26th AMOS

Advanced Maui Optical and Space Surveillance Technologies Conference

September 16-19, 2025 | Maui Hawaii

2025 CONFERENCE PROCEEDINGS

ISSN 2576-5965

PRESENTED BY mauieconomic DEVELOPMENTBOARD

PREFACE

This volume consists of papers from the oral and poster presentations of the AMOS Conference held in-person at the Wailea Beach Resort Marriott in Maui, Hawaii, as well as online, from September 16-19, 2025. The number and quality of the papers received attest to the increased recognition and exposure of the conference in the scientific and technical communities.

The topics contained in this year's proceedings include Astrodynamics, Atmospherics/Space Weather, Cislunar SDA, Conjunction/RPO, Machine Learning for SDA Applications, Satellite Characterization, SDA Systems & Instrumentation, Space-Based Assets, Space Debris and Space Domain Awareness.

We would like to express immense appreciation to the global SSA/SDA Community for continuing to advance this important field of study. We thank the technical and session chairs, the more than 1300 participants, conference sponsors and exhibitors, presenters, panelists, and authors who contributed to making the 26th AMOS Conference a success.

CITING A PAPER

When citing a paper from the proceedings please reference: 2025 Advanced Maui Optical and Space Surveillance Technologies Conference (AMOS) – www.amostech.com

HOW TO NAVIGATE THE CONTENTS

The document is set to open with a view of "Bookmarks' on right. These are linked to the various sessions and papers. Use the arrows next to session titles to reveal paper titles within that session.

The Contents pages also contain the links to each paper. Click on the title or page number of the paper to jump to that paper in the document.

To return to the Contents pages, click on a session name in the Bookmarks pane.

AMOS Conference
A program of Maui Economic Development Board, Inc.
1305 N. Holopono Street, Suite 1, Kihei, Maui, Hawaii 96753 | Telephone: 808.875.2300
E-mail: info@amostech.com | Web: www.amostech.com

2025 AMOS CONFERENCE PROCEEDINGS

SPACE-BASED ASSETS

Co-chaired by **Melrose Brown**, UNSW Canberra Space and **Orlando Diaz,** NASA Ames Research Center

SPACE DOMAIN AWARENESS

Co-chaired by **Jeremy Raley**, AFRL and **Ayla Reed,** AFRL/RDSM

Prototype Development of Web AI-Based Decision Support System: Insights and Recommendations for Satellite Anomaly Identification

Abdul Mutholib
ECE Department, Faculty of Engineering International Islamic University Malaysia
Kuala Lumpur, Malaysia
Department of System Information, Faculty of Science and Technology UIN Syarif Hidayatullah
Jakarta, Indonesia
antholib@gmail.com

***Nadirah Abdul Rahim**
ECE Department, Faculty of Engineering International Islamic University Malaysia
Kuala Lumpur, Malaysia
nadirahabdulrahim@iium.edu.my

Teddy Surya Gunawan
ECE Department, Faculty of Engineering International Islamic University Malaysia
Kuala Lumpur, Malaysia
tsgunawan@iium.edu.my

ABSTRACT

Satellites are vital for various applications, including communication, navigation, earth observation and any other research. Satellite reliability is important for mission continuity and space sustainability, as anomalies can cause costly failures, loss of operational capacity, and loss of sustainability in outer space. It results in crowded orbit and poses a risk to other active or future missions. Traditional methods of identifying and resolving anomalies are often reactive and limited in their ability to handle the complexity and volume of satellite data. Hence, this paper proposes a Web AI-based Decision Support System (DSS) designed to enhance the identification and resolution of satellite anomalies. The proposed Web AI-based DSS framework integrates Machine Learning (ML) based Trade-Space Exploration (TSE) as the model base and Generative AI as the knowledge base to offer insights and recommendations for anomaly prevention and decision making prior to the launch of the satellite into orbit. The system architecture includes a statistical data module for satellite anomaly identification and a decision support application. The Seradata database is utilized as the main source data for satellite anomalies which consist of around 4050 data since 1957. This system aims to identify anomalies to prevent any failures that mostly occur during satellite orbit and provide appropriate recommendations for counteractive actions. A detailed literature review highlights the current state of satellite anomaly identification and the application of the Web AI-based DSS in various fields. The review identifies gaps in existing research, emphasizing the need for a specialized DSS in satellite anomaly identification. The proposed framework addresses these gaps by incorporating state-of-the-art artificial intelligence (AI) and a decision-making system. This paper also summarizes the key findings of the case studies and discusses the benefits of using a Web AI-based DSS for satellite anomaly identification management. It also addresses potential challenges and limitations, such as the need for continuous updates to the model and the integration of diverse data sources. The contribution of this paper can be summarized as (i) outlines Web AI-based DSS for satellite anomaly identification, (ii) present comprehensive taxonomy of the Web AI-Based DSS methods applied to space situational awareness (SSA), which addresses the insight of factors leading to the loss of a satellite and its recommendation to prevent satellite failure during their operational. In a nutshell, the proposed Web AI-based DSS framework provides a robust solution to manage satellite anomalies, improves operational efficiency, and mitigates the risk of orbital satellite failure. The paper outlines prospective research directions that include the advancement of more sophisticated anomaly identification algorithms and the incorporation of additional data sources, such as data cost, to further improve system performance.

Keywords: **Artificial Intelligence (AI), Machine Learning (ML), Decision Support System, Satellite anomalies, Space Situational Awareness (SSA), Trade-space exploration, Seradata**

1. INTRODUCTION

Satellites are a fundamental component of modern society. Especially in the era of data, they play an important role in facilitating various applications, including data communication, navigation, or global positioning, meteorological forecasting, and Earth observation. McKinsey and Company projected substantial growth for the global space economy in the coming years. The space industry is projected to generate more than $1 trillion in revenue by 2040 [1], suggesting substantial growth over the next two decades. This expansion is driven by innovations in satellite technology, increased demand for data and connectivity, and the proliferation of commercial space endeavors [1]. This growth needs to be controlled and monitored to maintain space sustainability as initiated by the EU Space Program, which is known as space situational awareness (SSA) [2].

Over ten thousand objects in Earth's orbit pose a potential threat to satellites and their launches [3]. SSA refers to monitoring objects in orbit and forecasting their positions at specific times [2]. One component of SSA is orbiting space objects (OSA), which include active satellites and space debris [2]. A significant portion of SSA for OSA involves monitoring these entities to establish a catalog of such objects. The catalog serves as the foundation for orbital evolution and conjunction evaluation, the latter intended to protect active assets from collisions [2]. The production of space debris began with the launch of the first artificial satellite, Sputnik-1, in October 1957 and has accumulated to today [2]. Each space launch produces approximately 100 pieces of debris, including orbital remnants from launch vehicles, discarded shrouds, and smaller fragments resulting from pyrotechnic devices used to detach the satellite from the launch vehicle [2]. Based on the data from Seradata, from 1965 to March 2024 there were around 105,314 active satellites as shown in Fig.1 within various orbits including low Earth orbit (LEO), medium Earth orbit (MEO), geosynchronous and geostationary orbit (GSO & GEO) and high elliptical orbit (HEO); see Fig. 2. The current number of satellites and existing conditions may trigger a chain reaction, resulting in a significant increase in accidental collisions and the formation of a debris shell in low-Earth orbit, thereby rendering future operations in this orbital space unfeasible. This phenomenon has been designated as the Kessler syndrome [2].

