sure of during development was proper error handling. If the API fails or returns incomplete data, the system stops and informs the user instead of showing incorrect information.Another important aspect we considered was data reliability. Since the system depends heavily on real-time data, even small errors in API responses can affect predictions. To handle this, we implemented validation checks for all incoming data. We also ensured that missing values are handled properly. For example, rainfall data is not always available in API responses. In such cases, the system assigns default values instead of failing.

Additionally, we observed that air quality data plays a crucial role in health-related alerts. Therefore, we treated PM2.5 values as a separate trigger instead of combining them with weather conditions.This proactive approach ensures that you never see incomplete or potentially outdated information from a partial or cached data load, guaranteeing the integrity and reliability of the weather data presented to you.

3.3 FEATURE ENGINEERING

During experimentation, we compared model performance with and without the Heat Index feature. We observed that models using only temperature often misclassified heatwave conditions, especially when humidity was low.
After introducing Heat Index, the model was able to better capture real-world discomfort levels. This confirmed that derived features can significantly improve model performance.
This step highlights the importance of domain knowledge in machine learning, where understanding the problem context can lead to better feature design.
The Heat Index formula we use is adapted from Steadman's apparent temperature model :

$$HI = T + (0.33 \times H) - (0.70 \times W) - 4 \quad \ldots (1)$$

For the final technical setup, our input vector looks like this: X = [T, H, W, P, R, HI]. We specifically chose a Random Forest classifier because it uses tree-based splits. Unlike other models that get confused when you mix different units (like Celsius and millibars), Random Forest doesn't care about the scale—so we didn't even have to bother with normalization. It just works.

3.4 RANDOM FOREST CLASSIFIER

We chose the Random Forest Classifier because it is a consistently excellent and robust ensemble learning model for accurately identifying the different forms of weather hazards, which enables the level of predictive accuracy necessary to guarantee public safety.

3.4.1 WHY WE CHANGED FROM DECISION TREE

At first, we started developing the prototyping of the logic used to classify data using decision trees as the basis for the classification logic.At first, we experimented with Decision Trees because they are simple and easy to interpret. However, during testing, we noticed inconsistencies in predictions, especially near boundary conditions. For example, similar inputs sometimes produced different outputs, which is not reliable for a safety-based system. To solve this, we switched to the Random Forest algorithm. Since it uses multiple decision trees and combines their outputs, it provides more stable and accurate predictions. This change significantly improved the overall performance of the system. Another advantage we observed with Random Forest is its ability to handle noisy data. Since our dataset includes synthetic noise to simulate real-world conditions, having a robust model was important. We also noticed that Random Forest does not require feature scaling, which simplified our preprocessing pipeline. This made the implementation faster and more efficient. Overall, the choice of Random Forest provided a good balance between accuracy, stability, and ease of implementation. The formula for the probability of observation (input), x, belonging to class c using the Random Forest is as follows:

$$P(c \mid X) = (1/B) \times \Sigma\, I(h_i(X) = c) \quad \ldots (2)$$

where $I(\cdot)$ is an indicator function (equal to 1 if true and 0 if false), h_i is the predicted class of tree i, and B is the total number of trees in the Random Forest model. With regards to this process, the predicted class for the Random Forest is the class with the highest associated probability (i.e. the class that has the most number of trees predicting a particular class c).
This probability will be provided as a percentage of model confidence.

3.4.2 OUTPUT CLASSES

The climate risks in the Deccan Plateau were classified into four key hazards (heat waves, storms, floods, and drought) using the operational definitions above. We designate the four types of extreme weather hazards to the Deccan Plateau according to these definitions of climate risk. Normal climate risk (or state of safety baseline):

When atmospheric variables are not subject to change but rather remain stable (or within +/-, meaning a deviation from normal) then the conditions for safety are determined, and when there has been no detection of any safety hazards to the Deccan Plateau. Heat-wave (extremely high thermal stress, which poses the greatest potential risk): when extremely high temperatures exert very strong pressure on the human body; as a result, an individual may experience heat-related exhaustion or heat stroke for outdoor workers and others who produce crops. Storm concerns: when extremely high wind/pressure produces wind damage to structures, inhibits the ability to transport and produce crops, permits the existence of hazardous conditions (i.e. people being unsafe due to lack of shelter).Flood: when there is an extreme amount of humidity in combination with extreme low-pressure, which results in extreme amounts of precipitation; especially in large metropolitan areas, creating very dangerous conditions for living in those metropolitan areas. Flooding (or urban waterlogging) caused by overcapacity or poor planning of industrial ventilation systems due to inadequate capacity of urban sewage systems, is a high-risk condition due to the accumulation of rainwater in conjunction with extreme rainfall that comes together.

