

The Smart Accident System: An AI-Based Emergency Management System

Dr Pramod¹, Amith N², Ananya Sarvade³, Ankitha L⁴, Darshan Surwase⁵

^{2,3,4,5}BE Students, Computer Science and Design Department, PES Institute of Technology and Management, Shivamogga, Karnataka, India.

¹Professor & Head of Department, Computer Science and Design Department, PES Institute of Technology and Management, Shivamogga, Karnataka, India.

ABSTRACT

Road accidents are one of the leading causes of injuries and fatalities, and a major challenge in modern transportation systems is the delay in detecting accidents and initiating timely rescue responses. Traditionally, accident reporting depends on human witnesses or manual communication, leading to slow response times and potentially severe consequences. This project, Smart Accident Detection and Emergency Response System, presents a software-based real-time simulation platform that automatically generates accident events, visualizes vehicle movement on an interactive map, and performs intelligent routing for emergency response. The system makes use of GPS-based simulated tracking, accident interval generation, vehicle icons, and real-time updates to display accidents, nearby ambulances, and their navigation paths. The platform provides a dynamic dashboard where accidents appear every fixed interval (e.g., 30 seconds), vehicles such as cars and bikes are tracked, and emergency response flow is shown visually. By combining data simulation, mapping APIs, and algorithmic routing, the system demonstrates how an automated accident monitoring environment can support faster rescue operations and improve situational awareness in intelligent transport systems.

Keywords: Accident Simulation, Emergency Response, Real-Time Tracking, GPS, Routing Algorithm, Intelligent Transport System.

1. INTRODUCTION

Accident detection continues to be a challenging issue because it relies on unpredictable human behavior. When accidents occur, especially on highways or isolated landscapes, the absence of witnesses may result in no immediate support. Even in urban environments, panic, confusion, and lack of communication clarity often delay emergency response. The core problems addressed in this work include: Accidents often go unreported for long periods in remote or low-traffic regions due to the absence of observers. This significantly reduces survival chances.

Dependency on human reporting introduces inconsistencies, misinformation, and delays, which further worsen rescue timelines. Real accidents cannot be reproduced safely for academic experimentation, preventing the evaluation of emergency-response strategies. Hardware-based accident detection systems are expensive, require installation in vehicles, and are impractical for classroom and research laboratories. These limitations highlight the need for a software-based simulation model capable of producing realistic accident data and replicating full emergency workflows without relying on physical systems. The primary

objectives of the project are: **Automatic Accident Generation:** To simulate accidents at fixed time intervals for testing real-time response systems. **Real-Time Map Visualization:** To display moving vehicles (cars/bikes) and accident locations on an interactive world map. **Ambulance Routing:** To compute and display the shortest path from the nearest ambulance to the accident. **Dashboard Updates:** To show live accident details, timestamps, locations, and vehicle movements. **Efficient Emergency Response Simulation:** To provide a realistic environment for analyzing delay time, response patterns, and rescue flow.

Automatic Accident Generation: In real-world environments, accidents occur unpredictably, making it difficult to test emergency response systems under controlled conditions. Automatic accident generation allows the simulation of realistic accident events at consistent intervals, ensuring that the system can be thoroughly tested for responsiveness, scalability, and stability. It also provides a continuous stream of data essential for evaluating system behaviour in dynamic scenarios. **Real-Time Map Visualization:** Emergency response relies heavily on situational awareness. Displaying accidents, vehicles, and ambulances on a real-time map helps in visual understanding of incident locations, traffic flow, and rescue routing. Visual representation significantly reduces decision-making delays compared to raw textual coordinates. This objective ensures that responders can monitor movements instantly and act more efficiently.

Ambulance Routing (Shortest Path Calculation): Choosing the nearest ambulance and computing the shortest travel path is critical for minimizing response time, which directly impacts survival rates. The routing algorithm ensures mathematical accuracy and consistency in selecting optimal rescue routes. This objective supports the system's primary goal—reducing emergency arrival time and improving rescue efficiency. **Dashboard Updates and Monitoring:** A centralized dashboard consolidates all real-time information, including accident details, vehicle locations, timestamps, and route progress. This provides clarity, transparency, and control to the monitoring authority. Continuous updates allow stakeholders to analyze system performance and take corrective actions immediately. The dashboard also acts as a decision-support tool for emergency coordination. **Efficient Emergency Response Simulation:** Before deploying real-world accident management systems, simulation-based evaluation is essential. This objective ensures the creation of a safe, controlled environment to test different rescue strategies, identify limitations, and validate response algorithms. It enables experimentation without safety risks and provides measurable insights into system reliability, speed, and scalability.