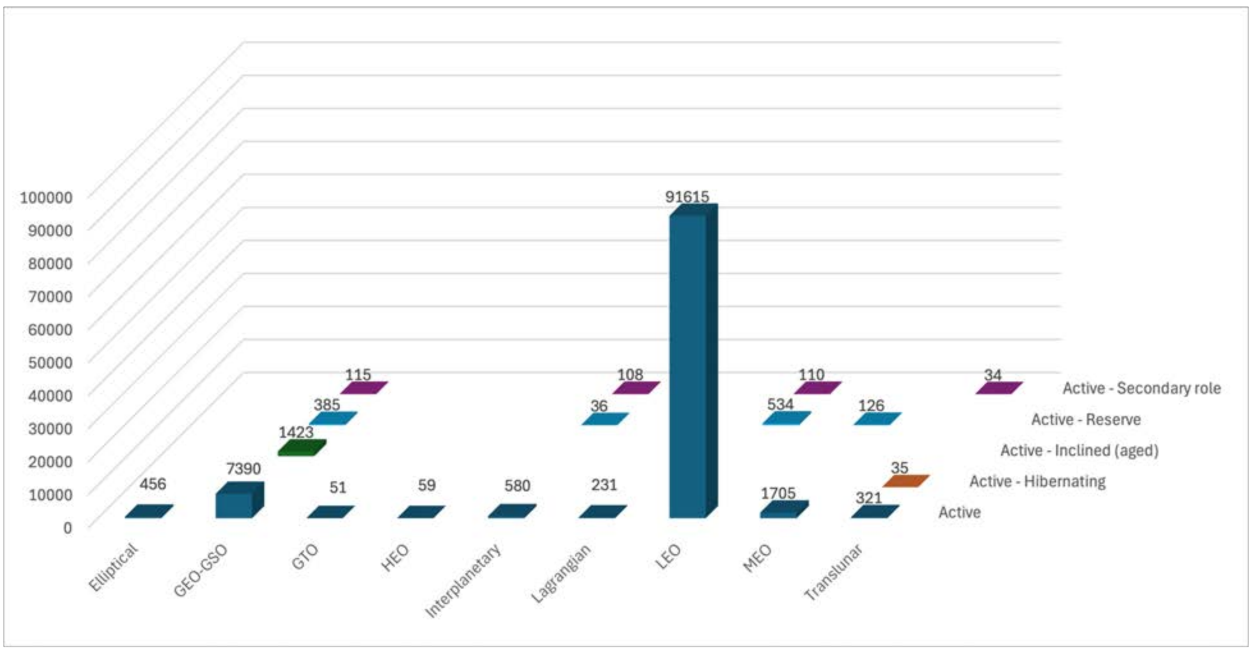

Fig. 1: Active Satellites Orbits in year 1965 - year 2024 (Seradata)

The types of satellite employed are dictated by the mission requirements of the stakeholders, including the owner, the manufacturer, and the operator. An operator, whether military, commercial, or civil, determines the purpose of the satellite, which typically determines the necessary orbital type required [4]. A subsequent analysis of the satellite's commercial viability will be conducted in the market, and financing will need to be secured. In both civil and military domains, it is customary to issue a request for proposals, which may subsequently result in the selection of one or more

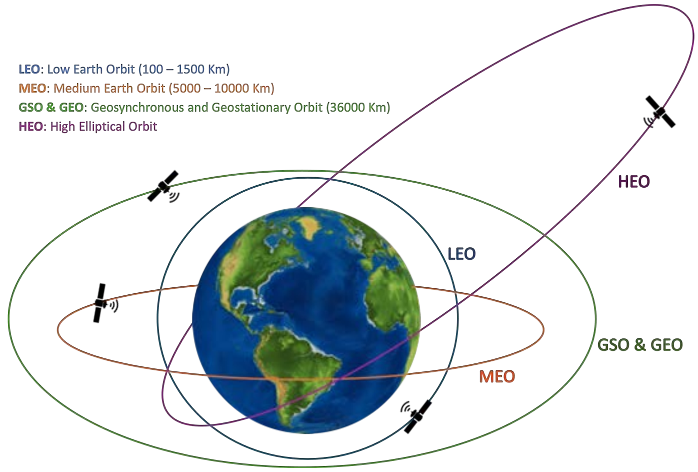

Fig. 2: Satellite Orbits

contractors to undertake the preliminary design of the prospective satellite. Provided that all or most of the operational requirements are satisfied, a construction order will be issued and a launch vehicle will be chosen [4].

Extensive and complicated missions are mostly conducted using large satellites and allow manufacturers to add more equipment or increase the safety margin of components for extra durability. Satellite operational lifespans are dependent on mission objectives, orbital and size, with smaller satellites typically lasting a few years, but geostationary communication satellites may function for up to 15 years [5]. This is important to the continuous operation of these complex systems, as it is essential to maintain critical infrastructure and ensure the seamless delivery of numerous services. Suppose a critical component breaks or an anomaly occurs during a satellite's operational lifespan; it may become partially or entirely inoperative and remain in orbit uncontrolled, exacerbating the congestion of Earth's orbits. As a result, outer space becomes less sustainable and poses a threat to active and future satellites [6, 7].

Some DSS research has previously been investigated to help solve their problems, including the management and protection of critical space infrastructure [8], optimizing the layout of satellite equipment [9], and solving the scheduling problem of a satellite constellation that observes Earth [10].

Integration of DSS with AI is proposed to improve the identification of satellite failures and reduce space debris produced by failed satellites that crowded the orbit. The prevention of satellite failure during its operational orbits is required. A robust approach is demanded to have a better point of view in supporting any decision made to prevent the failure of the satellite and to make the satellite's lifespan longer. The approach provides identifying failures, providing insight, and offering recommendations to improve the overall reliability of the satellite. Satellite reliability is a critical component on which this study is focused, especially for satellite owners and satellite manufacturers.

To contribute to SSA and help the satellite stakeholders, Web AI-based DSS is proposed and developed. Web AI-based DSS uses AI to process large volumes of data, recognize patterns, and generate actionable insights and recommendations. Two of the main components of DSS fully utilize AI, including the machine learning-based trade-space exploration model (TSE-ML), which is applied as a DSS model base to classify the reliability of the satellite according to the operational age over the expected design life. Generative AI is applied as the knowledge base acting as satellite experts to provide insight and recommendation to mitigate failures that occur on the satellites, allowing the stakeholder to make a quick decision with accurate information. Furthermore, the advancement of AI-driven DSS raises concerns while promoting space sustainability by preemptively addressing satellite failures prior to their intended lifespan; consequently, it mitigates or reduces the risks and even benefits from the potential resources present in man-made debris.

The remainder of the paper is organized as follows. Section II describes related works to this research, and Section III focuses on the system design of the proposed Web AI-based DSS. Section IV describes the development of Web AI-based DSS. Section V provides testing and evaluation. The conclusion is presented in the last section of this paper.

2. LITERATURE REVIEW

2.1 Decision Support System

The widespread use of computers and the internet facilitates access to extensive data on any subject, including space and satellite system. Before data is presented to users, data must be scrutinized and analyzed to ensure that users derive optimal benefits, for example, in decision-making support. As an instrument of information technology, DSS allows users to integrate analytical models with databases and user-friendly software to support users in improving their decision-making processes, especially when dealing with large datasets where identifying the optimal choice can be challenging [11]. The significance of DSS lies in their ability to improve the quality, speed, and efficacy of decisions, especially in contexts marked by uncertainty and complexity.

Over six decades, DSS have evolved significantly, became very practical, and garnered substantial academic attention, particularly in the last 20 years [12]. The literature shows that DSS is widely used in various industries, such as medical, manufacturing, energy, space, and various other industries. For example, in the space industry, the management and protection of critical space infrastructure [8], optimizing the layout of satellite equipment [9], and solving the scheduling problem of a satellite constellation that observes Earth [10].

DSS supports decision makers to address problems that require more effort to be solved using standard procedural methods or tools, employing both human judgment and computational resources. A DSS typically utilizes models for problem analysis, as modeling facilitates the exploration of various strategies under different configurations. A DSS can also be integrated with other systems and applications and can be disseminated through network and web technologies [12]. DSS is designed to reduce the time that system engineers spend interfacing the systems and to guarantee that the general requirements of the satellite are satisfied [13]. The distinction between a DSS and an expert system lies in the fact that a DSS solely takes into account the information provided from the database and the input from the users [13].

In developing the DSS, there are four pillar components of the DSS that need more attention [13]:

1. Accessibility to data, internal or external data.
2. Creation of modeling functions that utilize the data.
3. User interface design that facilitates an interactive experience.
4. Optimization of data using different mathematical or non-mathematical means.

2.2 Satellite Anomalies Reliability

By definition, satellite anomalies can be defined as unexpected events that affect the normal operation of satellites [3]. These anomalies are frequently classified according to their severity and impact on the satellite's performance, varying from minor glitches to total mission failures [6]. A minor anomaly may involve a transient communication failure, while a severe anomaly could lead to a complete loss of the satellite. Comprehending and analyzing these anomalies is crucial for improving satellite design and functionality, as well as to guarantee the reliability of satellite-based services [6].

Reliability, as mentioned in [14], is the probability that the system will perform its designated function without failure over a defined time frame. Reliability is considered an essential engineering technology, classified as the probability of performance over a specified period of time and the optimal functionality of a product after its service launch [14]. In other words, reliability refers to the probability of executing a series of tasks accurately and within a specified time frame, while adhering to established requirements [15].

The expanding utilization of electronic devices and their increasing complexity have integrated reliability into all stages of the concept, design, development, manufacturing, testing, and delivery processes. Reliability is an increasingly critical factor in system configuration, including in the design of communication satellites, where reliability is a paramount consideration [16].

