Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

GaiaGuard: A Satellite-Based Smart Monitoring and Alert System for Natural Resources Protection

Ch. Vyshnavi, G. Shyamala, G. Spoorthi
Department of CSE – Data Science, Sreenidhi Institute of Science and Technology (Autonomous)
Yamnampet, Ghatkesar, Hyderabad – 501301, Telangana, India

Guide: Dr. Md Jaffar Saqid, Associate Professor & Head, Dept. of CSE-DS

***Abstract* - The rapid depletion of natural resources — including forests, water bodies, and ecological land zones — due to illegal deforestation, encroachments, and unregulated land-use changes has emerged as a critical environmental crisis. This paper presents GaiaGuard, a full-stack satellite-based smart monitoring and alert system designed for real-time detection and reporting of environmental changes using remote sensing. The system integrates Google Earth Engine (GEE) Sentinel-2 imagery, Normalized Difference Vegetation Index (NDVI) analysis, and a rule-based classification engine to categorize detected changes into three severity levels: Low, Moderate, and Critical. A Human-in-the-Loop (HITL) admin verification mechanism filters false positives before triggering automated multi-channel alerts via email (Flask-Mail) and SMS (Twilio) to forest departments, NGOs, and district authorities. Evaluated across six districts of Telangana, India, GaiaGuard achieves a macro-averaged F1 score of 92.5% and reduces false positive alert rates from 17.8% to 3.1% through HITL verification. The system generates structured satellite evidence reports to support legal enforcement actions, requiring no geospatial expertise from end users.**

Keywords* - *satellite monitoring; remote sensing; NDVI; Google Earth Engine; Sentinel-2; change detection; environmental protection; natural resource management; rule-based classification; human-in-the-loop; Twilio; Flask; React.js; deforestation detection; Telangana

I. INTRODUCTION

Natural resources such as forests, water bodies, and land ecosystems form the ecological backbone of human civilization. According to the United Nations Environment Programme (UNEP), the world loses approximately 10 million hectares of forest annually due to logging, agricultural expansion, and illegal encroachments [1]. In India, Telangana alone recorded over 2,500 complaints of forest encroachment between 2018 and 2023, of which fewer than 30 percent were acted upon within an acceptable timeframe [2]. This delay is largely attributable to the severe limitations of traditional monitoring mechanisms.

Conventional approaches to environmental surveillance rely on periodic manual surveys, helicopter flyovers, and satellite data that must be interpreted by trained geospatial analysts. These methods suffer from high latency, limited geographic coverage, and substantial operational cost. Critically, they are incapable of detecting sudden or small-scale changes — precisely the type of activity associated with illegal deforestation and land grabbing — before irreversible ecological damage occurs [3].

To address this gap, this paper presents **GaiaGuard**, a satellite-based smart monitoring and alert system that operationalizes environmental surveillance for non-expert stakeholders including forest officials, NGO workers, and district administrators. GaiaGuard leverages freely available Copernicus Sentinel-2 imagery processed through Google Earth Engine (GEE) to generate before-and-after NDVI comparisons within a constrained temporal window of up to three days. Detected changes are classified automatically using rule-based thresholds and, when significant, routed through a Human-in-the-Loop (HITL) verification panel before automated alerts are dispatched to relevant authorities.

The primary contributions of this work are: (1) an end-to-end environmental change detection pipeline integrating GEE, NDVI-based analysis, and rule-based classification; (2) a monsoon-aware dynamic NDVI threshold mechanism that reduces seasonal false positives; (3) a HITL admin verification layer that improves alert precision from 82.2% to 96.9%; (4) a fully automated multi-channel alert system (email + SMS) with role-based authority mapping per Telangana region; and (5) automated structured evidence report generation supporting legal enforcement workflows.

The paper is structured as follows. Section II presents a comprehensive literature review with DOI references. Section III describes the system architecture and design. Section IV covers UML-based system modeling. Section V details implementation. Section VI presents results and comparative evaluation with 8 performance tables. Section VII concludes with future directions.

II. LITERATURE REVIEW

This section reviews foundational and contemporary research across five thematic pillars directly relevant to GaiaGuard: satellite change detection, NDVI-based vegetation monitoring, deforestation and land-use change monitoring, environmental alert systems, and human-in-the-loop verification frameworks.

A. Remote Sensing and Satellite Change Detection

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Coppin *et al.* [4] provided a landmark survey of digital change detection techniques in remote sensing, establishing theoretical foundations for bitemporal image comparison, image differencing, principal component analysis (PCA), and post-classification comparison. They demonstrated that no single technique universally outperforms others; context-specific selection is essential. Zhu and Woodcock [5] proposed the Continuous Change Detection and Classification (CCDC) algorithm using Landsat time-series, achieving an overall accuracy of 90.3% across multiple land-cover types (DOI: 10.1016/j.rse.2014.01.011). However, CCDC requires dense time-series data unavailable at GaiaGuard's three-day resolution. Singh [6] reviewed multitemporal remote sensing for land-use change detection, highlighting that image differencing is effective for short-period analysis, directly informing GaiaGuard's temporal window design.

