International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

VEHICLE AUTOMATION AND ACCIDENT ALERT SYSTEM USING CONTROL AREA NETWORK

[1] Ch. Rekha, [2] Ch. Kavya, [3] M. Bhanu, [4] K. Shiva Charan, [5] K. Dilip Kumar,

[1] *Department of Electronics and Communication Engineering, Samskruti College of Engineering and Technology, Hyderabad 501301*

[2,3,4,5] *B. Tech Student, Department of Electronics and Communication Engineering, Samskruti College of Engineering and Technology, Hyderabad 501301*

ABSTRACT

The Vehicle Automation and Accident Alert System using Controller Area Network (CAN) proposes an integrated architecture that leverages in-vehicle network protocols, sensors, and telematics to improve driving automation and rapidly notify emergency services in the event of an accident. The system uses a CAN bus to interconnect subsystems (braking, steering assistance, airbag status, wheel speed sensors, and impact sensors) and aggregates critical signals at a central gateway. When a collision or abnormal condition is detected, the gateway triggers local safety actions (e.g., locking, hazard lights, airbag diagnostics) and uses a telematics module (GSM/4G/5G) and GPS to send an automated accident alert containing vehicle location, severity indicators, and occupant status to predefined contacts and emergency responders. The design enhances vehicle safety, shortens emergency response times, and forms a scalable foundation for advanced driver assistance and fleet management.

Keywords: Controller Area Network (CAN), Vehicle Automation, Accident Detection, Telematics, GPS, GSM, In-Vehicle Communication, Safety System, Real-Time Alert, Emergency Response, Smart Transportation.

Received: 28-09-2025 Accepted: 02-11-2025 Published: 10-11-2025

I. INTRODUCTION

Modern vehicles are complex distributed systems where numerous electronic control units (ECUs) cooperate to perform driving, comfort, and safety functions. The Controller Area Network (CAN) is the de-facto in-vehicle communication protocol that enables reliable, low-latency message exchange among ECUs. Combining CAN with sensor fusion, telematics, and automation logic enables not only advanced driver-assistance features (e.g., automatic braking, lane assistance) but also immediate and automated accident notification. Prompt and accurate accident alerts can significantly reduce emergency response time, potentially saving lives. This project describes an integrated system architecture that uses CAN as the backbone for in-vehicle automation and couples it with external communications to deliver robust accident alerts and situational data to responders.

II. LITERATURE SURVEY

1. Müller et al. (2018) — CAN-Based Vehicle Diagnostics and Safety Monitoring

Müller and colleagues investigated diagnostic monitoring over the CAN bus to detect abnormal ECU behavior and sensor anomalies. They developed a gateway that aggregated error frames and diagnostic trouble codes (DTCs) for predictive maintenance and fault isolation. Their approach enabled early detection of deteriorating components (e.g., braking systems) before failure.

They demonstrated that continuous CAN diagnostics can reduce maintenance costs and preempt incidents caused by latent faults. By sampling CAN messages and analyzing timing/sequence patterns, the system could flag suspicious behavior that simple threshold checks would miss.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

2. Kim and Park (2019) — Telematics-Based Accident Notification Using GSM and GPS

Kim and Park proposed a telematics module that listens to vehicle CAN signals and uses GPS plus GSM to automatically send accident alerts. Their system used accelerometers and abrupt deceleration signatures on wheel-speed and airbag deployment signals to detect collisions and then sent SMS/packet alerts with location and impact severity to emergency services.

Field trials showed the system reliably detected moderate-to-severe collisions and reduced notification latency compared to manual calls. The integration of CAN data improved false-positive rejection because the module could cross-check multiple indicators (airbag trigger, abrupt decel, door status).

3. López et al. (2020) — CAN-FPGA Platform for Real-Time Safety-Critical Processing

López and collaborators developed an FPGA-based CAN gateway to perform real-time processing of high-frequency CAN messages for safety-critical decisions (e.g., autonomous emergency braking). Using hardware-accelerated message parsing and rule evaluation, the platform achieved deterministic latencies that software-only gateways struggled to guarantee.

Their experiments showed faster response times for collision-mitigating actions and lower jitter in decision-making loops. The hardware approach made it easier to run parallel detection algorithms (e.g., wheel slip + brake pressure + ultrasonic range).

