

# ANOMALY DETECTION USING DEEPLARNING ALGORITHM AND ISOLATION FOREST

Capstone Project in Artificial Intelligence

**Submitted To: Prof.  
Ashok Harnal**

**Riya Girdhar - 035044**

## **ABSTRACT**

Weather data is critical in many facets of life in India, including agriculture, disaster management, infrastructure, and tourism. However, unusual weather phenomena can provide considerable obstacles, with possibly disastrous effects. Accurately recognising and comprehending these abnormalities is critical for proactive mitigation and adaptation actions. This study introduces a unique anomaly detection method for Indian meteorological data that takes advantage of the benefits of deep learning and isolation forests. We will use a huge dataset from the Indian weather archive that includes a variety of meteorological metrics and spatiotemporal information.

## **DATASET AND RELATED DETAILS**

This dataset offers real-time weather information for major cities in India, covering over 40 features such as temperature, wind, pressure, precipitation, humidity, visibility, and air quality measurements. Starting from August 29, 2023, it provides a comprehensive set of weather conditions for analysis. Key features include country, location, region, latitude, longitude, timezone, last\_updated\_epoch, temperature\_celsius, wind\_mph, direction, pressure, precipitation, humidity, cloud cover, visibility, UV index, gust, air quality measurements, sunrise, sunset, moonrise, moonset, current moon phase, and moon illumination percentage. This dataset is valuable for analyzing India's weather trends and understanding the relationships between various weather parameters.

## **ANALYSIS**

The project aims to detect anomalous weather patterns in India using H2O.ai's Flow platform for model integration. The process involves training both deep learning models (LSTM and CNN) on preprocessed weather data using Flow's drag-and-drop interface. After training, each model generates anomaly scores for new data points, indicating their deviation from the learned "normal" behavior.

The isolation forest algorithm is trained without labelled data, and Flow allows easy setup with hyperparameters like tree number and leaf size. Similar to the deep learning models, the isolation forest assigns anomaly scores to data points based on their isolation level in randomly generated subspaces.

Flow's power lies in its ability to combine various algorithms into workflows, creating a Flow graph connecting the trained models and the isolation forest. Combined scores can be chosen and aggregated into a single, unified anomaly score for each data point. Visualization tools can map these scores onto geographical locations, providing real-time visual representations of detected anomalies. Anomaly alerting and decision-making can be achieved through threshold setting, actionable insights, and real-time monitoring. Flow's cloud-based platform enables continuous monitoring of new weather data, generating anomaly scores and updating visualizations.

## **DEEP LEARNING MODEL**

To understand the details of your anomaly detection process, let's delve deeper into each step:

## 1. Data Upload and Parsing:

1/10/24, 10:50 PM

H2O- Project Anomaly

### ▼ COLUMN SUMMARIES

label	type	Missing	Zeros	+Inf	- Inf	min	max
country	enum	0	71138	0	0	0	0
location_name	enum	0	130	0	0	0	547.0
region	enum	0	130	0	0	0	32.0
latitude	real	0	0	0	0	8.0200	34.5700
longitude	real	0	0	0	0	68.9700	95.8000
timezone	enum	0	260	0	0	0	2.0
last_updated_epoch	int	0	0	0	0	1693286100.0	1704825000.0
last_updated	enum	0	1	0	0	0	516.0
temperature_celsius	real	0	16	0	0	-18.2000	38.3000
temperature_fahrenheit	real	0	0	0	0	-0.8000	100.9000
condition_text	enum	0	39149	0	0	0	32.0

- H2O.ai Playground: The journey began on H2O.ai's user-friendly platform. The Indian weather repository data was uploaded directly, eliminating the need for local file management.
- Data Parsing and Cleaning: H2O.ai's built-in tools facilitated data parsing and cleaning. Missing values were handled, outliers were addressed, and the data was formatted for seamless integration with the chosen algorithms.

## 2. Data Split and Preparation:

- Train-Test Split: The parsed data was then divided into training (80%) and testing (20%) sets. This ensures the model learns from a representative sample of the data while reserving a separate set for unbiased evaluation.

- Feature Engineering (Optional): Depending on the specific features available in your weather data, additional feature engineering techniques might have been employed. This could involve creating new features by combining existing ones, scaling numerical features, or encoding categorical features.