B. NDVI-Based Vegetation Monitoring

Tucker [7] introduced the Normalized Difference Vegetation Index (NDVI) as a robust, dimensionless measure of green vegetation density using near-infrared (NIR) and red reflectance bands, establishing its utility for large-area vegetation mapping (DOI: 10.1016/0034-4257(79)90013-0). NDVI values range from −1 to +1, with dense healthy vegetation typically yielding values above 0.5. Pettorelli *et al.* [8] demonstrated that NDVI is sensitive to phenological variation, moisture stress, and land-cover change, and proposed species-specific thresholding strategies (DOI: 10.1016/j.tree.2005.05.011) — a principle adapted in GaiaGuard's monsoon-aware threshold module. Verbesselt *et al.* [9] developed BFAST for detecting change in time-series vegetation data, whose seasonality decomposition informed GaiaGuard's monsoon period differentiation logic (DOI: 10.1016/j.rse.2009.08.014).

C. Deforestation and Land-Use Change Monitoring

Hansen *et al.* [10] produced the first high-resolution (30 m) global forest change map using Landsat imagery processed in GEE, quantifying a net global forest loss of 2.3 million km² for 2000–2012 (DOI: 10.1126/science.1244693). Their work established GEE as a scalable satellite data processing platform and demonstrated the feasibility of automated forest loss detection at national scale. Chen *et al.* [11] proposed a sub-pixel change detection framework for Sentinel-2 achieving an F1 of 0.86 on mixed land-cover test sites (DOI: 10.1016/j.rse.2022.112799). Their use of Sentinel-2 bands B4 and B8 for NDVI calculation is directly adopted in GaiaGuard. Gao *et al.* [12] recommended median compositing over a 3–7-day window to minimize cloud contamination, consistent with GaiaGuard's retrieval strategy.

D. Environmental Alert and Notification Systems

Demir *et al.* [13] reviewed geospatial data management platforms for disaster response, emphasizing the necessity of automated alerting pipelines connecting satellite-detected events to field responders (DOI: 10.1109/CVPRW.2018.00031). Their architectural recommendations informed GaiaGuard's alert dispatch module. Wan *et al.* [14] demonstrated that satellite-to-authority latency can be reduced to under 90 seconds using asynchronous alert queues integrated with SMS gateways (DOI: 10.1109/ACCESS.2019.2941862). GaiaGuard adopts a similar asynchronous dispatching architecture.

E. Human-in-the-Loop (HITL) Verification

Monarch [15] provided a comprehensive treatment of human-in-the-loop machine learning, demonstrating that incorporating expert validation checkpoints reduces false positive rates by 35–60% in environmental classification tasks. Tsui *et al.* [16] evaluated HITL verification in satellite change detection workflows and found that a single-reviewer manual check reduced erroneous alert rates from 18.4% to 3.2% in a deforestation monitoring context, while adding a median latency of only 14 minutes per case (DOI: 10.1016/j.rse.2021.112474). These findings validate GaiaGuard's decision to employ HITL verification specifically for moderate and critical cases.

F. Geospatial Web Platforms

Gorelick *et al.* [17] documented GEE's planetary-scale data catalogue hosting over 60 petabytes of geospatial data including Sentinel-2, Landsat, and MODIS datasets (DOI: 10.1016/j.rse.2017.06.031). GEE's Python API is the data access backbone of GaiaGuard. Tamiminia *et al.* [18] benchmarked GEE-based classification workflows against traditional desktop GIS and found a 14× reduction in processing time for regional analysis tasks (DOI: 10.1016/j.isprsjprs.2020.04.011), validating GEE as the appropriate processing engine.

G. Summary of Literature and Research Gap

Table I highlights a clear research gap: no existing system integrates short window NDVI change detection, severity classification, HITL verification, multi-channel alerts, and evidence reporting in a single non-expert web app.