This research contributes a unified simulation framework that integrates accident generation, coordinate simulation, routing algorithms, and live visualization—a combination rarely addressed in a single academic project. The first major contribution is the development of a controlled accident simulation engine capable of repeatedly generating realistic accident scenarios independent of hardware. The second contribution is the implementation of a routing mechanism that identifies the nearest ambulance and computes the shortest emergency response path, enabling efficient rescue simulation. The third contribution is a centralized dashboard that visually represents real-time accidents, vehicle movement, and route paths, allowing users to study emergency operations seamlessly. The paper is structured to reflect standard academic research: Section 1 introduces the key ideas and problems. Section 2 reviews related studies. Section 3 outlines the system architecture and methodology. Section 4 provides results and analysis. Section 5 explains system advantages. Section 6 outlines future scope, and Section 7 concludes the study.

2. LITERATURE REVIEW

The development of a Smart Accident System for AI-based emergency management relies heavily on Computer Vision, Machine Learning (ML), and Intelligent Decision-Making algorithms. Prior research shows that accident detection, severity assessment, and automated alert generation benefit the most from deep learning-based image recognition, sensor data fusion, and real-time classification techniques. Studies in this domain consistently highlight three major pillars: accident detection models, real-time event classification, and intelligent alert routing—each essential in designing an automated emergency response system.

Accident Detection and Classification Methods. Early systems for accident detection depended on rule-based thresholds, such as fixed accelerometer values, pre-defined speed drop patterns, or simple motion cues extracted from surveillance videos. However, research indicates that these approaches fail to generalize across environments and cannot reliably differentiate between minor anomalies and real accidents, leading to high false-alarm rates [1]. **Traditional Machine Learning Approaches.** Several studies demonstrate that classical ML algorithms such as Support Vector Machines (SVMs), Random Forests, and Decision Trees can classify accident events using structured sensor or telematics data [2], [8]. These models work efficiently when the input variables are shallow (speed, vibration, gyroscope readings) and the patterns are linearly separable. Comparative evaluations show that SVMs and Random Forests remain popular choices for accident prediction due to their interpretability and robustness on small datasets [9]. **Modern Deep Learning Approaches.** Recent advancements show that Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based architectures significantly outperform classical models in analyzing real-time CCTV footage, dash-cam videos, and multimodal sensor streams [5], [7]. These models capture spatial features (vehicle deformation, collision points) and temporal cues (impact sequence, velocity transition), enabling far more accurate accident recognition. Studies consistently report substantial improvements in precision and response time when deep learning is integrated into emergency systems—especially in large, diverse datasets [7]. **Suitability for the Proposed System.** Despite the power of deep learning models, research indicates that Logistic Regression and Random Forests remain extremely effective for real-time, low-latency accident alert routing, where speed, transparency, and minimal computational cost are crucial [6], [9]. Since the proposed Smart Accident System involves continuous classification of event severity levels and routing alerts to hospitals or authorities, these lightweight models offer the best trade-off between accuracy and operational efficiency. **Data Preprocessing and Feature Engineering.** Accident detection systems rely on high-quality preprocessing to convert raw video, sensor values, and telematics data into meaningful features. Research highlights that noise removal, signal smoothing, and frame-wise standardization significantly enhance recognition accuracy—especially when dealing with noisy real-world footage or fluctuating IoT sensor inputs [4], [10]. **Feature Extraction Techniques.** Multiple studies compare traditional feature extraction techniques such as Histogram of Oriented Gradients (HOG), Optical Flow, and Wavelet-based features with modern deep learning-based embeddings. While CNN-derived feature maps deliver superior accuracy, research shows that handcrafted features like HOG and Optical Flow still remain highly efficient for real-time ML classifiers due to their simplicity and low computational load [3].

For emergency management tasks—such as detecting collisions, skidding events, or sudden object impact—optimized feature extraction helps systems recognize key indicators including deformation patterns, abrupt motion changes, and anomaly points in sensor streams [6]. These preprocessing steps form the backbone of the proposed system's ability to detect accidents with high precision and minimal delay.

3. METHODOLOGY

The methodology of the system revolves around a structured simulation workflow designed to imitate real-world emergency processes. The system initiates with a GPS simulation engine that produces continuous coordinate updates for various vehicles. The accident generator triggers accident events at defined intervals, ensuring constant testing opportunities. Once an accident is produced, its details are transmitted to the visualization module, where all vehicles, accidents, and ambulances are plotted on a real-time interactive map. The routing logic identifies the nearest ambulance using distance metrics and visualizes the optimal rescue path. A dashboard consolidates all system activities, making monitoring and analysis simple and comprehensive.