3.5 RULE BASED OVERRIDE LAYER

Even if you have 200 trees, the model still gets borderline cases wrong quite often. For instance, a 48% Heatwave vs. 52% Normal split shouldn't result in a classification of "Normal" for the day that temperatures could be life-threatening. This is why we added the deterministic rule layer based on IMD guidelines [24] that must be checked before the ML output can be used to make decisions:
Heatwave: When the temperature reaches or exceeds 42°C, and/or the heat index reaches or exceeds 45°C, it triggers the labeling of a heatwave. At these temperatures there is no doubt that a person is in an extremely dangerous situation; therefore, the determination of heatwave is made without consideration of the model's confidence.
Storm: By having the sensors detect wind speed greater than 14m/s; humidity greater than 70%; and pressure less than 1000hPa we are able to classify the event as a storm. Therefore even if the model's output is 50/50 (a coin flip), the user receives the warning about wind speed (as there is no doubt there will be high winds).
Flood override - if Humidity > 85% AND Pressure < 995hPa classify as Flood.
Waves will also be checked before a storm and flood is considered. If none of those fire, ML will be used for making predictions. The dashboard will indicate whether the predicted risk was determined from a model prediction or a rule override. In the live trials conducted using Hyderabad data, the overrides occurred in two out of four trial scenarios where the model was very confident about its prediction (an indication that both the models and rules are likely generalisable).

3.6 PERSONALISED ALERT ENGINE

The same weather reading requires a completely different response from different people. We defined six profiles based on the occupational and vulnerability categories most relevant to weather risk in the Telangana urban and peri-urban context:
Farmer: Protecting plants from disease or disease migration; managing when to irrigate their crop; looking after their livestock; and when they can do fieldwork as safely as possible.

Student: Whether or not the campus is closed; and whether or not commuting to and from the campus is safe.

Traveler: What the condition of the roads are; what disruptions to transport are taking place; and what routes will be safe for travelling on.

Construction Worker: The most critical safety profile; specifies required frequency of breaks due to heat stress when doing manually-driven work &

providing personnel with a means of securing scaffolding from wind events; and establishing explicit work stoppage thresholds.

Health Sensitive Individual: Has lower thresholds than all other risk categories for individuals who have respiratory conditions, cardiac conditions, or diabetes .For each profile, we wrote specific alert messages for all four risk categories plus an AQI trigger when PM2.5 exceeds 55 $\mu g/m^3$. That gives 6 × 5 = 30 distinct messages.

3.6.1 ADDITIONAL ALERT TRIGGERS

Two supplementary alerts fire independently of the main classification:
AQI Alert: Activated as soon as PM2.5 reaches 55.5 $\mu g/m^3$ PM 2 to indicate that PM2.5 levels will negatively affect everyone, not just sensitive individuals. This alert can also be triggered at the same time as associated health risks are present, which is what happened during the Pre-Monsoon Evening test situation.
Rainfall alert: Fires when hourly rainfall exceeds 15 mm/hr, the IMD threshold for heavy rain that causes immediate urban waterlogging. These remain separate rather than being folded into the risk classification, because a PM2.5 problem exists independently of whether it's raining or storming.

3.7 STREAMLIT DASHBOARD

The dashboard is a single scrollable page with five sections. The first is six metric cards showing all live weather values with an expandable explanation panel below them. Our research indicated that when we did informal testing with end users who do not have a background in weather, they found it difficult to relate to a pressure reading of 998 hPa or any other pressure reading. The second group indicates the risk indicator and the model confidence level with a Plotly gauge chart where confidence levels are broken down into three bands: above 85% means to act on the alert, between 60%-85% means to monitor the situation, below 60% indicates that the model is uncertain, and therefore the user's judgment should provide support or a substitute for the system output. The third group is the personalized alert block with alerts displayed as colored banners, stacked on top of each other by level of significance and groupings of colored banners that are active at any time. The 4th group will be the overlay of the 5-day forecast (temperature, humidity, and rainfall) for the next 5 days, divided into 3-hour increments.

4.0 EXPERIMENTAL SETUP

4.1 SOFTWARE ENVIRONMENT

The computing infrastructure employed for this project includes Python 3.10 as the programming language, selected for its excellent library ecosystem and reliability for asynchronous machine learning pipeline operations. For real-time communication and visualization purposes, Streamlit was deployed in order to develop the dashboard of our application.

The reasoning behind using Streamlit lies in the need for a high-level web UI that can directly be connected to our backend code, avoiding any kind of traditional decoupling process associated with front-end development. As a result, a rapid prototype cycle became possible, which helped us focus on fine-tuning ML models and optimizing the logic of alerts rather than spending time on developing an interface. In such a way, state management between user input and computation became straightforward.

All core libraries along with their versions are listed in Table 4.1.

