# LandSafe

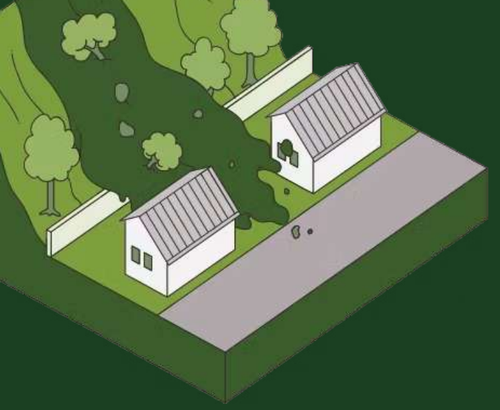
## A high-performance, generalizable landslide forecasting system

Arnav Mehta, Arsalaan Alam & Tanvi Sanghai

# Contents



# 01 The Problem



# Problem Statement

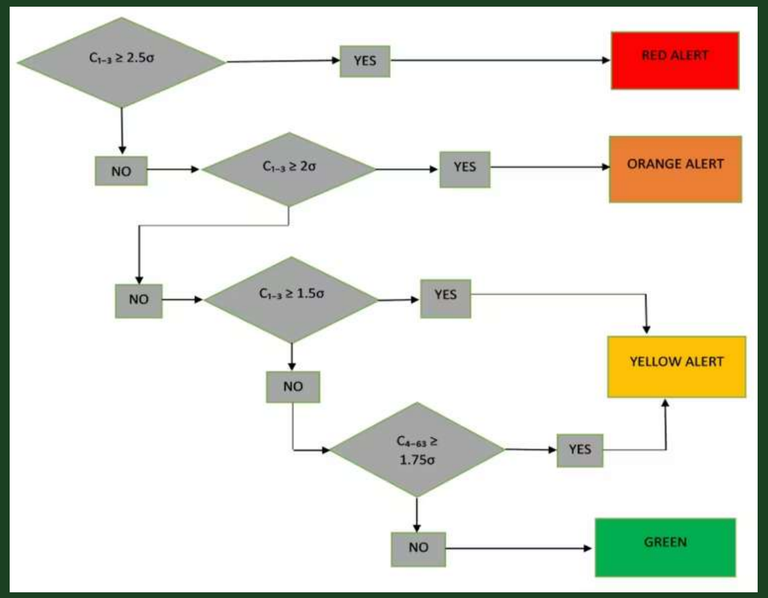
**Predicting Landslides Using Environmental and Infrastructural Data for disaster prone environments.**

## Why Are Landslides a Problem?

- **Highly Destructive:** Cause thousands of deaths and billions in damages yearly.
- **Widespread Impact:** Affect ~4.8 million people annually (UNDRR).
- **Major Consequences:** Infrastructure destruction, displacement, economic loss.

# 02 Literature Review

# Paper 1



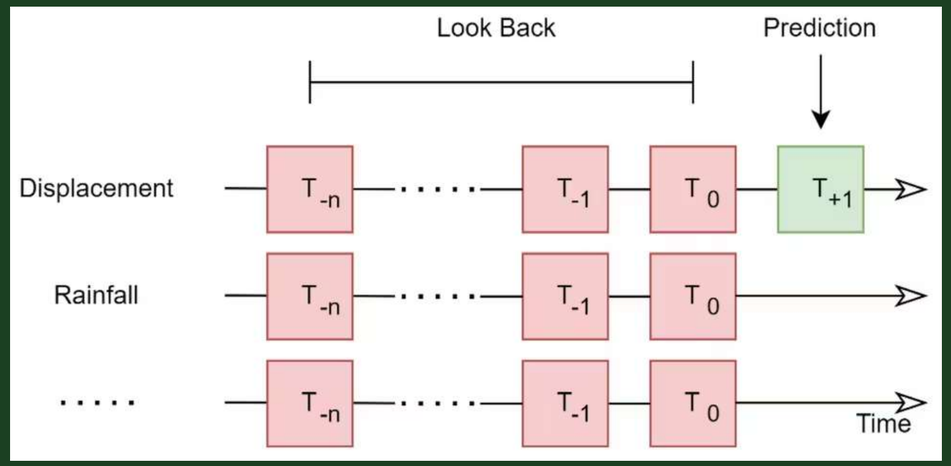

*Rainfall Threshold Estimation and Landslide Forecasting for Kalimpong, India Using SIGMA Model*

**Methodology:**
The paper primarily relies on rainfall data as the sole input for the SIGMA model, however ignores other critical factors such as soil moisture, geological conditions, slope stability, and land use patterns, which are known to influence landslide by other papers.

**Limitations:**
Limited to specific geographical areas and cannot be generalised.

*Abraham, M. T., Satyam, N., Kushal, S., Rosi, A., Pradhan, B., & Segoni, S. (2020). Rainfall Threshold Estimation and Landslide Forecasting for Kalimpong, India Using SIGMA Model. Water, 12(4), 1195. https://doi.org/10.3390/w12041195*

