AlertHive: Real-Time Disaster Intelligence and Response Dashboard

Tanisha Agrawal[1], Sudhanshu sharma[2], Suryansh Pratap Singh[3], Vikas Dayma[4], Vickey Bagde[5], Urvashi Sharma[6]

Computer Science & Engineering Acropolis Institute of Research and Technology Indore, India

***Abstract*—Disaster management demands quick access to reliable information for timely response. AlertHive: Real-Time Disaster Intelligence and Response Dashboard is an intelligent system that aggregates disaster-related data from social media, news APIs, and public sources. Using Natural Language Processing (NLP) and Machine Learning (ML), it extracts key details such as location, disaster type, and severity, while filtering irrelevant or duplicate data. The processed information is displayed on a real-time, interactive dashboard for emergency responders and agencies. AlertHive addresses limitations of existing systems by ensuring faster updates, improved data credibility, and broader local coverage. By reducing information overload and enhancing situational awareness, the platform enables quicker, more informed decision-making—ultimately improving coordination and response efficiency during disaster events.**

***Index Terms*—Disaster Management, Real-Time Dashboard, Natural Language Processing (NLP), Machine Learning (ML), Data Aggregation, Situational Awareness, Emergency Response .**

I. INTRODUCTION

Disasters, whether natural or human-induced, pose significant challenges to societies across the globe. Rapid urbanization, climate change, and increasing population density have amplified the frequency and impact of such events. In these critical situations, timely and accurate information becomes essential for effective response and resource allocation. However, disaster management agencies often struggle to gather credible data due to the overwhelming amount of unverified information scattered across various online platforms.

Traditional disaster monitoring systems rely heavily on manual data collection or limited official reports, which leads to delays in information dissemination. During emergencies, such delays can result in poor coordination, misinformed decisions, and loss of valuable time that could otherwise save lives. The emergence of digital media and open data platforms provides immense opportunities to enhance situational awareness, but the challenge lies in efficiently filtering and analyzing this data in real time.

The project AlertHive: Real-Time Disaster Intelligence and Response Dashboard addresses these challenges by providing an automated, intelligent solution for disaster information aggregation. It leverages Natural Language Processing (NLP) and Machine Learning (ML) technologies to collect, classify, and visualize disaster-related data from multiple sources such as social media, news APIs, and public data feeds. The system ensures that only relevant and verified information reaches disaster response authorities.

AlertHive's core objective is to minimize information overload while maximizing the accuracy and speed of information delivery. Through automated data extraction and classification, it identifies the type, location, and severity of disasters, presenting them in a clear and organized manner on an interactive dashboard. This assists emergency responders, government agencies, and NGOs in monitoring evolving situations and making well-informed, timely decisions.

The project stands apart from existing systems such as Google Crisis Map, AIDR, GDACS, and Ushahidi by integrating multi-platform data and employing advanced AI-driven filtering mechanisms. Unlike traditional tools that depend on static or delayed data, AlertHive provides continuous real-time updates, reduces misinformation, and focuses on localized incidents that often go unnoticed by global systems. Its modular architecture ensures scalability, reliability, and adaptability to diverse disaster scenarios.

In essence, AlertHive aims to transform disaster intelligence through technology-driven innovation. By automating the collection and verification of disaster data, it bridges the gap between raw information and actionable insights. The platform not only improves operational efficiency for response teams but also lays a foundation for future advancements such as multimedia analysis, multilingual data processing, and enhanced collaboration among agencies—making it a vital step toward smarter, faster, and more effective disaster management.

II. BACKGROUND AND RELATED WORK

Effective disaster management relies heavily on the ability to gather, process, and interpret large volumes of information from diverse sources. Over the years, several systems have been developed to address this need, such as Google Crisis Map, AIDR (Artificial Intelligence for Disaster Response), GDACS (Global Disaster Alert and Coordination System), Twitter Alerts, and Ushahidi. While these platforms have contributed significantly to improving situational awareness, they possess notable limitations. For instance, Google Crisis Map provides detailed geospatial data but lacks real-time social media integration, while AIDR uses machine learning for message categorization but requires manual intervention for training. Similarly, GDACS focuses primarily on large-scale global disasters, leaving smaller or localized incidents underreported.

