BHARATIYA ANTARIKSH HACKATHON 2025

Powered by H2S

Team Name :VYOMIANS

Team Leader Name : Athira S

Problem Statement : IDENTIFYING HALO CME EVENTS BASED ON PARTICLE DATA FROM SWIS-ASPEX PAYLOAD ONBOARD ADITYA-L1

Team Members

Team Leader:

Name: Athira S
College: Cochin University of Science and Technology

Team Member-1:

Name: Irene Sara Philipose
College:Cochin University of Science and Technology

Team Member-2:

Name:Adith R Lal
College:Cochin University of Science and Technology

Team Member-3:

Name:Ayana Anil
College:Cochin University of Science and Technology

Coronal Mass Ejections (CMEs) are among the most powerful solar phenomena, capable of impacting the near-Earth space environment and various technological systems. These massive ejections of solar plasma and magnetic fields can induce geomagnetic storms, disrupt satellite operations, impair GPS and radio communication, and even affect terrestrial power grids. Our project proposes a data-driven approach to modeling and predicting CME behavior using mission data from the Indian Space Science Data Centre (ISSDC), enabling proactive risk mitigation for both space- and ground-based assets.

SCIENTIFIC BACKGROUND

CMEs arise from the sudden release of magnetic energy stored in the Sun’s corona. This typically occurs through a process known as magnetic reconnection, wherein oppositely directed magnetic field lines realign and release vast quantities of energy.

Key Parameters to Monitor:

- Velocity (V): Determines transit time from Sun to Earth.
- Magnetic Field Strength (Bz): Negative Bz values (southward direction) lead to strong geomagnetic storms.
- Density & Temperature: Influence CME dynamics and detectability.

The Indian Space Science Data Centre (ISSDC), operated by ISRO, is the primary archive for space science mission data. For this project, ISSDC provides access to mission-critical datasets from Aditya-L1, India’s first dedicated solar mission, which carries seven payloads designed for comprehensive solar observation.

Data Access and Utility

- Formats: FITS, CSV, and netCDF.
- Levels: Raw (Level 0), Calibrated (Level 1), and Science-Ready (Level 2/3).
- APIs and Documentation: Provided via ISSDC website.

How We Will Use This Data:

- Training Predictive Models: Historical data from VELC, MAG, and SoLEXS will be used to train ML-based classification and forecasting models.
- Real-Time Tracking: Use near-real-time data from Aditya-L1 to detect CME events and forecast Earth impact.
- Event Correlation: Cross-reference between X-ray, UV, magnetic, and particle observations for holistic analysis. This multi-payload approach ensures a complete understanding of CME lifecycles

THE IDEA:

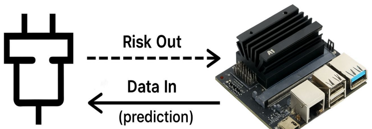
EDGE CME- Nano- Node AI on Ground for Local Grid Protection

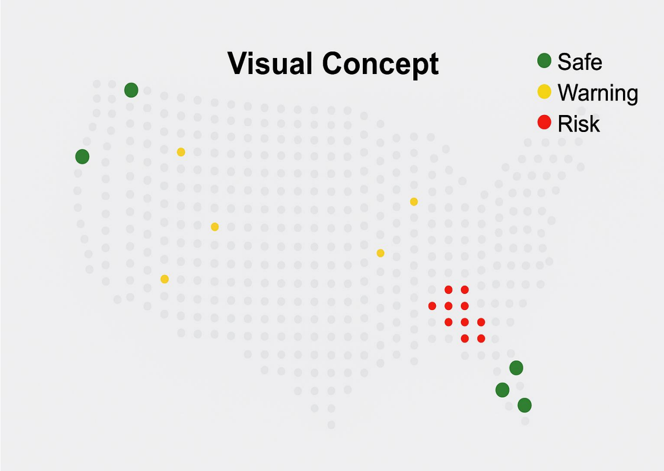
Current centralized CME alert systems frequently issue generalized or delayed warnings, making defence communication system, railroads, air traffic control, and regional power grids susceptible to unexpected geomagnetic disruptions. Power outages, damaged transformers, and interrupted communication infrastructure can all be consequences of these delays. The problem is solved by Edge CME's decentralized, artificial intelligence-driven nano-node system, which establishes direct local connections with radiation sensors and ground-based magnetometers. These nano-nodes enable immediate protective reactions by using real-time particle data and AI models trained to identify early indicators of CME-induced geomagnetic storms at the edge locally.