# Paper 2



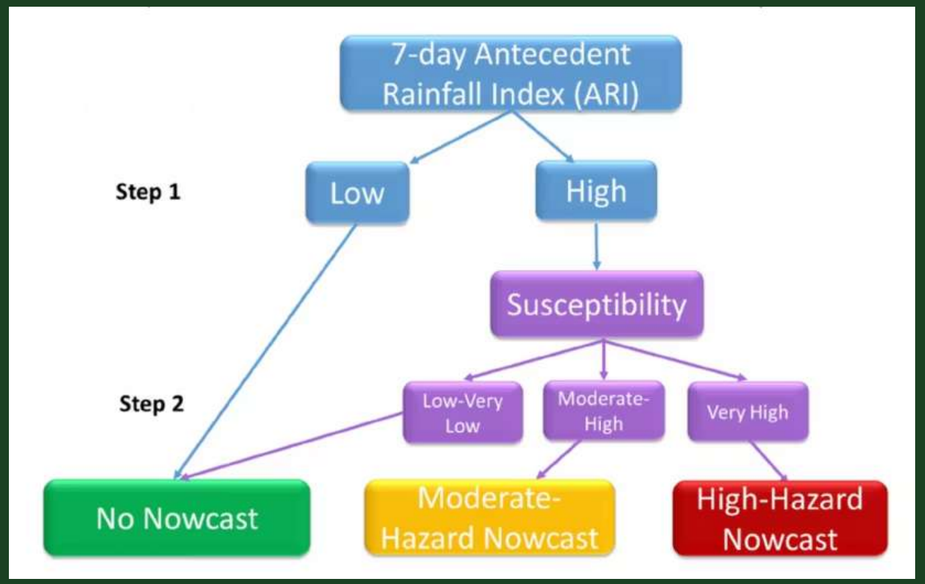

*Landslide displacement forecasting using deep learning and monitoring data across selected sites (in Italy)*

**Methodology**:
Evaluates 7 Deep Learning models (MLP, LSTM, GRU, 1D CNN, Bi-LSTM, Conv-LSTM, Stacked LSTM) are used for time series-based landslide displacement prediction. These models capture improved prediction accuracy, and detect seasonal patterns. Selects look-back windows (3, 5, 7, 9, 12 time steps) to optimize past data consideration.

**Limitations**:
No specific temporal prediction - only assesses landslide susceptibility. Limited Generalization of Conv-LSTM, which only worked well in seasonal landslides.

*Nava, L., Carraro, E., Reyes-Carmona, C. et al. Landslide displacement forecasting using deep learning and monitoring data across selected sites. Landslides 20, 2111-2129 (2023). https://doi.org/10.1007/s10346-023-02104-9*

# Paper 3




*Satellite-based assessment of rainfall-triggered landslide hazard for situational awareness (LHASA - by NASA)*

**Methodology:**
It uses a decision tree structure to issue nowcast warnings of 3 types of severity based on a 7-day Antecedent Rainfall Index (water accumulation) and a static susceptibility map.

**Limitations**:
The decision tree-based approach is very basic and as a result the LHASA system has True Positive Rates in the range of 8-60% and a 4-5 hour latency.

*Kirschbaum, Dalia & Stanley, Thomas. (2018). Satellite-Based Assessment of Rainfall-Triggered Landslide Hazard for Situational Awareness. Earth's Future. 6. 10.1002/2017EF000715.*

# Addressing the Gaps:

- **Narrow Feature Selection:**
  Most models only use rainfall and slope, ignoring key factors like forest loss, infrastructure, and humidity.

- **Poor Generalizability:**
  Region-specific training limits transferability; diverse datasets are needed for global performance.

- **Outdated Models:**
  Reliance on basic classifiers like logistic regression reduces accuracy. Modern hybrid models perform better.

- **Static & Stale Data:**
  Static maps don't reflect real-time changes like deforestation or construction.

# Presenting LandSafe:

- **Multi-feature ML Model** : Integrates climate, forest loss, elevation, & infrastructure data.
- **Generalizable** : Can determine occurence of landslide in next 5 days for (lat, long)
- **Early Warnings** : Enables proactive disaster mitigation and casualty reduction.

## Applications:

- **Predicts landslides** using **real-time climate** and **terrain data** to enable **early alerts** and reduce damage.
- Supports **safer infrastructure planning** by analyzing **terrain stability** and identifying **high-risk zones**.
- Helps **insurers assess risk** and **optimize policy pricing** using **environmental indicators** like **forest loss** and **weather patterns**.

# 03 Data Preprocessing

# Data Extraction - Climate & Forest Loss

**Landslide & Climate Data Collection:**

- Global Landslide Catalog (NASA): 11,093 landslides (lat, long, time, country).
- Synthetic Data: 10,000 non-landslide events, sampled locations/dates, fetched weather data.

**Forest Loss Data (Global Forest Change Dataset):**

- 5-mile radius check for forest loss before event year.
- 10° x 10° TIF tiles (40,000 x 40,000 pixels).
- Custom handler for TIF data extraction and analysis.

**Climate Features (Open-Meteo API):**

- 15 queries per event (15 days before to event day).
- 75 features: Precipitation, humidity, wind speed, air pressure, temperature.

# Data Extraction - Infrastructure, Elevation & Lithology

**Elevation Data (NASA SRTM)**

- Dataset used: NASA's Shuttle Radar Topography Mission (SRTM). Found on kaggle
- Processing: Downloaded .hgt files, ran gdaldem tool.
- Feature extracted: 90th percentile slope value within landslide bounding box.

**Infrastructure Data (OSM API) - OpenStreetMap API**

- Extracted counts of roads, highways, buildings, streets, and routes.
- Total count recorded as the OSM feature.

# Feature Preprocessing - Data Cleaning & Transformation

## 1 Data Cleaning & Balancing:

- Rows with missing or invalid values were removed.
- Landslide severity labels were filtered to only include "small", "medium", and "large" events.

## 2 Feature Engineering - Terrain & Infrastructure:

- Forest Loss was extracted from GFC TIF files: Binary feature: 1 if forest loss occurred before the event year, otherwise 0.
- Slope was extracted and stored as a continuous feature.
- OSM features counted the number of infrastructure tags at each location.
- Lithology was assigned as a categorical feature based on the rock type at the landslide location.