Satellite anomaly reliability can be defined as the probability of failure function period of the satellite compared to its design life [3, 6]. In Mathematical terms, reliability R (t) can be illustrated as follows:

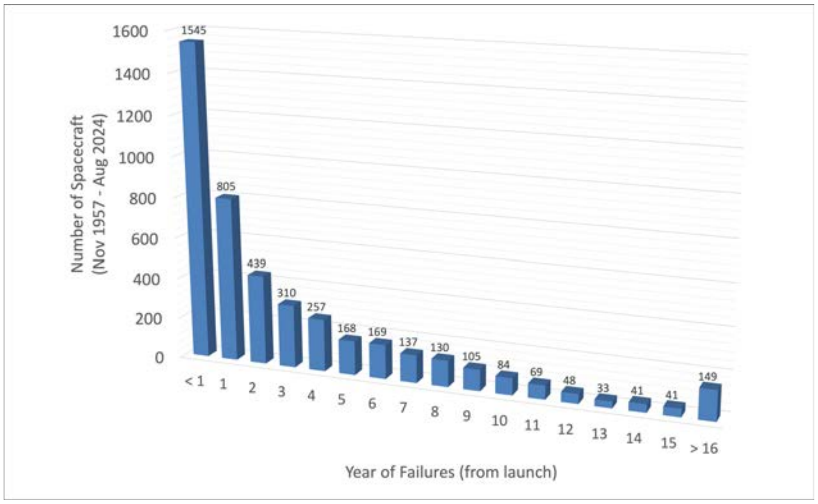

Fig. 3: Satellite Failures Age All Orbits (Seradata)

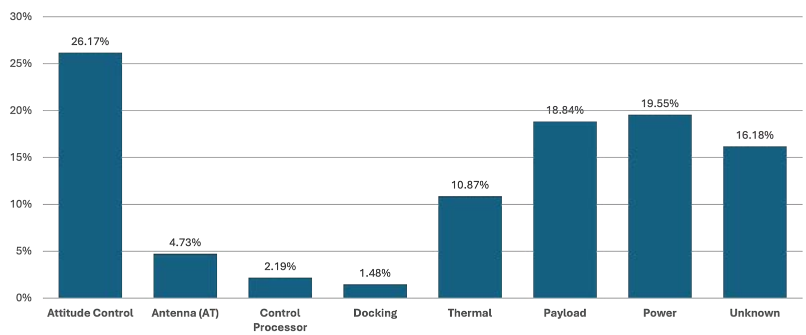

Fig. 4: Event Contribution in Satellite Failure $\leq$ 1 Year (Seradata)

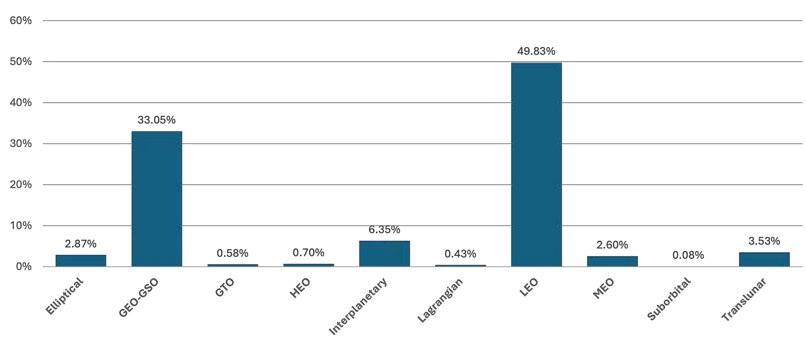

Fig. 5: Orbit Contribution in Satellite Failure $\leq$ 1 Year (Seradata)

$$R(t) = P(T > t) \tag{1}$$

Reliability problems in satellites during the last seven decades, as shown in Fig. 3, records suggest that up to 4530 satellites have encountered failures in various countries and orbits, resulting in substantial economic losses worldwide [6]. It also presents the number of satellites that failed during their year of operation from 1957 to 2024. It is obvious that around 50 percent of failure happened during their first year of operation. The most causes of these failures as shown in Fig. 4 are altitude control, power, and payload, which are around 26 percent, 20 percent, and 19 percent of the total failure during their first year, respectively. Unlike Fig. 5, the LEO orbit is the most prominent orbit that contributes to first year failures, which is around 50 percent. This means that small satellites are prone to failure during their first year of operation.

2.3 Artificial Intelligence

Artificial Intelligence (AI) can be defined as the ability of machines to emulate or replicate human behaviors, and has provided a comprehensive framework for various mathematical and technical inquiries [6]. As research has advanced over five decades, the parameters of this field have become increasingly indistinct, with humans progressively adopting and depending on machines that demonstrate varying levels of intelligence [6]. With the advancement of computational power, researchers are striving to improve their analysis by aggregating and employing increasingly extensive datasets while emphasizing enhanced collaboration and collective analysis [6]. With the ability of AI to integrate human performance and intelligence into machines, it is crucial in the development of intelligent, automated, decision support systems, and smart systems in various domains, including satellite anomalies.

Among the AI implementations in this study are ML and generative AI. ML, a subset of AI intimately associated with computational statistics, focuses primarily on prediction and pattern recognition within extensive datasets. Contemporary ML research encompasses diverse domains, such as natural language processing, computer vision, pattern recognition, cognitive computing, and knowledge representation, with predictive modeling serving as a fundamental application in industrial contexts [6]. ML techniques are classified into four categories: supervised, unsupervised, semi-supervised, and reinforcement learning. In supervised learning, models are trained on annotated data to accurately predict new inputs [6]. Unsupervised learning is used on unlabeled data to reveal hidden patterns or structures within the dataset [6]. Semi-supervised learning integrates labeled and unlabeled data to improve learning accuracy, particularly when labeled data is limited [6]. Reinforcement learning involves an agent engaging with its environment and acquiring knowledge to optimize rewards based on feedback from its actions, making it suitable for dynamic, decision-oriented contexts.

Generative AI denotes AI capable of producing novel content, as opposed to merely analyzing or responding to pre-existing data, similar to expert systems [17]. It was conceptually introduced with early ML algorithms that aimed to produce fresh data from previously learned patterns [18]. Modern generative AI incorporates a transformer model, which is trained on a corpus to map input information into a latent high-dimensional space, alongside a generator model that exhibits stochastic behavior, producing novel content with each test, even from identical input prompts [17]. However, expert systems specifically comprised a knowledge base and an inference engine that produced content through a database of if-else rules [17]. The following Table 1 presents well known Generative AI models such as Grok, Copilot, Gemini, Meta AI, Nvidia AI and GPT.

Meta AI with LLama 4 is the best option as a starting point for developing a prototype system. The reason being is that it is is fully open source, adaptable for research and free subscription.

2.4 Trade-Space Exploration

Trade-Space Exploration (TSE) is a systematic approach used to evaluate a range of possible solution alternatives, which encompasses a set of program parameters, system attributes, and performance characteristics. TSE is particularly valuable in complex engineering systems such as satellites, where it balances trade-offs between cost, schedule, risk, and performance to meet specific standards and requirements required by the stakeholders [37]. As a core technique in engineering design, TSE provides a structured framework to navigate intricate design decisions where trade-offs are inevitable [38]. By examining multiple dimensions of viable solutions, the TSE methodologies allow a thorough assessment of the strengths and limitations of each option, facilitating an informed selection of the most effective configuration for a given system [6]. Thus, TSE serves as a powerful tool for researchers to systematically evaluate and select design options, ensuring that the chosen solution aligns with the project goals and constraints [37].

The objective of TSE is to optimize a solution by balancing multiple parameters or objectives, such as classification accuracy, processing time, and computational cost in satellite anomaly detection. Mathematically, TSE can be formulated as a multi-objective optimization problem where we seek an optimal set of parameters for the satellite anomaly classification model.