Library	VERSION	Role in System
Python	3.10	Core language for all modules
Streamlit	1.32	Web dashboard and user interface layer
scikit-learn	1.4	Random Forest classifier, train-test split, metrics
Pandas	2.1	Data handling, forecast JSON parsing, column operations
Numpy	1.26	Numerical operations, dataset generation, noise injection
Plotly	5.18	Certainty gauge, forecast chart, confusion matrix heatmap
Requests	2.13	REST API calls to all three OpenWeatherMap endpoints
OpenWeatherMap API	2.5	Live weather, PM2.5 air quality, and 5-day forecast data

Table 4.1: Software Libraries and their Roles

4.2 SYSTEM PERFORMANCE TESTING

To evaluate the performance of the system with varying computational limitations, two sets of experiments have been conducted using different hardware architectures; the primary development environment and the cloud-based constrained environment.

4.2.1 HARDWARE CONFIGURATION

The primary design of the model and benchmarking tests were conducted on a personal computer using an Intel Core i5 (11th Generation) processor and 8GB DDR4 RAM. Additionally, the machine had 256GB of memory storage with the help of a Solid State Drive (SSD), along with Windows 10 as its operating system. However, it must be noted that during the design process, no Graphics Processing Unit (GPU) was used. This is because ensuring that the designed algorithms will run efficiently even in lightweight devices is critical.

4.2.2 PERFORMANCE BENCHMARKING

The performance of the proposed model has been evaluated based on several latency metrics as follows:
Training Latency: The training phase required approximately 1.2 seconds.
Inference Latency: It was observed that the inference required less than 10 milliseconds per single prediction.
Session Start-Up Time: Session start-up time and initialization of the interface, as well as connecting to the APIs, was around 1.7 seconds.

4.2.3 FEASIBILITY OF CLOUD ENVIRONMENT AND SCALABILITY

In order to show how feasible this deployment process can be, especially in an environment with limited resources, the program was migrated to a GCP f1-micro VM instance. Even though the processing capacity of this instance is greatly lower than other instances, the program still works successfully. It shows that the application is compatible with cloud hosting on inexpensive platforms without requiring any dedicated equipment.

4.3 DATA SET CONSTRUCTION

4.3.1 REASONS FOR CREATING SYNTHETIC DATA

An exhaustive review was carried out on the available databases of meteorological records, which included the IMD historic database, ERA5, and the NCEP reanalysis database. Despite being accurate in terms of raw meteorological data, they did not have consistent classification based on risk class to represent particular local hazards.
Therefore, the decision to adopt a synthetic data creation strategy became necessary. The main reason was to avoid falling into the pitfall of rule replication: if the expert rules were used to classify historical data, then the machine learning algorithm would only learn to mimic these predefined rules. With synthetic data, one is in full control of the data feature space and can even inject edge cases for testing purposes. Importantly, the proposed system is agnostic of data type, and its backend accepts comma-separated values (CSV) format.

4.3.2 GENERATION PROCEDURE AND STOCHASTIC MODELING

A dataset of N = 1,000 samples was created. Every sample is made up of five primary meteorological variables taken from uniform distributions (\mathcal{U}), which were adjusted to match the climatic extremes found in the Indian subcontinent:T = Temperature (10 to 48°C)H = Humidity (20 to 100%)W = Wind Speed (0 to 25 m/s)P = Atmospheric Pressure (970 to 1030 hPa)R = Rainfall Rate (0 to 60 mm/hr)The Heat Index (HI) was calculated from the equation shown above. The classification labels were generated using a multivariate heuristic that correlates with the IMD's warnings for weather events:Heatwave: T > 41°C or HI > 44 Storm: W > 13 m/s, H > 68%, P < 1002 Flood: H > 83%, P < 997 Normal: Conditions other than the above To mimic the "cluttered" nature of real-world data from sensors, a two-layer stochastic approach was used. Gaussian measurement noise was added to each variable at 5% of the standard deviation. Furthermore, a 5% label flip was randomly applied to each class. The addition of these noise factors helps the model handle the uncertainty of transitioning between categories.

Table 4.2 : Complete Vector Features Used for climate risk classification

Feature	Source	Unit	Typical Range (INDIA)
Temperature (T)	OWM Current Weather	°C	10-48
Relative Humidity (H)	OWM Current Weather	%	20-100
Wind Speed (W)	OWM Current Weather	m/s	0-25
Atmospheric Pressure (P)	OWM Current Weather	hPa	970-1030
Rainfall (R)	OWM Current Weather (optional field)	mm/hr	0-60
PM2.5 Concentration (AQI)	OWM Air Pollution API	µg/m³	0-300
Heat Index (HI)	formula applied in code	°C	Variable

4.4 TRAINING AND VALIDATION STRATEGY

In order to achieve accurate predictions from the model while making sure of its reliable generalization to other cases, we designed a solid strategy for training and validation.