## 3 Feature Engineering - ARI:

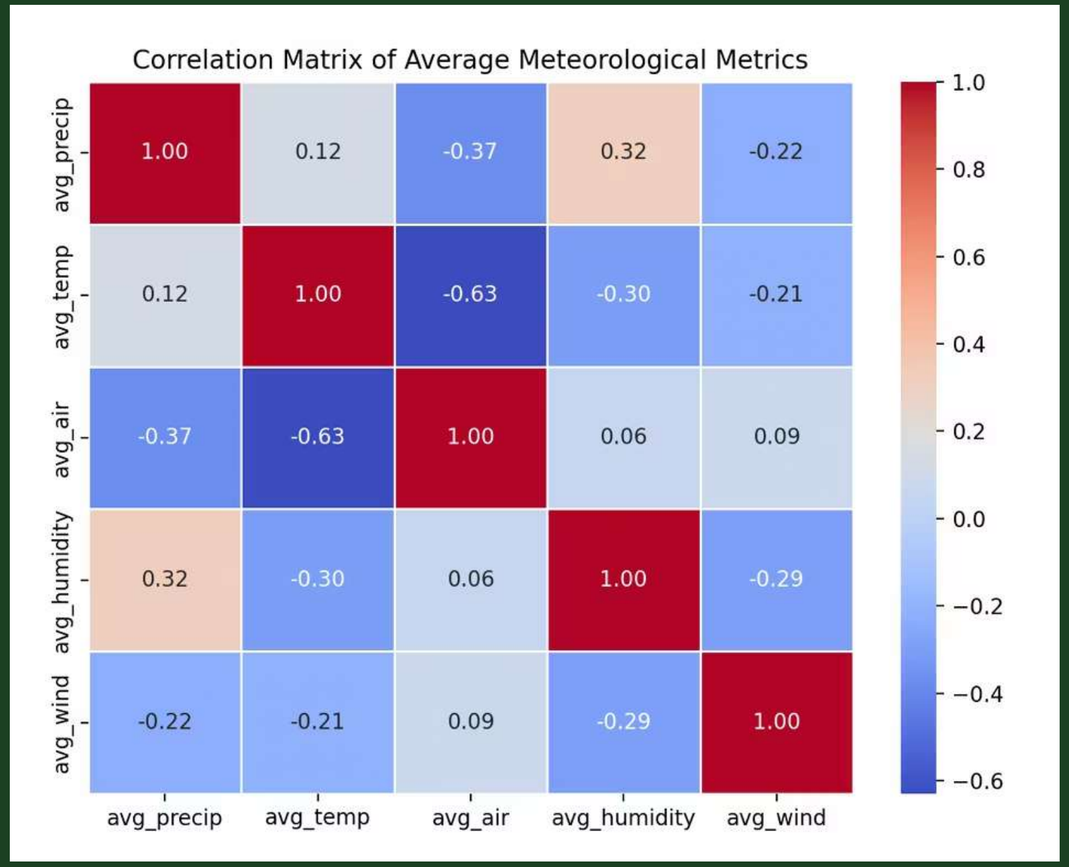
- Climate data was stored as 80 separate features, from 15 days before the event to the day of the event.
- Antecedent Rainfall Index (ARI) was computed using weighted moving averages of precipitation from 12 to 5 days before the event.

$$ARI = \frac{\sum_{t=0}^{6} p_t w_t}{\sum_{t=0}^{6} w_t}$$

## 4 Feature Scaling & Encoding

- Numerical Features (precipitation, temperature, humidity, wind speed, air pressure, slope, forest loss) were standardized using Z-score normalization.
- Categorical Features (lithology) were encoded using ordinal encoding.

# Correlation Matrix for Weather Metrics



# Some Snippets

## get_weather_data()

```python
def format_date(date_str):
    return datetime.datetime.strptime(date_str, "%m/%d/%y").strftime("%Y-%m-%d")

def get_weather_data(lat, lon, date):
    base_url = "https://archive-api.open-meteo.com/v1/archive"
    start_date = (datetime.datetime.strptime(date, "%Y-%m-%d") - datetime.timedelta(days=15)).strf
    end_date = (datetime.datetime.strptime(date, "%Y-%m-%d") - datetime.timedelta(days=7)).strftim

    params = {
        "latitude": lat,
        "longitude": lon,
        "start_date": start_date,
        "end_date": end_date,
        "daily": ["precipitation_sum", "temperature_2m_max", "temperature_2m_min", "surface_pressu
        "timezone": "auto"
    }

    response = requests.get(base_url, params=params)

    if response.status_code == 200:
        weather_data = response.json().get("daily", {})
        print(f"Weather data received for {lat}, {lon}")
        return {
            "precipitation": weather_data.get("precipitation_sum", [])[:9],  # Last 15 to 7 days
            "temperature": [(max_ + min_) / 2 for max_, min_ in zip(weather_data.get("temperature_
            "air_pressure": weather_data.get("surface_pressure_mean", [])[:9],
            "humidity": [(max_ + min_) / 2 for max_, min_ in zip(weather_data.get("relative_humidi
            "wind_speed": weather_data.get("wind_speed_10m_max", [])[:9]
        }
    else:
        print(f"Error fetching weather data for {lat}, {lon}: {response.text}")
        return None
```



