**ĐẠI HỌC PHENIKAA**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN**

A logo for a university

AI-generated content may be incorrect.

**BÁO CÁO BÀI TẬP LỚN**

**MÔN HỌC PHÂN TÍCH DỮ LIỆU VỚI PYTHON**

***ĐỀ TÀI:***

**HỆ THỐNG QUẢN LÝ VÀ DỰ ĐOÁN SẠT LỞ RỪNG**

**Họ và tên: Phạm Anh Tuấn**

**Mã sinh viên: 21011117**

**Lớp: CSE702031-1-3-24(N01)**

**Giảng viên hướng dẫn: ThS Nguyễn Ngọc Hùng**

**09 tháng 07 năm 2025**

Mục Lục

[Danh mục hình ảnh 3](#_Toc202916889)

[I. Giới thiệu 4](#_Toc202916890)

[II. Mô tả hệ thống 5](#_Toc202916891)

[1. Hệ thống cảm biến và truyền dữ liệu 5](#_Toc202916892)

[2. Trạm trung gian và truyền dữ liệu lên máy chủ 6](#_Toc202916893)

[3. Xử lý dữ liệu tại server 6](#_Toc202916894)

[III. Khám phá và tiền xử lý dữ liệu 7](#_Toc202916895)

[1. Nạp và khám phá dữ liệu ban đầu 7](#_Toc202916896)

[2. Mất cân bằng dữ liệu 8](#_Toc202916897)

[3. Chiến lược xử lý và dự đoán 8](#_Toc202916898)

[4. Ý nghĩa logic dự đoán 9](#_Toc202916899)

[IV. Mô tả hàm chuẩn bị dữ liệu chuan\_bi\_data() 9](#_Toc202916900)

[1. Mục đích 9](#_Toc202916901)

[2. Các bước thực hiện 9](#_Toc202916902)

[V. Mô tả hàm dự đoán du\_doan () 11](#_Toc202916903)

[1. Mục đích 11](#_Toc202916904)

[2. Đầu vào hàm 11](#_Toc202916905)

[3. Các bước thực hiện 11](#_Toc202916906)

[4. Kết luận 12](#_Toc202916907)

[VI. Trực quan hóa dữ liệu trên Dashboard 13](#_Toc202916908)

[1. Mục tiêu trực quan hóa 13](#_Toc202916909)

[2. Công nghệ sử dụng 13](#_Toc202916910)

[3. Nội dung hiển thị 13](#_Toc202916911)

[5. Kết luận 15](#_Toc202916912)

[VII. Đề xuất mở rộng 15](#_Toc202916913)

[1. Mở rộng tập dữ liệu và tăng độ chính xác dự đoán 15](#_Toc202916914)

[2. Tích hợp bản đồ số (GIS) 15](#_Toc202916915)

[3. Hệ thống cảnh báo sớm 16](#_Toc202916916)

[4. Tối ưu phần cứng cảm biến 16](#_Toc202916917)

[5. Truy cập từ xa và điều khiển tập trung 16](#_Toc202916918)

[Kết luận đề xuất 16](#_Toc202916919)

# Danh mục hình ảnh

Hình 1.1 sơ đồ tổng quan hệ thống………………………………………………....5

Hình 1.2: Giao diện dashboard 1…………………………………………………..14

Hình 1.3 Giao diện dashboard 2………………………………………………….. 14

# I. Giới thiệu

Quản lý và bảo vệ tài nguyên rừng là một nhiệm vụ cấp thiết trong bối cảnh biến đổi khí hậu và gia tăng các hoạt động xâm hại tài nguyên thiên nhiên. Để đáp ứng nhu cầu theo dõi, giám sát và thu thập dữ liệu môi trường tại các khu vực rừng sâu, nơi hạ tầng viễn thông còn hạn chế, dự án Hệ thống quản lý rừng thông minh đã được xây dựng nhằm ứng dụng công nghệ vào công tác quản lý và bảo vệ rừng hiệu quả hơn.