3.1. System Architecture Overview

The architecture consists of interconnected modules that communicate seamlessly to simulate emergency conditions: The GPS Movement Simulator generates smooth and continuous changes in coordinates for vehicles, allowing the system to mimic natural motion across a map. The Accident Generator introduces events autonomously at regular intervals, producing accident coordinates, time stamps, and vehicle identifiers. The Data Manager handles accident and vehicle data, ensuring proper flow between modules. The Map Visualization Layer displays all events on a dynamic map interface, giving users a clear, real-time understanding of the scenario. The Routing Engine determines the closest ambulance and computes the shortest route using geographical calculations. This architecture enables a realistic and scalable simulation model. Accident and GPS Simulation: Accidents are simulated using scheduled triggers that generate events autonomously. Each event includes the exact time and GPS location, which is then displayed on the dashboard. Simultaneously, all vehicles—including cars, bikes, and ambulances—are moved using continuously updated GPS coordinates.

Simulation ensures accidents can occur repeatedly without safety concerns. GPS-based movement creates a realistic environment where accident-response algorithms can be tested effectively. Routing Logic and Dispatch: Routing is a core component of the system. When an accident is registered: The routing engine calculates the distance from the accident site to each ambulance using formulas such as Haversine distance. The closest ambulance is selected and assigned to respond. The shortest path between the accident site and ambulance is computed and displayed on the map. This routing simulation matches real-world emergency dispatch approaches and allows performance evaluation. Classification Model: Logistic Regression, a supervised learning algorithm, is mainly used in the project to categorize the feedback; the Scikit-learn library is utilized for implementation.

This model is selected for its speed and straightforwardness in separating the feedback into the predefined groups like (e.g., academics, administration, technical). Training and Evaluation: The training for the model is done with the TF-IDF vectors. To check the system's accuracy, Model Evaluation is done with the target goal being 80–90% correct category identification. Finally, the model is preserved for future implementation using the joblib library. Functional Requirements: The functional requirements define the core actions, behaviours, and operations that the Smart Accident Detection and Emergency Response System must perform to simulate accident detection and routing in a realistic and reliable manner. These requirements ensure that the system behaves exactly as intended during accident simulation, vehicle tracking, and ambulance dispatch.

Accident Generation Functionality: The system must automatically generate accident events at predefined or configurable time intervals. Each accident must include relevant information such as location coordinates, time of occurrence, and vehicle involvement. This functionality is essential for supporting

continuous testing without relying on human input or real-world conditions. **GPS-Based Vehicle Movement Simulation:** The system must simulate the movement of cars, bikes, and ambulances using dynamically changing GPS coordinates. Each vehicle should appear to move naturally on the map, and its position must update in real time. This ensures realistic behaviour and enables accurate route computation between accidents and ambulances.

Real-Time Map Visualization: The system must display all vehicles, accidents, and ambulance routes on an interactive map interface. The visualization should clearly distinguish between normal vehicles, accident points, and responding ambulances, enabling users to understand the spatial relationships and event flow intuitively. **Nearest Ambulance Identification:** Upon detecting an accident, the system must calculate the distance between the accident site and all available ambulances to identify the closest responder. This ensures fast dispatch behaviour, similar to real-world emergency systems.

Shortest Path Routing: The system must compute the most efficient rescue path using appropriate algorithms such as the Haversine distance calculation or shortest-path algorithms. Once computed, the route should be immediately displayed on the map to illustrate the ambulance’s movement toward the accident location. **Dashboard-Based Monitoring:** The system must provide a unified dashboard to display accident logs, timestamps, vehicle coordinates, routes, and system messages. This ensures that users can track all activities within a single centralized interface, improving usability and monitoring efficiency. **Real-Time Updates and Notifications:** The system must continuously update all coordinates, routes, and accident details in real time. The dashboard must reflect every change instantly so users can observe the complete emergency workflow as it occurs.