## get_forest_loss()

```python
BASE_URL = "https://storage.googleapis.com/earthenginepartners-hansen/GFC-2023

def get_forest_loss(lat, lon, year):
    tile_lat = int(lat // 10) * 10
    tile_lon = int(lon // 10) * 10

    lat_suffix = "N" if tile_lat >= 0 else "S"
    lon_suffix = "E" if tile_lon >= 0 else "W"

    tile_lat = abs(tile_lat)
    tile_lon = abs(tile_lon)

    treecover_filename = f"Hansen_GFC-2023-v1.11_treecover2000_{tile_lat:02d}{
    treecover_path = os.path.join(TIF_DIR, treecover_filename)
    treecover_url = BASE_URL + treecover_filename

    if not os.path.exists(treecover_path):
        response = requests.get(treecover_url, stream=True)
        if response.status_code == 200:
            with open(treecover_path, "wb") as f:
                for chunk in response.iter_content(1024):
                    f.write(chunk)
        else:
            return None

    lossyear_filename = f"Hansen_GFC-2023-v1.11_lossyear_{tile_lat:02d}{lat_su
    lossyear_path = os.path.join(TIF_DIR, lossyear_filename)
    lossyear_url = BASE_URL + lossyear_filename
```

# Peek at the Compiled Dataset

(notice the features, 1st row)

GLIF dataset.csv > data

```
row,id,date,lat,lon,country,fatalities,injuries,type,trigger,severity,location,precip15,temp15,air15,humidity15,wind15,precip14,temp14,air14,humidity14,wind14,precip13,temp13,air13,humidity13,wind13
0,1069,9/4/16,46.726906,13.787332,Austria,0,0,landslide,dam_embankment_collapse,large,...,2.8,22,1017,98,9,49.6,13,1024,98,9,3.7,16,1029,96,12,0,20,1029,95,10,0,22,1027,95,7,0,23,1024,97,6,0,23,1022
1,1855,3/23/17,49.726406,-116.911834,Canada,0,0,mudslide,rain,small,above_road,3.3,-2,1024,100,9,6.8,-2,1026,100,6,4.4,3,1023,100,7,8.4,3,1023,100,6,2.9,3,1024,100,7,5,4,1024,100,7,13.4,4,1020,100,7
2,797,10/20/09,18.5347,-72.4097,Haiti,4,0,landslide,downpour,small,unknown,8.6,33,1015,97,16,16.7,31,1015,98,22,12.5,33,1017,97,17,16.4,32,1017,97,19,5.9,33,1016,97,13,16.6,31,1015,97,7,9.5,30,1015,
3,12967,12/31/09,4.421429046,-75.22070904,Colombia,0,0,rotational_slide,...,...,urban,0,29,1014,96,8,0,28,1014,88,9,0.1,26,1013,82,7,3.4,23,1016,96,5,6.9,24,1017,100,6,13.2,24,1017,100,5,3,27,1016,8
4,13089,5/1/14,39.2902,-76.6651,United_States,0,0,mudslide,rain,small,below_road,0,8,1035,85,26,0,10,1037,94,17,0,11,1035,95,10,0,17,1027,91,17,0,15,1030,87,19,0,21,1025,86,10,4.7,21,1011,92,20,0,13
5,9036,7/12/13,36.2629,-115.6158,United_States,0,0,debris_flow,rain,medium,unknown,0,36,1015,26,14,0,40,1013,28,15,0.2,39,1010,21,24,1.3,40,1009,23,21,1.9,38,1009,26,21,1.2,40,1012,29,21,2.5,39,1010
6,7990,5/10/09,4.44059512,-75.24390515,Colombia,0,0,unknown,...,...,urban,0.8,26,1015,79,12,0,26,1016,81,12,0,29,1016,81,11,1.9,30,1015,82,12,4.5,27,1015,89,8,14.5,23,1017,93,5,5.6,22,1018,87,6,5.6,
7,3954,10/25/10,15.5227,-85.265,Honduras,0,0,landslide,tropical_cyclone,medium,unknown,0.6,30,1015,99,9,2.6,30,1013,99,8,2.4,32,1014,100,6,1.4,31,1016,100,9,3.6,29,1016,98,8,4.4,28,1017,99,8,6,29,10
8,2288,3/29/14,42.3946,-122.2137,United_States,0,0,landslide,unknown,unknown,above_road,1.6,11,1028,93,13,0,15,1029,86,9,1.4,15,1027,93,19,8.4,2,1027,96,22,0,8,1027,91,9,0,11,1022,97,13,0,10,1023,97
```

```
,precip12,temp12,air12,humidity12,wind12,precip11,temp11,air11,humidity11,wind11,precip10,temp10,air10,humidity10,wind10,precip9,temp9,air9,humidity9,wind9,precip8,temp8,air8,humidity8,wind8,precip7,
,97,5,0,23,1022,96,7,11.7,24,1020,98,5,33.4,20,1021,96,7,0.2,18,1023,95,7,0,19,1023,90,6,3,21,1023,97,6,0.6,21,1023,95,7,0.3,23,1021,95,5,22.6,20,1019,97,6,1.047045721,0.737547857,7.789081337,24.027
,21.6,4,1014,100,8,9.4,4,1029,100,8,2.8,1,1029,100,6,22.3,4,1019,100,10,0.1,0,1031,100,6,0.1,0,1032,99,8,9.9,3,1020,100,9,0.8,4,1014,100,7,0.5,4,1020,100,8,10.53677015,17.38580139,11.2716969,5.84700
97,11,16.1,31,1015,98,9,24.3,31,1015,97,9,27.2,30,1013,98,8,20.3,30,1013,98,8,8.5,31,1015,96,12,4.9,32,1015,95,25,9.5,31,1015,95,20,18.2,30,1014,97,18,18.3,30,1013,96,13,10.89428943,14.5735388,20.74
8,8,0.7,26,1016,99,9,1.3,25,1016,99,7,5.7,26,1015,95,6,4.7,23,1015,99,9,6.4,28,1015,97,4,9.1,25,1016,98,8,5,28,1015,98,5,0.1,29,1016,100,10,0,28,1017,97,11,4.817555806,2.30632441,1.988242132,4.68272
,1016,89,31,0,16,1020,89,21,25.1,18,1017,98,22,4.1,21,1014,95,25,0,17,1020,92,15,4.2,16,1020,97,12,9.1,11,1024,98,25,61.6,17,1023,99,25,3.3,23,1010,99,18,3.108882898,0.777220724,0.345431433,16.79706
,32,26,3.6,39,1006,35,20,0.9,38,1006,31,30,0.3,35,1007,31,21,4.8,35,1008,31,19,0,35,1012,20,24,0,36,1014,17,18,0,35,1011,22,15,2.8,31,1011,50,16,1.1,31,1013,58,12,2.050778646,2.999502402,1.50084333,
25,1017,90,8,9.4,23,1018,92,5,5.2,25,1018,88,4,7.1,28,1016,94,9,4.1,26,1016,96,6,2.9,29,1016,98,8,5.4,28,1016,95,10,2.1,28,1015,85,10,2.8,26,1015,82,11,6.522090063,5.932252016,8.308820276,6.12912805
```

```
,humidity2,wind2,precip1,temp1,air1,humidity1,wind1,precip0,temp0,air0,humidity0,wind0,ARI9,ARI8,ARI7,ARI6,ARI5,ARI4,ARI3,ARI2,ARI1,ARI0,forest,forest_year,slope,osm,lithology,type,landslide
```