Hệ thống được thiết kế với mục tiêu thu thập dữ liệu từ các điểm đo được triển khai sâu trong rừng, bao gồm thông tin như nhiệt độ, độ ẩm, lượng mưa, chuyển động, và các tín hiệu liên quan đến nguy cơ cháy rừng hoặc phá rừng. Dữ liệu từ các điểm đo này được truyền về trạm trung gian thông qua sóng vô tuyến (RF - Radio Frequency). Đây là phương thức truyền thông tin hiệu quả trong điều kiện không có hạ tầng mạng, cho phép đảm bảo kết nối giữa các điểm đo cách xa nhau và nằm sâu trong vùng rừng núi.

Tại các trạm trung gian – những vị trí đã được khảo sát có vùng phủ sóng mạng di động hoặc Wi-Fi – hệ thống sẽ tự động đẩy dữ liệu lên nền tảng Firebase, nơi lưu trữ dữ liệu theo thời gian thực và phục vụ cho việc phân tích, cảnh báo, và hiển thị trên hệ thống giám sát trung tâm. Firebase đóng vai trò trung gian đồng bộ dữ liệu giữa các thiết bị thu thập thực địa và máy chủ hệ thống quản lý trung tâm, nơi dữ liệu được xử lý, trực quan hóa và hỗ trợ đưa ra các quyết định quản lý kịp thời.

Việc kết hợp giữa công nghệ truyền dữ liệu không dây (RF), nền tảng đám mây (Firebase) và hệ thống máy chủ giúp đảm bảo tính liên tục, ổn định và tiết kiệm năng lượng, phù hợp với điều kiện vận hành trong rừng – nơi thường xuyên thiếu điện, mạng và có điều kiện môi trường khắc nghiệt.

# II. Mô tả hệ thống

A diagram of a computer network

AI-generated content may be incorrect.

Hình 1.1 sơ đồ tổng quan hệ thống

## 1. Hệ thống cảm biến và truyền dữ liệu

Hệ thống được xây dựng với mục tiêu thu thập dữ liệu môi trường tại các khu vực rừng, từ đó phục vụ công tác giám sát và dự đoán nguy cơ sạt lở đất. Dữ liệu được lấy từ các cảm biến vật lý triển khai tại hiện trường, với các thông số bao gồm:

* Nhiệt độ
* Độ ẩm không khí
* Lượng mưa
* Tốc độ gió
* Và các yếu tố môi trường khác (nếu có mở rộng)

Các cảm biến này được kết nối trực tiếp với vi điều khiển ESP32, đóng vai trò là bộ thu thập và xử lý dữ liệu sơ cấp tại điểm đo. Sau khi xử lý, dữ liệu được truyền từ các điểm thu thập đến máy trạm (gateway) thông qua mô-đun truyền thông vô tuyến HC-12. Mô-đun HC-12 hoạt động ở dải tần 433 MHz, cho phép truyền dữ liệu với khoảng cách xa trong điều kiện không có mạng lưới viễn thông hoặc hạ tầng Internet sẵn có – rất phù hợp cho môi trường rừng núi hẻo lánh.

Trong giai đoạn mô phỏng hệ thống, dữ liệu lượng mưa được tạo ngẫu nhiên trong khoảng 0 đến 104 mm, với chu kỳ gửi cố định 10 phút một lần. Việc mô phỏng này nhằm đánh giá hiệu suất truyền nhận, kiểm tra quy trình lưu trữ, phân tích và dự đoán mà không cần thiết lập hệ thống cảm biến thực tế trong giai đoạn đầu.

## 2. Trạm trung gian và truyền dữ liệu lên máy chủ

Tại các máy trạm, thường được đặt ở những khu vực ven rừng hoặc có thể tiếp cận được mạng Internet (qua mạng di động hoặc Wi-Fi), dữ liệu được nhận từ các điểm đo thông qua sóng vô tuyến. Sau khi tiếp nhận, dữ liệu sẽ được tự động đẩy lên nền tảng Firebase – một dịch vụ backend theo thời gian thực được sử dụng để lưu trữ và đồng bộ dữ liệu.

Dữ liệu được truyền dưới dạng chuỗi đơn giản với cấu trúc

JSON:{"data":"thời gian: giá trị"}

Ví dụ: {"data":"22h46:31"}.