Features	Grok	Nvidia AI	Meta AI	Gemini	Copilot	GPT
Company	xAI [19]	Nvidia [20]	Meta [21]	Google [19]	Microsoft [19]	OpenAI [19]
Latest Model	Grok 3 [19]	NeMo 1.21.0 [20]	Llama 4 [22]	Gemini 2.0 [19]	Copilot [19]	GPT-4 [19]
Launch Date	November 2023 [19]	October 2023 [20]	September 2023 [23]	March 2023 [19]	February 2023 [19]	November 2022 [19]
Capabilities	Text generation, visual processing, analyzing trends and interaction on social network [24]	Text, image, speech, and 3D model generation, advanced reasoning [25]	Text, image, and speech generation, code generation [21]	Text, image, and speech generation, coding, real-time streaming [24]	Text generation, image generation, productivity tools [26]	Text generation human-like text across diverse styles and topics, reasoning, coding, image generation [19]
Training Data	Diverse internet data, X-based model [24]	Diverse internet data, synthetic data [27]	Open-source data, Meta platforms data [21]	Google data and user interactions [24]	Microsoft data and user interactions [26, 28]	Diverse internet data [24]
Unique Features	Unfiltered answers, contextualized responses based on real-time discussions, potentially useful tool for social interactions and trend analysis [24]	High-performance AI with Tensor Cores, agentic AI capabilities [29, 30]	Deep integration with social media platforms, open-source models [21]	Multimodal capabilities, agentic experiences [24]	Integrated with Microsoft 365, productivity-focused [26, 28]	Advanced reasoning, multilingual support. [31]
Use Cases	Creative tasks, visual content creation [24]	Content creation, data analysis, automation, AI factories [25, 29]	Social media interactions, content creation, research projects [21]	Personal assistants, creative tools, educational aids, marketers, offering advanced support for data scientists. [24]	Office productivity, content creation, task automation [26, 28]	Chatbots, content creation, coding assistance, sentiment analysis, dialog creation. [31]
Cost	Subscription-based [32]	Subscription-based [33]	Free and subscription-based options [23]	Integrated with Google services [34]	Integrated with Microsoft 365 subscriptions [35]	Subscription-based (e.g., ChatGPT Plus) [36]
Strengths	Real-time capabilities, multimodal functions, and strong performance in coding and math tasks. [24]	High-performance AI with Tensor Cores, scalable AI solutions, advanced reasoning capabilities [29]	Fully open-source, adaptable for research and commercial use, strong integration with social media platforms [21]	Multimodal capabilities (text, image, speech), strong reasoning and explanation abilities, and advanced coding support. [24]	Excellent integration with Microsoft 365, enhancing productivity through automation and content generation within Office tools. [26]	Advanced reasoning, high accuracy in text generation, and strong multilingual support. [31]
Limitations	New entrant, potential stability issues, limited user base since it dependence to X platform which may restrict its application ability to other contexts and introduce informational bias [24]	High cost for hardware, requires specialized knowledge for optimization. [25]	Limited for broader AI tasks outside social media, slower adoption in non-social media contexts [21]	Slower response times, experimental stability issues, and reliance on precise prompt engineering for optimal performance. [24]	Privacy concerns due to data integration with Microsoft 365, and potential over-reliance on AI suggestions. [26]	Prone to generating biased or nonsensical responses, struggles with understanding complex contexts like sarcasm and humor, and can produce hallucinations (confident but incorrect information). [31]

Table 1: Comparison of Generative AI Models

3. PROPOSED WEB AI-BASED DECISION SUPPORT SYSTEM

3.1 Proposed Framework Design

The proposed Web AI-based DSS framework comprises a database, TSE-ML as a model base, Generative AI as a knowledge base, and web-based user interfaces to facilitate user access to the system, as illustrated in Fig. 6. The framework aims to optimize decision support making in terms of processing time and comprehensive information provided.

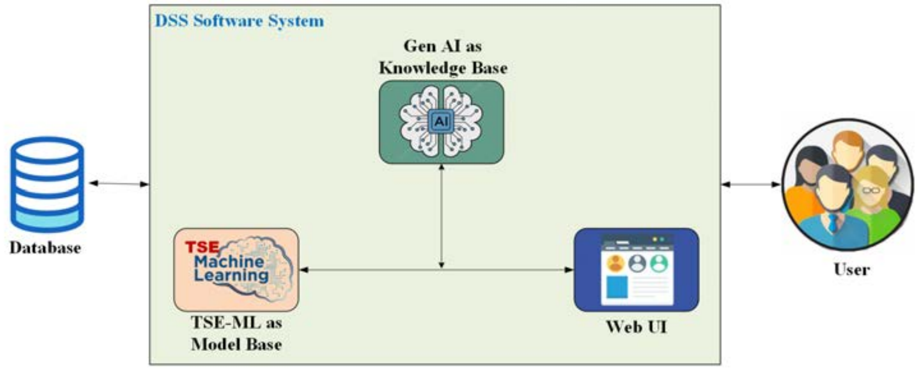

Fig. 6: Proposed Web AI-based DSS Framework

The database consists mainly of records of the set data from Seradata that have been refined based on needed research, including data on satellite failures during certain years and the final results of the DSS. Users will set their preferences for satellite information based on bus type, orbit, and mission. These preferences are accessible through web user interfaces (UI). The model base will process the data from the database query using TSE-ML to get the classification. The next step, both query result and model result, will be sent to the knowledge base to get insight and recommendations regarding of the satellite anomalies required by users. This result will be presented in the web UI as feedback from the user request. The detailed flow from user request to the final result of the proposed Web AI-based DSS is illustrated in Fig. 7.

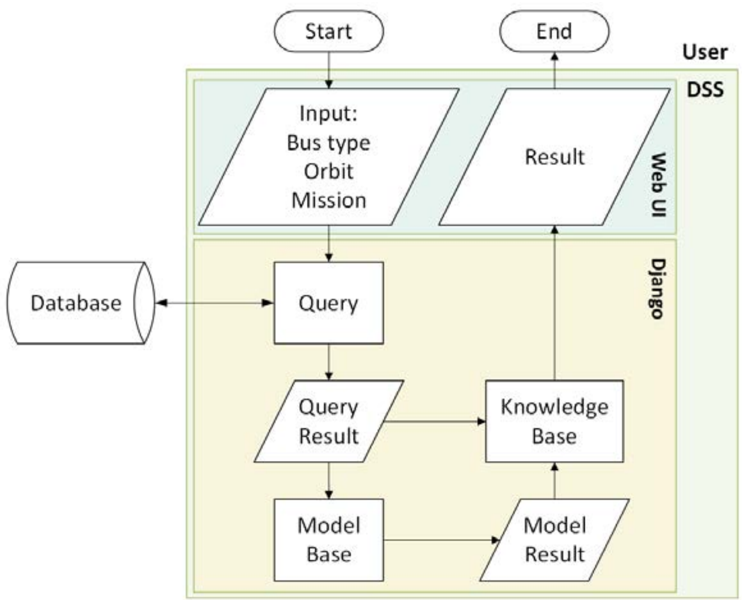

Fig. 7: Proposed Flow Chart Web AI-based DSS Framework

3.2 Environmental Setup

The experimental setup for this research is designed to ensure the repeatability and robustness of the Web AI-based DSS framework in satellite anomaly insight and recommendation. The experiments were carried out on a dedicated high-performance server with minimal background applications to eliminate external variables and ensure consistent results. The choice of hardware and software reflects the demanding computational requirements of large-scale satellite data analysis.

The hardware configuration as in Table 2 is optimized for intensive computational tasks including an Intel Core i9-

Category	Specification
CPU	Intel(R) Core i9-14900K, 3.20 GHz
GPU	NVIDIA GeForce RTX 4090
RAM	64.0 GB DDR5
Storage (SSD)	500 GB
Storage (HDD)	6 TB

Table 2: Hardware Specification

Category	Version
Windows Server Datacenter	2022
Python	3.11.9
Django	5.1.3
PostgreSQL	14
Bootstraps	5
Meta AI	LLama 4

Table 3: Software Specification

14900K processor, operating at a clock speed of 3.20 GHz, ensures high processing power and efficiency. With 64 GB of DDR5 RAM, the system supports large-scale memory-intensive operations, including handling the extensive Seradata dataset. Storage is divided between a 500 GB SSD for high-speed access to frequently used files and a 6 TB HDD for long-term storage, allowing seamless data management. The software environment as shown in Table 3 was curated to align with the computational need according to Web AI-based DSS, ensuring speed, efficiency, and reproducibility. Python 3.11.9 as a foundation, Django 5.1.3 as the back-end web framework, PostgreSQL 14 for the database management system (DBMS) and Bootstraps 5 for the front-end toolkit are applied in construction to produce a prototype of Web AI-based DSS. Meta AI LLama 4 is utilized as generative AI which has performance on par with OpenAI, but it is open-source and has fully open reasoning tokens. It is 671B parameters in size, with 37B active in an inference pass. The API for LLama 4 Meta AI is provided by openrouter.ai.

3.3 System Design

System design involves defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. The detailed explanation is given in the following sections.

3.3.1 System Architecture

In this architecture, the high-level structure of the system is defined by creating a blueprint that outlines how the components of the system will interact and work together to meet the requirements. The details of the system architecture are illustrated in Fig. 8. From Fig. 8, there are three sections involved. There are data, DSS and Openrouter as Meta AI LLama 4 models API provider.