4. MODELING AND ANALYSIS

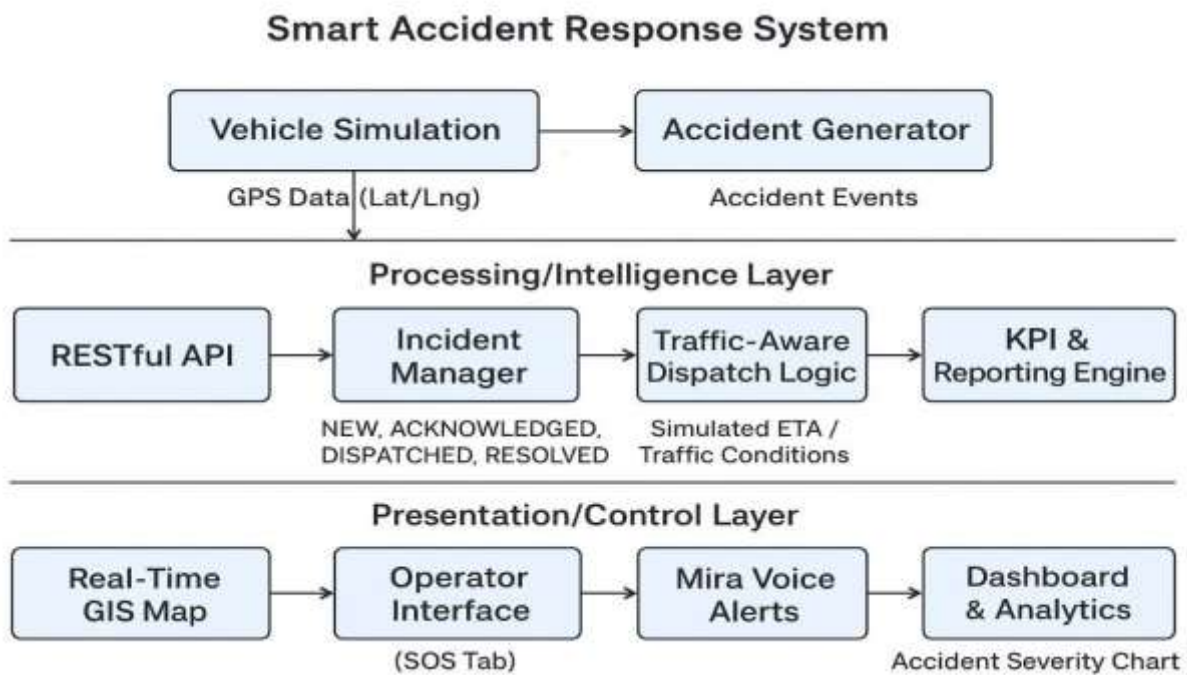


Figure 1: Block diagram of Smart Accident System.

Figure 1 illustrates the overall system architecture of the Smart Accident Response System, justifying how

the solution is organized into three coordinated layers to ensure realistic accident simulation and efficient emergency-response processing. The Simulation Layer is essential because it generates the core inputs of the system—vehicle movements and accident events—allowing continuous, repeatable testing without relying on real-world accidents. The Processing/Intelligence Layer is justified as the computational core of the architecture, responsible for managing incidents, dispatching ambulances using traffic-aware logic, and generating analytical performance metrics.

This layer ensures that the system behaves like an actual emergency control centre. Finally, the Presentation/Control Layer is required to translate backend intelligence into meaningful visual output, enabling operators to view the real-time GIS map, receive alerts, and analyze accident severity. Collectively, these layered components justify the architecture by demonstrating how raw simulation data is transformed into actionable emergency insights through structured processing and user-friendly visualization.

5. RESULTS AND DISCUSSION

5.1. Symptom Analysis Response Time and Accuracy

Interactive Map Visualization: The map view shows real-time accident locations across India, represented using prominent red warning icons. These markers update dynamically as new accidents occur. Vehicles such as cars, bikes, and ambulances are also shown with distinct icons, enabling users to clearly differentiate between normal traffic movement and active emergencies. The geospatial distribution of incidents demonstrates the system’s capability to simulate multiple accident points across wide geographic areas simultaneously. This visualization helps users understand spatial clustering, potential hotspots, and emergency load distribution.

