



International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)



ijerst.editor@gmail.com
editor@ijerst.com

Smart Vehicle Plate-Based Communication and Incident Reporting System Using Django Framework

POLISETTI LAKSHMI PRASANNA

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

K.Venkatesh

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

The rapid growth of urban traffic and vehicle density has created significant challenges in communication between vehicle owners and in reporting real-time road incidents. Traditional systems rely heavily on manual reporting or lack mechanisms for direct, secure communication between individuals using vehicle identifiers. This paper presents a **Smart Vehicle Plate-Based Communication and Incident Reporting System** developed using the Django web framework to address these challenges. The proposed system enables users to register their vehicle details and use license plate numbers as unique identifiers for communication. Users can send messages to other vehicle owners without requiring personal contact information, ensuring privacy and convenience. The system supports multiple message types such as alerts, emergency notifications, traffic updates, and general information, making it adaptable to various real-world scenarios. In addition to communication, the system incorporates an **incident reporting module** that allows users to report road-related issues such as accidents, potholes, congestion, flooding, and construction zones. Each incident is geo-tagged using latitude and longitude coordinates, enabling visualization on a city map interface. This feature enhances situational awareness and can assist authorities and commuters in making informed decisions. The system includes an analytics module that provides insights into message patterns, incident distribution, and user activity. Real-time statistics such as unread messages, open incidents, and message types are dynamically computed using database aggregation techniques. The architecture follows a modular design with separate components for user interaction, data processing, and storage. Django's ORM ensures efficient database management, while built-in security features enhance data protection. JSON-based APIs enable seamless integration with external applications or mobile platforms. The proposed solution demonstrates how web technologies can be leveraged to improve urban communication and road safety. By combining vehicle-based identification with real-time data sharing, the system offers a scalable and efficient platform for smart city applications.

Keywords: Vehicle Communication, Django Framework, Smart Transportation, Incident Reporting, License Plate System, Web Application, Smart City, Data Analytics

I. INTRODUCTION

Urban transportation systems are becoming increasingly complex due to rapid population growth and rising vehicle ownership. One of the major challenges in such environments is the lack of efficient communication between vehicle owners. Situations such as improper parking, accidents, or emergencies often require quick communication, but the absence of a direct and secure communication channel makes this difficult. Existing communication methods rely on indirect approaches such as public announcements or law enforcement intervention, which are often inefficient and time-consuming. Additionally, reporting road incidents such as potholes, congestion, or accidents typically involves manual processes that lack real-time updates and geolocation accuracy. To address these challenges, this paper proposes a **Smart Vehicle Plate-Based Communication and Incident Reporting System**. The system uses vehicle license plate numbers as unique identifiers, enabling users to send messages without revealing personal contact details. This ensures both privacy and convenience. The system is developed using the Django framework, which provides a robust and scalable environment for web application development. It integrates multiple functionalities, including messaging, incident reporting, data visualization, and analytics. Users can register their vehicles, send messages, report incidents, and view real-time updates through an intuitive interface. A key feature of the system is its ability to categorize messages and incidents, enabling better organization and analysis. The inclusion of geolocation data allows incidents to be displayed on a map, improving situational awareness for users and authorities. The proposed system aligns with the vision of smart cities by leveraging digital technologies to enhance urban mobility and safety. It provides a unified platform for communication and reporting, reducing delays and improving efficiency. Overall, the system demonstrates how modern web technologies can be used to create innovative solutions for real-world problems in transportation and urban management.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Recent advancements in smart transportation systems have emphasized the importance of real-time communication and data-driven decision-making. Various studies have explored the use of mobile applications, IoT devices, and artificial intelligence to improve traffic management and road safety. One common approach involves mobile-based reporting systems, where users manually report incidents through dedicated applications. While these systems provide real-time updates, they often require user authentication and may expose personal information, raising privacy concerns. Another approach utilizes IoT-enabled sensors and cameras to automatically detect incidents such

as accidents or traffic congestion. Although these systems offer high accuracy, they are expensive to deploy and maintain, limiting their scalability in developing regions. Vehicle-to-Vehicle (V2V) communication systems have also been proposed, enabling direct communication between vehicles using wireless technologies. These systems are effective in reducing accidents and improving traffic flow but require specialized hardware and infrastructure.