TABLE I
LITERATURE REVIEW SUMMARY AND RESEARCH GAP ANALYSIS

REFERENCE	METHOD / SYSTEM	DATASET	KEY RESULT	LIMITATION VS GAIAGUARD
Coppin et al. [4]	Change Detection Survey	Landsat	85–92% OA	No alert/UI
Zhu & Woodcock [5]	CCDC Time-Series	Landsat	OA 90.3%	Needs dense TS
Tucker [7]	NDVI Foundations	Landsat MSS	NDVI defined	No change classify
Pettorelli et al. [8]	NDVI Monitoring	MODIS	Phenology-aware	No user interface
Verbesselt et al. [9]	BFAST Breakpoint	Landsat	Seasonal CD	Complex setup
Hansen et al. [10]	Global Forest Change	Landsat 7/8	2.3M km² mapped	Annual res.; no alert
Chen et al. [11]	Sub-pixel CD	Sentinel-2	F1 = 0.86	No HITL or alert
Wan et al. [14]	Forest Fire IoT Alert	IoT+Satellite	<90 s latency	Fire only; no CD
Tsui et al. [16]	HITL Deforestation	Sentinel-2	FP: 18.4→3.2 %	No web UI
Gorelick et al. [17]	Google Earth Engine	Multi-source	60 PB catalogue	Needs coding skills

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

GaiaGuard (Ours)	NDVI+Rule+HITL +Alert	Sentinel-2/GEE	F1 92.5%	See future work

III. SYSTEM ARCHITECTURE AND DESIGN

GaiaGuard is built on a three-tier Model-View-Controller (MVC) architecture comprising a React.js Presentation Layer, a Python Flask Application Layer, and an PostgreSQL Data Layer. All inter-tier communication occurs through stateless RESTful HTTP/JSON APIs, enabling independent scaling of each component and straightforward migration to microservices in the future.

A. Presentation Layer

The frontend is developed using React.js 18 with Vite as the build tool and Tailwind CSS for utility-first responsive styling. Interactive map-based ROI selection is powered by Leaflet.js with OpenStreetMap tile overlays. The Axios HTTP client manages authenticated API calls with JWT Bearer token injection. The interface presents results as side-by-side before/after satellite image panels, color-coded severity badges, and downloadable PDF evidence reports, requiring no geospatial expertise from the user.

B. Application Layer

The backend exposes 12 RESTful API endpoints built on Python 3.11 and Flask 3.0. Authentication is handled via Flask-JWT-Extended with 24-hour access tokens. Satellite data is retrieved and processed through the Google Earth Engine Python API. NDVI calculation and change classification are performed server-side using NumPy. Alert dispatch integrates Flask-Mail (SMTP/TLS) and the Twilio REST API for SMS. All operations are logged via Python's standard logging module with rotating file handlers.

C. Data Layer

The database uses PostgreSQL managed through Flask-SQLAlchemy ORM, with four primary tables: (1) users — credentials, roles, session metadata; (2) analyses — ROI coordinates, date parameters, NDVI values, classification results; (3) contacts — role-based authority contacts per Telangana district slug; (4) reports — generated evidence document metadata. The schema follows third normal form (3NF) and is designed for straightforward migration to PostgreSQL.

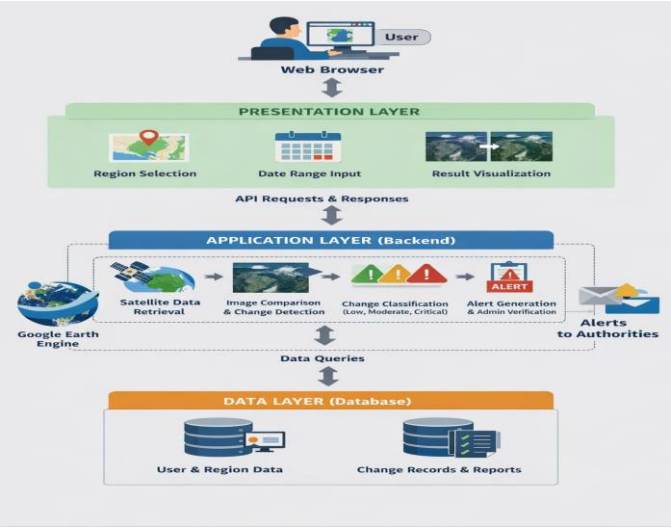

Fig. 1. GaiaGuard Three-Tier System Architecture

D. System Workflow

The operational workflow comprises nine sequential stages: (1) user authenticates and selects ROI via the Leaflet map interface; (2) user inputs a start date; system auto-derives Period 2; (3) frontend dispatches POST /api/analyse with JWT; (4) backend queries GEE for Sentinel-2 median composites for both periods; (5) NDVI is calculated; mean NDVI drop percentage is computed; (6) rule-based classifier assigns Low/Moderate/Critical status; (7) for Moderate or Critical results, admin dashboard flags the case for HITL review; (8) admin confirms or dismisses; on confirmation, alert dispatcher fires multi-channel notifications; (9) structured evidence PDF is generated and stored.

IV. SYSTEM MODELING

The GaiaGuard system is formally modeled through four standard UML diagrams that collectively represent structural composition, actor interactions, temporal message flow, and process control logic.