In the data section, it consists of a database that contains access control management, DSS analysis, and curated data from Seradata. This data is accessible via the local area network. It means that the data and the DSS application are in the same area network. This local access includes the authentication process, dashboard access, data presentation, and part of the data analysis. The other part of the data analysis required an external connection or internet access. The analysis of data as data support for decision making utilizes the API provided by openrouter.ai to access several generative AIs including GPT, Copilot, Gemini, Meta AI, and Grok.

3.3.2 Component

Components are the building blocks of a system. They are individual units that perform specific functions and interact with other components to form a complete system. Each component is designed to be self-contained and reusable.

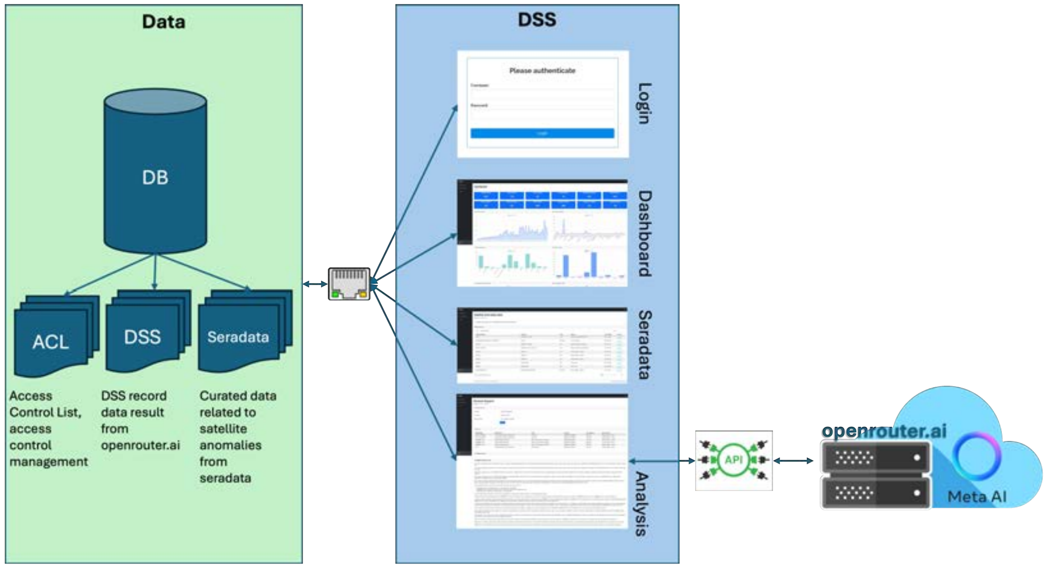

Fig. 8: System Architecture

The component, as illustrated in Fig.6, includes:

1. User Interfaces (UI): The part of the system with which users interact, such as the login page, dashboard page, form, and analysis page, as illustrated in Fig. 9 to Fig. 12.
2. Business Logic: The component that processes data and implements the rules and workflows of the system.
3. Database: The component that stores and manages data.
4. API (Application Programming Interface): The component that allows different parts of the system to communicate with each other and with external systems such as openrouter.ai as illustrated in Fig. 8.

3.3.3 Module

Modules are logical groupings of related components that work together to perform a specific function or set of functions within the system. They help to organize the system into manageable sections, making it easier to develop, maintain, and scale. The modules are illustrated in Fig. 8 as follows:

1. Authentication Module: Handles user login, registration, and access control.
2. Reporting Module: Collection of reports and visualizations based on system data, in this system it is grouped as a dashboard as illustrated in Fig. 9.
3. Data Module: Manages data collection from Seradata as shown in Fig. 10 and Fig. 11.
4. Analysis Module: Module that presents the result of the analysis as illustrated in Fig. 12. In this module, 5 sections include the form, table result, incident overview, reliability trend and insight, stakeholder recommendation, and summary.

3.3.4 Data

Data is the information that the system processes, stores, and manages. It is a critical part of the system design, as it drives the functionality and decision-making capabilities of the system. Data can be structured or unstructured and are typically stored in databases. Fig. 13 illustrates the schema of the database in this study.