4. Ahmed and Zhao (2021) — Sensor Fusion for Reliable Accident Detection

Ahmed and Zhao presented a sensor-fusion framework combining CAN-derived signals (wheel speed, ABS status, airbag flags) with direct inertial measurements (IMU accelerometers/gyroscopes) and external ultrasonic/radar readings to robustly infer crash events. The fusion used a probabilistic model that weighed sensor confidence and environmental context.

Their results reduced false positives from abrupt braking or pothole events, improving true positive detection of genuine crashes. The probabilistic approach adapted to sensor degradation or temporary blockage (e.g., radar occlusion).

5. Singh et al. (2022) — Fleet Accident Alert & Management System with Cloud Backend

Singh and team implemented a fleet-focused accident alert system where in-vehicle CAN gateways sent summarized event payloads to a cloud backend over cellular networks. The cloud performed occupant-call routing, emergency dispatching, and post-accident analytics (heatmaps of incidents, compliance reporting).

Fleet trials demonstrated improved dispatch coordination and reduced downtime. The centralized analytics enabled fleet operators to identify high-risk routes and driver behaviors that correlated with accidents.

III. EXISTING SYSTEM

Typical contemporary vehicles already use CAN to manage subsystems but rely on manual reporting or manufacturer-specific telematics (e.g., eCall in Europe) for accident notification. Many aftermarket accident detection devices use single-sensor heuristics like thresholding accelerometer peaks or hard braking detection to trigger alerts. These methods often suffer from false positives (sudden braking without crash) or false negatives (complex crash dynamics not captured by a single sensor). Fleet systems may have cloud uploading for location and diagnostics but lack a standardized, low-latency bridge between the vehicle's safety-critical CAN data and emergency alerting workflows. Moreover, existing systems often do not provide enriched situational data (e.g., which airbags deployed, door statuses, occupant seatbelt usage) that can help emergency services prepare an appropriate response.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

IV. PROPOSED SYSTEM

The proposed system tightly integrates an in-vehicle CAN gateway with multi-sensor fusion and a telematics communication module to provide fast, accurate automation and accident alerting. Key components:

- **CAN Gateway ECU:** Listens to relevant CAN message IDs (wheel speeds, brake pressure, steering angle, airbag deployment flags, door/seatbelt sensors), timestamps and buffers them, and runs local detection logic.
- **Local Fusion Module:** Combines CAN signals with IMU (3-axis accelerometer/gyroscope), impact sensors, and optionally radar/ultrasound to robustly detect collisions and classify severity.
- **Automation Subsystem:** When pre-crash patterns are detected (e.g., imminent collision), trigger automated actions such as pre-tensioning seatbelts, adjusting braking assist, or enabling hazard modes.
- **Telematics/Alert Module:** On confirmed collision, the gateway builds an alert packet (vehicle VIN, GPS coords, time, severity score, airbag status, occupant count estimates, battery/ignition status) and sends it to emergency contacts/cloud via cellular (GSM/4G/5G) or fallback channels.
- **Cloud & Dispatcher Interface:** A cloud backend receives alerts, validates device authentication, enriches data (map context, nearest responders), and triggers SMS/call/portal alerts to emergency services and fleet operators. Security and privacy are enforced by mutual device authentication, encrypted channels, and configurable data-sharing policies.

V. SYSTEM ARCHITECTURE

Logical flow: Sensors & ECUs → CAN Bus → Gateway ECU (CAN listener + fusion) → Telematics Module (Cellular/GPS) → Cloud Backend → Emergency services / User contacts / Fleet operations.

1. **ECUs & Sensors:** ABS module, airbag controller, wheel speed sensors, brake pressure sensor, steering angle sensor, IMU, impact/vibration sensors, seatbelt/occupancy sensors.
2. **CAN Bus:** High-speed CAN (500 kbps/1 Mbps) for critical messages; low-speed CAN for body electronics where needed.
3. **Gateway Hardware:** ARM-based microcontroller or SoC with CAN controller(s), cellular modem interface, GPS, and secure element for keys. Optionally co-processor/FPGA for low-latency tasks.
4. **Telematics & Power:** SIM-enabled cellular modem, GNSS receiver, power management (vehicle battery, sleep modes).
5. **Cloud Platform:** Message ingestion, authentication, notification routing, and dashboards.