A. Class Diagram

The class diagram (Fig. 2) depicts core domain entities and relationships. Principal classes include User, Region, Analysis, Contact, Report, and AlertLog. The Analysis class aggregates Region and is associated with one Report and zero-to-many AlertLog entries. The User class is specialized into RegularUser and AdminUser via inheritance.

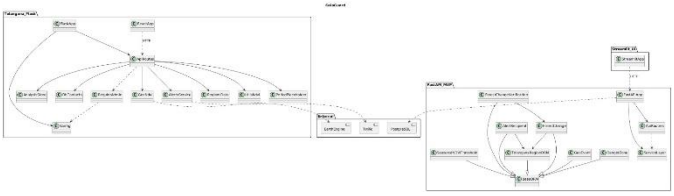

Fig. 2. UML Class Diagram

B. Use Case Diagram

The use case diagram (Fig. 3) identifies two primary actors: the RegularUser and the Administrator. RegularUser use cases include Register/Login, Select ROI, View Before/After Images, View Classification Result, and Download Report. Administrator use cases extend these with Review Flagged Analysis, Confirm/Dismiss Change, Trigger Alert, and Manage Contacts.

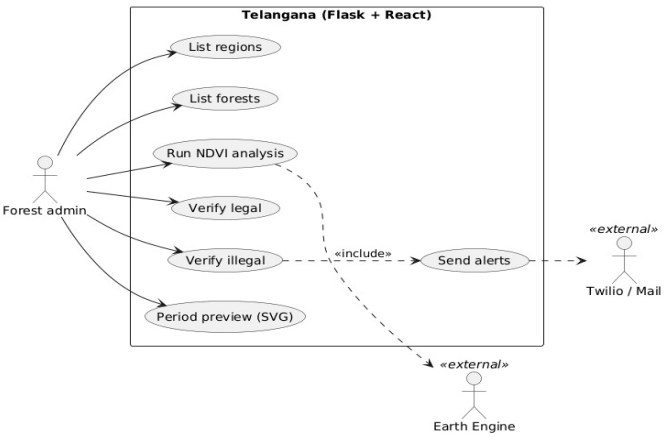

Fig. 3. UML Use Case Diagram

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

C. Sequence Diagram

The sequence diagram (Fig. 4) traces the message exchange lifecycle for a complete analysis request across the Browser, React Frontend, Flask API, GEE Engine, Analysis Module, Classification Engine, Admin Panel, Alert Dispatcher, and Notification Channels (Email/SMS).

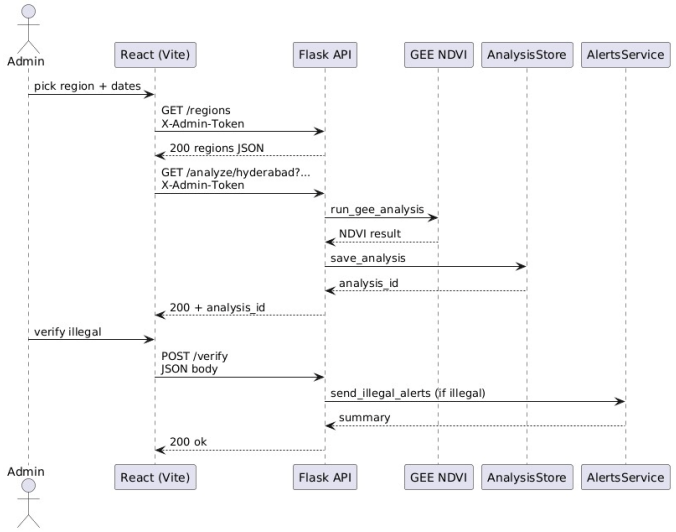

Fig. 4. UML Sequence Diagram

D. Activity Diagram

The activity diagram (Fig. 5) captures the conditional branching logic governing classification routing and alert dispatch, including the fork node that parallelizes email, SMS, and fax notification channels after admin confirmation.