Try Pitch

# 04 Proposed ML Methodology

# Approaches

## Binary Classification

Used to determine whether a landslide will occur at a given location and time (0 or 1).

**Models Used:**

1. K-Nearest Neighbors (KNN)
2. Support Vector Classifier (SVC)
3. Random Forest (RF)

**Purpose:** Evaluate model accuracy on historical data using multi-feature environmental input.

## Temporal Forecasting

Used to predict how soon a landslide might occur, based on recent climate patterns.

**Models Used:**

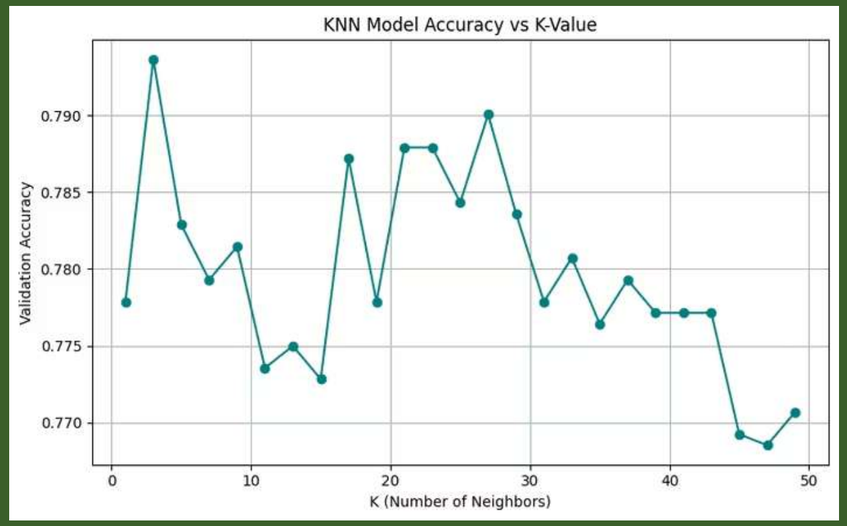
1. LSTM (Recurrent Neural Network)
2. XGBoost (Time-aware tabular prediction)

**Purpose:** Capture temporal dependencies in weather and terrain features to enable proactive alerts.

# 04(a) Binary Classification

## K Nearest Neighbours (KNN)

KNN predicts landslides by finding past cases with similar conditions. If most led to landslides, it predicts one will occur.




KNN achieved peak accuracy around when KNN value was around 3 to 5.
It was slow to compute during training and didn't scale well to 18,000+ samples.

## Support Vector Machine (SVM)

SVC finds the optimal boundary (hyperplane) that best separates landslide and non-landslide points in high-dimensional space.

```
nSV = 7465, nBSV = 7156
Total nSV = 7465
```

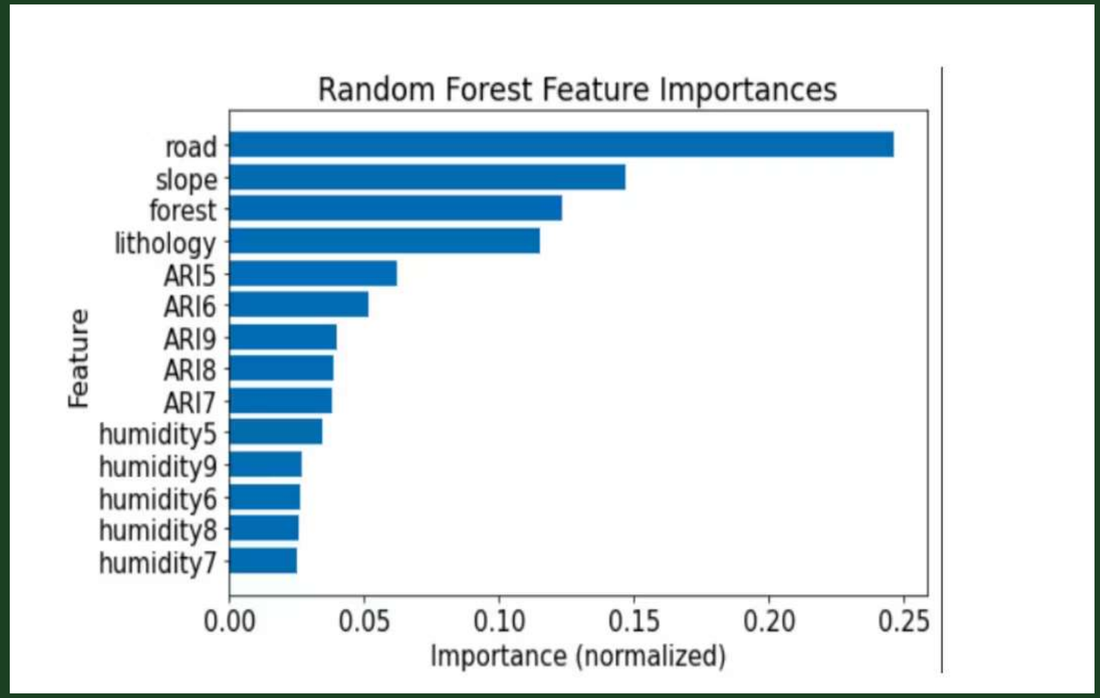
The model used 7465 support vectors, nearly 41% of the training data, reflecting a highly complex decision boundary.