Trong đó:

thời gian: là thời điểm gửi dữ liệu đi

giá trị: là lượng mưa đo được tại thời điểm đó

## 3. Xử lý dữ liệu tại server

Trên server hệ thống, dữ liệu được truy xuất từ Firebase theo thời gian thực. Sau đó, một quy trình xử lý sẽ được kích hoạt để:

* Tách dữ liệu thành 2 phần riêng biệt: thời gian và giá trị đo được.
* Ghi dữ liệu vào file .csv để lưu trữ, phục vụ mục đích theo dõi lâu dài, kiểm tra lịch sử và backup.
* Trực quan hóa dữ liệu: các giá trị đo theo thời gian được hiển thị trực tiếp trên dashboard giúp người dùng theo dõi xu hướng biến đổi của môi trường.

Ngoài ra, dữ liệu sau xử lý còn được đưa vào mô hình học máy (AI/ML) đã được huấn luyện từ trước, nhằm dự đoán nguy cơ sạt lở tại thời điểm hiện tại hoặc trong tương lai gần. Mô hình này có thể đánh giá khả năng xảy ra sạt lở theo mức độ “cao” hoặc “thấp” dựa trên các tham số đo được và lịch sử dữ liệu.

Trên mức độ mô phỏng, em sử dung file data trên kagle : [Dữ liệu tương quan lượng mưa theo giờ và sạt lở.](https://www.kaggle.com/datasets/hoibonguyn/d-liu-tng-quan-lng-ma-theo-gi-v-st-l)

# III. Khám phá và tiền xử lý dữ liệu

## 1. Nạp và khám phá dữ liệu ban đầu

Dữ liệu được nạp vào từ một tệp Excel sử dụng thư viện pandas, tạo thành một DataFrame bao gồm các trường chính sau:

* Ngay: Ngày ghi nhận dữ liệu
* Gio: Giờ trong ngày (theo định dạng số nguyên)
* Luong mua: Lượng mưa đo được tại thời điểm đó (đơn vị mm)
* Trượt lở: Biến phân loại đánh dấu hiện tượng trượt lở (0: Không xảy ra; 1: Xảy ra trượt lở)

Để thuận tiện cho việc xử lý và mã hóa thống nhất trong hệ thống, cột "Trượt lở" được đổi tên thành truot\_lo.

Sau khi kiểm tra dữ liệu ban đầu, phát hiện có 2.547 giá trị bị thiếu (NaN) trong cột Luong mua. Các giá trị thiếu này có thể làm sai lệch thống kê và ảnh hưởng đến kết quả phân tích. Do đó, tác giả áp dụng phương pháp xử lý như sau:

* Với mỗi dòng bị thiếu giá trị lượng mưa, hệ thống sẽ tự động tính trung bình của ba giá trị gần nhất theo thời gian, và bỏ qua các giá trị NaN nếu có trong ba điểm đó.
* Phương pháp này đảm bảo tái tạo dữ liệu một cách hợp lý, hạn chế làm méo mó xu hướng thực tế của lượng mưa.

## 2. Mất cân bằng dữ liệu

Sau quá trình thống kê phân bố nhãn truot\_lo, nhận thấy sự mất cân bằng rõ rệt giữa số trường hợp xảy ra và không xảy ra sạt lở, cụ thể:

* Tổng số dòng dữ liệu: ~87.672 dòng
* Trong đó:
  + Không xảy ra sạt lở (truot\_lo = 0): 87.551 dòng
  + Xảy ra sạt lở (truot\_lo = 1): 121 dòng

Tỷ lệ giữa hai nhãn là 87551:121 (~724:1), thể hiện mức độ mất cân bằng dữ liệu rất cao.

## 3. Chiến lược xử lý và dự đoán

Do mất cân bằng dữ liệu, mô hình học máy sẽ dễ bị lệch và đánh giá sai kết quả nếu không có biện pháp xử lý đặc biệt (như oversampling, under-sampling hoặc điều chỉnh trọng số). Tuy nhiên, thay vì áp dụng mô hình học máy truyền thống, hệ thống sử dụng một mô hình dự đoán dựa trên thống kê và ngưỡng thực nghiệm, cụ thể như sau:

* Xác suất sạt lở (xac\_suat\_satlo) tại một thoi\_diem (tháng-ngày-giờ) được tính dựa trên tỷ lệ truot\_lo tại thời điểm đó.
* Lượng mưa trung bình gây sạt lở (mua\_tb) và lượng mưa thấp nhất từng gây sạt lở (mua\_min) được tính toán từ tập dữ liệu lịch sử.