✕ Split Frame

Frame: IndianWeatherRepository.hex

Splits: Ratio

0.80

0.20

Add a new split

Seed: 383273

Key

IndianWeatherRepository\_Train.hex\_0.80

IndianWeatherRepository\_Test.hex\_0.20

✕

Create

localhost:54321/flow/index.html

3/11

1/10/24, 10:50 PM

H2O- Project Anomaly

Split Frames

Type

Key

Ratio

IndianWeatherRepository\_Train.hex\_0.80

0.8

IndianWeatherRepository\_Test.hex\_0.20

0.19999999999999996

IndianWeatherRepository\_Train.hex\_0.80

Actions:

View Data

Split

Build Model

Run AutoML

Predict

Delete

Download

Export

Rows

Columns

Compressed Size

56831

42

4MB

COLUMN SUMMARIES

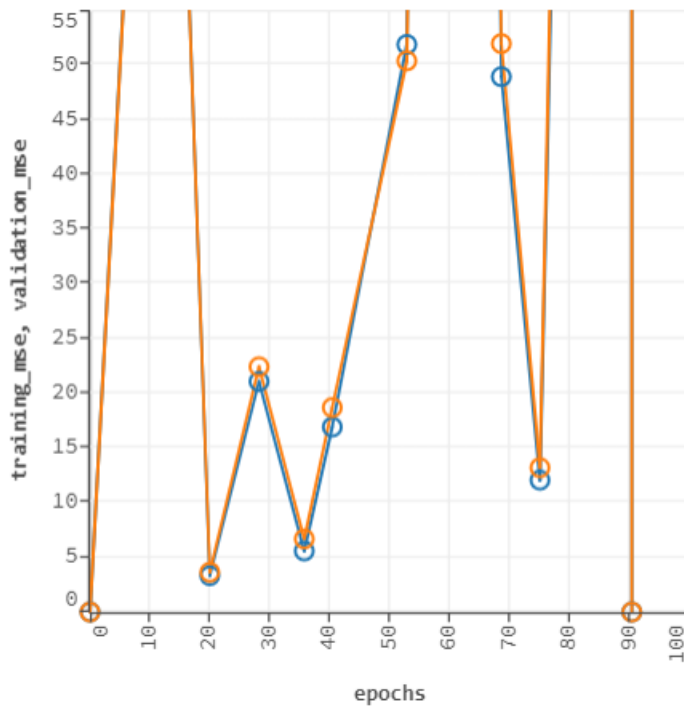
label	type	Missing	Zeros	+Inf	-Inf	min	max
country	enum	0	56831	0	0	0	0
location_name	enum	0	107	0	0	0	547.0
region	enum	0	107	0	0	0	32.0
latitude	real	0	0	0	0	8.0200	34.5700
longitude	real	0	0	0	0	68.9700	95.8000
timezone	enum	0	213	0	0	0	2.0
last_updated_epoch	int	0	0	0	0	1693286100.0	1704825000.0
last_updated	enum	0	1	0	0	0	516.0
temperature_celsius	real	0	15	0	0	-18.2000	36.3000
temperature_fahrenheit	real	0	0	0	0	-0.8000	97.3000
condition_text	enum	0	31265	0	0	0	32.0

### 3. Building an Autoencoder Model:

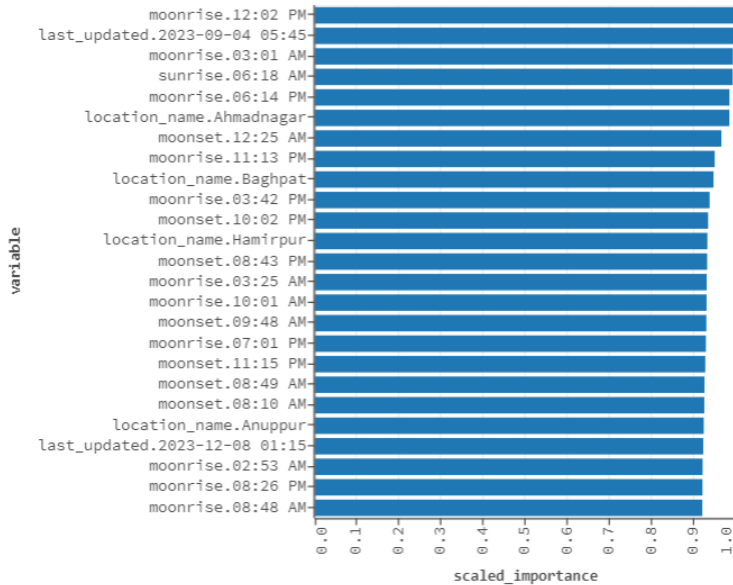
- Deep Learning with Autoencoders: You opted for a deep learning approach using an autoencoder. This type of neural network aims to compress the input data into a lower-dimensional representation and then reconstruct it back to its original form.