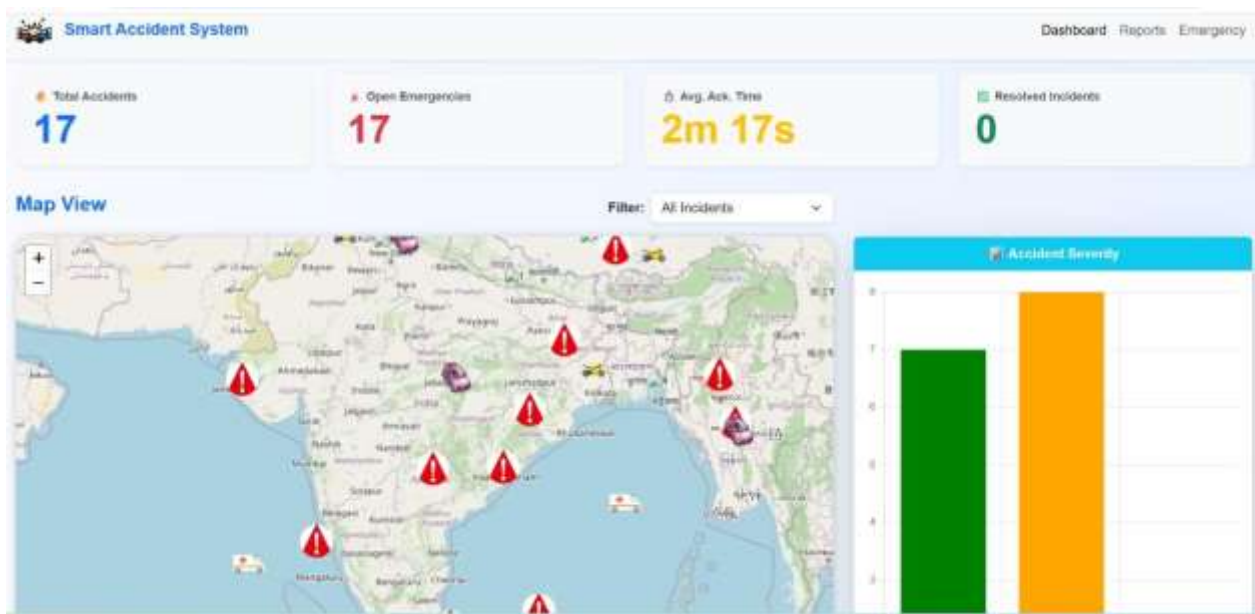


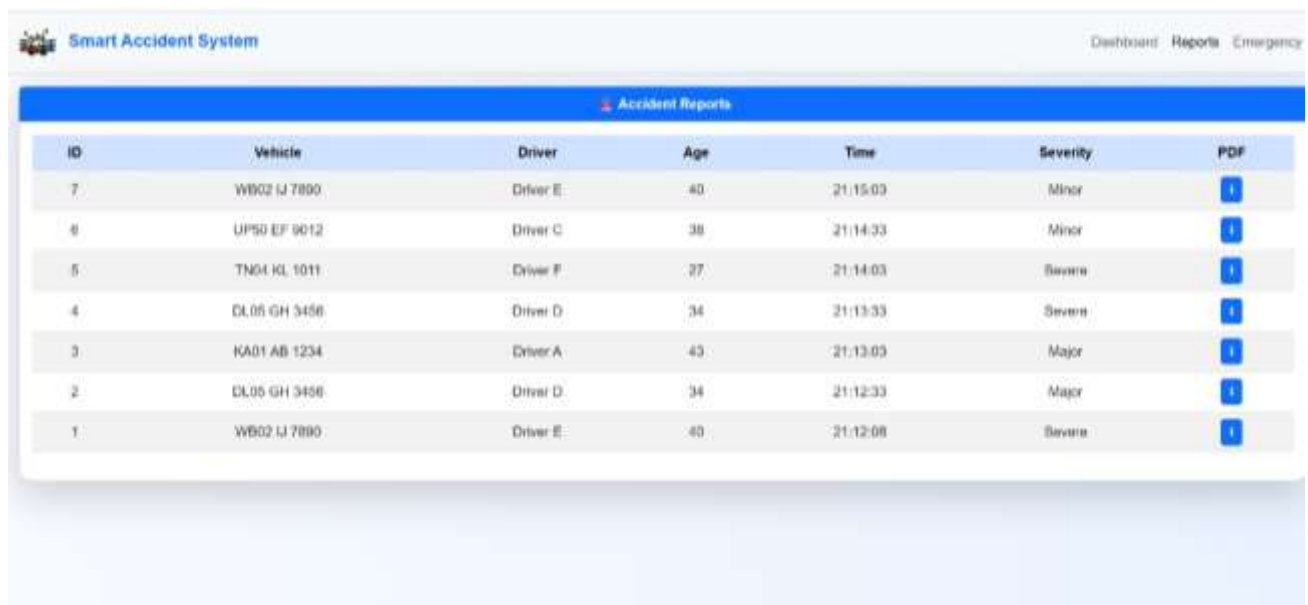
Figure 2: A highly informative dashboard that furnishes the output from Smart Accident system.

