IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, May 2025
Impact Factor: 7.67

Accident Detection and Emergency Response System

Sawardekar Omkar Pravin, Pawar Aditya Dadasaheb, Pharate Ansh Chetan
Rohad Tanushree Rajesh, Shah Griva Minesh
All India Shri Shivaji Memorial Society's Institute of Information Technology, Pune, India

Abstract: *Road traffic accidents continue to be a major cause of death and injury globally. Fatal accidents are increasing to 47.3% of all crashes up to 2023. To solve this, we introduce a new intelligent accident detection system that combines on-board IoT sensors with a machine-learning (ML) model to dynamically adjust detection thresholds depending on driving conditions. The system constantly tracks vehicle movement (e.g. acceleration, rotation, speed) and environment, employing an ML classifier to identify actual crashes versus regular driving scenarios. When it detects an accident, it notifies emergency contacts automatically and talks to nearby intelligent traffic lights to provide a green corridor for ambulances. We deploy the system on Arduino/Raspberry Pi platforms and compare its performance via the iFogSim simulator. Our performance measures are detection rate, false alarm rate, mean response latency, and network overhead. Experimental outcomes exhibit greater than 95% detection accuracy, minimal false positives, and timely alerting, greatly bettering static- threshold baselines. These outcomes indicate that adaptive IoT-ML systems have the potential to radically enhance emergency response while keeping the cost of networks and latency low.*

Keywords: traffic accidents

I. INTRODUCTION

Road crashes cause more than 1.35 million deaths and 50 million injuries each year globally 1. The World Health Organization cites traffic accidents as the fifth global leading cause of death 1. In most regions the percentage of fatal crashes is on the increase; for instance, fatal crashes in Pakistan rose from 40.1% of all crashes in 2014 to 47.3% in 2023. Quick detection and intervention of accidents are therefore paramount to preventing loss of lives. Conventional emergency response depends on witnesses or delayed calls, commonly resulting in tardy dispatch of ambulances. To enhance this, researchers have proposed automated accident detection through car or mobile phone sensors. IoT sensors such as accelerometers and GPS can alert crashes, whereas machine learning (ML) can evaluate sophisticated sensor patterns to eliminate spurious alarms.

Current solutions have shortcomings. Most smartphone-based detectors employ plain thresholds on accelerometer data and suffer frequent false positives at low speeds. 4. Hardware-based systems with fixed thresholds often cannot distinguish minor jolts from real crashes.

5. Vision-based or deep-learning methods (e.g. dash-cam CNN/GRU models) need large datasets and will not work if the camera is broken\t. Further, while some studies emphasize the accuracy of detection, they rarely interface with traffic management to accelerate ambulance response. Current emergency vehicle priority systems often pre-empt traffic lights by setting them green along the route, but they do not adjust dynamically to dynamic traffic or road conditions.

In this work, we introduce an extensive IoT-ML based accident detection and response system. Major contributions are: (1) Adaptive Detection: an ML algorithm which dynamically changes the detection thresholds according to current driving circumstances (speed, road type, weather) in order to reduce unnecessary alarms; (2) Integrated

Alerting: a built-in alert module that sends automatic notifications to predefined contacts with crash location and severity, and syncs with proximate smart signals to clear an ambulance route; (3) Prototype Implementation: implementation of Arduino/Raspberry Pi with accelerometer, GPS, and communication modules, and fog computing

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, May 2025
Impact Factor: 7.67

simulation (iFogSim) for real-time testing; (4) Performance Analysis: quantitative assessment of accuracy in detection, latency, rate of false positives, and network overhead, in comparison to baseline approaches. The remainder of the paper discusses related work, our system design and implementation, results, and future directions

II. LITERATURE REVIEW

A number of accident detection methods based on IoT have been researched. Smartphone-based methods leverage native accelerometers, GPS, and microphones to deduce crashes, but generally depend on single- sensor thresholds and exhibit high rates of false alarms (particularly at low speeds). For instance, Patel et al. used an Android application based solely on acceleration and GPS; its main drawback is single-sensor reliance resulting in crashes if the accelerometer fails. Hardware-based solutions place specific sensors (e.g. vibration, pressure) on the vehicle. These are more reliable but tend to utilize static thresholds: if impact force or braking is above a set threshold, an alarm is produced. In most such systems the precision is limited and expense is great because of onboard processing units.

Machine learning has been utilized to enhance detection. Choi et al. utilized a dashboard camera and ensemble deep learning (CNN+GRU) to determine accidents, but this is dependent upon video information which could be unavailable or unutilizable (e.g. damaged cameras). Another strategy blended CNN and SVM on visual features with ~85% accuracy ; however, this is not adequate for real-world safety-critical applications.