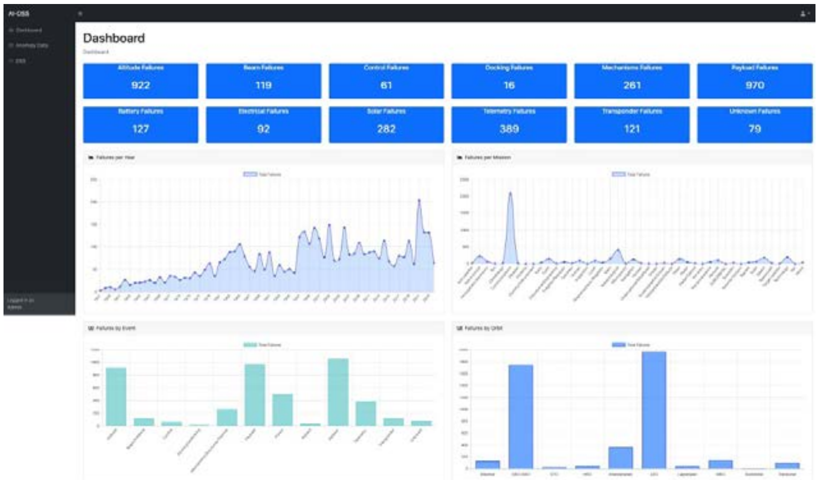

Fig. 9: Dashboard

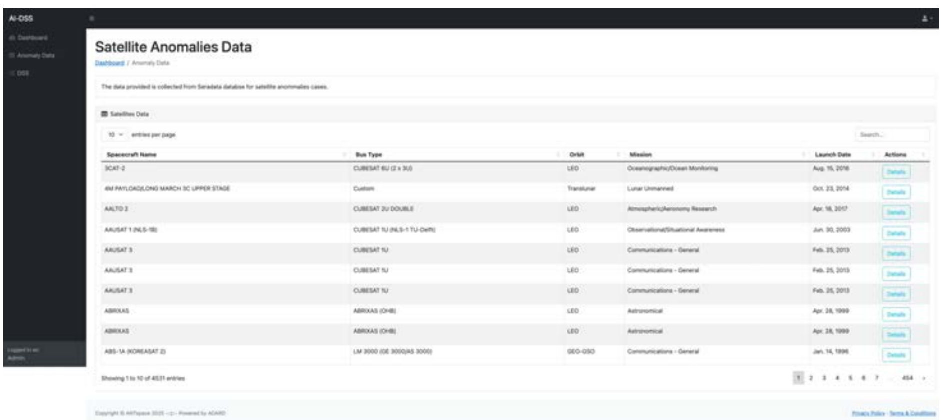

Fig. 10: Collection of Data

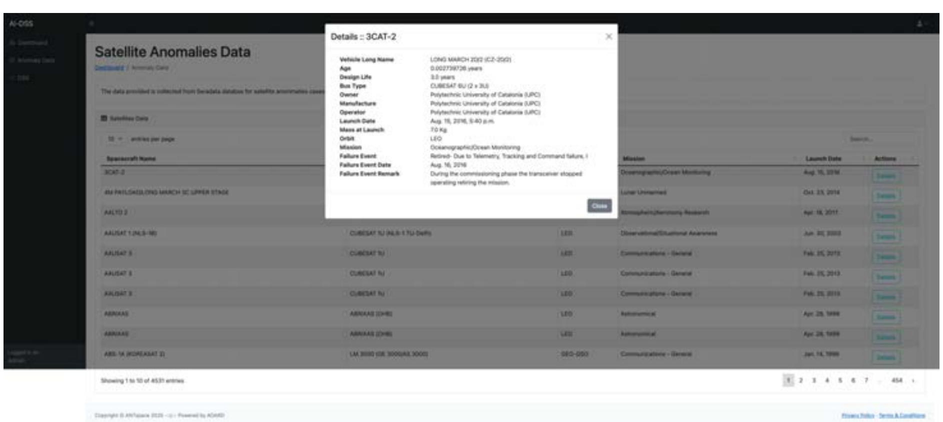

Fig. 11: Details of Data

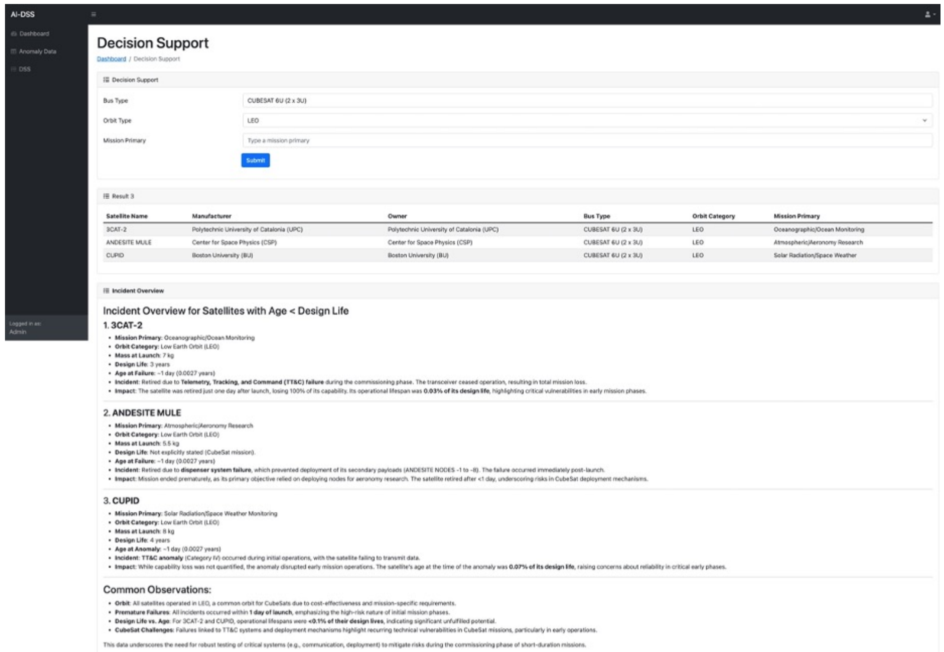

Fig. 12: Data Analysis

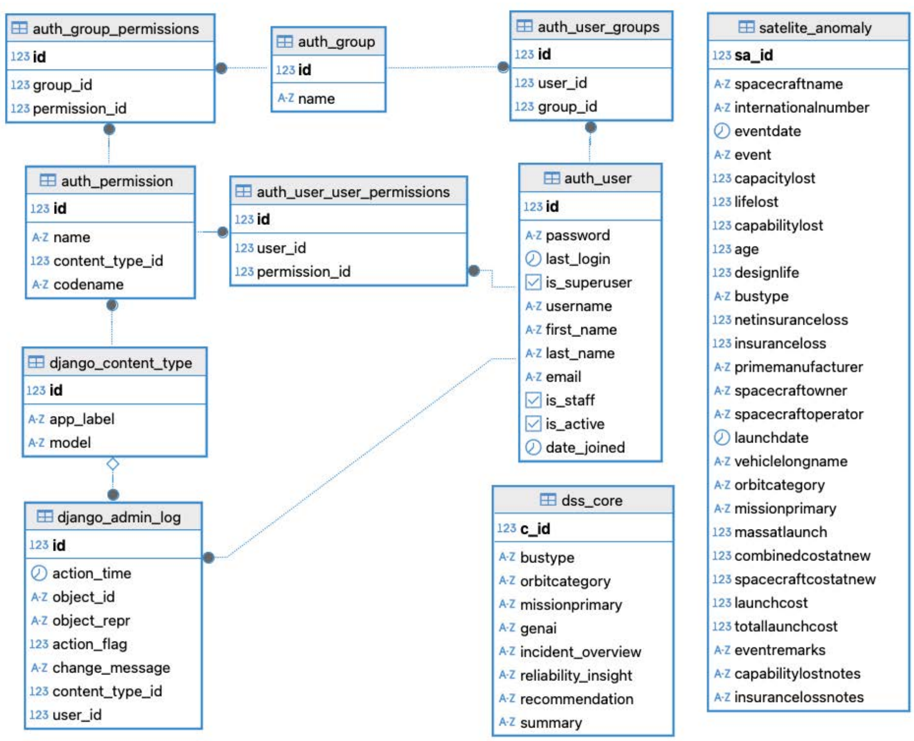

Fig. 13: Database Diagram

4. RESULT AND DISCUSSION

4.1 Web AI-based DSS Prototype Development

The development process, namely environment setting, code building, and testing, is explained in the next sections.

4.1.1 Environment Setting

To guarantee a seamless and effective workflow, setting up an environment for Django development is an essential step. Since Django is a framework based on Python, the first step in the procedure is to install Python. The most recent version of Python must be downloaded from the official website and added to the PATH on the computer.

```
pip --version
```

Listing 1: PIP Version

Installing pip, the Python package installer, comes next after installing Python. Pip is usually included with Python, but it can also be installed individually if necessary. Listing 1 shows how to verify the *pip* version.

For Django projects, setting up a virtual environment is strongly recommended. This procedure maintains the project separate from the global Python environment and aids in dependency management. The *venv* module allows to build a virtual environment that contains all the required libraries and packages needed. To create a new virtual environment, it can follow a command-line as illustrated in Listing 2.

```
python -m venv antspace
```

Listing 2: New Virtual Environment Creation

```
source antspace/bin/activate
```

Listing 3: Virtual Environment Activation

With the virtual environment activated as in the Listing. 1, we can proceed to install Django using pip. This installation is straightforward, as can be seen in Listing 4 and ensures that Django and its dependencies are correctly set up within the virtual environment. Verifying the installation by checking the Django version helps as shown in Listing 5 confirm that everything is in place. Finally, we can start a new Django project using the *django-admin startproject* command as illustrated in Listing 6, which sets up the basic structure of your project.

```
pip install django
```

Listing 4: Django Installation

```
django-admin --version
```

Listing 5: Django Installation Verification

```
django-admin startproject satellite_dss
```

Listing 6: Django Project Initialization

The command in the listing 6 will produce a new directory called *satellite_dss* with the following structure:

```
satellite_dss/
    manage.py
    satellite_dss/
        __init__.py
        settings.py
        urls.py
```

```
            asgi.py
            wsgi.py

```

Listing 7: Structure Directory

Under *satellite_dss* directory we have several files and a directory with the same name as our project folder. *manage.py* which consists of a set of command-line utility for interacting within the project such as starting the server as shown in Listing 8 and synchronizing the database. *settings.py* is the website or project setting that includes the registration of any created applications, the location of static files, and the details of the database configuration. *urls.py* is the file in which all URL declarations and functions to call are stored within the projects. While *asgi.py* and *wsgi.py* are both entry points for ASGI and WSGI compatible web servers.

To run the development server, navigate to the project directory and run the development server using the following code:

```
cd myproject
python manage.py runserver

```

Listing 8: Running Development Server

```
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
May 01, 2025 - 08:12:08
Django version 5.1.3, using settings 'satellite_dss.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

```

Listing 9: Server Running Message

The output message will appear, indicating that the server is running as in Listing 9. To access the web page, open a web browser and go to http://127.0.0.1:8000 to see the Django welcome page.

In general, setting up a dedicated environment for Django development not only streamlines the development process but also enhances project management, security, and reproducibility. It allows developers to work on multiple projects with different dependencies without interference, making it an essential practice for any Django developer.

4.1.2 Code Construction

Django is well-known for its unique and well-managed application structure. An application can be developed as a fully standalone module with all the features. For instance, distinct modules for login/logout, dashboard, data, and analyze.

4.1.2.1 Create New DSS App

The first step in the code construction is creating a new application named as *dss*. The following command line in Listing 10 was used to create the DSS app for this study.

```
python manage.py startapp dss

```

Listing 10: Creation of a New App

The above command line will produce a new directory called **dss** as illustrated in Listing 11 below:

```
dss/
    __init__.py
    admin.py
```

```
            apps.py
            models.py
            tests.py
            views.py
            migrations/

```

Listing 11: New Application Directory Structure

There are five important files in the *dss* directory, namely *admin.py*, *apps.py*, *models.py*, *test.py*, *views.py* and 1 folder with *migration* name. *admin.py* is a configuration file for the Django admin interface, *apps.py* consists of configuration for the *dss* application, *models.py* is where we define our database models, *test.py* to write the tests, *views.py* will be the file to define the views and folder *migration* contains the database migrations for the models.

4.1.2.2 Defining Models

The next step is to define the models as the whole process in this DSS system will store and use data to and from the database. Django models are a powerful tool for defining the structure of database tables with Python classes. Each model corresponds to a single database table and includes fields representing the table's columns. Django models also provide an automatically built database-access API, making it easy to communicate with the database without writing SQL queries.