Dựa trên các ngưỡng này, mức độ nguy cơ được phân loại như sau:

if luong\_mua >= mua\_min:

return 'Cao.'

if xac\_suat == 0:

return 'Thap.'

if luong\_mua >= mua\_tb:

return 'Cao'

else:

return 'Thap'

## 4. Ý nghĩa logic dự đoán

* Nếu lượng mưa hiện tại lớn hơn hoặc bằng lượng mưa thấp nhất từng gây sạt lở, hệ thống đánh giá nguy cơ "Cao" bất kể xác suất.
* Nếu tại thời điểm đó chưa từng xảy ra sạt lở, và ngưỡng lượng mưa an toàn xác suất bằng 0 → nguy cơ "Thấp".
* Nếu lượng mưa cao hơn trung bình tại các thời điểm đã xảy ra sạt lở → nguy cơ "Cao".
* Nếu không thỏa các điều kiện trên → nguy cơ "Thấp".

Phương pháp này tuy đơn giản nhưng hiệu quả trong bối cảnh thiếu dữ liệu và yêu cầu tính toán nhẹ (phù hợp cho hệ thống nhúng và thời gian thực).

# IV. Mô tả hàm chuẩn bị dữ liệu chuan\_bi\_data()

## 1. Mục đích

Hàm chuan\_bi\_data() có nhiệm vụ xử lý dữ liệu gốc từ cảm biến và file dữ liệu, trích xuất thông tin thời gian, và tính toán thống kê về xác suất sạt lở và lượng mưa gây sạt lở theo từng thời điểm cụ thể. Kết quả của hàm được lưu vào file du\_doan.csv để phục vụ cho việc dự đoán sau này.

## 2. Các bước thực hiện

**a) Xử lý thời gian**

df['Ngay'] = pd.to\_datetime(df['Ngay'], format='%Y-%m-%d', errors='coerce')

df['month'] = df['Ngay'].dt.month

df['day'] = df['Ngay'].dt.day

df['hour'] = df['Gio']

* Chuyển đổi cột Ngay sang kiểu datetime để trích xuất các thành phần ngày, tháng.
* Kết hợp với Gio để tạo thông tin thời gian chính xác đến từng giờ.

**b) Tạo khóa thời gian (thoi\_diem)**

df['thoi\_diem'] = df.apply(

lambda row: f"{int(row['month']):02d}-{int(row['day']):02d} {int(row['hour']):02d}h",

axis=1

)

* Tạo cột mới thoi\_diem dạng chuỗi MM-DD HHh để gom nhóm dữ liệu dễ dàng theo từng khung giờ.

**c) Thống kê trên dữ liệu sạt lở**

df\_sl = df[df['truot\_lo'] == 1]

mua\_min = df\_sl['Luong mua'].min()

mua\_tb = df\_sl.groupby('thoi\_diem')['Luong mua'].mean().reset\_index(name='luong\_mua\_tb\_khi\_satlo')

* Lọc các dòng xảy ra sạt lở.
* Tính lượng mưa thấp nhất từng gây sạt lở (mua\_min).
* Tính lượng mưa trung bình gây sạt lở theo từng thời điểm (mua\_tb).

**d) Tính xác suất sạt lở theo thời điểm**

xac\_suat=df.groupby('thoi\_diem')['truot\_lo'].mean().reset\_index(name='xac\_suat\_satlo')

* Tính xác suất (tỷ lệ) xảy ra sạt lở tại mỗi thời điểm (tháng-ngày-giờ).

**e) Gộp dữ liệu và lưu kết quả**

tk = xac\_suat.merge(mua\_tb, on='thoi\_diem', how='left')

tk = tk.fillna(0)

tk['xac\_suat\_satlo'] = (tk['xac\_suat\_satlo'] > 0).astype(int)

tk['luong\_mua\_min\_satlo'] = mua\_min

tk.to\_csv("du\_doan.csv", index=False)

* Gộp bảng xác suất và bảng lượng mưa trung bình.
* Gán thêm cột lượng mưa tối thiểu gây sạt lở (một giá trị duy nhất).
* Chuyển xác suất về dạng nhị phân (0 hoặc 1).
* Xuất kết quả ra file du\_doan.csv.