Figure 2 represents the visual analysis: The Smart Accident Detection and Emergency Response System successfully generate, visualizes, and analyses accident events in real time. The result dashboard provides a comprehensive overview of ongoing emergencies, system performance, and spatial distribution of

incidents across the map. As shown in the result interface, the system recorded a total of 17 accidents, all of which are currently active and awaiting resolution. The dashboard also displays an Average Acknowledgement Time of 2 minutes 17 seconds, indicating the overall response efficiency of the simulation engine. Each metric tile—Total Accidents, Open Emergencies, Acknowledgement Time, and Resolved Incidents—is visually emphasized to allow quick situational assessment.

The streamlined UI helps users identify critical parameters instantly, improving decision-making during emergency simulations. **Emergency Response Tracking:** The system highlights the presence and movement of ambulances in real time. Ambulance icons are positioned strategically across the map, allowing users to observe how rescue vehicles relate to ongoing accident locations. Although all 17 accidents remain unresolved in this screenshot, the system is designed to dispatch the nearest ambulance and compute the shortest response route during active simulation. This ability to visualize ambulance availability and their proximity to incidents is crucial for performance analysis and emergency planning.

Filtering and Incident Categorization: The result page includes an incident filter, enabling users to view All Incidents, Active Emergencies, or specific categories. This feature supports detailed examination of accident patterns and offers a more structured understanding of the simulation results. Filtering ensures that users can focus on particular types of events without navigating the entire map manually. **Accident Severity Analysis:** On the right-side panel, the Accident Severity Bar Chart provides a statistical view of the intensity of accidents recorded during simulation. Different color bars indicate varying severity levels, helping users analyse the nature of incidents rather than just their frequency. This statistical component is essential for evaluating incident distribution and understanding how severe accidents correlate with geographic areas or vehicle types.

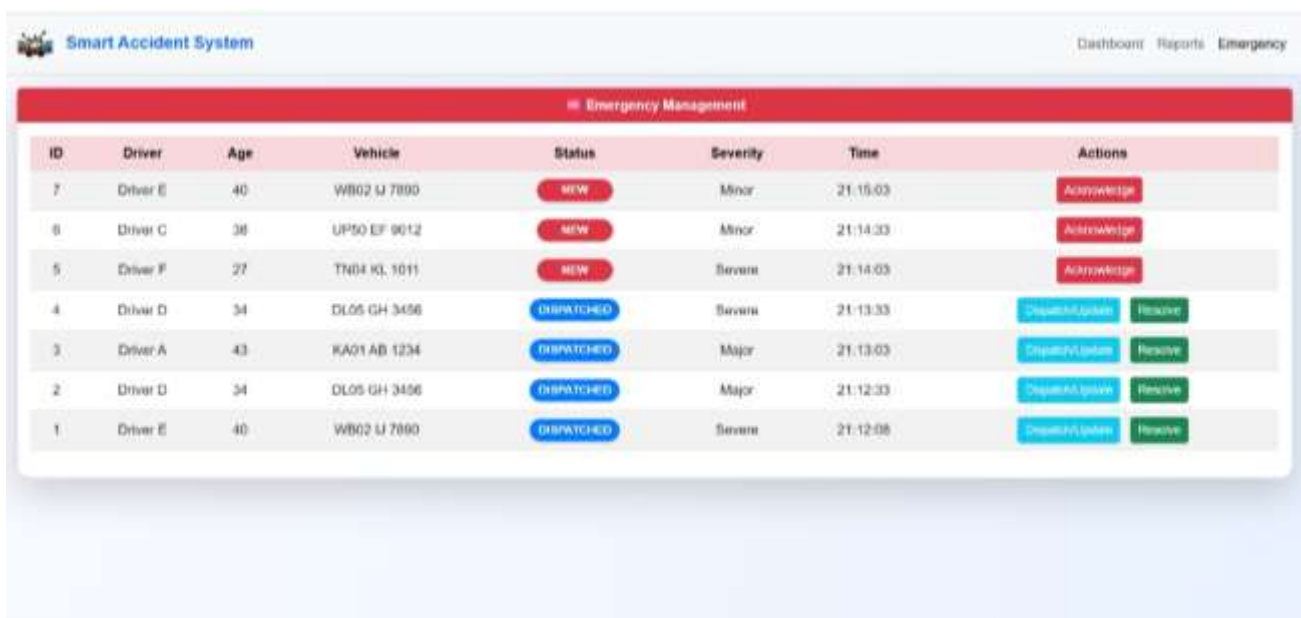


ID	Vehicle	Driver	Age	Time	Severity	PDF
7	WB02 IJ 7890	Driver E	40	21:15:03	Minor	PDF
8	UP50 EF 9012	Driver C	38	21:14:33	Minor	PDF
5	TN04 KL 1011	Driver F	27	21:14:03	Severe	PDF
4	DL05 GH 3456	Driver D	34	21:13:33	Severe	PDF
3	KA01 AB 1234	Driver A	43	21:13:03	Major	PDF
2	DL05 GH 3456	Driver D	34	21:12:33	Major	PDF
1	WB02 IJ 7890	Driver E	40	21:12:08	Severe	PDF

Figure 3: Accident Reports Module Interface.

