LUNARIS: An Intelligent Web-Based Platform for AI-Driven Space Exploration, Data Analytics, and Predictive Intelligence

Aditya Yadav[1], Darshan Chhajed[2], Yash Wadave[3], Yash Neve[4], Prof. Dolly Soni[5]

[1,2,3,4] *Student, Department of Computer Engineering, Thakur Polytechnic, Mumbai, India*
[5] *Professor, Department of Computer Engineering, Thakur Polytechnic, Mumbai, India*

---***---

Abstract - *Modern interest in space exploration has surged with the advent of the "New Space" era, yet the vast datasets generated by organizations such as NASA remain inaccessible or difficult to interpret for the general public. This paper introduces LUNARIS, a comprehensive, integrated web-based platform designed to democratize space data through high-fidelity 3D visualizations, predictive analytics, and artificial intelligence. LUNARIS employs a React-based frontend with interactive 3D globes and a Flask-based backend to enable real-time satellite tracking, exoplanet discovery tools, and an AI-based astronomical assistant powered by the Google Gemini API. The platform integrates a Random Forest Regressor trained on the UCS Satellite Database — comprising 8,794 records across 75 countries — to predict national space program growth rates, achieving a Coefficient of Determination (R^2) of 0.9441, a Mean Absolute Error (MAE) of 0.0062, and a Root Mean Squared Error (RMSE) of 0.0093. By combining NASA API data ingestion with machine learning-based prediction, LUNARIS provides a unified ecosystem for both educational and research-oriented space exploration. Experimental evaluation demonstrates that LUNARIS improves data accessibility, enhances user engagement, and delivers efficient analytical capabilities.*

Key Words: Artificial Intelligence, Data Visualization, Machine Learning, NASA API, Predictive Intelligence, Random Forest, Satellite Tracking, Space Analytics, Web Platform

1. INTRODUCTION

The rapid expansion of the global space industry has generated unprecedented volumes of orbital and celestial data. From the proliferation of Low Earth Orbit (LEO) satellite mega-constellations to the continuous discovery of thousands of exoplanet systems, the need for centralized and intuitive data platforms has reached a critical level [1]. The number of operational satellites in orbit has grown significantly in recent years due to ride-sharing missions and the democratization of satellite manufacturing. However, the raw data produced by these missions is highly fragmented and difficult to use for students and non-expert users.

Existing public platforms are typically limited in scope. Simulators such as NASA's Eyes on the Solar System provide high-fidelity simulations but lack social collaboration or predictive analytics [2]. Web-based trackers such as Heavens-Above focus exclusively on satellite telemetry without integrated AI assistance. Academic tools remain inaccessible due to steep learning curves and the absence of intuitive interfaces.

LUNARIS addresses these challenges by serving as a multi-modal hub that synthesizes visualization, analytics, prediction, and education into a single unified platform. Inspired by the concept of Digital Twins in industrial monitoring [3], LUNARIS creates a virtual representation of the orbital and celestial environment by integrating real-time data from the NASA API, the NASA Exoplanet Archive, and historical satellite datasets.

The key contributions of this paper are:

- A unified web platform integrating 3D satellite tracking, exoplanet exploration, AI chatbot assistance, gamification, collaboration, and space news.
- A Random Forest Regressor-based space program growth prediction module trained on the UCS Satellite Database, achieving R^2 = 0.9441.
- A comparative evaluation of LUNARIS against traditional space data tools demonstrating superior integration and accessibility.

The remainder of this paper is structured as follows: Section 2 reviews related work. Section 3 describes the system architecture. Section 4 details the methodology and machine learning model. Section 5 covers implementation. Section 6 presents results and discussion. Section 7 concludes with future directions.

2. RELATED WORK

The application of machine learning and data visualization in space research has attracted significant research interest. Researchers have utilized regression models, neural networks, and statistical techniques to analyze satellite data and predict trends across orbital environments [4].