This leads to higher memory use and slower prediction, making it harder to scale.

# Approach 3: Random Forest (RF)

**Random Forest** predicts if a **landslide** will occur by combining the results of many **decision trees**, each trained on different **environmental features** (like **ARI**, **slope**, **forest loss**, etc.) and **data subsets**. The final prediction is based on a **majority vote** across all trees.

**RF** captures **complex, non-linear patterns** across features to **robustly predict landslides** across diverse **terrain** and **climate conditions**.




LandSafe's RF model relies more on terrain and infrastructure than short-term weather, boosting generalization across regions.

# 04(b) Temporal Forecasting

# Approach 4: LSTM Neural Network

**Input:**

- 10-day time-series window with 10 features per day

**Architecture (attached image)**

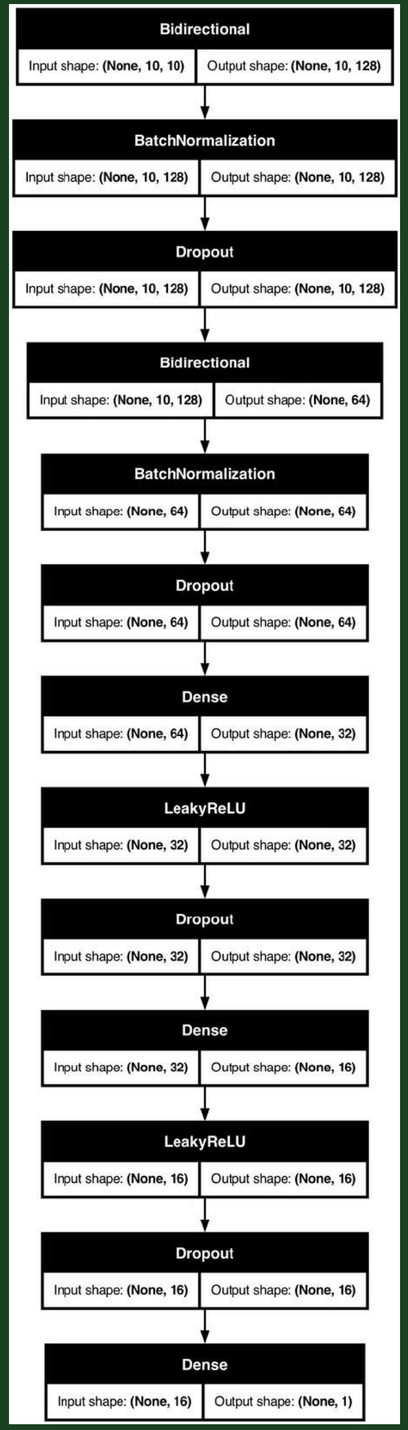
- Bidirectional LSTM-based Neural Network with stacked layers
- Dropout applied after each layer (p = 0.2)
- LeakyReLU activations used after each Dense layer

**Compilation:**

- Loss Function: Binary Crossentropy
- Optimizer: Adam

**Training Configuration:**

- Epochs: 25
- Batch Size: 64



# Approach 5: XGBoost

Used XGBoost Classifier for binary classification to predict if a landslide will occur within the next 24 hours based on the preceding 15 days of climate and terrain data.

**Features**

- Total of 89 features.
- Past 15-day sequences of Precipitation, Temperature, Air Pressure, Humidity, Wind Speed
- Static features: slope, forest loss, OSM infrastructure, lithology

**Training Details**

- Ordinal encoding for categorical features
- Feature scaling applied before model input
- Model trained using XGBoost DMatrix format for efficiency

# Hyperparameter Tuning

```
# XGBoost parameters
params = {
    'objective': 'binary:logistic',
    'eval_metric': 'logloss',
    'eta': 0.1,
    'max_depth': 6,
    'min_child_weight': 1,
    'subsample': 0.8,
    'colsample_bytree': 0.8,
    'gamma': 0.1,
    'alpha': 0.1,
    'lambda': 1.0,
    'seed': 42
}
```

We used early stopping. Stopped at 20 rounds to avoid overfitting!

# Challenges

- **Domain Knowledge**
  Understanding terrain-specific features like lithology and slope required deep geospatial insight.
  **Our solution:** We conducted extensive literature reviews and expert consultation to guide feature relevance.