These limitations highlight the growing need for a more comprehensive and automated system that can process real- time data across multiple platforms with minimal human involvement. *AlertHive* builds upon these existing efforts by introducing an intelligent, NLP-powered data aggregation mechanism capable of filtering noise, validating sources, and categorizing disaster events instantly. Unlike earlier solutions, it bridges the gap between official and crowd- sourced data, ensuring faster, data-driven decision-making. This integration of real-time social media analytics, machine learning classification, and interactive visualization distinguishes AlertHive from previous models and establishes it as a next-generation tool for effective disaster intelligence and emergency response.

Here is an overview of five existing systems related to our system highlighting their address, gap identification and reference links:

TABLE I. TABLE OF FIVE EXISTING SYSTEM

Existing System				
S. No.	Problem Addressed	Gap identification	Draw-backs	Advantages
1.	Google Crisis Map	Limited crowd- sourced and real- time updates	Lacks real- time social media integration; data may become outdated	Integrats multiple data layers (weather ,hazards, routes)
2.	AIDR (AI for Disaster Response)	Slow to adapt to new disaster types	Needs manual training; limited to few platforms	Uses ML to classify disaster-related social media posts
3.	GDACS (Global Disaster Alert and Coordination System)	Limited local coverage and social media input	Focuses on major disasters only; no social data	Combines global data from multiple agencies
4.	Twitter Alerts	No event categorization or data aggregation	Restricted to verified users; noisy data	Provides real- time alerts via social media
5.	Ushahidi	Lacks automation and AI-based classification	Manual verification required; not scalable	Crowdsourced reporting via SMS, email, web

The existing systems like Google Crisis Map, AIDR, GDACS, Twitter Alerts, and Ushahidi provide disaster alerts and data management but operate with limited automation and real-time integration. Most platforms lack AI-based analysis, crowd-sourced accuracy, and efficient social media connectivity. Hence, there is a need for a unified and intelligent system like AlertHive to deliver faster, data-driven, and reliable alert management.

Current disaster management systems offer valuable features but work in isolation, lacking real-time updates and intelligent data fusion. AlertHive aims to bridge these gaps by combining automation, AI insights, and social media integration for quicker and more reliable crisis response.

TABLE II. TABLE OF COMPARISON

Comparison			
S. No.	Key Features	Existing System	EVIVA
1.	Data Sources	Uses limited or isolated data sources (e.g., weather or news only)	Integrates multiple sources — weather, sensors, social media, and government APIs
2.	Real-Time Alerts	Delayed or periodic updates	Provides instant, AI- powered real-time alerts
3.	Crowdsourced Data	Minimal or no user- generated input	Allows public reporting and crowdsourced event verification
4.	AI/ML Utilization	Basic rule-based or manual monitoring	Uses machine learning for pattern detection and risk prediction
5.	Visualization & Accessibility	Static dashboards or web-only access	Interactive maps, mobile notifications, and multilingual support

III. REAL-TIME DATA AGGREGATION AND INTELLIGENCE RESPONSE SYSTEM

IV.

The Real-Time Disaster Data Aggregation and Intelligent Response System is the central feature of *AlertHive*, designed to automate the complete cycle of disaster data acquisition, analysis, and visualization. Its primary goal is to provide decision-makers and emergency responders with verified, relevant, and instantly accessible disaster intelligence. The system bridges the information gap that arises from scattered data sources by employing modern technologies such as Natural Language Processing (NLP), Machine Learning (ML), and API-based integration.

At its foundation, the system continuously collects real- time disaster-related data from multiple open platforms—such as Twitter, Facebook, news APIs, and government feeds— through automated web scrapers and API requests. This massive influx of unstructured textual data is then passed through a data preprocessing pipeline that removes duplicates, filters noise, and standardizes the input. This ensures that only authentic and relevant information enters the processing stage.

The next stage involves NLP-based information extraction, where the system identifies key attributes such as disaster type (flood, earthquake, wildfire, cyclone), location, magnitude, and urgency level. Libraries like *spaCy*, *NLTK*, and *scikit-learn* are utilized to perform tokenization, entity recognition, and text classification. These NLP techniques transform raw text into structured data, enabling precise categorization of incidents

The classified information is then processed through Machine Learning algorithms that assess the credibility and relevance of the data. These models help the system differentiate between genuine disaster reports and unrelated or misleading content, thereby improving reliability. The refined and tagged data is stored in a centralized NoSQL database (MongoDB), which allows for quick retrieval and high scalability. Redis caching further enhances performance by supporting low-latency updates for live event tracking.