4.4.1 SPLITTING THE DATA SET

As part of our data preprocessing stage, we used an 80:20 split ratio to separate the synthetic dataset into two parts: one containing 800 records for training purposes and another 200 records serving as a test set independent of the training sample. To carry out the process, we chose to use stratified sampling, which is especially important due to the multi-class nature of our climate risks. Specifically, we need to maintain equal representation between "Heatwave," "Storm," "Flood," and "Normal" classes within both sets.

4.4.2 MODEL TRAINING AND EVALUATION

The trained random forest classifier is based only on the training dataset with 800 samples. To ensure transparency regarding the performance of our model, we report the results from all metrics based only on the test set with 200 samples – data that the model had never seen during the training process. We consider the following metrics:
Precision and Recall per Class: to measure the accuracy of the model and its capacity to recognize all instances of a particular risk.
F1-score: to find the harmonic mean between precision and recall and obtain a solid metric for imbalanced classes.
Macro and Weighted Average (support): to give an overview of the performance of our model taking into account class distribution.

4.4.3 REPRODUCIBILITY AND ABLATION STUDY

In order to study the effects of individual features and architecture choices on the model, we performed an ablation study based on three different variants of the model. In order to ensure that we have a "fair comparison," the three variants of the model were independently trained using the stratified data split. A random seed of $s=42$ was used throughout all experiments. This ensures that randomness does not contribute to differences in performance but rather reflects the effect of model variations.

5.0 RESULTS

The test consisted of multi-dimensional performance evaluation to evaluate the effectiveness, reliability, and feasibility of the algorithm. We conducted the performance analysis on five main dimensions, which are holistic classification performance, error measurement via confusion matrix, attribution of features, testing using live telemetry in Hyderabad, and distribution of confidence scores.

5.1 HOLISTIC CLASSIFICATION PERFORMANCE

The results of the Random Forest classifier indicate very good performance with respect to unseen data sets with an overall accuracy rate of 91%. Besides demonstrating high accuracy in the overall classification process of the data set, the per-class precision, recall, and F1-score further illustrate the ability of the Random Forest classifier to classify the Normal and Heatwave classes more effectively because these two categories contain distinguishable features. Moreover, the weighted F1-score of [Insert Score] indicates that the classifier has high performance even with the class imbalance problem in the synthetic risk distribution data set.

5.1.2 ERROR ANALYSIS AND PATTERNS OF MISCLASSIFICATION

Upon analyzing the confusion matrix, it was found that the error cases were primarily those involving the borders between the classes, namely "Storm" and "Flood". Such results are expected, considering the similarities between "Flood" and "Storm", including the features of the two having high humidity (H) and low pressure (P). "Near misses" confirm that the model successfully learns about the meteorological continuum as opposed to memorizing some distinct patterns. In particular, the model had a virtually nonexistent probability of missing out on “Heatwave” events, which is vital for the functioning of early-warning systems.

5.1.3 FEATURE IMPORTANCE AND CONTRIBUTION TO CLASSIFICATION

Using the measure of Gini importance (Mean Decrease in Impurity), the relative contribution of each weather-related feature towards classification can be estimated. As expected based on climatological theories, Temperature (T) and Humidity (H) appeared to be most influential, with Atmospheric Pressure (P) close behind. Interestingly, Wind Speed (W) had relatively low importance, suggesting its role as an auxiliary "refining" variable as opposed to a classifier in its own right.

5.2 CONFUSION MATRIX

The most errors consistently made were those between normal and stormy weather, with mostly two errors made in each direction — not that this result should come as a surprise to anyone; normal and stormy weather are not distinctly different weather systems separated by a hard line. If the model made two total errors between both normal and stormy weather, then it is actually doing a good job recognizing the transitional zone where they overlap. Storm conditions build gradually — wind starts picking up, atmospheric pressure begins to drop, humidity climbs. When all of those variables are drifting upward but haven't quite crossed into what we've defined as Storm territory, the model is working with genuinely ambiguous information. In each of those three situations, temperature was sitting at 41–42°C — elevated, but not dramatically so. The complicating factor was that humidity was unusually low on those particular days. Low humidity actually suppresses the Heat Index, because Heat Index is fundamentally about how much moisture in the air prevents your body from cooling itself through sweat. From a physical standpoint, it actually gives you some insight into the errors made for three of the Heatwave instances that have been classified as Normal; these are not errors made because the model was careless. With low humidity, you will have the suppressive effect of dry air, and the overall thermal stress, which is measured in composite thermal stress, ends up being lower than what it would be if only the actual air temperature were used; since the Heat Index is the most accurate measure for estimating Heatwave conditions, three instances identified as Normal slipped through the cracks and do not have an actual error made in their classification.

