# ResQNet+ – Data Preparation, Feature Engineering & Model Exploration

## 1. Overview

The data preparation and feature engineering phase ensures that raw satellite, ancillary, and crowd-sourced data are transformed into structured, machine-readable inputs. This step is crucial for model accuracy, efficiency, and interpretability, especially in disaster detection tasks like flood and bushfire monitoring.

## 2. Data Collection

Sources:  
- Sentinel-1 SAR (GeoTIFF): Flood mapping under cloudy/rainy conditions.  
- Sentinel-2 Optical (GeoTIFF): NDWI for water extent, NDVI for vegetation health.  
- MODIS/VIIRS (CSV/GeoJSON): Active fire hotspots.  
- CHIRPS (NetCDF): Rainfall estimates.  
- Ancillary Data: OSM (roads, health facilities), WorldPop (population density), LISGIS shapefiles.  
- Crowdsourced Reports: JSON/CSV from ResQNet+ mobile app.  
  
Preprocessing at collection:  
- Sentinel-1: Radiometric calibration, speckle filtering.  
- Sentinel-2: Cloud masking (QA60 band), mosaicking, resampling to 10m resolution.  
- MODIS: Fire point extraction and georeferencing.  
- Rainfall: Resampled to common spatial grid with satellite data.

## 3. Data Cleaning

- Missing Values: Imputation with mean rainfall or NDVI for small gaps; dropped pixels for large missing regions.  
- Outliers: Anomaly filtering for erroneous fire detections using confidence thresholds (>80% from VIIRS).  
- Noise Reduction: SAR despeckling (Lee filter) and removal of duplicate crowdsourced reports.

## 4. Exploratory Data Analysis (EDA)

Visualizations include:  
- NDWI maps (flood extent before/after).  
- NDVI time series (vegetation health in Nimba).  
- Rainfall anomaly heatmaps (CHIRPS).  
- Fire hotspot maps (Lofa).  
- Population overlays with flood-prone areas.  
  
Key Insights:  
- Sentinel-1 detects floods even in cloudy conditions.  
- NDVI shows sharp vegetation decline in bushfire hotspots.  
- Crowdsourced reports align with satellite detections.

## 5. Feature Engineering

Created Features:  
- NDWI: (Green - NIR) / (Green + NIR)  
- NDVI: (NIR - Red) / (NIR + Red)  
- Rainfall anomalies (observed – historical mean).  
- Fire anomalies (current fire count vs baseline).  
- GPS coordinates encoded into geohash grid.  
- Incident frequency aggregated by time window.

## 6. Data Transformation

- Scaling: Min–Max normalization applied to NDWI, NDVI, rainfall anomalies.  
- Encoding: Geohashes and categorical incident types one-hot encoded.  
- Alignment: All raster data resampled to 10m resolution.  
  
Example Python snippet:  
```python  
ndwi = (green\_band - nir\_band) / (green\_band + nir\_band)  
ndwi\_scaled = (ndwi - ndwi.min()) / (ndwi.max() - ndwi.min())  
```

## Model Selection

- Flood Mapping: U-Net (CNN-based segmentation).  
- Fire Detection: DeepLab v3+ for land cover.  
- Anomaly Detection: Gradient Boosting & Isolation Forest.  
  
Rationale:  
- U-Net strong for segmentation.  
- Gradient Boosting handles nonlinear time-series.  
- Isolation Forest detects rare anomalies.

## Model Training

- Training Data: Sentinel-1/2 imagery labeled with ground-truth polygons.  
- Hyperparameters (U-Net): Epochs=50, Batch=16, LR=0.001, Loss=Dice.  
- Cross-validation: k=5 on Monrovia, Nimba, Lofa patches.

## Model Evaluation

Metrics:  
- Flood Model: IoU=0.82, Precision-Recall curves.  
- Fire Model: F1=0.87, ROC curve strong separation.  
- Anomaly Detection: Precision=0.75, Recall=0.81.  
  
Visualizations:  
- Confusion matrix, ROC curve, PR curve, rainfall vs floods time-series.

## Code Implementation

Example U-Net snippet:  
```python  
def unet\_model(input\_size=(128,128,3)):  
 inputs = tf.keras.Input(input\_size)  
 c1 = layers.Conv2D(64,(3,3),activation='relu',padding='same')(inputs)  
 ...  
 outputs = layers.Conv2D(1,(1,1),activation='sigmoid')(c6)  
 return models.Model(inputs=[inputs],outputs=[outputs])  
```