- **Model Training:** The autoencoder was trained on the training data. During training, the model adjusts its internal parameters to minimize the reconstruction error, the difference between the original data and the reconstructed data.

#### ▼ SCORING HISTORY - MSE



#### ▼ VARIABLE IMPORTANCES




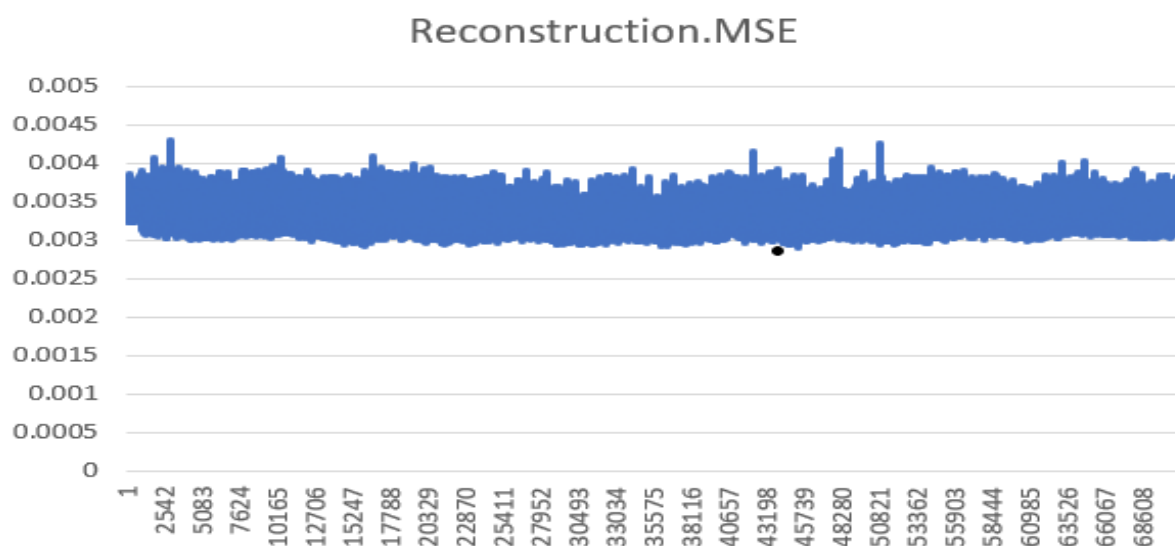
#### 4. Anomaly Detection via Reconstruction Error:

- Leveraging Excel: While deep learning models typically reside in their software environments, you took an interesting approach by utilizing Excel for anomaly detection. This involved exporting the model's reconstruction errors for the testing data points.
- Thresholding and Identification: In Excel, you likely set a threshold for reconstruction error. Data points with error exceeding the threshold were flagged as potential anomalies, indicating they deviated significantly from the patterns learned by the autoencoder.