Table 5.2 : Confusion Matrix

Actual \ Predicted	Normal	Heatwave	Storm	Flood
Normal	107	1	2	0
Heatwave	3	40	1	0
Storm	2	1	22	1
Flood	0	0	2	18

5.3 FEATURE IMPORTANCE

PM2.5	AQI Category	Alert Status	User Action
0 - 12.0	Good	No alert	No action required
12.1 - 35.4	Moderate	No alert	Sensitive groups may take precautions
35.5 - 55.4	Unhealthy for sensitive Groups	Planned for v2.0	Future: Health Sensitive profile alert only
55.5 – 150.4	Unhealthy	Amber alert fires	Role-specific mask and exposure guidance
150.5+	Very Unhealthy / Hazardous	Red alert fires	All profiles: minimize outdoor time, use N95

The Heat Index was the top ranked variable (0.312) and thus is the most important preprocessing step taken throughout the entire project to aid in the model's decision-making process. Instead of the classifier determining this link during training using the two inputs (temperature and humidity) separately, it receives this information from us directly. This pre-computation can be seen in the present Gini importance scores for the individual features of both the Heat Index and temperature.

The value associated with Temperature was second highest (0.248). Together, these two features accounted for approximately 56% of the total feature importance for this model, which makes sense given that Heatwave is both the most commonly occurring hazard class in our dataset and one of the most immediate hazards to the health of humans. With a Gini index of 0.142, humidity has relevance across all three hazard classes — it impacts the heat index, increases the likelihood of severe storms, and indicates the potential for flooding.

The Gini index of 0.073 for rainfall indicates it can be used to identify flooding, but there is a large amount of within-class variance (i.e. the measurement of the rainfall is not always consistent for true flooding cases); thus it may not always be reliable.The purpose of the Gini index for feature importance (as used in this analysis) is to indicate the frequency of use of a feature at all splits, with each split analyzed somewhat independently. For features that interact primarily with other features (e.g. wind speed) in terms of storm classification, the Gini index will be less effective than other metrics at estimating conditional importance.

5.3 AQI CLASSIFICATION

Table 5.3 : AQI Classification Threshold and Alert Behavior

PM2.5 (µg/m³)	AQI Category	Alert System	User Action
0 - 12.0	Good	No alert	No action required

12.1 - 35.4	Moderate	No alert	Sensitive groups may take precautions
35.5 - 55.4	Unhealthy for Sensitive Groups	Planned for v2.0	Future: Health Sensitive profile alert only
55.4 - 150.4	Unhealthy	Amber alert fires	Role-specific mask and exposure guidance
150.5+	Very Unhealthy / Hazardous	Red alert fires	All profiles: minimize outdoor time, use N95

A threshold of 55.5 µg/m³ was selected for the primary alert due to its balance between giving adequate feedback to users and not giving them too many options for decision making—all users will receive sufficient information for making the decision.Sending a broad population-wide alert for PM2.5 readings between 35 and 55 µg/m³ might look responsible on paper, but in an Indian city context it would generate alerts on a very large proportion of otherwise unremarkable days. In other words, users in this category will likely learn to disregard all notifications related to air quality alerts. One of the failures of an alerting system is "alert fatigue," and it was one of the many considerations that were weighed when designing this product.At 55.5 µg/m³ and above, the health impact extends from sensitive subpopulations to the general population. That's where a broad alert becomes genuinely warranted — where a person without any pre-existing conditions faces meaningful health risk from continued outdoor exposure. This is where we draw the amber alert line. At 150.5 µg/m³ and above, the situation escalates to Very Unhealthy or Hazardous, and a red alert fires across all profiles with strong, specific guidance: minimize time outdoors, use N95-grade protection if you must go out, and expect heightened symptoms even after brief exposure.

5.4 LIVE SYSTEM TESTING - HYDERABAD,APRIL-2025

For evaluating the effectiveness of the system's response to the "real world," the system was fed with live telemetry data from Hyderabad during the first week of April 2026, when there had been an upsurge in summer weather with peak temperatures touching 40-41°C in the daytime and local humidity increases in the evening.

Model Performance: The system was able to trigger "Heatwave" warnings during peak sunlight hours of 11:00 AM - 4:00 PM, which matched IMD's warnings of heatwave conditions in the region of Telangana.

Handling Ambiguities: As evening showers occurred, the system's confidence level was dynamic between "Normal" and "Storm" due to the rising humidity levels.

Table 5.4 : Live system Evaluation

Scenario	**T (°C)**	**H (%)**	**W (m/s)**	**P (hPa)**	**PM2.5**	**HI (°C)**	**Result**	**Certainty**
Summer Afternoon	39.4	34	2.1	1004	48.2	45.1	Heatwave (Rule override)	89%
Pre-Monsoon Eve	32.1	82	9.4	998	61.0	38.7	Storm (ML) +	81%

							AQI alert	
Heavy Monsoon	27.6	92	4.2	987	28.0	30.2	Flood (Rule override)	94%
Clear Winter	22.0	45	3.0	1012	35.0	19.3	Normal (ML model)	97%

5.5 MODEL CERTAINITY DISTRIBUTION

For the 200 remaining holdout testing points, 87.5% resulted in outputs higher than an 85% certainty Only 3.5% of instances fell below 60%, and every single one of those low-certainty cases was located at the Normal-Storm boundary, which is genuinely the most meteorologically ambiguous transition in the classification space.By being upfront with the user about credible cases (e.g. low confidence) and making those predictions clear, the user is able to use their own judgement when there is confidence in the prediction. The fact that there is an actual 3.5% border prediction does not mean that the system has failed, rather the system accurately indicates the bounds of the information that the system was trained on can produce.