# V. Mô tả hàm dự đoán du\_doan ()

## 1. Mục đích

Hàm du\_doan () có nhiệm vụ dự đoán khả năng xảy ra sạt lở dựa vào thời gian và lượng mưa đầu vào. Hàm sử dụng dữ liệu được xử lý và lưu bởi chuan\_bi\_data ().

## 2. Đầu vào hàm

def du\_doan (gio, ngay, thang, luong\_mua, file\_path='du\_doan.csv'):

* gio, ngay, thang: thời điểm cần dự đoán. Lấy cả ngày tháng và giờ vì lượng mưa và khả năng sạt lở sẽ có đặc trưng theo mùa qua từng tháng.
* luong\_mua: lượng mưa đo được tại thời điểm đó (đơn vị mm).
* file\_path: đường dẫn file dự đoán (du\_doan.csv), mặc định là cùng thư mục.

## 3. Các bước thực hiện

**a) Tạo khóa thời gian**

thoi\_diem = f"{int(thang):02d}-{int(ngay):02d} {int(gio):02d}h"

* Chuyển thời gian đầu vào về đúng định dạng đã xử lý trước đó để truy xuất.

**b) Đọc dữ liệu dự đoán**

df = pd.read\_csv(file\_path)

row = df[df['thoi\_diem'] == thoi\_diem]

if row.empty:

return f"Khong"

* Tìm hàng tương ứng trong bảng du\_doan.csv. Nếu không có dữ liệu tại thời điểm đó → trả về "Khong" (không có dữ liệu).

**c) Lấy giá trị tham chiếu**

xac\_suat = row['xac\_suat\_satlo'].values[0]

mua\_tb = row['luong\_mua\_tb\_khi\_satlo'].values[0]

mua\_min = row['luong\_mua\_min\_satlo'].iloc[0]

* Trích xuất ba giá trị cần thiết cho dự đoán: xác suất sạt lở, lượng mưa trung bình, lượng mưa thấp nhất gây sạt lở.

**d) Dự đoán nguy cơ sạt lở**

if luong\_mua >= mua\_min:

return 'Cao.'

if xac\_suat == 0:

return 'Thap.'

if luong\_mua >= mua\_tb:

return 'Cao'

else:

return 'Thap'

Logic dự đoán như sau:

* Nếu lượng mưa hiện tại ≥ lượng mưa thấp nhất từng gây sạt lở → nguy cơ "Cao".
* Nếu xác suất sạt lở bằng 0 → chưa từng xảy ra → "Thấp".
* Nếu lượng mưa ≥ lượng mưa trung bình gây sạt lở → "Cao".
* Nếu không thỏa điều kiện nào trên → "Thấp".

## 4. Kết luận

Hàm du\_doan() sử dụng các quy tắc ngưỡng đơn giản dựa trên thống kê để đánh giá nguy cơ sạt lở, phù hợp với yêu cầu chạy nhẹ, thời gian thực trong môi trường tài nguyên hạn chế (ví dụ: thiết bị nhúng tại trạm đo).

# VI. Trực quan hóa dữ liệu trên Dashboard

## 1. Mục tiêu trực quan hóa

Sau khi dữ liệu được chuẩn bị và tiến hành dự đoán nguy cơ sạt lở, hệ thống tiếp tục thực hiện bước cuối cùng trong chu trình xử lý: trực quan hóa kết quả trên giao diện dashboard. Mục tiêu của dashboard là giúp người vận hành dễ dàng:

* Theo dõi các giá trị đo đạc theo thời gian (nhiệt độ, độ ẩm, lượng mưa…)
* Quan sát các thời điểm có nguy cơ sạt lở cao
* Đánh giá xu hướng thay đổi môi trường theo giờ, ngày
* Tra cứu bảng dữ liệu kết quả dự đoán để hỗ trợ ra quyết định kịp thời

## 2. Công nghệ sử dụng

Giao diện dashboard được xây dựng bằng HTML và JavaScript với giao diện đơn giản, dễ triển khai và dễ tích hợp trên trình duyệt hoặc các hệ thống nhúng (ví dụ như hiển thị qua màn hình LCD hoặc WebApp).

Dữ liệu hiển thị được lấy từ file du\_doan.csv – file chứa kết quả đã được xử lý và dự đoán trước đó.