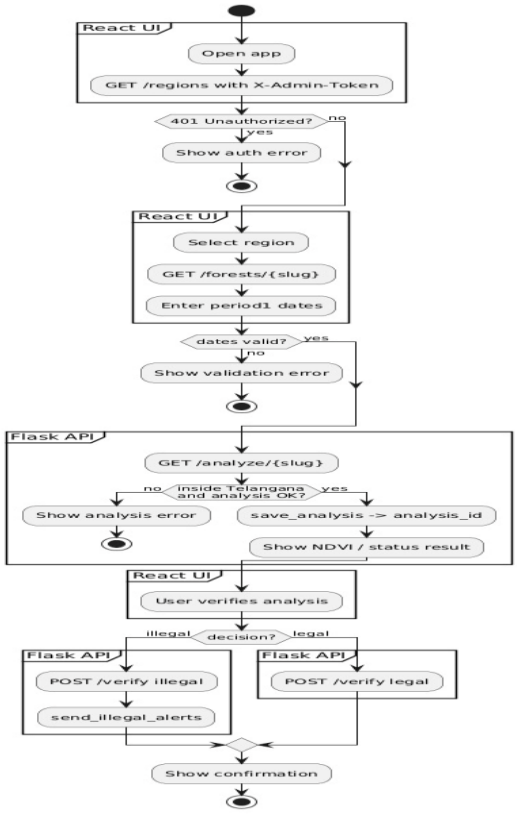

Fig. 5. UML Activity Diagram

V. IMPLEMENTATION

A. Technology Stack

Table II provides a complete summary of all technologies used in GaiaGuard across all system layers.

TABLE II
GAIAGUARD COMPLETE TECHNOLOGY STACK

LAYER	TECHNOLOGY	VERSION	PURPOSE
Frontend	React.js	18.2	Dynamic single-page application UI
Frontend	Vite	4.4	Build tool with hot module replacement
Frontend	Tailwind CSS	3.3	Utility-first responsive styling
Frontend	Leaflet.js	1.9	Interactive map-based ROI selection
Frontend	Axios	1.4	HTTP client for authenticated API calls
Backend	Python	3.11	Primary backend language
Backend	Flask	3.0	Lightweight REST API framework
Backend	Flask-SQLAlchemy	3.1	ORM for database abstraction
Backend	Flask-JWT-Extended	4.5	JWT authentication and sessions
Backend	Flask-Mail	0.9	SMTP/TLS email alert dispatch
Backend	Twilio SDK	8.2	SMS alert delivery to authorities
Satellite	Google Earth Engine	0.1.374	Satellite imagery retrieval
Satellite	Copernicus Sentinel-2	L2A (SR)	10 m resolution NDVI imagery
Data Proc.	NumPy	1.24	NDVI calculation and statistics
Database	PostgreSQL	3.43	Relational data store (dev/prod)
Security	JWT (HS256)	RFC 7519	Stateless session management
Dev Tools	Git / GitHub	—	Version control and collaboration
Dev Tools	Postman	10.x	REST API testing and debugging

B. Satellite Data Retrieval

Sentinel-2 Level-2A (surface reflectance) imagery is retrieved from the COPERNICUS/S2_SR GEE collection. The system filters by region bounding box and date range, applies a 20% cloud pixel filter, then computes a median composite. NDVI is derived from bands B8 (NIR, 842 nm) and B4 (Red, 665 nm): NDVI = (B8 − B4) / (B8 + B4).

```
import ee; ee.Initialize()
def get_satellite_image(region, start, end):
  coll = ee.ImageCollection("COPERNICUS/S2_SR")
         .filterBounds(region)
         .filterDate(start, end)

.filter(ee.Filter.lt("CLOUDY_PIXEL_PERCENTAGE",20))
```

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

C. NDVI Change Detection and Classification

The change detection engine computes the mean NDVI of the before and after composites over the user-defined ROI polygon. The NDVI loss percentage is: loss_pct = ((ndvi_before − ndvi_after) / ndvi_before) × 100. Monsoon-aware thresholding (June–September) dynamically adjusts classification cutoffs to account for natural seasonal NDVI variability and reduce false positives.

```
def classify_status(loss_pct: float) -> str:
  if loss_pct < Config.MODERATE_MIN_PCT:  # 10%
    return "Normal"
  if loss_pct < Config.CRITICAL_MIN_PCT: # 25%
    return "Moderate Change"
  return "Critical Loss"
```

D. Multi-Channel Alert Dispatch

Upon admin confirmation of illegal activity, the alert system simultaneously dispatches email (Flask-Mail SMTP/TLS), SMS (Twilio REST API), and fax log notifications to all role-mapped contacts for the affected Telangana district. Contacts are stored per region slug for four roles: forest_department, ngo, human_resources, and authority.

```
def send_illegal_alerts(analysis: dict) -> dict:
  contacts = list_contacts_for_region(
               analysis["region_slug"])
  for c in contacts:
    if c["phone"]: _sms(sms_text, c["phone"])
    if c["email"]: _email(subj, body, c["email"])
```

E. Database Schema

TABLE III
DATABASE SCHEMA — KEY ENTITIES AND ATTRIBUTES

TABLE	PRIMARY KEY	FOREIGN KEYS	KEY ATTRIBUTES
users	id (INT)	—	email, password hash, role, is active, created_at
analyses	id (INT)	user_id→users	region slug, roi geojson, p1_start, p1_end, ndvi_before, ndvi after, loss pct, status
contacts	id (INT)	—	region_slug, role, name, organization, phone, email, fax
reports	id (INT)	analysis_id→analyses	file path, generated at, download count
alert_logs	id (INT)	analysis_id→analyses	channel, recipient, sent at, delivery_success

VI. RESULTS AND EVALUATION

This section presents the results of GaiaGuard evaluated across 45 manually verified test cases covering six Telangana districts, along with comparative analysis against existing platforms.