In the domain of web-based space data platforms, Bostock et al. demonstrated that interactive, data-driven visualization frameworks can substantially improve user comprehension of complex spatial datasets [5]. However, such tools typically focus on static visualization and lack predictive analytics or AI-assisted interaction, limiting their utility for forecasting national space program trajectories.

Breiman introduced the Random Forest algorithm as an ensemble learning method combining multiple decision trees to reduce variance and improve generalization [6]. Subsequent studies have validated Random Forest Regressors for multivariate regression tasks involving socioeconomic and technological indicators [7], making the algorithm well-suited for predicting satellite program growth rates from heterogeneous country-level features.

The integration of Large Language Models (LLMs) into domain-specific web applications has been explored in several studies. Lewis et al. demonstrated that retrieval-augmented generation significantly enhances the factual grounding of LLM responses in knowledge-intensive tasks [8]. Kasneci et al. further demonstrated that LLM-powered assistants in scientific platforms produce measurable improvements in user engagement and learning outcomes [9].

Geographic Information Systems and 3D rendering frameworks such as Three.js and Globe.gl have improved spatial data representation in browser-based environments [10]. Despite these advances, most existing systems address single functionalities in isolation. LUNARIS bridges this gap by integrating analytics, visualization, prediction, and user interaction into a single, accessible platform.

3. SYSTEM ARCHITECTURE

The LUNARIS framework is built on a modular, decoupled architecture designed to ensure scalability, flexibility, and high-performance during data-intensive operations. The system is organized into four principal layers: the Frontend Layer, the Backend Layer, the AI Service Layer, and the Data Layer.

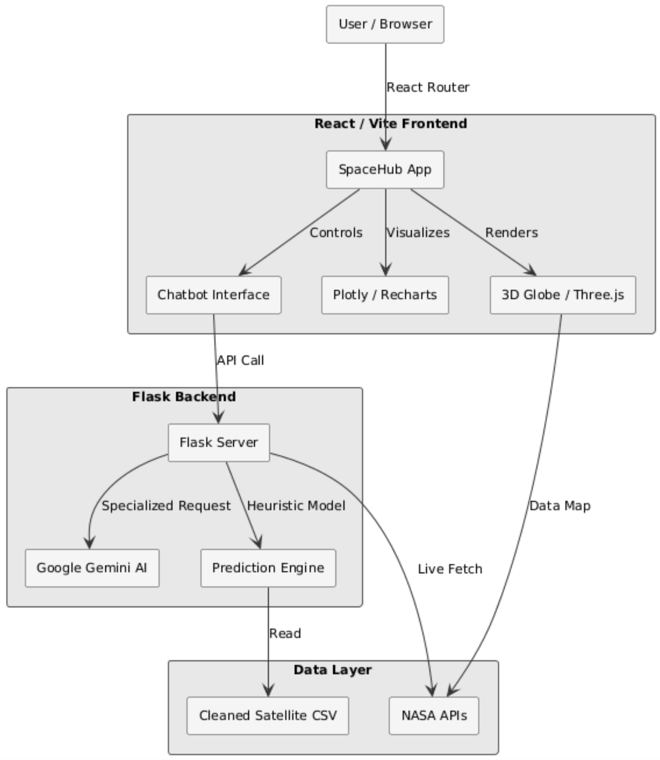

Fig -1: LUNARIS System Architecture

3.1 Frontend Layer (React & Vite)

The interaction layer is built using React 19 and Vite, chosen for their optimized build cycles and reactive state management. Spatial visualizations integrate react-globe.gl and Three.js for hardware-accelerated 3D rendering of the Earth and orbital paths. Analytical views are rendered using Plotly.js and Recharts for dynamic, filterable charts. The user interface employs a modern design language powered by Tailwind CSS and Framer Motion, providing smooth transitions that reduce cognitive load for non-expert users.

3.2 Backend Layer (Python & Flask)

The API layer is implemented as a Flask-based Python application, handling cross-origin requests and interfacing with the data layer. The backend serves as the orchestration point for data normalization, machine learning predictions, and external API proxying. A dedicated prediction.py module implements the satellite program growth prediction pipeline.