## 3. Nội dung hiển thị

Dữ liệu đầu ra từ mô hình được trình bày dưới dạng bảng với các cột chính như sau:

A screenshot of a graph

AI-generated content may be incorrect.Hình 1.2: Giao diện dashboard 1

A screenshot of a computer

AI-generated content may be incorrect.

Hình 1.3 Giao diện dashboard 2

## 5. Kết luận

Giao diện dashboard là phần quan trọng giúp chuyển hóa dữ liệu kỹ thuật thành thông tin có thể hành động, hỗ trợ cán bộ quản lý rừng đánh giá nhanh tình trạng môi trường và nguy cơ sạt lở tại từng thời điểm cụ thể. Việc sử dụng HTML giúp đảm bảo tính đơn giản, dễ bảo trì, có thể mở rộng tích hợp thêm bản đồ, biểu đồ, hoặc các công cụ cảnh báo tự động.

# VII. Đề xuất mở rộng

Mặc dù hệ thống hiện tại đã hoàn thiện quy trình thu thập, xử lý, dự đoán và hiển thị dữ liệu dựa trên các cảm biến môi trường, vẫn còn nhiều hướng mở rộng tiềm năng để nâng cao hiệu quả và độ chính xác của hệ thống. Một số đề xuất mở rộng bao gồm:

## 1. Mở rộng tập dữ liệu và tăng độ chính xác dự đoán

* Thu thập thêm dữ liệu từ nhiều vùng rừng, đặc biệt là các khu vực thường xuyên xảy ra sạt lở, để tăng tính tổng quát của mô hình.
* Bổ sung các yếu tố địa hình như độ dốc, loại đất, độ bão hòa nước trong đất, chỉ số thực vật (NDVI) từ ảnh vệ tinh.
* Thu thập thêm dữ liệu về sạt lở. Từ đó giảm tình trạng mất cân bằng dữ liều rồi nâng cấp áp dụng từ phương pháp thống kê ngưỡng đơn giản sang các mô hình học máy như Random Forest, XGBoost hoặc mạng nơ-ron sâu để cải thiện khả năng dự đoán.
* Kết hợp dữ liệu từ nhiều nguồn như nhiệt độ, độ ẩm, tốc độ gió, độ dốc của chất, chất liệu địa chất,….

## 2. Tích hợp bản đồ số (GIS)

* Hiển thị dữ liệu dự đoán trực tiếp trên bản đồ địa lý kết hợp tọa độ GPS của các trạm cảm biến.
* Dùng màu sắc cảnh báo (xanh-vàng-đỏ) để phân vùng mức độ nguy hiểm theo thời gian thực.
* Hỗ trợ cán bộ quản lý rừng ra quyết định nhanh tại những vị trí nguy cơ cao.

## 3. Hệ thống cảnh báo sớm

* Kết nối với các hệ thống gửi cảnh báo qua tin nhắn SMS, email, hoặc zalo bot API khi phát hiện nguy cơ sạt lở cao tại một điểm cụ thể.
* Thiết lập các ngưỡng cảnh báo động theo mùa (ví dụ mùa mưa, mùa khô) hoặc thời gian thực.

## 4. Tối ưu phần cứng cảm biến

* Sử dụng các mô-đun truyền thông tầm xa như LoRa, NB-IoT thay vì HC-12 để tăng độ ổn định và giảm tiêu thụ năng lượng.
* Thiết kế mạch cảm biến hoạt động siêu tiết kiệm điện kết hợp với pin mặt trời, giúp hệ thống vận hành lâu dài tại khu vực không có nguồn điện.

## 5. Truy cập từ xa và điều khiển tập trung

* Tạo giao diện quản trị có thể truy cập qua trình duyệt từ xa, phân quyền người dùng, tải về dữ liệu CSV, cấu hình hệ thống và xem lại lịch sử cảnh báo.
* Cho phép cập nhật firmware từ xa cho các node cảm biến.

# Kết luận đề xuất

Những đề xuất mở rộng trên nhằm hướng tới xây dựng một hệ thống quản lý rừng cảnh báo sạt lở thông minh, tự động, chính xác, dễ triển khai và dễ bảo trì, phù hợp với nhu cầu thực tiễn tại các vùng rừng núi Việt Nam. Hệ thống không chỉ phục vụ công tác quản lý rừng mà còn có thể nhân rộng để ứng dụng trong cảnh báo thiên tai, nông nghiệp thông minh, và giám sát môi trường.