The Accident Reports module (Figure X) presents a structured and well-organized tabular interface that displays detailed information about all simulated accident events recorded within the system. This page acts as the central repository where each accident entry is logged with essential metadata, enabling operators and researchers to analyze incident patterns with clarity. The table includes fields such as

Accident ID, Vehicle Number, Driver Name, Driver Age, Time of Occurrence, and Severity Level, all of which are automatically populated by the system during simulation. The system categorizes every accident into predefined severity classifications such as Minor, Major, and Severe. This classification helps operators quickly assess the priority of each incident. The inclusion of a PDF generation button for every entry further enhances the utility of this module, allowing users to download and store detailed accident reports, which can be used for documentation, evaluation, or external review. The design prioritizes readability, ensuring that operators can monitor multiple incidents simultaneously without confusion. Overall, this interface demonstrates the system’s capability to maintain a comprehensive log of accidents and provide easily accessible reports, fulfilling essential requirements of reliability, traceability, and documentation within an emergency-response evaluation platform.



ID	Driver	Age	Vehicle	Status	Severity	Time	Actions
7	Driver E	40	WB02 UJ 7890	NEW	Minor	21:15:03	Acknowledge
6	Driver C	38	UP50 EF 9012	NEW	Minor	21:14:33	Acknowledge
5	Driver F	27	TN04 KL 1011	NEW	Severe	21:14:03	Acknowledge
4	Driver D	34	DL05 GH 3456	DISPATCHED	Severe	21:13:33	Dispatch/Update, Resolve
3	Driver A	43	KA01 AB 1234	DISPATCHED	Major	21:13:03	Dispatch/Update, Resolve
2	Driver D	34	DL05 GH 3456	DISPATCHED	Major	21:12:33	Dispatch/Update, Resolve
1	Driver E	40	WB02 UJ 7890	DISPATCHED	Severe	21:12:03	Dispatch/Update, Resolve

Figure 4: Emergency Management and Incident Control Interface.

The Emergency Management module (Figure Y) provides the operational control center for managing and responding to ongoing incidents detected by the system. This interface emphasizes real-time decision making by allowing operators to update the state of each accident through a structured workflow. Each accident entry displays the Driver Name, Age, Vehicle Number, Severity, Time of Occurrence, and most importantly, the current Status of the incident, which can be NEW, ACKNOWLEDGED, DISPATCHED, or RESOLVED. This module enables the operator to perform key actions through dedicated control buttons such as Acknowledge, Dispatch/Update, and Resolve, thereby simulating the real-world workflow followed in emergency dispatch centers. The color-coded status labels enhance situational awareness—red indicating new cases, blue representing dispatched cases, and green for resolved cases. This visual differentiation ensures that operators can immediately identify pending emergencies and act accordingly. The system’s ability to update incident states in real time reflects a realistic emergency-response workflow and demonstrates the strength of the simulation engine in handling dynamic scenarios. This interface also serves as an educational tool for understanding how actual command centers coordinate ambulance dispatches, track high-severity incidents, and close cases after resolution. As a result, it highlights the

flexibility and effectiveness of the Smart Accident Response System in modeling complex emergency-management behavior.

6. Results Comparison Table

Parameter	Traditional Accident Detection Systems	Proposed Smart Accident Response System
Data Collection Method	Hardware sensors installed on vehicles (accelerometers, GSM modules).	Fully simulated GPS data and auto-generated accident events.
Cost and Deployment	High cost due to hardware installation and maintenance.	Very low cost; software-based system with no physical setup.
Safety in Testing	Real-world crash testing is risky and difficult to reproduce.	Completely safe simulation environment with repeatable scenarios.
Scalability	Limited — each added vehicle requires new hardware.	Highly scalable — unlimited simulated vehicles and locations.
Visualization	Little to no real-time map visualization.	Full GIS map with dynamic icons and accident markers.
Response Workflow	Manual, slow, and unstructured.	NEW → ACKNOWLEDGED → DISPATCHED → RESOLVED workflow.
Reporting	No automated route calculation.	Automated nearest-ambulance routing using Haversine formula.
Real Time Updates	Reports created manually.	One-click PDF reports generated for each incident.