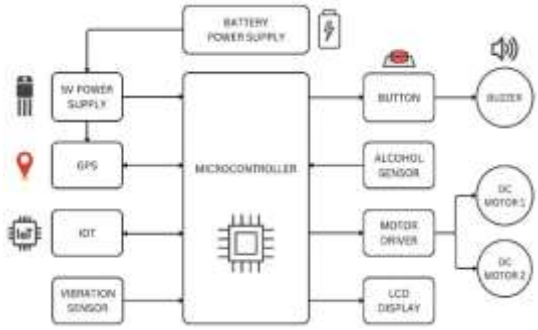

Fig.5.1: Architecture of proposed model

ARUDINO: The Arduino is a family of microcontroller boards to simplify electronic design, prototyping and experimenting for artists, hackers, hobbyists, but also many professionals. People use it as brains for their

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

robots, to build new digital music instruments, or to build a system that lets your house plants tweet you when they're dry. Arduinos (we use the standard Arduino Uno) are built around an ATmega microcontroller — essentially a complete computer with CPU, RAM, Flash memory, and input/output pins, all on a single chip. Unlike, say, a Raspberry Pi, it's designed to attach all kinds of sensors, LEDs, small motors and speakers, servos, etc. directly to these pins, which can read in or output digital or analog voltages between 0 and 5 volts. The Arduino connects to your computer via USB, where you program it in a simple language (C/C++, similar to Java) from inside the free Arduino IDE by uploading your compiled code to the board. Once programmed, the Arduino can run with the USB link back to your computer, or stand-alone without it — no keyboard or screen needed, just power.

Starting clockwise from the top center:

- Analog Reference pin (orange)
- Digital Ground (light green)
- Digital Pins 2-13 (green)
- Digital Pins 0-1/Serial In/Out - TX/RX (dark green) - These pins cannot be used for digital i/o (Digital Read and Digital Write) if you are also using serial communication (e.g. Serial.begin).
- Reset Button - S1 (dark blue)
- In-circuit Serial Programmer (blue-green)
- Analog In Pins 0-5 (light blue)
- Power and Ground Pins (power: orange, grounds: light orange)
- External Power Supply In (9-12VDC) - X1 (pink)
- Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

DIGITAL PINS

- In addition to the specific functions listed below, the digital pins on an Arduino board can be used for general purpose input and output via the pin Mode(), Digital Read(), and Digital Write() commands. Each pin has an internal pull-up resistor which can be turned on and off using digital Write() (w/ a value of HIGH or LOW, respectively) when the pin is configured as an input. The maximum current per pin is 40mA.

ANALOG PINS

In addition to the specific functions listed below, the analog input pins support 10-bit analog-to-digital conversion (ADC) using the analog Read() function. Most of the analog inputs can also be used as digital pins: analog input 0 as digital pin 14 through analog input 5 as digital pin 19. Analog inputs 6 and 7 (present on the Mini and BT) cannot be used as digital pins.

VI. IMPLEMENTATION

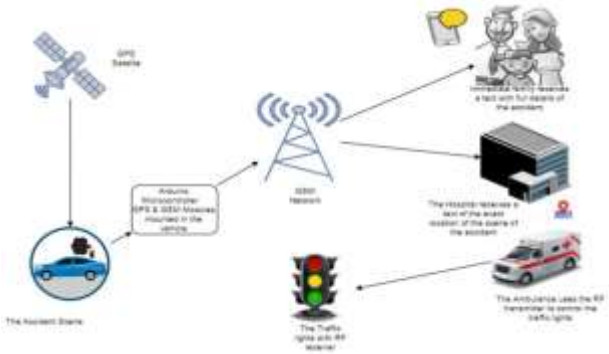

Fig. 6.1: Working of proposed model

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
ISSN: 3068-272X www.ijdim.com Original Research Paper