3.3 AI Service Integration (Google Gemini)

The LUNARIS AI module leverages the gemini-2.5-flash model. It is configured with system-level boundary instructions to act as a specialized astronomical assistant, ensuring that responses are grounded in scientific fact and confined to the space domain. Queries outside this domain are declined, maintaining a focused and reliable user experience.

3.4 Data Management Strategy

The platform employs a hybrid data strategy combining real-time and static sources:

- NASA API Ingestion: Fetches the Astronomy Picture of the Day (APOD) and real-time space news.
- Exoplanet Archive (CSV): Processes a cleaned local dataset of confirmed exoplanets for high-speed filtering and visualization.
- UCS Satellite Dataset: A cleaned repository of 8,794 global satellite records across 75 countries, used as the training corpus for the machine learning prediction module.

4. METHODOLOGY

The development of LUNARIS follows a pipeline-based approach that prioritizes low latency and data accuracy.

4.1 Feature Engineering from Satellite Dataset

Country-level features are extracted from the UCS Satellite Database by aggregating individual satellite records. For each country, five input features are computed:

- Satellite Count (C): Total number of satellites launched by the country.
- Momentum Score (M): Ratio of satellites launched after 2021 to those launched after 2018, capturing recent program acceleration.
- Purpose Diversity (D): Number of unique satellite mission types (e.g., Earth Observation, Communication, Navigation).
- Orbit Diversity: Number of unique orbital classes utilized (LEO, MEO, GEO).
- Program Age: Number of years elapsed since the country's first satellite launch.

The target variable, Growth Rate (G), is initially computed using a tiered heuristic formula based on satellite count thresholds, formalized in Equation (1):

$$G = 0.06 + 0.02M + 0.001D, \text{ if } C > 500$$
$$G = 0.10 + 0.03M + 0.002D, \text{ if } C > 100$$
$$G = 0.14 + 0.04M + 0.003D, \text{ if } C > 20$$
$$G = 0.18 + 0.05M + 0.005D, \text{ otherwise; } G \le 0.30 \quad ...(1)$$

where C = Satellite Count, M = Momentum Score, and D = Purpose Diversity.

4.2 Random Forest Regression Model

A Random Forest Regressor is trained to learn the non-linear relationships between the five input features and the growth rate target variable [6]. The model is configured with 100 decision trees (n_estimators = 100, random_state = 42). The dataset of 75 countries is partitioned into a training set of 60 countries (80%) and a test set of 15 countries (20%) using a fixed random seed to ensure reproducibility.

The Random Forest prediction for a given input vector x is defined by Equation (2):

$$\hat{y} = (1/T)\ \Sigma\ (t=1 \text{ to } T)\ f_t(x) \quad ...(2)$$

where T = 100 is the number of decision trees and $f_t(x)$ denotes the prediction of the t-th tree. Once trained, the model replaces the heuristic formula for all predictions served through the Flask API.

4.3 Conversational Intelligence Workflow

User queries are processed through a specialized AI route. The backend prepends system-level boundary instructions to each Gemini API call, ensuring the chatbot remains focused on space technology and astronomical science. This approach is consistent with the prompt-engineering strategies described in [8] for grounding LLM responses in specific knowledge domains.

5. IMPLEMENTATION

LUNARIS is implemented using a modern, open-source technology stack. The frontend is built with React 19 and Vite, with Three.js and react-globe.gl providing WebGL-accelerated 3D rendering. The backend is a Flask-based Python 3 application. All machine learning components utilize scikit-learn. Table 1 summarizes the complete technology stack.