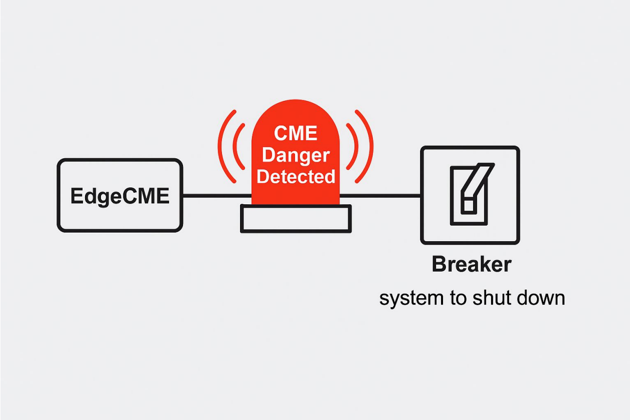
Why EdgeCME?

1. Mini Inference Engines at the Edge of AI: Small AI models on edge devices (such as Jetson Nano or Raspberry Pi) situated at strategic grid points like such as power plants and substations. Aditya-L1's SWIS-ASPEX particle data is used to train the AI model. There is no need to wait for a central server because it operates locally. It detects anomalous increases in the magnetic field or particle radiation.It has decentralized detection,no cloud lag or internet delay. It functions even in the event of partial communication outages. Easily expandable to thousands of substations. It is reliable and scalable
2. Sensor Fusion for Localized CME Prediction: To generate a unique risk profile for every site, we integrate SWIS-ASPEX particle data with local magnetometer readings. How Real-time fusion of space and ground data is possible with AI. It determines whether that particular substation will be impacted by the current CME event and assign a severity score (Safe, Warning, Danger, etc.). It provides a hyper-local prediction,a location-specific alert rather than a general one.Improves power grid operators' accuracy and actionability and allows for proactive decision-making prior to the storm.

3. Damage Prevention through Automatic Signal Breaker Control:When a hazardous CME is anticipated, our system has the ability to immediately send emergency shutdown signals to transformers.At the substation, connect to the SCADA system or signal relay. It sends a breaker command to cut off or reroute power flow when danger is anticipated.It stops equipment damage, overheating, and surges.It eliminates human delays in responding to crises. helps safeguard electrical equipment valued at millions of dollars,lays the groundwork for automated systems that protect against solar storms.







THE SOLUTION

Step 1: Retrieving particle Information from SWIS-ASPEX Aditya-L1's SWIS-ASPEX payload provides real-time particle data (such as proton/electron flux) to the EdgeCME system. Early indications of the formation of a Halo CME event are provided by this data.

Step 2: AI-Powered Early CME Pattern Detection We train a lightweight machine learning model to identify patterns that resemble Halo CME signatures. It continuously tracks real-time SWIS-ASPEX data. It immediately marks an anomaly or suspicious spike as a "CME Risk."

Step 3: Auto signal breaker system to protect transformers during peak solar storms. Local Grid Nodes for Edge AI Inference. EdgeCME installs AI software at power substations directly rather than waiting for central agencies. Control rooms for rail signals Towers for communication at airports With nearly no delay, these local AI nodes analyse the data locally .

USP OF THE SOLUTION:

- Architecture Decentralized This system employs edge-based nano-nodes at regional locations to detect and react to solar disturbances instantly, as opposed to depending on centralized ISRO CME alerts, which may be delayed or generalized.
- Local Analysis Driven by AI Based on information from ground-based sensors such as radiation and magnetometers, each node employs lightweight machine learning models to infer possible CME risks in real time.
- Direct Communication with the Local Grid In order to enable automated protective actions without waiting for centralized instructions, the system directly interfaces with air traffic control, defense communication infrastructure, railway systems, and local power grid controllers.
- Both scalable and resilient Because of their low cost, modular design, and scalability, nano-nodes can be deployed in a variety of crucial geographic locations, particularly in remote or vulnerable rural areas.
- Dashboard in Real Time and Mobile App Real-time data visualization via a dashboard that responds and enables early warnings and monitoring of solar activity threats for engineers and operators.
- ISRO-Independent but Integrated The central delay or blackout issue is resolved by the core system operating independently, even though it can use ISRO space weather APIs for validation.
- Prepared for disasters and failsafe During a CME event, local action guarantees quicker grid isolation, safeguarding power plants, transformers, and satellite communications, lowering risk, expense, and downtime.