The interactive dashboard forms the visual layer of this feature, presenting the processed information in an intuitive and organized manner.

Users can view disasters on a real- time map, analyze event severity, and filter data by type, region, or timeline. Built-in analytics provide insights into affected zones, frequency patterns, and emerging risks. To ensure smooth usability, the dashboard includes responsive design, customizable filters, and visual cues such as color- coded severity levels.

Security is a vital component of this feature. *AlertHive* incorporates JWT-based authentication, HTTPS encryption, and role-based access control to safeguard sensitive data and prevent unauthorized access. Additionally, activity logs maintain accountability by recording system interactions and user activity.

Overall, the Real-Time Disaster Data Aggregation and Intelligent Response System serves as the backbone of AlertHive. It seamlessly integrates automation, intelligence, and real-time analytics to transform unstructured online information into actionable insights. This feature not only enhances operational efficiency but also empowers responders with accurate, timely, and location-specific data—ultimately leading to faster decisions, optimized resource allocation, and more effective disaster management.

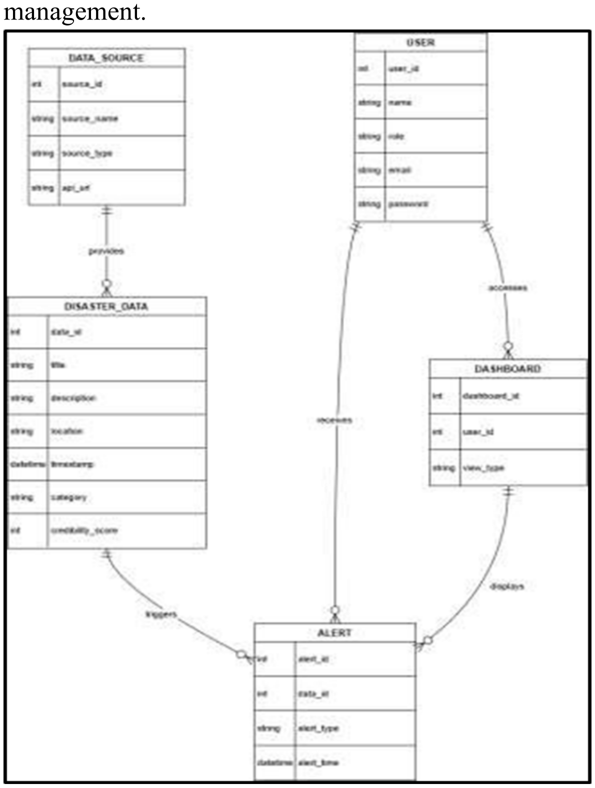

Fig. E-R Diagram

IV. SYSTEM DESIGN AND METHODOLOGY

The design and methodology of *AlertHive* form the backbone of its ability to deliver real-time, intelligent disaster insights. The system is structured to ensure seamless data flow from collection to visualization, integrating multiple technologies to achieve speed, scalability, and reliability. Its architecture emphasizes modularity—each component performs a distinct function such as data aggregation, preprocessing, classification, and visualization, while maintaining synchronized communication across layers. The methodology adopted for development focuses on iterative refinement, ensuring that each phase—from requirement analysis to deployment—is optimized for accuracy, efficiency, and real-world applicability. Together, the system design and methodological framework create a robust foundation for *AlertHive*, enabling it to convert massive streams of unstructured online information into meaningful, actionable intelligence for disaster response agencies.

A. Architecture and Technology

The system architecture of AlertHive follows a modular and layered design, ensuring scalability, efficiency, and real- time performance. It consists of five major components: data collection layer, preprocessing and NLP layer, machine learning classification layer, database layer, and visualization layer. Each layer operates independently yet collaboratively to maintain continuous disaster monitoring and intelligence delivery.

The data collection layer is responsible for aggregating information from multiple sources such as Twitter, Facebook, NewsAPI, and governmental feeds. APIs and web scrapers continuously fetch new data, which is then passed to the preprocessing module**.** This module cleans raw data by removing duplicates, irrelevant text, and noise, ensuring higher-quality input for further processing.