Listing 12 is the code construction for the model to define the dss_core based on the table from the database, as shown in Fig. 12 above.

```python
from django.db import models

class DSSCore(models.Model):
    c_id = models.AutoField(primary_key=True)
    bustype = models.CharField(max_length=100, blank=True, null=True)
    orbitcategory = models.CharField(max_length=100, blank=True, null=True)
    missionprimary = models.CharField(max_length=100, blank=True, null=True)
    genai = models.CharField(max_length=100, blank=True, null=True)
    incident_overview = models.TextField(blank=True, null=True)
    reliability_insight = models.TextField(blank=True, null=True)
    recommendation = models.TextField(blank=True, null=True)
    summary = models.TextField(blank=True, null=True)

    class Meta:
        managed = False
        db_table = 'dss_core'

```

Listing 12: Model for DSS Core

After defining the models, it is necessary to create the corresponding database tables. This is accomplished through Django's migration system. First, generate the migration files with the *makemigrations* command as shown in Listing 13:

```
python manage.py makemigrations

```

Listing 13: Corresponding Database Tables

If for an example we do not have any tables in database yet, use the *migrate* command to apply the migrations and construct the tables in the database:

```
python manage.py migrate

```

Listing 14: Database Table Generation

The command in Listing 14 will generate the necessary SQL statements to create the tables and apply them to the database.

4.1.2.3 Constructing Views

A view function is a Python function that accepts a web request and provides a web response. This response could be the HTML contents of a web page, a redirect, a 404 error, an XML document, an image, or anything else. The view itself contains whatever arbitrary logic is necessary to get the response. This code can be placed anywhere as long as it is on the Python path. Views are usually saved in a file called *views.py*, placed in a project or an application directory.

The Listing 15 a class contains a function to obtain data from the model *SateliteAnomaly* in the *models.py* file.

```
class AnomalyData(TemplateView):
template_name = 'dss/anomalydata.html'

def get_context_data(self, **kwargs):
    context = super().get_context_data(**kwargs)
    context['anomaly'] = SateliteAnomaly.objects.all()
    return context

```

Listing 15: Loading the Satellite Anomaly Data

The following code in Listing 16 is the way to call the Listing 15 function to present the data value in the web browser.

```
{% for anomaly in satellite_anomaly %}
    <tr>
        <td>{{ anomaly.spacecraftname }}</td>
        <td>{{ anomaly.bustype }}</td>
        <td>{{ anomaly.orbitcategory }}</td>
        <td>{{ anomaly.missionprimary }}</td>
        <td>{{ anomaly.launchdate|date:"M. d, Y" }}</td>
    </tr>

```

Listing 16: Satellite Anomaly Data Presentation

It can be seen from the above code that the statement of *for anomaly in satellite_anomaly* means to extract data value to be presented on the web page properly.

4.1.3 Testing and Evaluation

Testing and evaluation are performed to ensure that the system is feasible and reliable. This section was conducted during May and June 2025. The testing is done for every module functionality including the whole unit program in modules to evaluate whether the module is functioning as intended. The purpose of this test is to conduct an investigation to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective independent view of the software to allow stakeholders to appreciate and understand the risks of software development and implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or defects).

The method used in this testing and evaluation is the black-box. Black-box testing treats software as a “black box” without knowing of the internal implementation. In addition, the black-box tester does not need a “bind” to the code, and the perception of the tester is very simple: a code must have bugs. Thus, the tester simply enters the data and focuses only on the output of the test object. This method simplified the system for end users with different backgrounds to verify [39].

The testing and evaluation are focused on 4 units program including Login/Logout, Dashboard, Anomaly Data, and Analysis pages as shown in Table 2 to Table 5. As a result, Table 2 describes the test and evaluation for the Login/Logout unit program. This part is very important, as the login is the gate of a system that has many vital functions for the DSS. Whereas both Table 3 and Table 4 are Dashboard and Anomaly Data, respectively, which focus on the successful retrieval of data from the database and their presentation in chart and table. While Table 5 demonstrates the analysis unit program which focuses on functionality of the form including auto complete input in both Bus Type and Mission Primary, connectivity between system and Openrouter API to send request and get response, presentation of the result, and PDF format downloadable file. Finally, by this testing and evaluation, the development and the end-user expectation are met.

No	Unit Program	Expected Result	Actual Result
1	Login/Logout	1.) User login to enter the system with encryption. 2.) Logout from the system with destroy the session.	Ok

Table 4: Login Testing Result

No	Unit Program	Expected Result	Actual Result
2	Dashboard	1.) Successfully retrieve the data from database. 2.) All charts and tabled are well presented.	Ok

Table 5: Dashboard Testing Result

No	Unit Program	Expected Result	Actual Result
3	Anomaly Data	1.) Successfully retrieve the data from database. 2.) All data for satellite anomaly are well presented in table.	Ok

Table 6: Anomaly Data Testing Result

No	Unit Program	Expected Result	Actual Result
4	Analysis	1.) Auto correct is well performed for Bus Type and Mission Primary input form. 2.) Successfully retrieve the data from database based on filter form (Bus Type, Orbit Type and/or Mission Primary. 3.) Result data from database is well presented in Table. 4.) Successfully send prompt based on the result data to Openrouter API (Meta LLama 4 Maverick - free) and get response from it. 5.) Analysis result from Openrouter API response is well presented. 6.) Download the analysis report as PDF format.	Ok

Table 7: Analysis Testing Result

No	Unit Program	Expected Result	Actual Result
5	Generative AI Analysis	1.) Is the analysis result presented in an easy to understand language? 2.) Is the analysis result address potential biases? 3.) Is the analysis result have potential real-world impacts? 4.) How is the accuracy and quality of the analysis result? 5.) How coherence of the analysis result?	Ok

Table 8: Generative AI Analysis Testing Result

Prompt for Incident Overview

”As an expert in the satellite area, write the incident overview related to mission primary, orbit category, mass at launch, age and design life especially for age less than design life based on the data as follows: table result provided by Seradata based in user input.”

As referenced for the test and evaluation results in Table 8, the prompt above was used to generate the result analysis. This prompt serves as input to the language model which guided it in producing a comprehensive evaluation based on the specified parameters and context of the experiment, such as generated incident overview as follows:

Incident Overview

Incident Overview:
Mission Primary: Oceanographic/Ocean Monitoring
Orbit Category: LEO (Low Earth Orbit)
Mass at Launch: 7.0 kg
Age: 0.002739726 years (approximately 1 day)
Design Life: 3.0 years
The 3CAT-2 satellite, launched on August 15, 2016, experienced a critical failure just a day after launch, resulting in the loss of its mission. The failure was attributed to a Telemetry, Tracking, and Command (TT&C) failure, which led to the retirement of the spacecraft.

Key Observations:
The age of the spacecraft at the time of failure (0.002739726 years) is significantly less than its design life (3.0 years), indicating a premature failure.

The mission primary was Oceanographic/Ocean Monitoring, suggesting that the satellite was intended to play a crucial role in monitoring ocean health and dynamics.

The satellite's mass at launch was 7.0 kg, which is relatively small, consistent with its classification as a 6U CubeSat. The orbit category was LEO, which is a common orbit for Earth observation and remote sensing missions.

Incident Summary:
The 3CAT-2 satellite failed due to TT&C failure during its commissioning phase, resulting in a total loss of capability and mission termination. The incident highlights the challenges associated with ensuring the reliability and longevity of small satellites, particularly during the early stages of their operation. Given that the age of the spacecraft at failure was significantly less than its design life, this incident underscores the need for robust design, testing, and validation procedures to ensure that satellites can withstand the rigors of launch and early operation.

5. CONCLUSION

The prototype of the Web AI-based DSS system has been developed and the necessary information related to satellite anomaly identification has been provided, which can help stakeholders support their decisions for their next satellite launch program. By its design, it covers all the needs and requirements of the DSS, including incident overview, trend, insight, and recommendation. This study also plays a role in supporting space situational awareness (SSA) to make outer space safe and sustainable by reducing debris caused by satellite failures after their launch.

Moreover, this prototype provides actionable information on satellite design, operational planning, and anomaly mitigation, allowing for more reliable and efficient systems in increasingly congested orbital environments. Furthermore, this work advances the integration of AI for satellite anomaly identification in Web AI-based DSS, laying the groundwork for scalable, transparent, and successful anomaly identification. Future work will focus on expanding the integration of further Generative AI and investigating its applicability to larger aerospace systems to improve satellite reliability and operational resilience while enabling SSA.

The testing and evaluation showed positive results, where no bugs or defects were found. The validation of the analysis result has been validated by the expert and has good feedback.

Finally, this prototype of Web AI-Based DSS can be used as the foundation for the next development, for example, by comparing some of the Generative AI models mentioned in Table 1.

6. ACKNOWLEDGMENT

The authors would like to thank the Asian Office of Aerospace Research (AOARD) for the monetary assistance under the grant scheme numbers FA2386-23-1-4073 and SPI23-179-0179. The authors also thank the IIUM CPS Khair Award for the tuition fee coverage.