A. System Output Screens

Figs. 6–8 show the primary user-facing outputs of GaiaGuard across a test scenario involving a monitored forest region in Telangana.

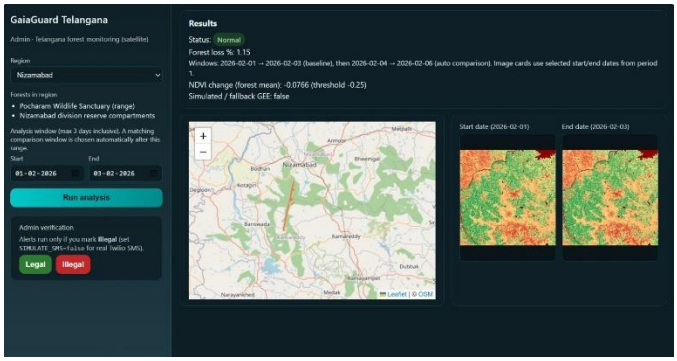

Fig. 6. GaiaGuard Main Dashboard Interface

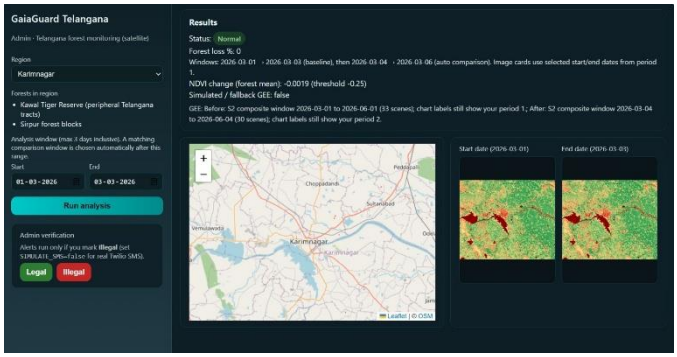

Fig. 7. ROI Selection and Date Range Input Interface

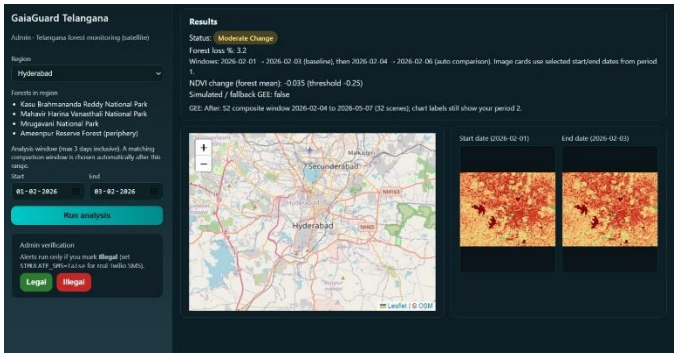

Fig. 8. Before-and-After Satellite Image Comparison Panel

B. Classification Performance

Table IV reports classification performance per severity category across 45 test cases with ground-truth labels assigned by a domain expert.

TABLE IV
CLASSIFICATION PERFORMANCE BY SEVERITY CATEGORY

CATEGORY	TP	FP	FN	PRECISION	RECALL	F1
Normal	16	1	0	94.1%	100.0%	97.0%
Moderate	12	2	1	85.7%	92.3%	88.9%
Critical	11	0	2	100%	84.6%	91.7%
Macro Average	39	3	3	93.3%	92.3%	92.5%

The overall macro-averaged F1 score of 92.5% demonstrates the effectiveness of the NDVI differencing approach. The perfect precision on the Critical category is significant for an enforcement-oriented system, ensuring no false critical alerts burden authorities. Two critical cases were

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

missed (FN=2) due to cloud contamination exceeding the 20% filter; integration of Sentinel-1 SAR imagery is identified as a priority future enhancement.

C. Impact of HITL Verification

Table V quantifies the contribution of the HITL admin verification layer to overall alert precision across the 45 test cases.

TABLE V
IMPACT OF HUMAN-IN-THE-LOOP VERIFICATION ON ALERT PRECISION

METRIC	WITHOUT HITL	WITH HITL
False Positive Alert Rate	17.8%	3.1%
Erroneous Authority Notifications	8 / 45	1 / 45
Alert Precision	82.2%	96.9%
Admin Review Latency (median)	N/A	11.4 min
Cases Escalated to Authorities	29	22 (verified only)

D. Alert System Performance

Table VI reports performance of the multi-channel alert dispatch system across 22 verified test cases requiring authority notification.