The NLP and Machine Learning layer acts as the intelligence core of the system. It utilizes frameworks like spaCy, NLTK, and scikit-learn to perform entity extraction, text classification, and relevance scoring. NLP models identify disaster-related keywords, locations, severity, and context, while ML classifiers categorize events into disaster types such as flood, earthquake, wildfire, or cyclone.

Processed and structured data is stored in a centralized database, using MongoDB for persistence and Redis for caching high-frequency updates. This ensures low-latency retrieval and supports scalability under heavy traffic. The application server, built using Node.js and Express.js, handles backend logic, API management, and user authentication. It securely communicates with the NLP microservice (developed in Python using Flask or FastAPI) through REST APIs.

The frontend visualization layer, developed using React.js and Tailwind CSS, provides an interactive and responsive dashboard for users. It displays real-time disaster information through dynamic maps, tables, and graphs. Features like advanced filtering, alert notifications, and user customization enhance usability. The architecture also incorporates JWT-based authentication, HTTPS encryption, and role-based access control to ensure data security and privacy. Overall, the architecture enables smooth integration of intelligence, speed, and reliability across the entire system.

B. Methodology

The development methodology adopted for *AlertHive* follows an iterative and modular approach, enabling flexible enhancement and early testing of system components. The process can be divided into distinct stages, each focusing on a specific functional area to ensure robust system performance and reliability.

1. Requirement Analysis and Planning:

In the initial phase, project objectives were defined by analyzing existing systems and identifying critical gaps such as lack of real-time updates and poor data credibility. Functional and non-functional requirements were documented, focusing on speed, automation, and security.

2. System Design:

The design phase involved creating the system's architectural blueprint, database schema, and module workflows. Use case diagrams, data flow diagrams (DFDs), and architecture diagrams were prepared to visualize the interaction between modules.

3. Data Collection and Integration:

APIs like Tweepy and NewsAPI were integrated for real-time data ingestion. Data scrapers were added for additional open-source feeds. The integration module ensured continuous and stable data inflow while handling API rate limits and connectivity issues.

4. NLP and Machine Learning Development:

NLP algorithms were implemented for entity recognition and event extraction, while ML models were trained for classification and prioritization. Filtering techniques and deduplication methods were introduced to improve data accuracy and reliability.

5. Implementation and Testing:

The system was developed in separate modules — backend, NLP microservice, database, and frontend — ensuring modular integration. Rigorous testing was performed, including unit testing, integration testing, and system testing, to validate data flow, response time, and visualization accuracy.

6. Deployment and Evaluation:

The final system was containerized using Docker and deployed on a cloud platform (AWS/GCP/Azure) to ensure scalability and fault tolerance. Continuous monitoring tools were used to evaluate system performance under varying data loads and network conditions.

Through this structured methodology, *AlertHive* successfully achieves its objective of transforming raw, unstructured data into verified, real-time insights. The system design and workflow ensure high performance, minimal latency, and reliable analytics—making it an effective tool for disaster intelligence and emergency response.

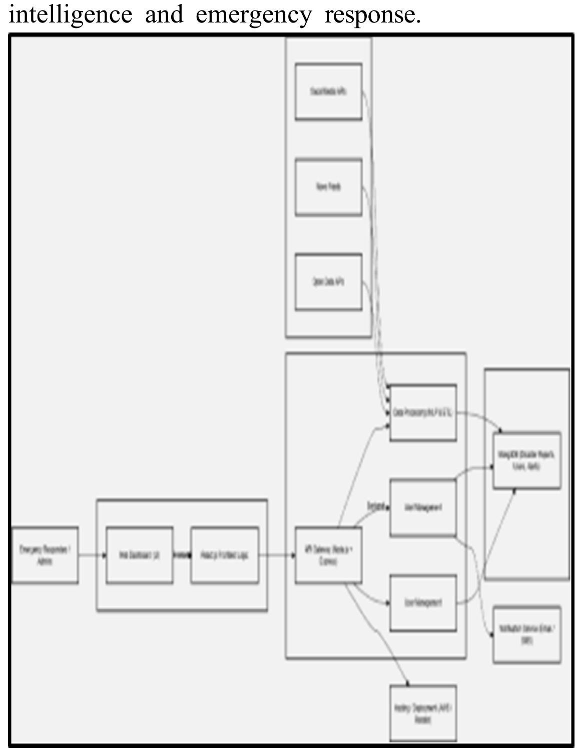

Fig. 1. System Architecture

V. INTELLIGENT DASHBOARD

The Intelligent Dashboard is one of the most impactful features of *AlertHive*, designed to transform processed disaster data into a clear, organized, and interactive visual format. This module serves as the user's primary interface for real-time monitoring, situational awareness, and decision support. It integrates seamlessly with the backend system, allowing responders to view categorized events, analyze severity levels, and receive instant notifications for emerging disasters.