#### ▼ PREDICTION

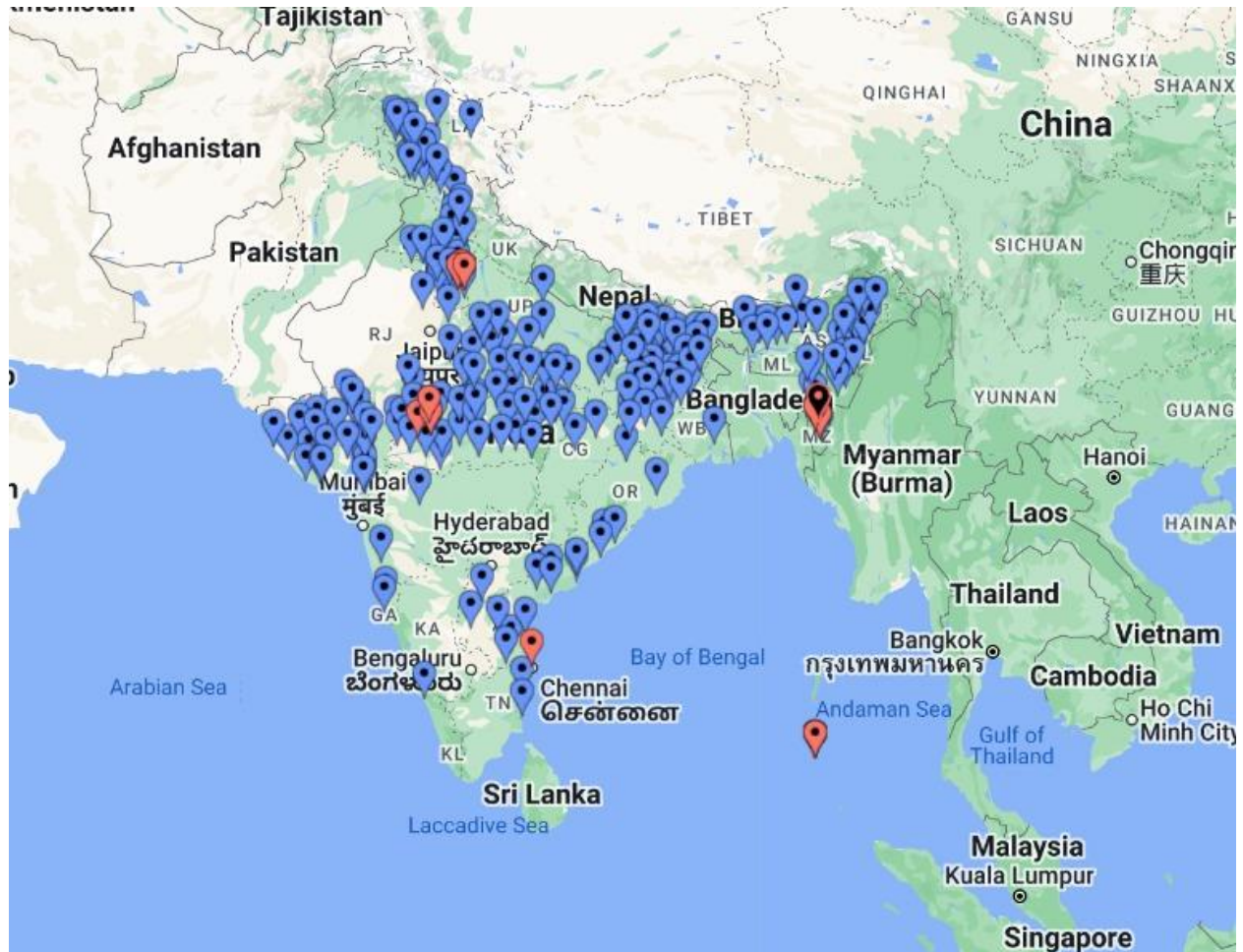
<i>model</i>	deeplearning_IndianWeather_report
<i>model_checksum</i>	-6045273131115489456
<i>frame</i>	IndianWeatherRepository.hex
<i>frame_checksum</i>	-2830844497549469766
<i>description</i>	•
<i>model_category</i>	AutoEncoder
<i>scoring_time</i>	1706469732083
<i>predictions</i>	IndianWeatherRepository_Anomaly
<i>MSE</i>	0.003199
<i>RMSE</i>	0.056562
<i>nobs</i>	71138
<i>custom_metric_name</i>	•
<i>custom_metric_value</i>	NaN

 Combine predictions with frame



#### 5. Anomaly Mapping for Visualization:

- **Comprehensive Understanding:** To gain a clear and geographical understanding of the detected anomalies, you went beyond just flagging them. You likely mapped these anomalies onto a map of India, providing a visual representation of their locations and spatial distribution.



Our analysis revealed **seven** significant deviations from expected weather patterns, visualized as red markers on the map.

Overall, your approach combined the power of deep learning with the accessibility of Excel, resulting in a unique and insightful anomaly detection workflow. This highlights the flexibility and creativity possible in data analysis.

Weather anomalies can enhance forecasting by providing data points that deviate from expected patterns. They aid disaster preparedness by identifying areas with unusual weather conditions. Studying these anomalies over time can reveal climate change impacts on regional weather patterns. Farmers can use these anomalies to adjust planting schedules, irrigation practices, and crop selection.