Several web-based grievance and reporting systems have been developed for municipal services. These systems allow users to report issues such as potholes or waterlogging. However, they typically lack integration with vehicle-based identification and do not support direct communication between users. Recent research has also highlighted the importance of data analytics in transportation systems. By analyzing user behavior and incident patterns, authorities can make informed decisions to improve infrastructure and reduce congestion. Despite these advancements, there is a gap in systems that combine **vehicle-based communication with incident reporting in a unified platform**. The proposed system addresses this gap by integrating messaging, reporting, and analytics into a single web-based solution.

III. EXISTING SYSTEM

Existing systems for vehicle communication and incident reporting are generally fragmented and limited in functionality. Traditional methods rely on manual intervention, such as contacting authorities or using public announcements, which are inefficient and time-consuming. Mobile-based reporting applications have been introduced to address some of these limitations. These applications allow users to report incidents and receive updates. However, they often require users to share personal information, which raises privacy concerns. Additionally, these systems do not support direct communication between vehicle owners. IoT-based solutions provide automated incident detection using sensors and cameras. While these systems are highly effective, they involve significant infrastructure costs and are not feasible for widespread deployment in all regions. Web-based grievance systems used by municipal authorities focus primarily on issue reporting and tracking. These systems lack real-time communication features and are not designed for vehicle-specific interactions. Another limitation of existing systems is the absence of integrated analytics. Most platforms do not provide insights into user behavior, message trends, or incident patterns, limiting their effectiveness in decision-making. Overall, current systems fail to provide a comprehensive solution that combines **secure communication, real-time incident reporting, and data analytics**. This highlights the need for an integrated platform like the proposed system, which addresses these limitations and offers a more efficient and scalable approach.

IV. PROPOSED METHOD

The proposed system is a **Smart Vehicle Plate-Based Communication and Incident Reporting Platform** designed to enhance urban mobility, safety, and communication. The system leverages vehicle license plate numbers as unique identifiers, enabling secure and privacy-preserving communication between vehicle owners without requiring

personal contact details. The platform integrates two core functionalities: **vehicle-to-vehicle messaging** and **real-time incident reporting**. Users can send messages categorized into types such as emergency alerts, parking notices, traffic updates, and general communication. This improves coordination in real-world scenarios such as accidents, congestion, or improper parking. The second major component is an **incident reporting module**, where users can report road conditions such as accidents, potholes, flooding, and congestion. Each incident is geo-tagged using latitude and longitude, allowing visualization on a city map. This supports real-time awareness and decision-making. The system is built using the Django framework and follows a modular architecture. It includes role-based functionalities such as user interaction, data processing, and analytics. The platform also provides dashboards displaying key metrics such as message counts, incident types, and user activity. Modern research highlights the importance of integrating communication technologies and data analytics in transportation systems to improve safety and efficiency. Additionally, automated incident reporting systems significantly enhance response time and decision-making. The proposed system addresses gaps in existing solutions by combining communication, reporting, and analytics into a unified platform. It is scalable, cost-effective, and suitable for smart city applications.

V. IMPLEMENTATION

The implementation of the system is carried out using the Django framework, which follows the Model-View-Template (MVT) architecture. The backend is developed in Python, and the database is managed using Django ORM for efficient data handling.

1. System Architecture

The system is divided into three main layers:

- **Presentation Layer:** HTML templates and user interface
- **Application Layer:** Django views and business logic
- **Data Layer:** Models representing database tables

This layered architecture ensures modularity and scalability.

2. Database Design

The system uses three main models:

- **PlateOwner:** Stores vehicle and owner details
- **Message:** Stores communication data
- **Incident:** Stores incident reports with geolocation

Django ORM is used to perform CRUD operations, ensuring security and reducing SQL injection risks.