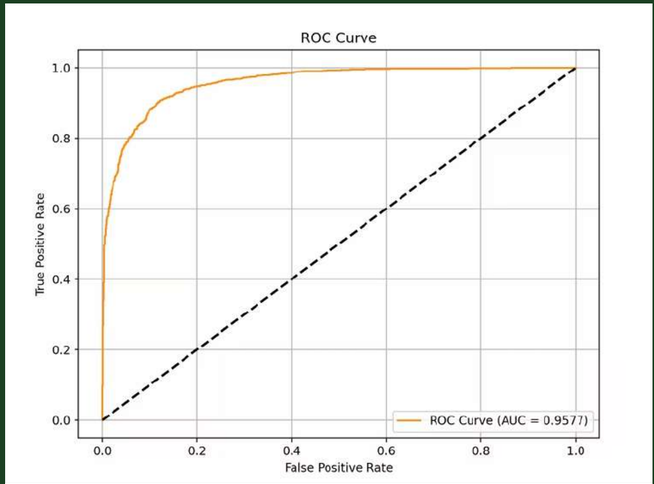
- **Feature Preprocessing**
  Data from APIs and TIF raster files was noisy and inconsistent across sources.
  **Our solution:** We built custom parsers and applied thresholding, scaling, and encoding pipelines.

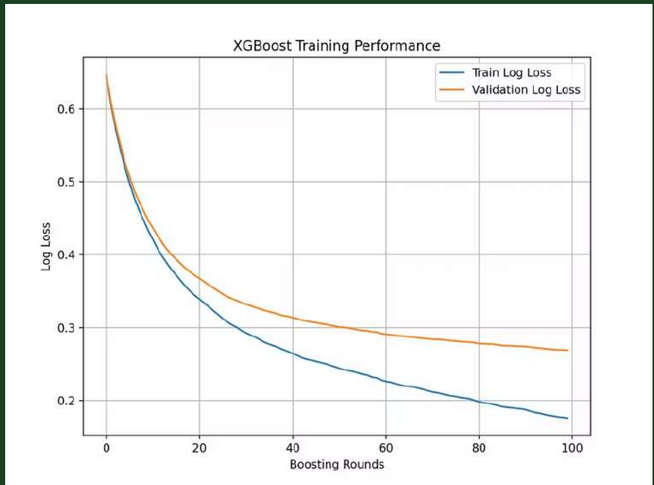
- **Real-Time Integration**
  Weather APIs had quota limits and inconsistent temporal coverage.
  **Our solution:** We designed a lightweight pipeline and explored sensor-based alternatives for edge inference.

# 05 Performance Metrics & Deployability

# The Best Performer is...

# XGBoost!

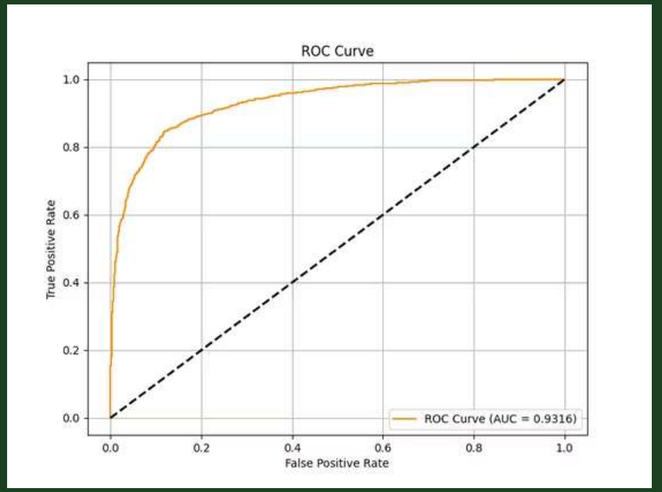


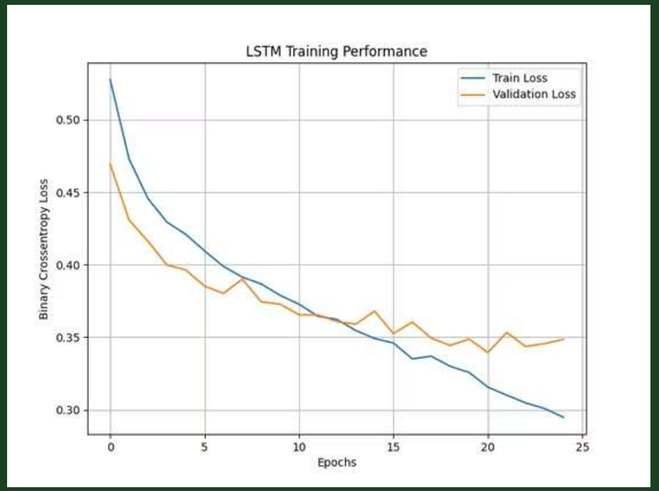




```
============================== MODEL EVALUATION ==============================
Train Accuracy : 0.9471
Test Accuracy  : 0.8896
Precision      : 0.8950
Recall         : 0.8816
F1 Score       : 0.8882
ROC AUC        : 0.9577

Confusion Matrix:
[[1814 207] <- [True Negatives, False Positives]]
[[237 1764] <- [False Negatives, True Positives]]

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.90      0.89      2021
           1       0.89      0.88      0.89      2001

    accuracy                           0.89      4022
   macro avg       0.89      0.89      0.89      4022
weighted avg       0.89      0.89      0.89      4022
```