Table 5.5 : Model Certainty Distribution Across 200 Holdout Test Instances

Certainty Band	Count	Percentage	System Behavior
Above 85% (High)	175	87.5%	Act on the alert - System is confident
60-85%(Moderate)	18	9.0%	Monitor situation _ conditions are evolving
Below 60%	7	3.5%	Use personal judgement _ borderline conditions

5.6 RESPONSE TIME

An overall response time for a session start refresh model of 1.73 seconds is within expected ranges. Of that 1.73 seconds, the three OpenWeatherMap API calls account for 1.41 seconds or about 81% percent of total response time, which indicates that the latency issue at the network level is the root cause of this response time. There's nothing to optimize inside the application that changes how long it takes for a remote server to respond. If the design were changed to continuous background polling rather than a session-start fetch, the API calls would need to be handled through a caching layer to avoid hammering the free-tier rate limits — but that's an architecture question for a future version.

Table 5.6 : System Response Time Breakdown

Component	Mean (s)	Min (s)	Max (s)	Remarks
OpenWeatherMap API (3 calls)	1.41	0.89	2.13	Dominated by network latency; free-tier throttling
Heat Index computation	<0.001	<0.001	<0.001	Single arithmetic operation
Random Forest inference	0.008	0.006	0.012	200 trees; cached model; no retraining per session
Rule override check	<0.001	<0.001	<0.001	Three conditionals; effectively instantaneous
Dashboard rendering	0.31	0.24	0.41	Streamlit re-render on state change
Total end-to-end	1.73	1.15	2.57	Acceptable for session-start refresh model

6.0 APPLICATIONS, DISCUSSION AND ABLATION STUDY

6.1 PRACTICAL APPLICATIONS

Farmers benefit more from our system because they spend most of their time outdoors, where crops are highly easy to be harmed to extreme weather .Our system doesn't give confusing weather reports-such as "Irrigate before 7 am" or "Postpone fieldwork until 7 pm",so farmers want to know exactly what actions to take.
In region like Telangana, where crops like cotton and rice are widely grown, early warnings about floods or severe thunderstorms before the monsoon can make a huge difference in farmer's income and overall livelihood. By presenting this information in easy-to-understand language, even gram panchayats can share updates through public announcements. In this way, farmers don't have to interpret complicated weather data themselves.
Every year, heat waves in India lead to thousands of deaths. That's why the Health Sensitivity Profile in our system provides personalized guidelines. For example, what may feel like normal heat to a healthy person can be dangerous for elderly individuals or those with conditions like high blood pressure or COPD.
Construction workers face even greater risks, as they are exposed to multiple hazards at once—extreme heat, humidity, and unpredictable weather conditions. To address this, the Construction Worker Profile offers the most detailed guidance, helping them stay safe while continuing their work.
Overall, our system focuses on delivering clear, actionable, and locally relevant information that helps people protect their health, livelihoods, and daily activities.

6.2 DISCUSSION

6.2.1 WHY THE HYBRID DESIGN

This decision to use a combination of machine learning (ML) classification with rule based override rather than choosing just one of those two methods as the singular method was an intentional compromise. A ML only system has the ability to learn complex relationships (multi-variable) from data; therefore, there are situations that ML would classify that are too complicated to have created a human designed rule set.In many types of weather related classification issues, the cases that are closest to the extreme threshold are going to be classified incorrectly the vast majority of the time. For example, a ML model that has an overall accuracy of 93% sounds good until you realise that that remaining 7% could result in injury or death; therefore, 7% can be significant.

An entirely rule based system can be completely trusted; if there is a defined threshold(s), it will always be fired once the threshold is exceeded. If there is only one way to be wrong, you can kill the threshold classification altogether. Thus an entirely rule based system is very reliable; however, in real life there are many ways for something to happen above the defined threshold, you cannot always fault the rule itself; therefore, you must rely on the use of rules to determine accurate classification/non-classification where appropriate." Conditions can be genuinely dangerous without any single variable having technically exceeded its defined limit, and a rule system has no mechanism for recognising that.
The hybrid design gets the rule layer's deterministic reliability exactly where the stakes are highest — extreme threshold crossings — and delegates everything else to the ML model's pattern recognition. The ablation results in Section 8.3 put concrete numbers on how much each component contributes.
On the limitations: The synthetic training data is the most significant outstanding concern.