Key features

1. Real-Time Halo CME Detection Analyzes live particle flux data from Aditya-L1's SWIS-ASPEX uses AI/ML models to identify early-stage Halo CME patterns. monitors variations in ion, electron, and proton activity.

2. Edge AI-Powered Inference Engine ML models that are lightweight and installed on edge devices, such as the Raspberry Pi and Jetson Nano permits local substations and control centers to make predictions in real time. It operates offline when there is a communication outage.

3. Combining Sensors with Ground Data combines satellite data with local magnetometer sensor readings. Creates CME risk scores based on location. Increases the relevance and accuracy of predictions

4.Real-time Risk Management System Web-based user interface for visualizing: Real-time CME data feed Levels of severity (Safe, Warning, Critical) Impact likelihood at different locations shows action logs, prediction confidence, and time-series graphs.

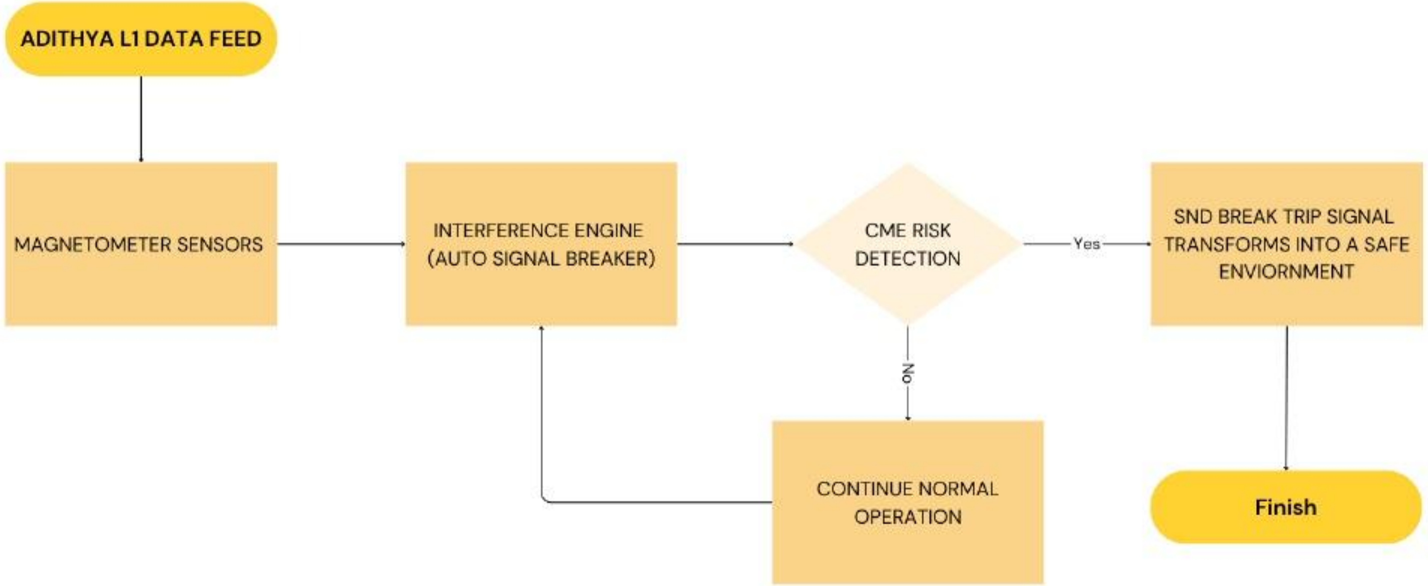
5. Self-Sustained Safety Measures transmits instructions to isolate grid segments or activate circuit breakers. allows for automated response through SCADA integration. lowers the possibility of cascading failures or transformer damage

6. Alert System with Multiple Channels Instant alerts through Telegram, SMS, or email Adjustable thresholds to notify various stakeholders enables the escalation of alerts (Operator → Grid Manager → Control Center).