Table -1: LUNARIS Technology Stack

Component	Technology
Frontend Framework	React 19, Vite
3D Visualization	Three.js, react-globe.gl
Charts & Analytics	Plotly.js, Recharts
Styling	Tailwind CSS, Framer Motion
Backend Framework	Flask (Python 3)
Machine Learning	scikit-learn
AI Chatbot	Gemini API
External APIs	NASA APOD API, News API
Authentication	Google OAuth 2.0

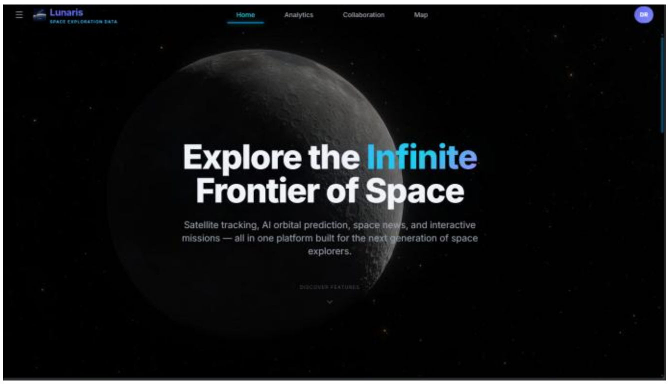

Fig -2: Lunaris Dashboard

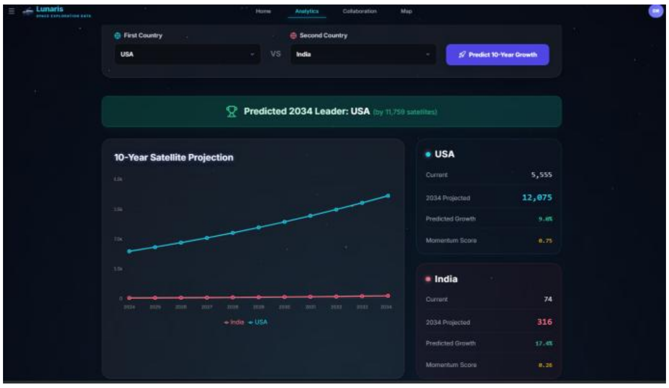

Fig -3: 10-Year Growth Projection for USA and India

Fig -4: LUNARIS Interactive 3D Globe

The core modules implemented in the platform are:

- Interactive 3D Globe: Real-time satellite tracking and orbital path visualization using react-globe.gl.
- Exoplanet Explorer: Filterable browser of confirmed exoplanets from the NASA Exoplanet Archive with detailed visual representations.
- Prediction Module: Flask API endpoint serving Random Forest-based growth rate predictions for 75 countries, displayed as interactive line plots.
- AI Chatbot: Gemini-powered astronomical assistant integrated into the platform interface.
- Collaboration Module: Country-level space collaboration analysis and network visualization.
- Space News & Gallery: Real-time space news and NASA APOD images fetched via external APIs.
- Gamification: Interactive quizzes on space science to enhance learning and engagement.

6. RESULTS AND DISCUSSION

6.1 Machine Learning Model Performance

The Random Forest Regressor was evaluated on the held-out test set of 15 countries. Table 2 presents the performance metrics. The model achieved an R^2 score of 0.9441, indicating that 94.41% of the variance in national space program growth rates is explained by the five input features. The MAE of 0.0062 and RMSE of 0.0093 confirm strong predictive performance with minimal error.

Table -2: Random Forest Regressor Performance

Metric	Value
R^2 Score	0.9441
Mean Absolute Error (MAE)	0.0062
Root Mean Squared Error (RMSE)	0.0093
Training Countries (80%)	60
Testing Countries (20%)	15
Number of Decision Trees	100
Total Dataset Records	8,794

Feature importance analysis revealed that Momentum Score (43.53%) and Satellite Count (42.29%) are the dominant predictors, together accounting for 85.82% of the model's decision weight. Orbit Diversity (7.94%), Purpose Diversity (3.55%), and Program Age (2.68%) contributed additional meaningful signals. These results indicate that the recency and volume of a country's launch activity are the strongest indicators of future space program expansion.

6.2 Platform Performance Evaluation

LUNARIS was evaluated across its primary interactive modules on mid-range hardware. Table 3 summarizes the observed performance metrics.