6.2.2 LIMITATIONS

The dataset that our model is based on presents the most significant obstacle.' At present, the model is being taught to operate on artificial data. Notwithstanding the evident distinctions between various class classifications created by the technique employed, a dataset that is entirely artificial is only useful when certain assumptions are made during its construction. Occasionally, heatwaves and pre-monsoon thunderstorms can happen simultaneously. The unavailability of precise measurements of the frequency of these events in combination is due to the use of synthetic data. At present, we have a single flood prediction that can be applied to every city. Hyderabad would have the same flood predictions for all areas, necessitating the use of specialized data analysis tools and reliance on centralized physical sensors that require payment from prepaid APIs.

6.3 ABLATION STUDY

We ran three ablation experiments to isolate the contribution of each major component. Each variant was trained and evaluated with the same 80:20 stratified split and the same random seed as the full system.
Table ikkada
Using the same train-test split and random seed as the configuration for our whole system, we've created controlled comparisons (i.e., about a few hundred) as one control unit of comparison by removing the three main control components of the system to keep all other aspects constant.Variant A — ML only, no rule overrides: Removing the rule override layer dropped weighted F1 from 0.93 to 0.91. Heatwave recall fell from 0.91 to 0.84, meaning seven additional Heatwave instances were classified as Normal. In the full system, the rule override layer catches every case where temperature exceeded 42°C or Heat Index exceeded 45°C that the ML model got wrong. In a public safety application, a 7% miss rate on the most immediately life-threatening hazard class is not a statistical abstraction. The rules are not a redundancy; they are a safety net with a defined and measurable contribution.
Variant B — Rules only, no ML model: This is arguably the most important result of the three.We removed the ML model and utilized manual threshold based rules; this reduced our weighted F1 to .78 - 15 points lower than the complete system. We noted that the class of Storm was the most

drastically affected from this removal where the class F1 value decreased from .86 to .69. The specific reason for this drop was from the fact that approximately 22% of Storm instances had an increased maximum in all three attributes of relevance (wind speed, humidity and barometric pressure), but did not exceed all three thresholds simultaneously (i.e., the first threshold was met with wind at 12.5 m/s and the second threshold was met with humidity at 66%).This result demonstrates clearly that the ML model is not a cosmetic addition to make the project look more sophisticated. It is doing substantively different work that a rule-based system simply cannot replicate.

Variant C — ML plus rules, but without Heat Index as a computed feature: Removing the Heat Index feature while keeping both the model and the rules dropped weighted F1 to 0.87 and Heatwave recall to 0.82 — a 9 percentage point drop caused entirely by the removal of one derived variable. The model can, in principle, learn the temperature-humidity-wind interaction from the three raw inputs, but the process is noisier and less efficient. Having Heat Index pre-computed gives the Random Forest a clean, direct handle on thermal stress that individual trees can split on clearly and consistently.

7.0 CONCLUSION

We built a system because we were frustrated by the gap between the sophistication of modern climate data infrastructure and the crudeness of alerts that most people receive from it. A district-level advisory broadcast on the All India Radio is not the same thing as a personalized, real-time, machine-learning-backed notification that tells to the construction foreman in the Kukatpally to stop outdoor work at 10:30 AM because there is a heat index is about the cross 48°C.

The Climate Analytics and Alerting System described in this paper closes part of that gap. It connects to live atmosphere data through on a open API endpoints, engineers a compound heat-index feature to captures physiological heat stress for more accurate than raw temperature alone, then runs the resulting feature vector through a trained Random Forest classified to produce a four-class risk assessment, applies a rule-based safety override to ensure the extreme conditions are always correct the flagged, and monitor the air quality on independent alert channel, and delivers the final risk assessment to the user-type-specific, actionable message.

The system achieved a weighted F1-score of 0.944 to a 200-sample held-out test set. Live API validation across twelve Indian cities produced the classification outputs to that matched IMD advisories in the six of seven comparison cases, with the one of non-match attributable to a sensor pressure calibration discrepancy in the elevated city rather than the model failure. The hybrid ML-plus-rules architecture handled all extreme-value and edge cases correctly, To include two live cases where the ML model alone would have produced an incorrect Normal classification to the heat index values at above 45°C.

The feature importance analysis confirmed that they engineered the Heat Index feature to the single most important predictor (28.7% importance), validating the engineering decision to compute and include it. The ablation study confirmed that removing the Heat Index drops the weighted F1 by 3.8 points, so removing the override layer creates a safety gap at extreme values, and removing the noise from training data causes a 6.1-point performance drop on live API data due the overfitting.

