HIMASHIELD 2024: Early Warning Detection System for GLOFs

Title of Innovation:

GLOFSense: Integrated Early Warning Detection System for Glacial Lake Outburst Floods (GLOFs)

1. Introduction

Overview

GLOFSense is an advanced solution to monitor and mitigate the risks of Glacial Lake Outburst Floods (GLOFs), specifically targeting the Hindu Kush Himalaya (HKH) region. By utilizing state-of-the-art remote sensing technologies, machine learning (ML), and IoT-enabled hardware, the system predicts, monitors, and provides early warnings for potential GLOF events, thereby enhancing disaster preparedness and response capabilities.

Scope

This document covers:

1. Satellite-based detection using Synthetic Aperture Radar (SAR) images and ML simulations.
2. IoT-enabled floating devices for real-time monitoring.
3. End-to-end integration with disaster response systems and local alert mechanisms.

2. Architecture Design

2.1 System Overview

System Description: GLOFSense integrates ML algorithms trained on satellite data and IoT hardware deployed on-site to monitor and predict glacial lake behavior. The system generates early warnings through a web interface and alerts relevant disaster management agencies.

2.2 System Architecture

1. **High-Level Architecture Diagram**:
   - **Satellite Processing Pipeline**:

        - Input: Sentinel-1 SAR images from 1995-2021 via Copernicus.
        - Processing: Image enhancement, pixel-based segmentation using convolutional neural networks (CNNs).
        - Output: Area growth trends and breach probability predictions based on time-series data.
    - **ML Prediction Module**:
        - Model:
            - **CNN (Convolutional Neural Network)** for spatial analysis and feature extraction of SAR images.
            - **LSTM (Long Short-Term Memory)** for time-series forecasting of lake area changes based on historical trends.
        - Training: Dataset of 6000 SAR images annotated for area segmentation (128x128 resolution, 15 epochs).
        - Validation: Mean Intersection over Union (mIoU) and Root Mean Square Error (RMSE) for prediction accuracy.
    - **IoT Device Network**:
        - Floating PCB arrays equipped with multi-modal sensors for real-time data collection.
        - Communication via LoRa (Long-Range Radio) for transmitting data to central servers.
    - **Alert Systems**:
        - Dashboard: Web interface for visualizing risk metrics, thresholds, and time-series predictions.
        - Alerts: Real-time notifications via SMS, email, and integration with NDAP.
2. **Technology Stack**:
    - **Programming**: Python (ML and backend), C++ (firmware), JavaScript (frontend).

- **Frameworks**: TensorFlow and PyTorch (ML models), Flask (web app), Arduino IDE (hardware control).
- **Hardware Components**:
  - Sensors: MS5803-14BA (water level), YFS201 (flow rate), DHT11 (temperature), ADXL345 (tilt), BME280 (pressure).
  - Microcontroller: ESP32 for on-site data processing.
  - Communication: LoRa SX1276 for long-range communication.
- **Cloud Services**: AWS S3 for data storage, EC2 for real-time ML model inference, and CloudWatch for monitoring.

3. **Scalability & Performance Considerations**:
   - Real-time SAR data processing using cloud-based GPUs.
   - Energy-efficient IoT devices optimized for remote and extreme environments with solar power.
4. **Security Considerations**:
   - End-to-end encryption of data transmissions.
   - OAuth-based secure API access for web and mobile users.
5. **Fault Tolerance & Redundancy**:
   - Redundant LoRa nodes ensure communication reliability in the event of device failure.
   - Scheduled model retraining to adapt to evolving glacial dynamics.
6. **Integration Points**:
   - Copernicus and Sentinel-1 satellite APIs for SAR data retrieval.
   - National and regional disaster management platforms for automated alert dissemination.