Data mining across large accident databases has revealed patterns (e.g. cause vs. environment) via association rules and clustering, but these are descriptive analyses and not in real-time detectors. In general, deep learning techniques demand large training data, which is frequently limited in on- board environments.

Integrating emergency response is another research focus. Conventional systems alert one rescue center upon crash. More sophisticated IoT-alert architectures (e.g. Shaik et al.) collect GPS and accelerometer data, transmit to the cloud, then push notifications (such as accident severity and location) to all registered emergency contacts. Traffic management strategies prioritize emergency vehicles by making signals green along their route 8. Kaisar et al. observe that most current schemes establish fixed routes or disregard the congestion around them 8. Therefore, there is room for a general system that not only identifies accidents with minimal false alarms, but also clears ambulances' paths dynamically.

In short, the existing work emphasizes the requirements of multi-sensor fusion and adaptive algorithms. Most systems are plagued with single-point failure or rigid thresholds. ML models using vision can spot accidents but rely on high-quality data and won't work in certain scenarios. IoT-based emergency routing (e.g., smart traffic lights) has been explored, but tends to rely on predetermined paths. Our proposed system combines these strands by using on-vehicle IoT sensors, an ML classifier for flexible thresholding, and IoT-enabled traffic signal control, aiming to overcome the limitations of each.

III. PROPOSED METHODOLOGY

System architecture is split into perception, processing (fog), and application layers. Sensor Module: There are several sensors per vehicle, e.g. a tri-axial accelerometer/gyroscope (MPU-6050) and GPS module. They capture sudden impact/deceleration and position. Extra sensors (heartbeat monitor, alcohol sensor, etc.) can optionally provide additional data. Edge Processing (Fog Node): A Raspberry Pi or a microcontroller connected to the vehicle continually reads sensor data. An onboard ML model (a decision tree or compact neural network) inspects the arriving sensor stream. The model is trained against labeled crash vs. non-crash data that includes features such as impact force, velocity change, and context variables (current speed, time of day, weather). Most importantly, rather than having a static threshold on acceleration, the model provides a dynamic "crash score" that varies with conditions – i.e., it needs a greater acceleration change to indicate a crash if the vehicle is traveling at high speeds, and needs a lower threshold on slippery roads where braking distances are longer. This adaptive thresholding reduces false alarms over fixed rules. Alert Module: In the event of a detected crash (score above threshold), the system initiates an alert process. It utilizes GSM/Wi-Fi to automatically send the car's GPS coordinates, time, and estimated severity to pre-registered emergency contacts (family or first responders). At the same time, it interacts with a connected traffic management system through IoT messages. Particularly, it makes a priority request to the local smart traffic lights along the best route to the closest

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, May 2025
Impact Factor: 7.67

hospital, forcing those lights to turn green in succession. This is an extension of current "ambulance preemption" systems, but is event- triggered by our detection and may have to adapt if traffic conditions improve on the journey.

Major technical elements are: - Adaptive ML Model: A supervised classifier (e.g. random forest) trained on past crash sensor traces. Input characteristics are accelerometer peaks, deceleration rate, gyro rotation, speed, and environmental factors. The decision boundary of the model effectively deploys a context-sensitive threshold. - Communication: The edge device in the vehicle communicates through cellular or dedicated short-range communications to a fog server or cloud. Alerts take the form of short packets (location, ID) to keep network loads low. - Fog Simulation (iFogSim): For testing, we simulate a smart city scenario using iFogSim.

Cars produce sensor data streams; fog nodes (edge servers) execute accident inference, and traffic signals are coordinated by a central controller.

The overall strategy combines sensing, machine learning, and networked infrastructure. Through in-time detection sensitivity adjustment and integration with traffic control, the system seeks to optimize true detection and reduce response time.

IV. SYSTEM IMPLEMENTATION

A prototype was built with an Arduino Uno and a Raspberry Pi as the fog/edge node. The Arduino (or substitute microcontroller) connects with an MPU-6050 gyroscope/accelerometer and a GPS module. Sample sensor readings (speed, acceleration, orientation) are input into the Raspberry Pi via a serial connection. The Pi executes a Python-based ML model (scikit- learn) pre-trained on crash data sets. The Pi sends SMS notifications using a GSM/GPRS shield or Wi-Fi when the model is triggered. An MQTT message is also sent to a local server that controls traffic lights.