Table 1: Results Comparison Table

Summary of Findings

Table 1 presents a comprehensive comparison between traditional accident detection systems and the proposed Smart Accident Response System. Traditional hardware-based systems depend heavily on installed sensors such as accelerometers and GSM modules for detecting collision events. While this approach is practical for real vehicles, it involves significant installation costs and cannot be safely tested for academic or research purposes. In contrast, the proposed system eliminates these constraints by adopting a simulation-driven design where accidents, GPS movements, and incident data are generated programmatically. This allows safe, repeatable, and flexible experimentation.

Another critical difference lies in visualization and workflow management. Traditional systems typically provide limited real-time display features and rely on manual reporting. The proposed system enhances operational clarity by integrating an interactive GIS map, a dashboard with real-time updates, and a structured emergency workflow (NEW → ACKNOWLEDGED → DISPATCHED → RESOLVED). This structured flow mimics real emergency-response centers and demonstrates the practicality of the system for training and analysis.

Furthermore, routing and reporting capabilities are significantly improved in the proposed model. Traditional systems lack automated route planning, whereas the proposed system calculates the nearest ambulance using the Haversine distance formula and enables quick PDF report generation for each

incident. Overall, the comparison highlights that the proposed system offers superior scalability, safety, flexibility, and real-time operational features, making it highly suitable for intelligent transportation research and emergency-response simulations.

7. CONCLUSION

The Smart Accident Detection and Emergency Response System successfully demonstrate how a simulation-driven approach can replicate real-world accident behaviour, emergency routing, and situational monitoring without relying on costly hardware. By integrating automatic accident generation, GPS-based vehicle movement, real-time visualization, and shortest-path ambulance routing, the system provides a coherent and comprehensive platform for understanding emergency response workflows. The dashboard interface enhances situational awareness by presenting all critical information—accident locations, timestamps, vehicle movement, and rescue routes—in a visually rich and interactive manner. The system proves to be highly scalable, cost-effective, and suitable for academic and research environments where real-world accident data is difficult to obtain and unsafe to reproduce. Overall, the project establishes a strong foundation for intelligent transportation research and offers significant potential for expansion into real-time deployments, predictive analytics, and city-level emergency management systems.

8. REFERENCES

1. M. R. H. K. S. P. A. L. T. R. O. J. P. “Real-Time Accident Detection Systems Using GPS and Sensor Simulation,” 2021 International Conference on Intelligent Transportation and Smart Mobility (ITSM), pp. 112–118, 2021.
2. S. A. Kumar and D. R. Patel, “Emergency Response Optimization Using Shortest Path Algorithms in Smart Cities,” 2022 4th International Conference on Smart Infrastructure and Transportation (ICSIT), pp. 325–331, 2022.
3. Y. F. N. T. L. W. Z. H. P. S. M. “Simulation-Based Accident Generation and Visualization for Intelligent Transport Systems,” IEEE Access, vol. 9, pp. 148920–148934, 2021.
4. L. B. Shah and R. M. Khadilkar, “GIS-Based Accident Mapping and Route Planning for Emergency Services,” 2020 International Conference on Geospatial Technologies and Applications (GeoTech), pp. 56–62, 2020.
5. P. K. Verma and S. Raj, “Automated Accident Monitoring Using Road Traffic Simulation Models,” 2023 International Conference on Computing, Networking and Intelligent Systems (ICCNS), pp. 1–7, 2023.
6. A. K. Das, B. Roy, M. Sen, and N. Gupta, “Optimizing Ambulance Dispatch Strategy Using Haversine and Dijkstra Algorithms,” 2021 International Conference on Computational Intelligence and Communication Networks (CICN), pp. 233–238, 2021.
7. F. L. C. T. D. R. S. M. “Real-Time Visualization Techniques for Emergency Transportation Analytics,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 4, pp. 3897–3908, Apr. 2022.
8. K. S. Rao and A. M. Jayanth, “Smart Emergency Routing System Using Dynamic GPS Simulation,” 2019 International Conference on Advanced Communication Technologies (ICACT), pp. 91–96, 2019.
9. N. P. Joseph, T. Abraham, and H. Mathew, “Evaluation of Simulation Frameworks for Accident Detection and Response,” 2020 International Conference on Computing and Information Sciences (ICCIS),

10. pp. 141–147, 2020.
11. S. Devi and R. K. Mishra, “A Comparative Study of Rescue Path Optimization Techniques in Accident Simulation Models,” 2018 International Conference on Computing, Power and Communication Technologies (ICCPCT), pp. 1–6, 2018.