7. REFERENCES

[1] Jesse Klempner Andrew Sierra Alizée Acket-Goemaere, Ryan Brukardt and Brooke Stokes. Space: The $1.8 trillion opportunity for global economic growth. Technical report, Mckinsey Company, 2024.

[2] John A Kennewell and Ba-Ngu Vo. An overview of space situational awareness. In *Proceedings of the 16th International Conference on Information Fusion*, pages 1029–1036, 2013.

[3] Abdul Mutholib, Nadirah Abdul Rahim, Teddy Surya Gunawan, and Mira Kartiwi. Trade-space exploration with data preprocessing and machine learning for satellite anomalies reliability classification. *IEEE Access*, 13:35903–35921, 2025.

[4] Philip Hylands. Spacetrak briefing: Introduction to space general information. Technical report, Seradata, 2014.

[5] Taha Tetik, Gulesin Sena Daş, and Burak Birgoren. A multi-objective perspective to satellite design and reliability optimization. *Expert Systems with Applications*, 246:123178, 2024.

[6] Abdul Mutholib, Nadirah Abdul Rahim, Teddy Surya Gunawan, and Ahmad Afif Ahmarofi. Performance comparison of data preprocessing methods for trade-space exploration with ai model: Case study of satellite anomalies detection. *2024 IEEE 10th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, pages 271–275, 2024.

[7] J. C. Green, Robert Quinn, T. Paul O'Brien, Jakob Likar, Yuri Y. Shprits, Alexander J. Boyd, Gascoin S., Claudepierre, Stuart L. Huston, P. Whelan, and N. Reker. Developing a comprehensive application for satellite anomaly analysis and attribution. 2020.

[8] Devon L Brown, Danielle Ciesielski, and Duli Chand. Decision support systems for critical space infrastructure assets. *A Journal of Strategic Airpower & Spacepower*, 2:76–90, 2023.

[9] Liang Yan'gang and Qin Zheng. A decision support system for satellite layout integrating multi-objective optimization and multi-attribute decision making. *Journal of Systems Engineering and Electronics*, 30(3):535–544, 2019.

[10] Pei Wang, Gerhard Reinelt, Peng Gao, and Yuejin Tan. A model, a heuristic and a decision support system to solve the scheduling problem of an earth observing satellite constellation. *Computers Industrial Engineering*, 61(2):322–335, 2011. Combinatorial Optimizatiion in Industrial Engineering.

[11] Edrisi Munoz, Elisabet Capon-Garcia, and Luis Puigjaner. Advanced model design based on intelligent system characterization and problem definition. In Anton A. Kiss, Edwin Zondervan, Richard Lakerveld, and Leyla Özkan, editors, *29th European Symposium on Computer Aided Process Engineering*, volume 46 of *Computer Aided Chemical Engineering*, pages 1045–1050. Elsevier, 2019.

[12] Maria Rashidi, Maryam Ghodrat, Bijan Samali, and Masoud Mohammadi. *Decision Support Systems*, pages 19–38. IntechOpen, 10 2018.

[13] Ahmed Ali Ahmed Albuainain. Reliability enhancement decision support system tool for small satellites. Master's thesis, Khalifa University of Science and Technology, May 2021. Available at `https://khazna.ku.ac.ae/ws/portalfiles/portal/6813710/file`.

[14] N. S. A. Abdul Rahman and N. A. Rahim. Analysis of reliability prediction and maintainability activities of an earth station system using parallel configurations. In *WCSE 2022 Spring Event: 2022 9th International Conference on Industrial Engineering and Applications)*, pages 1647–1655, 2022.

[15] Nur Shazana Abdul Rahman and Nadirah Abdul Rahim. Sustainable framework for a geostationary satellite control earth station system using parallel configuration. *Indonesian Journal of Electrical Engineering and Computer Science*, 30(3):1498–1508, 2023.

[16] Nadirah Abdul Rahim and Nor'afifah Nordin. Reliability model and proposed maintainability activities of earth station system. *International Journal of Electrical Electronics and Data Communication*, 8:20–24, 05 2020.

[17] Roberto Gozalo-Brizuela and Eduardo C. Garrido-Merchan. Chatgpt is not all you need. a state of the art review of large generative ai models, 2023.

[18] Mahayaudin M. Mansor, Nurain Ibrahim, Noor Fadhilah Ahmad Radi, Nadirah Abdul Rahim, Syarul Heiry

Yahaya, Mohd Aftar Abu Bakar, and Roslinazairimah Zakaria. Discrepancy and thematic bibliometric analyses of the remaining limitations in artificial intelligence. In *2024 IEEE International Conference on Computing (ICOCO)*, pages 485–490, 2024.

[19] Álvaro Cabezas-Clavijo and Pavel Sidorenko-Bautista. Assessing the performance of 8 ai chatbots in bibliographic reference retrieval: Grok and deepseek outperform chatgpt, but none are fully accurate. *arXiv preprint arXiv:2505.18059*, 2025.

[20] Nemo 1.21.0. https://github.com/NVIDIA/NeMo/releases/tag/v1.21.0. Accessed: 2025-06-10.

[21] Glen Nolasco and John Dave Dicuangco. A review article – facebook's meta ai: A potential boon or doom for learning? *International Journal of Metaverse*, 3(1):1–7, Apr. 2025.

[22] Llama 4. https://www.llama.com/. Accessed: 2025-06-10.

[23] Meta plans paid subscription service for chatbot meta ai, source says. https://www.reuters.com/technology/artificial-intelligence/meta-plans-release-standalone-meta-ai-app-cnbc-reports-2025-02- Accessed: 2025-06-10.

[24] Murillo Edson de Carvalho Souza and Li Weigang. Grok, gemini, chatgpt and deepseek: Comparison and applications in conversational artificial intelligence. *INTELIGENCIA ARTIFICIAL*, 2(1), 2025.

[25] Exploring nvidia nemo framework: An in-depth overvie. https://www.linkedin.com/pulse/exploring-nvidia-nemo-framework-in-depth-overview-mike-shen-7p71c. Accessed: 2025-06-10.

[26] R.D. Balaji, Said Jaboob, Senthil Kumar, and Samskrati Gulvady. *AI-Powered Academia Enhancing Efficiency Through Copilot and Microsoft Integration*, pages 293–302. 02 2025.

[27] Curating custom datasets for llm training with nvidia nemo curator. https://developer.nvidia.com/blog/curating-custom-datasets-for-llm-training-with-nvidia-nemo-curator/. Accessed: 2025-06-10.

[28] Harry Peter. The future of digital collaboration integrating copilot across the microsoft 365 ecosystem. 04 2025.

[29] Nvidia nemo agent toolkit. https://developer.nvidia.com/nemo-agent-toolkit. Accessed: 2025-06-10.

[30] Nvidia nemo framework. https://docs.nvidia.com/nemo-framework/user-guide/24.07/overview.html. Accessed: 2025-06-10.

[31] Moatsum Alawida, Sami Mejri, Abid Mehmood, Belkacem Chikhaoui, and Oludare Isaac Abiodun. A comprehensive study of chatgpt: Advancements, limitations, and ethical considerations in natural language processing and cybersecurity. *Information*, 14(8), 2023.

[32] xai models and pricing. https://docs.x.ai/docs/models?cluster=us-east-1#detailed-pricing-for-all-grok-models. Accessed: 2025-06-10.

[33] Nvidia ai enterprise licensing guide. https://docs.nvidia.com/ai-enterprise/planning-resource/licensing-guide/latest/index.html. Accessed: 2025-06-10.

[34] Gemini subscription. https://gemini.google/subscriptions/?hl=en. Accessed: 2025-06-10.

[35] Copilot subscription. https://www.microsoft.com/en/microsoft-365/copilot/pricing. Accessed: 2025-06-10.

[36] Chatgpt pricing. https://openai.com/chatgpt/pricing/. Accessed: 2025-06-10.

[37] Nadirah Binti Abdul Rahim and Timothy L. J. Ferris. A method to establish a trade-space of system requirements and life cycle cost. *IEEE Systems Journal*, 14(1):1257–1264, 2020.

[38] Julia Daniels, Cameron Turner, J. Wagner, Nafiseh Masoudi, Malena Agyemang, Gregory Hartman, Denise Rizzo, David Gorsich, Annette Skowronska, and Rachel Agusti. Designing the design space: Evaluating best practices in tradespace exploration, analysis and decision-making. In *SAE International Journal of Advances and Current Practices in Mobility*, volume 4, 03 2022.

[39] Abdul Mutholib, Teddy Surya Gunawan, and Mira Kartiwi. Development of fyp online system for outcome based education. In *TENCON 2011 - 2011 IEEE Region 10 Conference*, pages 1312–1316, 2011.