To simulate, we used the iFogSim toolkit. We simulated 50 vehicles as IoT nodes producing sensor readings, one fog server for each 10 vehicles, and a cloud data center. Two smart traffic lights were simulated as fog actuators. Random crash events were injected during simulation runs at different speeds and locations. We evaluated: (a) Detection Accuracy– percentage of actual crashes correctly identified

(b) False Positive Rate– normal occurrences incorrectly signaled, (c) Detection Latency– latency from crash to alert delivered, and (d) Network Overhead– information transferred per event. The parameters of the ML model were optimized using cross-validation. Everything is event-driven in how the software components (sensor reading, ML inference, communications) are implemented to permit real-time performance testing.

V. RESULTS AND DISCUSSION

The analysis demonstrates the system achieves high reliability and low latency. Table 1 summarizes important metrics comparing our adaptive ML approach to a baseline static-threshold system (fixed acceleration cutoff). Metric Proposed System Baseline System

- Detection Accuracy: 96.0% to 90.2%
- False Positive Rate: 4.0% to 10.0%
- Average Latency (ms): 120 200
- Network Load (KB/alert): 20.5 to 30.0

Our adaptive ML model detected 96% of test crashes correctly, much higher than the 90% achieved by fixed-threshold logic. The false positive rate was also decreased (<5% vs. 10%), showing improved discrimination (i.e., fewer false alarms in everyday driving). Detection latency was 120 ms on our hardware (more than sufficient for real-time requirements), compared to 200 ms in the baseline, because the model decision-making and fog processing were faster. Overhead per alert was reasonable: just the basic crash summary (ID, location) – roughly 20 KB – was transmitted, compared to larger packets in the baseline experiment. In our simulation using iFogSim, deployment of fog reduced network congestion: edge server coordination allowed ~30% less data transfer to the cloud than an all-cloud architecture (in line with Gupta et al.'s observations regarding fog latency and congestion).

These findings confirm the suggested system adheres to real-time requirements. Machine learning successfully adjusts the detection threshold; e.g., at high speed it can accept larger sensor spikes before triggering, and at low speed it is still

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 7, May 2025
Impact Factor: 7.67

sensitive enough to detect soft bumps. This adaptation kept "gray-zone" misses seen in fuzzy-logic implementations to a minimum. The integrated control of traffic signals also worked well: as soon as an accident alert was triggered, simulated phase of signals was instantaneously shifted to green on the selected path, reducing estimated ambulance travel time by ~20%. In comparison to alert systems that only trigger a single rescue center, our multi-recipient alert and intelligent-light clearing can significantly shorten response time.

In summary, the metrics of performance (accuracy, latency, overhead) validate the viability of our design. The outcomes validate earlier findings that fog-based IoT systems can enhance emergency response performance. In contrast to certain deep-learning proposals demanding extensive video databases, our sensor/ML method is lightweight and scalable. A comparison with current solutions reveals that our system's accuracy as well as false alarm rate are competitive or better: e.g. earlier accelerometer-based systems had high false positives in low-speed conditions, which our adaptive model prevents.

VI. CONCLUSION

We have introduced an intelligent accident detection and response system which utilizes IoT sensors and machine learning for enhanced safety. By calibrating detection thresholds to driving situation, the system significantly lowers false alarms without compromising crash detection accuracy. The automated alert module quickly alerts emergency contacts and collaborates with intelligent traffic lights to reduce ambulance travel time. Our Arduino/Raspberry Pi prototype and iFogSim-based simulation prove the feasibility of the approach: the system attained ~96% accuracy, low latency, and negligible network load. In practice, which equals faster, more efficient emergency alerts and, perhaps, lives saved. With the rising number of fatal accidents, intelligent systems like this are crucial.

Future work will involve real-world trials with expanded sensor sets (e.g. road condition inputs), and exploration of adaptive learning (online calibration of thresholds). Further integration with urban traffic management platforms could allow dynamic route optimization beyond simple signal control. The proposed framework is extensible, and by combining IoT, ML, and fog computing it provides a robust foundation for next-generation emergency response in smart cities.

REFERENCES

[1]. https://thesai.org/Downloads/Volume16No3/Paper_65-IoT_Based_Smart_Accident_Detection.pdf

[2]. [1606.02007] iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments

[3]. https://ar5iv.labs.arxiv.org/html/1606.02007

[4]. AI Enabled Accident Detection and Alert System Using IoT and Deep Learning for Smart Cities

[5]. https://www.mdpi.com/2071-1050/14/13/7701

[6]. IoT-Based Emergency Vehicle Services in Intelligent Transportation System - PMC

[7]. https://pmc.ncbi.nlm.nih.gov/articles/PMC10256047/