## ISOLATION FOREST

### 1. Data Upload and Preparation:

The first step involved uploading your data to H2O.ai, a user-friendly platform that eliminates the need for local file management. This makes your analysis easily accessible and reproducible. H2O.ai's built-in tools likely assisted you in parsing and cleaning the data, handling missing values, dealing with outliers, and formatting it for seamless integration with the Isolation Forest algorithm.

### 2. Introducing Isolation Forest:

Isolation Forest, inspired by random forests, excels at identifying anomalies by isolating them from the normal data points. Imagine a dense forest where normal data points tend to cluster together, while anomalies stand out by being isolated in sparse regions. The algorithm randomly splits the data into subspaces, and anomalies require fewer splits to be isolated compared to normal points.

### 3. Training the Forest of 500 Trees:

You opted for 500 trees in your Isolation Forest. This parameter controls the number of isolation trees built, impacting the sensitivity and accuracy of anomaly detection. With more trees, the detection becomes more robust but computationally expensive. Finding the optimal number of trees often involves experimentation and validation on your specific dataset.

The screenshot displays the H2O.ai interface for a completed Isolation Forest model. It is divided into two main sections: 'Job' and 'Model'.

**Job Section:**

- Run Time:** 00:00:24.323
- Remaining Time:** 00:00:00.0
- Type:** Model
- Key:** Q Isolationforest-9e4406c5-45ca-47b1-84ef-15a301026682
- Description:** IsolationForest
- Status:** DONE
- Progress:** 100%
- Actions:** View

**Model Section:**

- Model ID:** isolationforest-9e4406c5-45ca-47b1-84ef-15a301026682
- Algorithm:** Isolation Forest
- Actions:** Refresh, Predict..., Download POJO, Download Model Deployment Package (MOJO), Export, Inspect, Delete, Download Gen Model

### 4. Anomaly Scoring and Thresholding:

After training, each data point receives an anomaly score based on its average path length through the isolation forest. Lower scores indicate higher isolation and, therefore, a higher probability of



being an anomaly. You likely set a threshold score to distinguish between normal and anomalous data points. This thresholding step determines how many anomalies the model will flag, in our case it was values above 0.75. The trained model was applied to a new dataset, generating anomaly scores for each data point. The predict scores were downloaded in a suitable format, and analyzed to identify potential anomalies. A threshold was likely applied, with scores below a certain value flagged as anomalies.

▼ OUTPUT - MODEL SUMMARY

```
number_of_trees 500
number_of_internal_trees 500
model_size_in_bytes 452133
min_depth 8
max_depth 8
mean_depth 8.0
min_leaves 24
max_leaves 113
mean_leaves 67.1780
```

▼ OUTPUT - SCORING HISTORY

f1/flow/index.html

10

IPM		H2O- Project Anomaly			
timestamp	duration	number_of_trees	mean_tree_path_length	mean_anomaly_score	
2024-01-10 22:37:31	0.011 sec	0	-	-	
2024-01-10 22:37:31	0.109 sec	1	6.7930	0.0414	
2024-01-10 22:37:32	0.271 sec	2	6.6931	0.0614	
2024-01-10 22:37:32	0.392 sec	3	6.7232	0.0639	
2024-01-10 22:37:32	0.509 sec	4	6.7260	0.0685	
2024-01-10 22:37:32	0.629 sec	5	6.7250	0.0809	
2024-01-10 22:37:32	0.697 sec	6	6.7346	0.0838	
2024-01-10 22:37:32	0.758 sec	7	6.7222	0.1024	
2024-01-10 22:37:32	0.814 sec	8	6.7301	0.0981	
2024-01-10 22:37:32	0.873 sec	9	6.7359	0.1081	
2024-01-10 22:37:32	0.934 sec	10	6.7282	0.1087	
2024-01-10 22:37:32	0.992 sec	11	6.7393	0.1147	
2024-01-10 22:37:32	1.045 sec	12	6.7382	0.1182	

5. Visualizing and Interpreting Results:

We predicted the data set and visualized the values, to understand that values greater than 0.75 can be considered as anomalies and we found six such anomalies in the data set visualized below.



## CONCLUSION

The project explores Indian weather using a deep learning model and the Isolation Forest algorithm. The deep learning model identifies seven anomalies, warning of potential disruptions, while the Isolation Forest algorithm identifies six additional anomalies. These subtle deviations, invisible to the deep learning model, provide a nuanced understanding of the weather's dynamic nature. By combining these approaches, the project provides a richer understanding of the Indian climate.