3. Messaging Module:Users can send messages by entering sender and recipient plate numbers. Validation ensures:

- Fields are not empty
- Sender and recipient are different Messages are stored with attributes such as type, subject, body, and timestamp. The inbox feature allows filtering based on plate number and message type.

4. Incident Reporting Module

Users can report incidents by providing:

- Plate number
- Incident type
- Description
- Location and coordinates

The system stores this data and displays it on a map interface. Research shows that real-time incident reporting improves traffic response efficiency and reduces delays .

5. Dashboard and Analytics

The dashboard aggregates data such as total messages, incidents, and active plates. Django's aggregation functions (e.g., Count) are used to generate insights.

Analytics include:

- Messages by type
- Incidents by type and status
- Top senders
- Distribution by city

6. API Integration

The system provides JSON APIs for:

- Incident data
- System statistics

This enables integration with mobile applications or external services.

7. Security Features

- Input validation
- Django ORM protection

- Session handling

8. Scalability

The system can be extended with:

- AI-based incident detection
- Notification systems ,Cloud deployment

VI. ALGORITHMS

1. Message Sending Algorithm

Input: Sender plate, recipient plate, message details

Steps:

1. Validate input fields
2. Check sender \neq recipient
3. Store message in database
4. Display success message

2. Incident Reporting Algorithm

Input: Plate number, incident details

Steps:

1. Capture user input
2. Assign default coordinates if not provided
3. Store incident data
4. Update dashboard

3. Filtering Algorithm (Inbox)**

Steps:

1. Retrieve all messages
2. Apply filters (plate/type)
3. Return filtered results

4. Analytics Algorithm

Steps:

1. Group data using aggregation (Count)
2. Convert results into JSON format
3. Display charts

5. Read Status Algorithm

Steps:

1. Fetch message by ID
2. Update is_read = True
3. Save changes

6. API Data Conversion Algorithm

Steps:

1. Fetch records
2. Convert datetime to ISO format
3. Return JSON response

VII. SYSTEM DESIGN

The system follows a **modular and scalable design**, ensuring flexibility and maintainability.

1. Architecture

The system uses the **Model-View-Template (MVT)** architecture:

- **Model:** Defines data structure
- **View:** Handles logic
- **Template:** Renders UI

This design improves separation of concerns.

2. Functional Modules

a) *User Module*

- Register vehicle
- Send messages
- View inbox

b) *Messaging Module*

- Message creation
- Message filtering
- Read/unread tracking

c) Incident Module

- Report incidents
- Track status
- Map visualization

d) Analytics Module

- Data aggregation
- Visualization

3. Data Flow

1. User inputs data
2. Request sent to Django view
3. Data processed and stored
4. Response returned to UI

4. Database Structure

Tables include:

- PlateOwner
- Message
- Incident

Relationships are logical rather than foreign-key-based, ensuring flexibility.

5. Security Design

- Input validation
- ORM usage
- Session management

Modern smart city systems emphasize cybersecurity due to increasing digital threats .

6. Scalability and Integration

The system supports:

- API integration

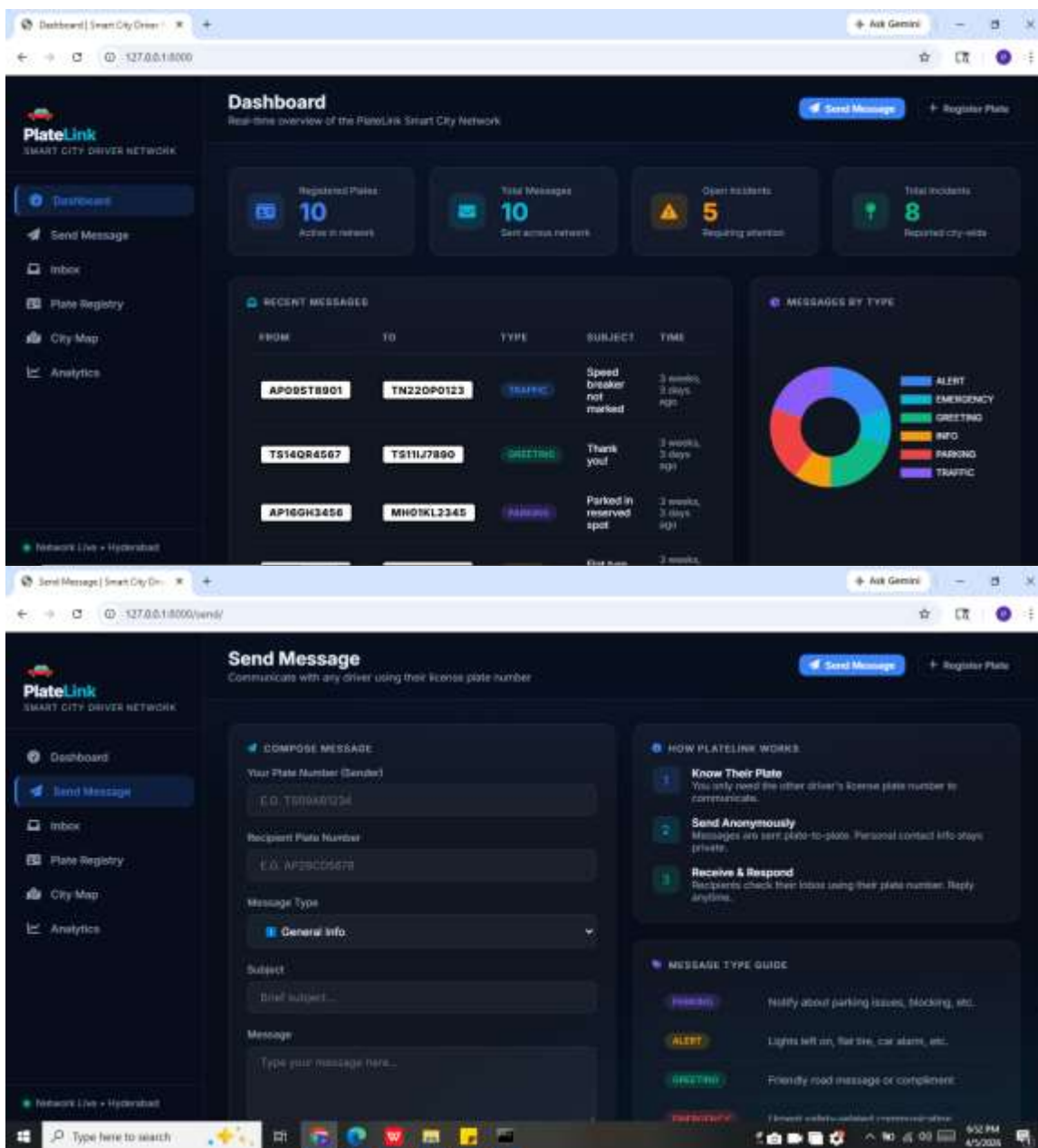
- Cloud deployment
- AI extensions

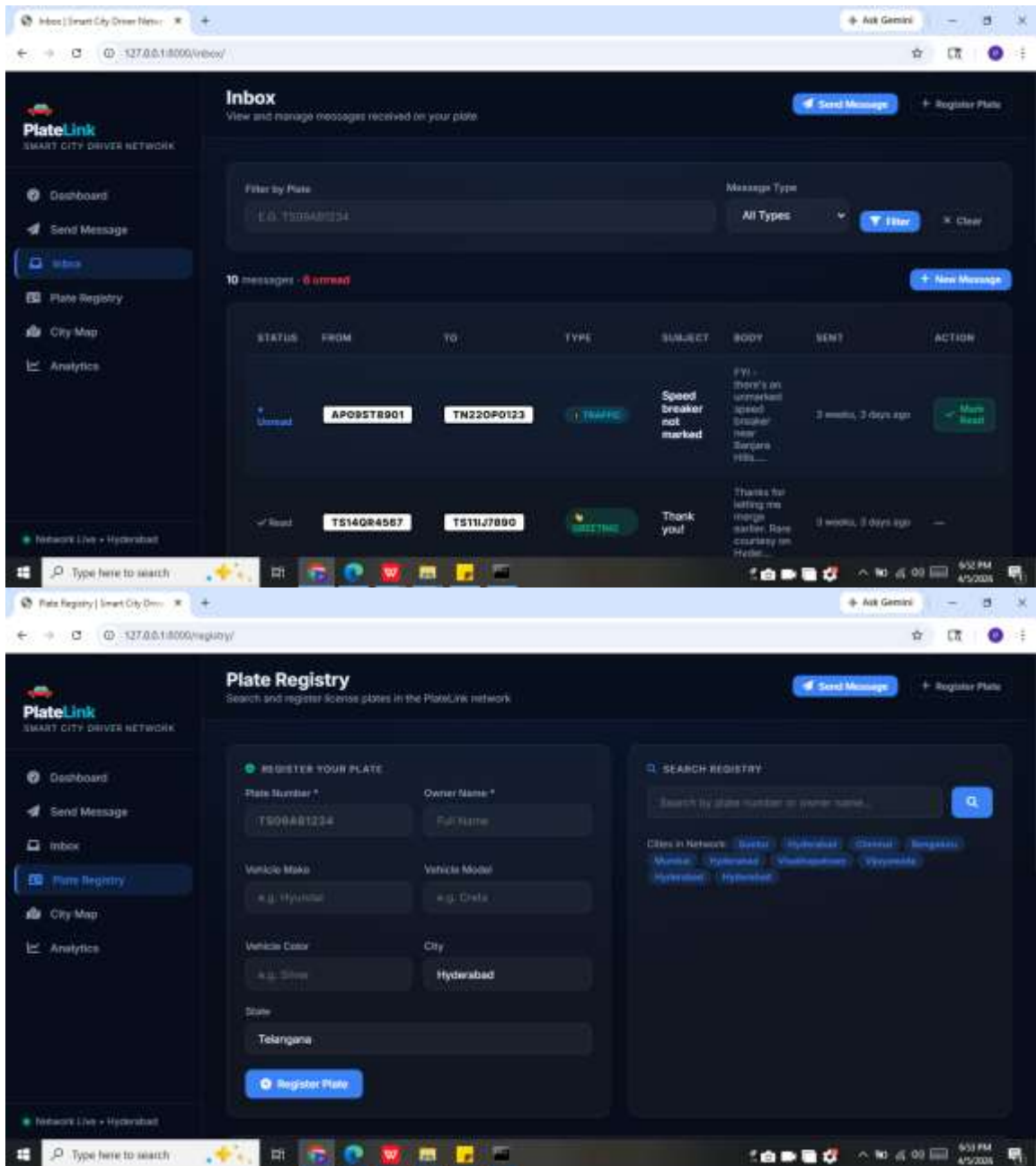
Research shows that integrating AI and communication technologies enhances transportation systems .