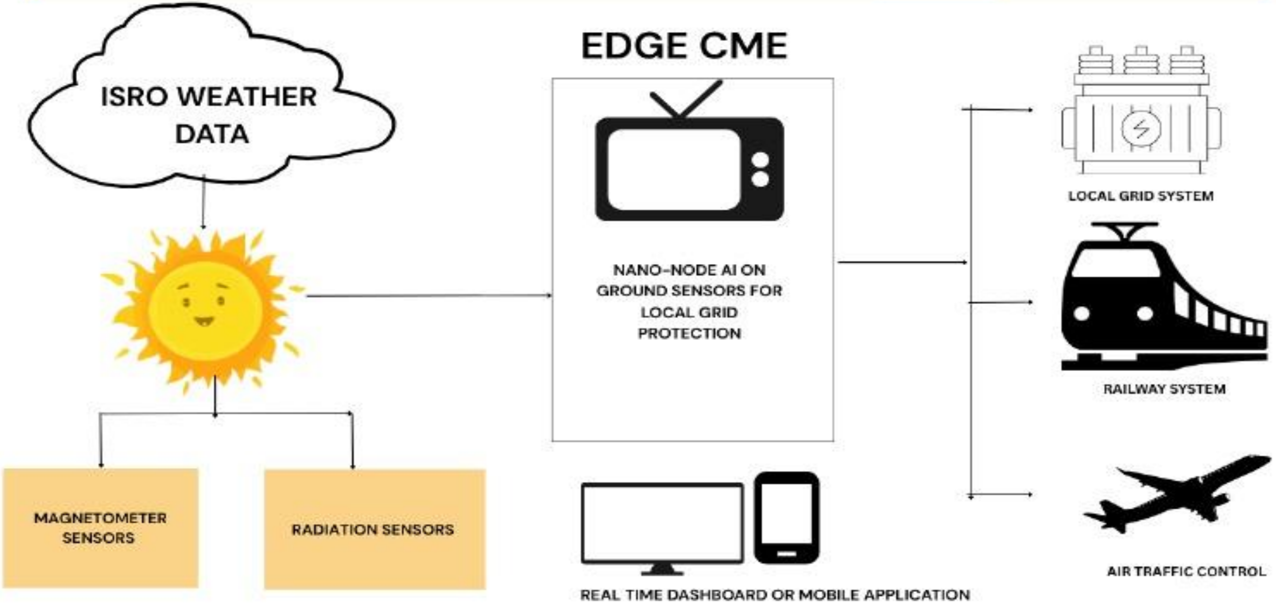
7. Modular and Scalable Architecture Developed to facilitate deployment over thousands of nodes Every edge node operates separately or in concert with the cloud. Replicating, updating, and remotely monitoring is simple.

8. Safety and Dependability In the event of a cloud disconnect, local fallback models Data logs that are resistant to tampering (with the potential to be expanded via blockchain) Sensor communication encryption is optional

Workflow Diagram



MOCK DIAGARAM



AI BASED MINI INTERFERENCE ENGINE TO DETECT CME

Powered by

CME AI.py > read_satellite_data

```python
import time
import numpy as np
import joblib

# STEP 1: Load Pre-trained Model

try:
    model = joblib.load("cme_inference_model.pkl")
    print(" Model loaded successfully")
except Exception as e:
    print(f"Error loading model: {e}")
    exit()

# STEP 2:Simulated Sensor Inputs

def read_magnetometer():
    # Simulate magnetic field readings data
    return [25.1, 48.3, -17.2]

def read_satellite_data():
    # Simulate particle data from Aditya-L1 (electron & proton flux) data
    return [1200.0, 950.0]


# STEP 3:Inference Loop creating

def run_inference_loop():
    while True:
        try:
            # input simulation
            local_mag = read_magnetometer()      # 3 values
            particle_flux = read_satellite_data()  # 2 values

            # Combining all input features
            input_features = np.array(local_mag + particle_flux).reshape(1, -1)

            # prediction running code
            prediction = model.predict(input_features)

            # result display code
            if prediction[0] == 1:
                print(" ALERT: CME Risk Detected! Be alerted!!!")
            else:
                print(" All Clear: No CME Threat detectded")

            time.sleep(3)

        except KeyboardInterrupt:
            print("\n Inference loop stopped by user.")
            break

# STEP 4: Run the Engine

if __name__ == "__main__": ...
# time to time printing happens here
```