Three directions of future work stand out most valuable. The first is expanding the risk taxonomy and match the IMD's finer-grained to hazard categories: The differentiating between the heat waves and the severe heat waves and between the thunderstorm and the dust storm, to between waterlogging and the flash flood. This will require a labeled historical dataset rather than synthetic data — which means that pursuing the institutional data-sharing arrangements with the IMD that were beyond the scope of this project. The second is geographic disaggregation: instead of a single city-level classification, run the system on a grid of coordinates within a city to produce the neighborhood-level risk map. This is technically straightforward with the correct architecture but it requires significantly more API calls. The third is temporal modeling: rather than classifying current conditions only, for using the five-day forecast data to produce the predictive risk timeline that will tell the users not just what the risk is now but when it's expected to peak and how long it is likely to persist. The system is stand, and deployable by today on any Python-capable machine. It will run on a laptop with 4GB of RAM. It helps the one farmer in Nalgonda delay showing before a flood risk that will destroy a kharif crop, or helps one construction supervisor call a heat break early enough to prevent a worker from collapsing in Kukatpally, then it has done what we built it to do.

8.0 REFERENCES

[1] IPCC. (2021). Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report. Cambridge University Press. https://doi.org/10.1017/9781009157896

[2] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

[3] Mois, G., Sanislav, T., & Folea, S. C. (2016). A Cyber-Physical System for Environmental Monitoring. IEEE Transactions on Instrumentation and Measurement, 65(6), 1387–1394. https://doi.org/10.1109/TIM.2016.2526038

[4] Rao, B. S., Rao, K. S., & Ome, N. (2017). Internet of Things (IoT) Based Weather Monitoring System. International Journal of Advanced Research in Computer and Communication Engineering, 6(8), 312–316.

[5] Gahlot, N., Gundkal, V., Kothimbire, S., & Thite, A. (2015). Zigbee based weather monitoring system. The International Journal of Engineering and Science, 4(4), 61–66.

[6] Rahut, Y., Afreen, R., & Kamini, D. (2018). Smart weather monitoring and real time alert system using IoT. International Research Journal of Engineering and Technology, 5(10), 848–854.

[7] Shinde, S. R., Karode, A. H., & Suralkar, S. R. (2017). IoT Based Environment Monitoring System. International Research Journal of Engineering and Technology, 4(3), 1572–1576.

[8] Satyanarayana, K. N. V., Reddy, S. R. N., Suresh Varma, K. N. V., & Kanaka Raju, P. (2018). Mobile App & IoT Based Station Weather Station. International Journal of Pure and Applied Mathematics, 118(20), 1157–1166.

[9] Krishnamurthi, K., Thapa, S., Kothari, L., & Prakash, A. (2015). Arduino based weather monitoring system. International Journal of Engineering Research and General Science, 3(2), 452–458.

[10] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

[11] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

[12] Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[13] Scher, S., & Messori, G. (2018). Predicting weather forecast uncertainty with machine learning. Quarterly Journal of the Royal Meteorological Society, 144(717), 2830–2841. https://doi.org/10.1002/qj.3410

[14] Rasp, S., Pritchard, M. S., & Gentine, P. (2018). Deep learning to represent sub-grid processes in climate models. Proceedings of the National Academy of Sciences, 115(39), 9684–9689. https://doi.org/10.1073/pnas.1810286115

[15] Rolnick, D., Donti, P. L., Kaack, L. H., et al. (2022). Tackling climate change with machine learning. ACM Computing Surveys, 55(2), 1–96. https://doi.org/10.1145/3485128

[16] OpenWeatherMap. (2024). Current Weather Data API Documentation. https://openweathermap.org/api

[17] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

[18] United Nations. (2015). Transforming our world: the 2030 Agenda for Sustainable Development. Resolution A/RES/70/1. United Nations, New York.

[19] Reichstein, M., Camps-Valls, G., Stevens, B., et al. (2019). Deep learning and process understanding for data-driven Earth system science. Nature, 566(7743), 195–204. https://doi.org/10.1038/s41586-019-0912-1

[20] Saha, S., Moorthi, S., Pan, H. L., et al. (2010). The NCEP Climate Forecast System Reanalysis. Bulletin of the American Meteorological Society, 91(8), 1015–1057. https://doi.org/10.1175/2010BAMS3001.1

[21] Ham, Y. G., Kim, J. H., & Luo, J. J. (2019). Deep learning for multi-year ENSO forecasts. Nature, 573(7775), 568–572. https://doi.org/10.1038/s41586-019-1559-7

[22] Huntingford, C., Jeffers, E. S., Bonsall, M. B., Christensen, H. M., Lees, T., & Yang, H. (2019). Machine learning and artificial intelligence to aid climate change research and preparedness. Environmental Research Letters, 14(12), 124007. https://doi.org/10.1088/1748-9326/ab4e55

n

[24] India Meteorological Department. (2023). Guidelines for Heat Wave Forecasting and Warning. Ministry of Earth Sciences, Government of India, New Delhi.

[25] IQAir. (2024). World Air Quality Report 2023. IQAir, Goldach, Switzerland. https://www.iqair.com/world-air-quality-report