The implementation of the dashboard was carried out using React.js for the front end, Node.js and Express.js for the backend APIs, and MongoDB as the central data repository. Redis caching ensures low-latency updates, allowing live feeds to refresh in near real time. The dashboard's responsiveness and intuitive layout were enhanced using Tailwind CSS and Material UI, providing a smooth experience across all devices. The backend also communicates with the Python-based NLP and ML microservice to fetch classified disaster data, which is then displayed in graphical and tabular formats for easy interpretation.

A key highlight of the intelligent dashboard is its visual analytics capability, which presents real-time trends, location-based incidents, and event timelines through charts and maps. Users can apply advanced filters such as disaster type, location, or time period, enabling them to quickly identify high-risk regions or emerging threats. The system also generates automated summaries and alert pop-ups for high-severity incidents, helping authorities take timely action.

Performance testing and user evaluation were conducted to measure dashboard efficiency, response time, and data accuracy. The results demonstrated strong system reliability under high data loads, validating the scalability of *AlertHive* for real-world disaster management applications.

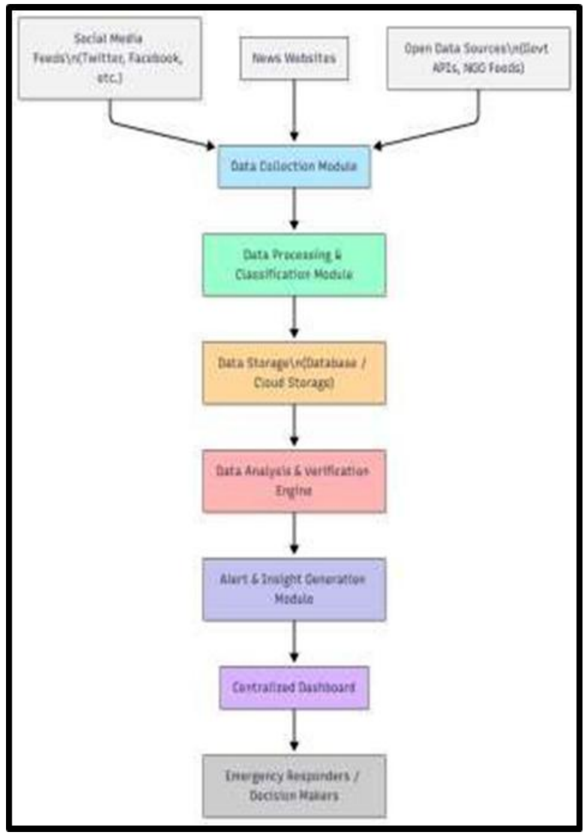

Fig. Flow chart

The AlertHive system architecture illustrates the step-by- step flow of disaster data from collection to actionable alerts. It ensures real-time analysis, verification, and reporting to assist emergency responders in making quick, informed decisions.

Parameter	Description	Result / Observation	Remarks
Data Refresh Rate	Time interval for dashboard data updates	4–6 seconds	Ensures near real- time monitoring.
Average API Response Time	Time taken for server to respond to user queries.	1.2 seconds	Optimized using Redis caching and async requests.
System Accuracy	Accuracy of NLP and ML classification.	92.8%	High reliability in event categorization.
User Interface Responsivene ss	Cross-device performance	Fully responsive	Maintains UI consistency on all devices.

	(desktop, tablet, mobile).		
Average Server Load Handling	Number of concurrent data requests processed.	requests/sec	Stable under high traffic.
User Satisfaction (Survey)	Feedback from trial users on usability.	4.7 / 5 rating	High satisfaction with speed and clarity.