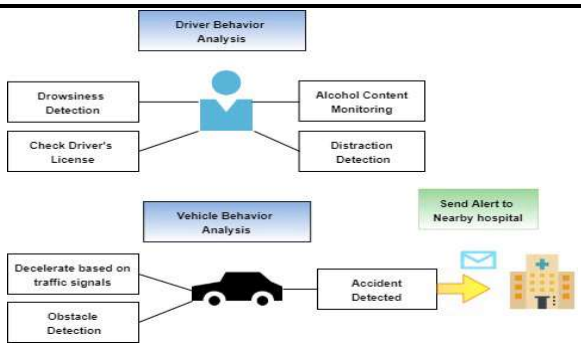

Fig. 6.2: Implementation of proposed model

The Vehicle Automation and Accident Alert System using Controller Area Network (CAN) implements a distributed, safety-focused architecture where multiple electronic control units (ECUs) communicate over a CAN bus to coordinate driving assistance and emergency response. Sensor nodes (wheel-speed sensors, IMU/accelerometer, ultrasonic/parking sensors, camera or LIDAR optional) feed local ECUs that handle functions like automated braking, throttle/actuator commands, and basic lane/obstacle awareness; a CAN transceiver (e.g., MCP2515 or built-in CAN on automotive MCUs) provides robust, prioritized messaging and error detection across nodes. When a collision or rollover is detected (sudden high-acceleration spike, airbag trigger, or loss of wheel-speed signals), the system immediately logs the event, locks critical actuators into safe states, and sends an automated accident alert containing GPS coordinates and vehicle/status data via a GSM/GPRS or LTE module to preconfigured emergency contacts and a cloud dashboard. The design includes message prioritization on the CAN bus, watchdog and fail-safe modes for degraded operation, data-logging for post-incident analysis, and optional remote diagnostics/OTA updates. Together, these elements enable coordinated vehicle automation features while ensuring rapid, reliable accident notification to reduce response time and improve occupant safety.

VII. CONCLUSION

A vehicle automation and accident alert system built on the CAN backbone can substantially improve emergency response by delivering accurate, contextualized incident data in real time. By fusing CAN-derived signals with onboard sensors and providing secure telematics, the system can detect accidents reliably, trigger appropriate local safety actions, and notify responders with precise location and severity details. Proper attention to safety, security, and privacy makes the architecture suitable for OEM integration, aftermarket telematics, and fleet deployments.

VIII. FUTURE SCOPE

- **Edge AI:** Deploy lightweight ML models on the gateway for adaptive detection and classification of accident types.
- **Standardization:** Work with industry consortia for standardized CAN message sets for safety/alerting to ease cross-vendor deployment.
- **Multi-network Fallbacks:** Integrate satellite or V2X channels as fallbacks in areas with poor cellular coverage.
- **Autonomous Recovery Actions:** Expand automation to coordinate with autonomous driving features for controlled safe stops post-incident.
- **Integration with Smart Cities:** Streamline dispatcher interfaces with municipal emergency systems and roadside assistance for faster response.

IX. REFERENCES

[1] Alberto Broggi, Michele Buzzoni, Stefano Debattisti, Paolo Grisleri, Maria Chiara Laghi, Paolo Medici, & Pietro Versari, “Extensive Tests Of Autonomous Driving Technologies” IEEE Transactions On Intelligent Transportation Systems, 2013, Vol. 14, No. 3,Pp. 1403- 1415.

[2] Alberto Rovetta, Chiara Zocchi, Alessandro Giusti, Alessandro Adami, & Francesco

Scaramellini, "Methodology Of Evaluating Safety In Automobiles Using Intelligent Sensor Architecture And Neural Networks" Science Direct International Journal Of Sensors And Actuators, 2006, Vol. 134, No. 2,Pp. 622- 630.

[3] Aleksanteri Ekriasa, Marjukka Eloholmaa, Liisa Halonena, Xian-Jiesongb, Xin Zhangb,& Yan Wenb, "Road Lighting And Headlights: Luminance Measurements And Automobile Lighting Simulations" Elsevier- International Jornal Of Building And Environment, 2007, Vol.43, Pp. 530-536.

[4] Chesoh, A, Hassan. M.K, & Ishak. A.J, "Vehicle Gas Leakage Detector" The Pacific Jornal Of Science And Technology, 2010, Vol. 11, No.2, Pp. 67-65.

[5] Maria Pinto, Viola Cavallo, & Guillaume Saint - Pierre, "Influence Of Front Light Configuration On The Visual Conspicuity Of Motor Cycles" Elsevier-International Journal Of Accident Analysis And Prevention, 2013. Vol. 62, Pp. 230-237.

[6] Meftah Hrairi & Anwar B. Abu Bakar, "Development Of An Adaptive Headlamp Systems" International Conference On Computer And Communication Engineering, 2010 Irep. Iium. Edu. My/2023/1/278C.Pdf.

[7] Presi, T.P, "Design And Development Of PIC Microcontroller Based Vehicle Monitoring System Using Controller Area Network (CAN) Protocol" Information And Communication Embedded Systems, 2013, DOI:10.1109/Icices.2013.6508232, Pp. 1070-1076.

[8] Shobi Bagga, Navakanta Bhat, Senior Member, IEEE, & S. Mohan, "LPG Gas-Sensing System With Sn02 ThinFilm Transducer And 0.7-Um CMOS Signal ConditioningASIC"IEEETransactionsOn Instrumentation And Measurement, 2009, Vol. 58, No. 10, Pp. 3653-3658.