3. Execution Plan

3.1 Development Strategy

1. **Phases of Execution**:
   - Phase 1: Feasibility Study & Dataset Preparation (Oct 2024).
   - Phase 2: Prototype Development (Nov 2024).
   - Phase 3: System Testing & Optimization (Dec 2024).
   - Phase 4: Deployment and Pilot Studies (Jan 2024).
2. **Timeline**:
   - Dataset Collection & Preprocessing: 15 days.
   - ML Model Development: 1 month.
   - Hardware Testing: 1 month.
   - Full Integration & Deployment: 1.5 months.
3. **Task Allocation**:
   - Venkatesh R: SAR image preprocessing and ML model training.
   - Nithish T: Sensor integration and hardware prototyping.
   - Naveen Krishnaa S: Communication protocol development.
   - Mukeshraj: Web interface design and alert system integration.

3.2 Development Process

Methodology: Agile development with bi-weekly sprints and milestones.

Tools: GitHub (version control), Docker (containerization), Jira (task management).

4. List of Submodules

Submodule 1: Satellite-Based Detection

- **Description**: Analyze SAR imagery to detect and predict lake area expansion.
- **Responsibilities**:
  - Image preprocessing: Noise reduction and contrast enhancement.

- ML model training: Pixel-based area segmentation and time-series forecasting.

- **Technologies**: TensorFlow, GDAL (Geospatial Data Abstraction Library), OpenCV.

- **Dependencies**: Sentinel-1 data, pretrained CNN models.

Submodule 2: IoT-Based Monitoring

- **Description**: Deploy IoT devices for real-time lake parameter monitoring.

- **Responsibilities**: Sensor data acquisition, threshold-based alerting, and real-time communication.

- **Technologies**: Arduino IDE, LoRa SX1276, ESP32.

- **Dependencies**: Sensor hardware, LoRaWAN network infrastructure.

Submodule 3: Web Interface

- **Description**: Centralized dashboard for visualizing data and managing alerts.

- **Responsibilities**:

    - Real-time visualization of lake parameters and predictions.

    - User notifications and alert management.

- **Technologies**: Flask, JavaScript, HTML/CSS.

- **Dependencies**: Cloud-hosted database and REST APIs.

5. System Integration Plan

5.1 Integration Strategy

- **Module Integration**: Validate interoperability of ML models, IoT devices, and dashboard components.

- **Integration Phases**: Unit testing, system testing, field trials.

- **Integration Testing**: Simulate GLOF scenarios using historical data and synthetic thresholds.

5.2 Data Flow & Communication

1. **Data Flow Diagram**:
    - SAR imagery > Preprocessing > ML predictions > Dashboard.
    - IoT sensor data > LoRa > Cloud database > Dashboard.
2. **Communication Protocols**: LoRaWAN for IoT, HTTPS for secure cloud interactions.
3. **Error Handling & Logging**: Robust logging for anomalies, with automatic retry mechanisms.

5.3 External Systems Integration

- **Integration with NDAP**: Automated data feeds for national disaster management systems.
- **Authentication**: OAuth-based secure access.

6. Test Plan

6.1 Testing Objectives

1. Validate SAR-based predictions for accuracy and reliability.
2. Ensure IoT-based sensor data matches real-time lake conditions.
3. Confirm seamless integration of all components.

6.2 Test Strategy

- **Test Types**:
    - Unit tests for individual modules.
    - Integration tests for system-wide interactions.
    - Field tests for IoT devices in simulated environments.
- **Tools**: PyTest, Selenium (for web testing), Arduino IDE (for hardware testing).

6.3 Test Cases

1. **Image Processing Validation**:
    - Input: Historical SAR images.

- Expected Output: Accurate segmentation (mIoU > 85%) and area predictions.

2. **IoT Sensor Alerts**:
   - Input: Simulated threshold breaches.
   - Expected Output: Timely alerts (<10 seconds latency).
3. **Dashboard Integration**:
   - Input: Combined data streams.
   - Expected Output: Synchronized visualization and actionable insights.

6.4 Test Execution

Issue Tracking: Automated issue tracking through Jira with real-time resolution updates.

References

1. https://glofsense.com
2. Copernicus Sentinel-1: https://www.copernicus.eu
3. ICIMOD Reports: https://icimod.org
4. Springer Journal: https://link.springer.com/article/10.1007/s12665-021-09740-1