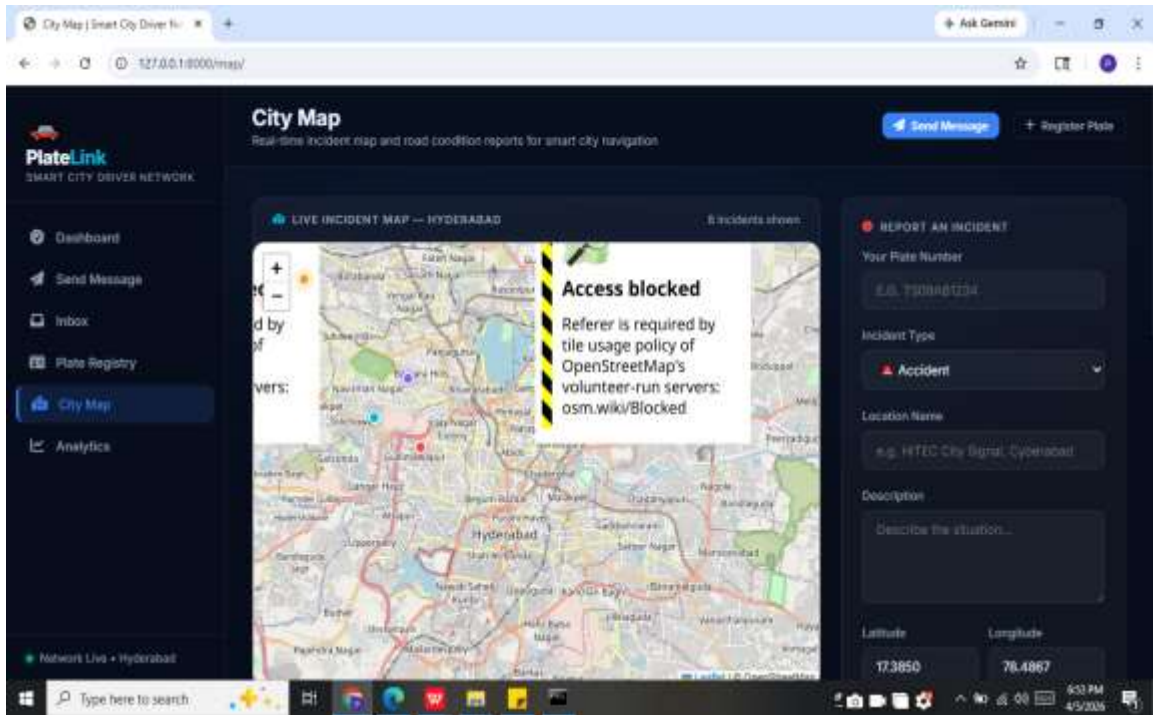
7. System Workflow

The system ensures efficient communication and reporting by integrating all modules into a unified platform.

SYSTEM DESIGN IMAGES







VIII. CONCLUSION

The proposed Smart Vehicle Plate-Based Communication and Incident Reporting System provides an innovative solution for improving communication and safety in urban transportation environments. By utilizing vehicle license plate numbers as unique identifiers, the system ensures privacy while enabling direct interaction between users. The integration of messaging and incident reporting features creates a comprehensive platform that addresses real-world challenges such as traffic congestion, accidents, and emergency communication. The use of geolocation enhances situational awareness and supports efficient decision-making. The implementation using Django ensures scalability, security, and ease of development. The modular design allows future enhancements such as AI-based analytics, automated incident detection, and mobile integration. Research indicates that intelligent transportation systems significantly improve efficiency, safety, and resource utilization in urban environments. The proposed system aligns with these advancements and contributes to the development of smart cities. Despite its advantages, the system can be further improved by incorporating advanced security mechanisms and real-time notification systems. Future work may also include machine learning techniques for predictive analytics and automated incident classification. In conclusion, the system demonstrates the potential of integrating web technologies and smart transportation concepts to create a practical and scalable solution for modern urban challenges.

REFERENCES

1. Shaffi & Sidhick (2025). *Real-time Incident Reporting Framework*.
2. Ashraf & Idrisi (2024). *Smart Transportation Systems*.
3. Ahmed et al. (2024). *AI Framework for Traffic Incident Response*.
4. MDPI (2023). *Smart Transportation Technologies Survey*.
5. Springer (2026). *Cybersecurity in Smart Cities*.
6. ScienceDirect (2025). *Industry 4.0 in Transportation*.
7. Thomas et al. (2025). *Transportation Cyber Incident Analysis*.
8. Li et al. (2025). *Drone-based Incident Detection System*.
9. Creß et al. (2021). *ITS Infrastructure Survey*.
10. Dabiri & Heaslip (2018). *Smart Transportation Data Sources*.
11. IEEE (2025). *LLM Integration in ITS*.
12. ACM (2024). *Multimodal AI in Transportation*.
13. Springer (2025). *Vehicular Communication Systems*.
14. IEEE ICCA (2023). *AI in Smart Mobility*.
15. IJETED (2025). *Django-based Smart Systems*.