SIGNAL BREAK CONTROL SYSTEM

Powered by H2S

```python

ELECTRON_FLUX_THRESHOLD = 1300  # e/cm²·s
PROTON_FLUX_THRESHOLD = 1100    # p/cm²·s
MAG_VARIATION_THRESHOLD = 26    # nanoTesla


# simulation of sensor functions with data

def read_satellite_data():
    # solar flux readings
    electron_flux = np.random.normal(1150, 200)
    proton_flux = np.random.normal(1000, 150)
    return electron_flux, proton_flux

def read_magnetometer():
    # Simulate magnetic field (X, Y, Z) in nT
    mag_x = np.random.normal(25, 3)
    mag_y = np.random.normal(-45, 3)
    mag_z = np.random.normal(5, 1)
    return mag_x, mag_y, mag_z

# Breaker Trigger Function

def trip_breaker():
    print(" BREAKER TRIPPED: Transformer Isolated")
    # On real Raspberry Pi, use GPIO output here
    # Example: GPIO.output(BREAKER_PIN, GPIO.HIGH)

def restore_breaker():
    print("BREAKER RESTORED: Normal Operation")
    # On real Pi: GPIO.output(BREAKER_PIN, GPIO.LOW)


# Decision

def detect_cme_risk(e_flux, p_flux, mag_x, mag_y, mag_z):
    total_flux = e_flux + p_flux
    mag_variation = max(abs(mag_x), abs(mag_y), abs(mag_z))

    if (e_flux > ELECTRON_FLUX_THRESHOLD and
        p_flux > PROTON_FLUX_THRESHOLD and
        mag_variation > MAG_VARIATION_THRESHOLD):
        return True
    return False

# Main Inference Loop

def run_auto_protection():
    try:
        while True:
            e_flux, p_flux = read_satellite_data()
            mag_x, mag_y, mag_z = read_magnetometer()

            print(f"\nElectron: {e_flux:.2f}, Proton: {p_flux:.2f}, Mag_XYZ: {mag_x:.2f}, {mag_y:.2f}, {mag_z:.2f}")

            if detect_cme_risk(e_flux, p_flux, mag_x, mag_y, mag_z):
                trip_breaker()
            else:
                restore_breaker()

            time.sleep(3)

    except KeyboardInterrupt:
        print("\n System Stopped by User")


# Start the system

if __name__ == "__main__":
    run_auto_protection()
  # time to time looped printing
```

Technologies used:

Frontend:

React Core is a framework for creating dynamic, interactive user interfaces. **SSR/SSG** for performance and the **Next.js React** framework with routing. **TypeScriptStrong** typing for increased safety and developer experience. For quick and responsive styling, use **Tailwind CSS**, which prioritizes usability. **Chart.jsfor** sensor data graphs in real time.**Motion of the framer** for animations and to improve user interface.

Verification & Onboarding:

Firebase Auth securely manages user sign-up, login and session administration.

Backend + Database:

Firebase Auth securely handles session administration, login, and user sign-up.

Application Programming Interface: ISRO Aditya-L1 Data

Hardware:

Raspberry Pi + Sensor : For real-world implementation of edge sensing and inference.

Estimated implementation cost :12000

HARDWARE	COST
Raspberry Pi(microcontroller)	2500
Ground Based Sensor	4000
Power supply	1000
Casing	500
TOTAL	8000

SOFTWARE	COST
Frontend Next.js	-
Backend Node.js	-
API ISRO SOLAR DATA	-
Firebase hosting	1200
Edgecme.in	800

DESIGN	COST
UI/UX	-

TESTING AND VALIDATION	COST
Local field setup(travel)	1000

MISCELLANEOUS(DATA)	1000

TOTAL COST = 12000

This foundationaSl research provides the necessary inputs for subsequent data modelling, feature extraction, and real-time prediction systems, ultimately contributing to India’s space weather preparedness and operational safety in space.

The ability to predict Coronal Mass Ejections and assess their Earth-directed impact is crucial for building space-weather-resilient systems. Leveraging the cutting-edge observational data from Aditya-L1 via ISSDC, and grounded in well-established CME physics, our project proposes a scientifically robust approach to forecasting solar eruptions using modern data analytics and machine learning techniques.

Powered by H2S

BHARATIYA ANTARIKSH HACKATHON 2025

THANK YOU