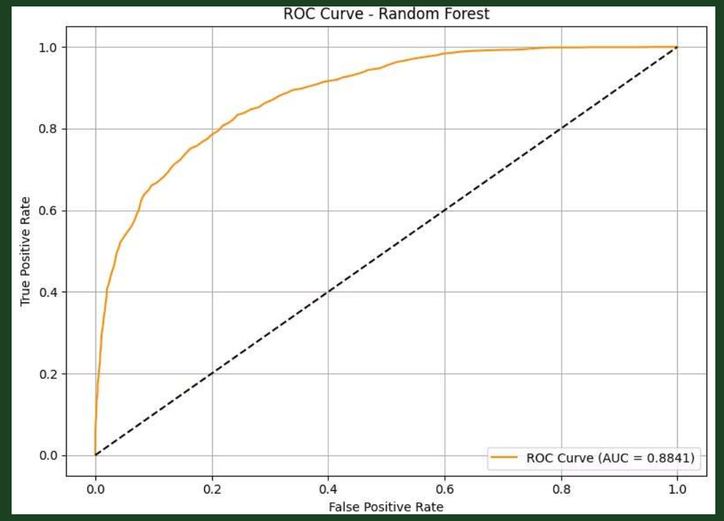
# LSTM

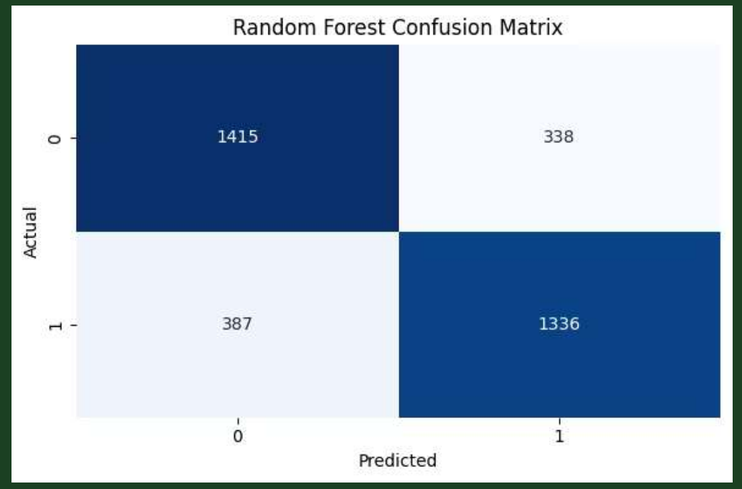
```
======================== MODEL EVALUATION ========================
Train Accuracy : 0.8960
Test Accuracy  : 0.8533
Precision      : 0.8605
Recall         : 0.8416
F1 Score       : 0.8509
ROC AUC        : 0.9278

Confusion Matrix:
[[1748 273] <- [True Negatives, False Positives]]
[[317 1684] <- [False Negatives, True Positives]]

Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.86      0.86      2021
           1       0.86      0.84      0.85      2001

    accuracy                           0.85      4022
   macro avg       0.85      0.85      0.85      4022
weighted avg       0.85      0.85      0.85      4022
```



# Random Forest




```
Training Random Forest Classifier...

Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.83      0.84      1753
           1       0.83      0.85      0.84      1723

    accuracy                           0.84      3476
   macro avg       0.84      0.84      0.84      3476
weighted avg       0.84      0.84      0.84      3476

Accuracy: 0.8357307249712314
```

Try Pitch

# And last but not the least...

# KNN

```
Training KNN...
k parameter: 100%|██████████| 25/25 [00:01<00:00, 21.59it/s]

Best k value: 37

Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.75      0.73      1756
           1       0.73      0.67      0.70      1720

    accuracy                           0.71      3476
   macro avg       0.71      0.71      0.71      3476
weighted avg       0.71      0.71      0.71      3476
```

# SVC

```
Total nSV = 7913
[[1292  464]
 [ 510 1210]]
              precision    recall  f1-score   support

           0       0.72      0.74      0.73      1756
           1       0.72      0.70      0.71      1720

    accuracy                           0.72      3476
   macro avg       0.72      0.72      0.72      3476
weighted avg       0.72      0.72      0.72      3476
```

# Deployability of LandSafe

## Deployment Potential at Plaksha:

**LandSafe** can be deployed as an **early warning system** to assess **landslide risk** in **real-time** using **local environmental data**.

**Plaksha's proximity** to the **Shivalik hills**, combined with **seasonal monsoon rainfall**, makes it a **relevant testbed** for **landslide forecasting**.

A **lightweight dashboard** or **alerting system** could be integrated with **local weather stations** and **satellite feeds** for **live prediction**.

## How it'll work:

- **Integrate LandSafe** with **on-site weather sensors** (*humidity, rainfall, temperature*).
- **Feed this data** into the **trained model** to predict **landslide risk** for the **next 24 hours**.
- **Trigger warnings** for **campus authorities** via **SMS** or **web dashboard** if **high risk** is detected.

## Scalability Challenges:

- We'll have to create a **real-time data processing pipeline** to fetch **features** required for **inference**; for that we'll need to find **open-source APIs** or develop **custom sensors**.
- Even though we've trained on the **largest dataset**, there's a chance that **feature values** may differ and lead to **wrong predictions**.

# References


https://www.usgs.gov/natural-hazards/landslide-hazards

Abraham, M. T., Satyam, N., Kushal, S., Rosi, A., Pradhan, B., & Segoni, S. (2020). Rainfall Threshold Estimation and Landslide Forecasting for Kalimpong, India Using SIGMA Model. Water, 12(4), 1195. https://doi.org/10.3390/w12041195

Nava, L., Carraro, E., Reyes-Carmona, C. et al. Landslide displacement forecasting using deep learning and monitoring data across selected sites. Landslides 20, 2111-2129 (2023). https://doi.org/10.1007/s10346-023-02104-9

Kirschbaum, Dalia & Stanley, Thomas. (2018). Satellite-Based Assessment of Rainfall-Triggered Landslide Hazard for Situational Awareness. Earth's Future. 6. 10.1002/2017EF000715.

Nava et al. (2023) - Use of BiLSTM, CNN, etc.

https://doi.org/10.1007/s10346-023-02104-9

https://www.kaggle.com/datasets/kazushiadachi/global-landslide-data

https://earthenginepartners.appspot.com/science-2013-global-forest

https://www.openstreetmap.org

https://open-meteo.com

https://landslides.nasa.gov/

https://education.nationalgeographic.org/resource/landslide

# Thank You