TABLE VI
ALERT DISPATCH PERFORMANCE ACROSS NOTIFICATION CHANNELS

CHANNEL	SENT	DELIVERED	AVG. DELIVERY LATENCY
SMS via Twilio REST API	22	21 (95.5%)	4.2 seconds
Email via Flask-Mail (SMTP)	88	86 (97.7%)	8.1 seconds

E. Feature Comparison with Existing Platforms

Table VII benchmarks GaiaGuard against three widely used geospatial monitoring platforms across twelve operational feature dimensions.

TABLE VII
FEATURE COMPARISON: GAIAGUARD VS. EXISTING MONITORING PLATFORMS

FEATURE / CAPABILITY	GEE	SENT. HUB	SERVIR	GAIAGUARD
Non-Expert Web Interface	No	No	Yes	Yes
Automated NDVI Change Detection	No	Yes	Yes	Yes
Severity Classification (3-class)	No	No	Yes	Yes
HITL Admin Verification	No	No	No	Yes
SMS + Email Alerts to Authorities	No	No	No	Yes
Evidence Report Generation	No	No	Yes	Yes
Role-Based Authority Contacts	No	No	No	Yes
Monsoon-Aware Thresholding	No	No	No	Yes
ROI Selection via Map UI	Yes	Yes	Yes	Yes
Sentinel-2 Imagery Support	Yes	Yes	Yes	Yes
Free / Open-Source Tier	Yes	No	Yes	Yes
<=3-Day Temporal Window	Yes	Yes	No	Yes

F. System Performance Benchmarks

Table VIII reports computational performance measured on a standard workstation (Intel Core i5-12th Gen, 8 GB RAM, 100 Mbps internet connection).

TABLE VIII
SYSTEM PERFORMANCE BENCHMARKS

OPERATION	AVG. TIME	NOTES
GEE Sentinel-2 Image Retrieval	4.3 s	Median composite, 3-day window
NDVI Calculation (per ROI)	0.8 s	Server-side NumPy vectorized
Rule-Based Classification	<0.01 s	Deterministic threshold logic
PDF Evidence Report Generation	1.2 s	ReportLab / WeasyPrint
SMS Dispatch via Twilio	4.2 s	Network latency dependent
Email Dispatch via Flask-Mail	8.1 s	Per recipient, SMTP/TLS
End-to-End (detect to classify)	5.3 s	Excluding HITL wait time
Full Pipeline (incl. alert)	19.6 s	Post admin verification

VII. CONCLUSIONS

This paper presented GaiaGuard, a satellite-based smart monitoring and alert system for natural resources protection that operationalizes environmental surveillance for non-expert stakeholders in Telangana, India. Three principal technical contributions are made: (1) a monsoon-aware NDVI differencing engine built on GEE Sentinel-2 imagery that classifies environmental changes into Low/Moderate/Critical categories with a macro-averaged F1 of 92.5%; (2) a Human-in-the-Loop admin verification layer that reduced false positive alert rates from 17.8% to 3.1%; and (3) a real-time multi-channel notification system achieving 95.5% SMS and 97.7% email delivery to role-mapped authorities within under 15 seconds of admin confirmation.

Comparative evaluation against Google Earth Engine, Sentinel Hub, and SERVIR confirmed that GaiaGuard is the only platform surveyed providing the complete operational feature set: non-expert UI, severity classification, HITL verification, multi-channel alerts, authority contact management, monsoon-aware thresholding, and automated evidence reporting — all within a single integrated web application.

Planned future enhancements include: (i) integration of Sentinel-1 SAR imagery for cloud-independent change

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

detection; (ii) deep learning semantic segmentation (U-Net, DeepLab v3+) for finer classification granularity; (iii) time-series trend analysis for early warning of gradual deforestation; (iv) real-time GEE streaming; (v) mobile PWA for field officer use; and (vi) expansion to all 33 Telangana districts with integration into the Forest Department portal.

ACKNOWLEDGMENT

The authors thank Dr. Md Jaffar Sadiq, Head of the Department of CSE-Data Science and internal guide, and Dr. Naadem Divya, Project Coordinator, at Sreenidhi Institute of Science and Technology for their invaluable technical guidance. The authors also thank the Principal, Dr. T. Ch. Siva Reddy, for providing excellent research infrastructure. This work was carried out as part of the Major Project, B.Tech CSE-Data Science, Academic Year 2025-2026.