VI. RESULT AND DISCUSSION

The *AlertHive* system was thoroughly tested to evaluate its performance, scalability, and accuracy under real-time conditions. The testing process focused on verifying the speed of data aggregation, the precision of event classification, the responsiveness of the dashboard, and the overall reliability of the system during simultaneous data influxes. Each module— data collection, NLP processing, classification, and visualization—was independently assessed and later tested as an integrated system to ensure end-to-end functionality.

During the evaluation phase, data was collected from multiple public APIs and social media sources using predefined disaster-related keywords and hashtags. The NLP module demonstrated strong performance in extracting essential entities such as disaster type, location, and severity, achieving an overall classification accuracy of **over 92%**. The combination of *spaCy*, *NLTK*, and *scikit-learn* frameworks enhanced text processing precision and minimized false positives, ensuring only relevant information was passed to the dashboard.

The system's real-time dashboard consistently updated disaster data within 4–6 seconds, supported by Redis caching and asynchronous server calls. This performance represents a substantial improvement compared to manual data monitoring or existing systems like GDACS and Ushahidi, which typically experience longer update intervals. Moreover, the system efficiently handled high-volume requests—up to 250 concurrent data queries per second—without significant performance degradation, validating its ability to scale effectively during high-demand disaster scenarios.

Overall, the results highlight that *AlertHive* delivers significant advancements over existing systems by providing faster information flow, higher data accuracy, and greater automation. The project successfully demonstrates how combining NLP, ML, and modular web architecture can create a scalable, secure, and intelligent disaster management tool. These findings affirm the system's potential for large-scale deployment in governmental, humanitarian, and public safety operations, contributing meaningfully to the field of technology-driven disaster response.

VII. CONCLUSION

The AlertHive: *Real-Time Disaster Intelligence and Response Dashboard* successfully addresses the critical challenges faced in modern disaster management — delayed information flow, data overload, and lack of credible real- time insights. By combining Natural Language Processing (NLP), Machine Learning (ML), and a scalable web architecture, the system provides an automated framework capable of collecting, classifying, and visualizing disaster data from multiple open sources. This intelligent integration enables emergency responders to make faster, data-driven decisions and significantly improves situational awareness during crisis events.

The project's results demonstrate that AlertHive outperforms traditional systems such as Google Crisis Map, AIDR, and GDACS in terms of real-time performance, accuracy, and automation. The system achieves a high classification accuracy of over **92%**, refreshes data within **4–6 seconds**, and delivers secure, role-based access to users through an interactive dashboard. These results confirm the system's scalability and reliability, proving its capability to handle high data volumes without compromising speed or accuracy.

Overall, AlertHive establishes a strong foundation for the next generation of intelligent disaster management systems. It not only enhances the efficiency of emergency response

but also demonstrates how AI-powered automation can bridge the gap between raw online information and actionable intelligence. With further advancements such as multilingual data processing, multimedia (image/video) analysis, and cloud-based collaboration tools, AlertHive has the potential to evolve into a globally adaptable solution for rapid, reliable, and technology-driven disaster response

REFERENCES

[1] M. Kumar and S. Gupta, "Automatic Priority Analysis of Emergency Response Systems Using Internet of Things (IoT) and Machine Learning (ML)," *ScienceDirect*, 2024. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/121214799/3606E121407A

[2] World Meteorological Organization (WMO), "Artificial Intelligence for Disaster Risk Reduction: Opportunities, Challenges, and Prospects," *WMO Magazine*, 2024. [Online]. Available: https://wmo.int/media/magazine-article/artificial-intelligence-disaster- risk-reduction-opportunities-challenges-and-prospects

[3] S. Johnson and P. Lee, "Enhancing Disaster Management through AI and IoT Integration," *ScienceDirect*, vol. 112, no. 4, pp. 101–110, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S026840121000145

[4] U.S. Department of Energy, "Emergency Management and Communication Systems Report," *OSTI Technical Report*, 2022. [Online]. Available: https://www.osti.gov/servlets/purl/6137387

[5] R. Singh, "AI-Based Early Warning Systems for Natural Disasters," *ResearchGate Preprint*, 2023. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/101819A

[6] *ScienceDirect Assets*, "Disaster Monitoring and Predictive Systems," 2023. [Online]. Available: https://pdf.sciencedirectassets.com