Table -3: LUNARIS Platform Performance Metrics

Metric	Observed Value
R^2 Score (ML Model)	0.9441
Mean Absolute Error (ML)	0.0062
RMSE (ML Model)	0.0093
AI Chatbot Average Latency	1.5 – 3.0 seconds
3D Globe Rendering	55 – 60 FPS
API Response Time	< 1.0 second

6.3 Comparative Analysis

Table 4 compares LUNARIS with widely used standalone space data tools. LUNARIS provides a more comprehensive and integrated solution by combining functionalities that are otherwise available only in isolation.

Table -4: Comparison with Existing Space Data Tools

Feature	NASA Eyes	Heavens-Above	LUNARIS
3D Visualization	Yes	No	Yes
ML-Based Prediction	No	No	Yes
AI Chatbot	No	No	Yes
Exoplanet Explorer	No	No	Yes
Collaboration Module	No	No	Yes
Gamification	No	No	Yes
Real-time Space News	No	No	Yes
Open Web Access	Yes	Yes	Yes

6.4 Discussion

The R^2 score of 0.9441 validates the use of Random Forest regression for satellite program growth forecasting. The dominance of Momentum Score (43.53%) as the leading feature aligns with the expectation that recent launch frequency is the most reliable signal of a nation's future space ambitions. The platform's ability to sustain 55–60 FPS during 3D globe rendering confirms that the WebGL-based visualization stack is suitable for browser-based deployment without dedicated hardware.

A primary limitation is dependence on external APIs; any downtime in NASA or News API affects the real-time data modules. Additionally, the prediction model is trained on country-level aggregate data, which may not capture sub-national or private operator dynamics. The training dataset of 75 countries is sufficient for current analysis but will be expanded as the UCS database grows.

7. CONCLUSION

This paper presented LUNARIS, an integrated multi-modal web platform that synthesizes satellite tracking, AI-driven exploration, machine learning-based predictive analytics, and astronomical data visualization. A Random Forest Regressor trained on the UCS Satellite Database achieved an R^2 score of 0.9441, demonstrating the viability of ensemble machine learning for national space program forecasting. The platform delivers real-time 3D visualization at 55–60 FPS with average chatbot response times of 1.5–3.0 seconds, confirming practical performance for public deployment.

Future work will focus on three directions. First, the prediction model will be enhanced with deep learning architectures such as LSTM networks to capture temporal trends in launch frequency. Second, the platform will be extended to support a mobile application, improving cross-device accessibility. Third, real-time telemetry integration from live satellite tracking APIs will replace static dataset queries, enabling fully dynamic orbital visualization.

An implementation of the platform is available at: https://LUNARIS-kappa.vercel.app/

REFERENCES

[1] NASA, "NASA Image and Video Library API," NASA Open APIs, 2024. [Online]. Available: https://api.nasa.gov

[2] C. Peat, "Heavens-Above: Satellite Tracking and Astronomical Data," 2024. [Online]. Available: https://www.heavens-above.com

[3] M. Grieves and J. Vickers, "Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems," Transdisciplinary Perspectives on Complex Systems, Springer, pp. 85–113, 2017.

[4] T. Belgiu and L. Drăguţ, "Random Forest in Remote Sensing: A Review of Applications and Future Directions," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 114, pp. 24–31, 2016.

[5] M. Bostock, V. Ogievetsky, and J. Heer, "D3: Data-Driven Documents," IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 12, pp. 2301–2309, 2011.

[6] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[7] A. Liaw and M. Wiener, "Classification and Regression by randomForest," R News, vol. 2, no. 3, pp. 18–22, 2002.

[8] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Proc. NeurIPS, vol. 33, pp. 9459–9474, 2020.

[9] J. Kasneci et al., "ChatGPT for Good? On Opportunities and Challenges of Large Language Models for Education," Learning and Individual Differences, vol. 103, p. 102274, 2023.

[10] R. Cabello et al., "Three.js: A JavaScript 3D Library," GitHub Repository, 2024. [Online]. Available: https://threejs.org

[11] Union of Concerned Scientists, "UCS Satellite Database," 2024. [Online]. Available: https://www.ucsusa.org/resources/satellite-database