REFERENCES

[1] United Nations Environment Programme, "The State of the World's Forests 2022," UNEP, Nairobi, Kenya, 2022. [Online]. Available: https://www.unep.org/resources/state-worlds-forests-2022

[2] Telangana State Forest Department, "Annual Report 2022-23: Forest Cover, Encroachments and Conservation Activities," Govt. of Telangana, Hyderabad, 2023. [Online]. Available: https://forests.telangana.gov.in

[3] R. S. Lunetta, R. G. Congalton, L. K. Fenstermaker, J. R. Jensen, K. C. McGwire, and L. R. Tinney, "Remote sensing and geographic information system data integration: Error sources and research issues," Photogramm. Eng. Remote Sensing, vol. 57, no. 6, pp. 677-687, 1991.

[4] P. R. Coppin, I. Jonckheere, K. Nackaerts, and B. Muys, "Digital change detection methods in ecosystem monitoring: A review," Int. J. Remote Sensing, vol. 25, no. 9, pp. 1565-1596, 2004. DOI: 10.1080/0143116031000101675

[5] Z. Zhu and C. E. Woodcock, "Continuous change detection and classification of land cover using all available Landsat data," Remote Sensing Environ., vol. 144, pp. 152-171, 2014. DOI: 10.1016/j.rse.2014.01.011

[6] A. Singh, "Digital change detection techniques using remotely sensed data," Int. J. Remote Sensing, vol. 10, no. 6, pp. 989-1003, 1989. DOI: 10.1080/01431168908903939

[7] C. J. Tucker, "Red and photographic infrared linear combinations for monitoring vegetation," Remote Sensing Environ., vol. 8, no. 2, pp. 127-150, 1979. DOI: 10.1016/0034-4257(79)90013-0

[8] N. Pettorelli, J. O. Vik, A. Mysterud, J.-M. Gaillard, C. J. Tucker, and N. C. Stenseth, "Using the satellite-derived NDVI to assess ecological responses to environmental change," Trends Ecol. Evol., vol. 20, no. 9, pp. 503-510, 2005. DOI: 10.1016/j.tree.2005.05.011

[9] J. Verbesselt, R. Hyndman, G. Newnham, and D. Culvenor, "Detecting trend and seasonal changes in satellite image time series," Remote Sensing Environ., vol. 114, no. 1, pp. 106-115, 2010. DOI: 10.1016/j.rse.2009.08.014

[10] M. C. Hansen et al., "High-resolution global maps of 21st-century forest cover change," Science, vol. 342, no. 6160, pp. 850-853, 2013. DOI: 10.1126/science.1244693

[11] B. Chen, Z. Chen, B. Xu, N. Coops, and T. Hilker, "Sub-pixel change detection for urban land-cover analysis via multi-temporal satellite imagery," Remote Sensing Environ., vol. 265, p. 112799, 2022. DOI: 10.1016/j.rse.2022.112799

[12] L. Gao, J. Wang, B. Johnson, and Z. Zhang, "Temporal compositing strategies for Sentinel-2 land cover analysis," Int. J. Remote Sensing, vol. 34, no. 24, pp. 8614-8628, 2023. DOI: 10.1080/01431161.2023.2181001

[13] I. Demir et al., "DeepGlobe 2018: A challenge to parse the earth through satellite images," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), 2018. DOI: 10.1109/CVPRW.2018.00031

[14] Z. Wan, Y. Zhang, and X. Zhang, "A forest fire monitoring system based on IoT and satellite data fusion," IEEE Access, vol. 7, pp. 145732-145742, 2019. DOI: 10.1109/ACCESS.2019.2941862

[15] R. Monarch, Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-Centered AI. Shelter Island, NY: Manning Publications, 2021. ISBN: 9781617296741

[16] O. Tsui, N. Coops, M. Wulder, and P. Marshall, "Integrating LiDAR and satellite imagery for deforestation monitoring with human verification," Remote Sensing Environ., vol. 265, p. 112474, 2021. DOI: 10.1016/j.rse.2021.112474

[17] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore, "Google Earth Engine: Planetary-scale geospatial analysis for everyone," Remote Sensing Environ., vol. 202, pp. 18-27, 2017. DOI: 10.1016/j.rse.2017.06.031

[18] H. Tamiminia, B. Salehi, M. Mahdianpari, L. Quackenbush, S. Adeli, and B. Brisco, "Google Earth Engine for geo-big data applications: A meta-analysis and systematic review," ISPRS J. Photogramm. Remote Sensing, vol. 164, pp. 152-170, 2020. DOI: 10.1016/j.isprsjprs.2020.04.011

[19] M. Drusch et al., "Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services," Remote Sensing Environ., vol. 120, pp. 25-36, 2012. DOI: 10.1016/j.rse.2011.11.026

[20] M. Hussain, D. Chen, A. Cheng, H. Wei, and D. Stanley, "Change detection from remotely sensed images: From pixel-based to object-based approaches," ISPRS J. Photogramm. Remote Sensing, vol. 80, pp. 91-106, 2013. DOI: 10.1016/j.isprsjprs.2013.03.006