IoT Vehicle Accident Detector

Michael Leib, Yuval Yacov Said
Ben Gurion University
Be'er Sheva, Israel

***Abstract*— Vehicle accidents pose significant challenges to emergency response systems. To address delayed responses, we introduce an Automated Vehicle Accident Alert System leveraging IoT technologies. This system monitors vehicle movements in real-time using advanced sensors and triggers instant alerts upon accident detection. Challenges include sensor accuracy, interoperability, and security. Future efforts should focus on enhancing these aspects. Our project contributes to proactive accident detection, enhancing emergency response and overall safety.**

I. INTRODUCTION

In today's fast-paced world, vehicular accidents continue to be a significant concern, not only due to their potential for causing injury or loss of life but also because of the challenge they pose to emergency response systems. One of the critical factors contributing to the severity of accidents is the time it takes for emergency services to arrive at the scene. Delayed emergency response can exacerbate injuries, increase the likelihood of fatalities, and result in further damage to property. This problem is particularly pronounced in situations where accidents occur in remote locations or when victims are unable to call for help themselves.

To address the issue of delayed emergency response in vehicle accidents, innovative solutions leveraging Internet of Things (IoT) technologies have emerged. One such solution is the Automated Vehicle Accident Alert System, which aims to revolutionize the way accidents are detected, reported, and responded to. By combining advanced sensor technologies with cloud-based communication and analysis platforms, this system enables real-time monitoring of vehicle movements and instant alert generation in the event of an accident.

Vehicle Accident Alert System Overview

The Automated Vehicle Accident Alert System utilizes state-of-the-art hardware components, including the ESP32 microcontroller, MPU6050 accelerometer and gyroscope, and GY-GPS6MV2 GPS module, to gather crucial data related to vehicle motion, orientation, and location. These sensors continuously monitor the vehicle's behavior, allowing for the detection of abnormal events such as sudden accelerations, decelerations, or changes in orientation that may indicate an accident.

Once an accident is detected, the system securely transmits real-time sensor data to the AWS IoT Core, a cloud-based service that facilitates seamless communication between IoT devices and the cloud. Within the AWS environment, incoming data is analyzed in real-time, and if an accident is confirmed, instant alerts and notifications are generated and sent to designated emergency contacts, including emergency services, family members, and relevant authorities. Additionally, all sensor data, including pre- and post-accident information, is logged and stored in AWS databases for further analysis and forensic purposes.

II. BACKGROUND:

Vehicle accidents represent a significant public health concern worldwide, with millions of fatalities and injuries occurring annually. Prompt emergency response plays a crucial role in mitigating the consequences of these accidents. Studies have shown that timely medical intervention significantly improves the chances of survival and reduces the severity of injuries sustained in accidents. However, the effectiveness of emergency response is often hindered by delays in reporting and dispatching assistance to accident scenes.

In many cases, victims may be incapacitated and unable to call for help themselves. Additionally, accidents that occur in remote or isolated areas may face challenges in accessing emergency services promptly. The importance of reducing response times in such situations cannot be overstated, as every minute saved can make a difference in the outcome for accident victims.

Review of Existing Literature and Technologies:

In recent years, there has been a growing interest in leveraging IoT technologies to enhance emergency response systems for vehicle accidents. Various research studies and technological innovations have explored the use of IoT devices and cloud-based platforms to detect accidents, transmit distress signals, and facilitate rapid response from emergency services.

IoT-based accident alert systems typically utilize sensors such as accelerometers, gyroscopes, and GPS modules to monitor vehicle movements and detect abnormal events indicative of accidents. These sensors are integrated into the vehicle's onboard systems or attached externally to enable real-time data collection and analysis. Upon detecting an accident, the system triggers automatic alerts to predefined contacts, including emergency services and designated individuals.

Furthermore, cloud based IoT platforms have emerged as essential components of accident alert systems, providing scalability, reliability, and real-time communication capabilities. Platforms like AWS IoT, Microsoft Azure IoT, and Google Cloud IoT offer robust infrastructure for securely transmitting, storing, and processing sensor data, enabling seamless integration with emergency response protocols.

Gaps and Limitations in Current Approaches:

Despite the advancements in IoT-based accident alert systems, several challenges and limitations persist. One significant limitation is the reliability and accuracy of sensor data in accurately detecting accidents. False positives and false negatives can occur due to sensor errors, environmental factors, or system malfunctions, leading to unnecessary alerts or missed incidents.

Moreover, interoperability and standardization issues may arise when integrating IoT devices from different manufacturers or deploying systems across diverse geographical regions. Ensuring seamless communication and compatibility between hardware components and software platforms is crucial for the effective operation of accident alert systems.

Furthermore, privacy and security concerns surrounding the collection, transmission, and storage of sensitive data pose additional challenges. Safeguarding user privacy and protecting against unauthorized access or data breaches requires robust encryption, authentication, and access control mechanisms.

Addressing these gaps and limitations is essential for the widespread adoption and effectiveness of IoT-based accident alert systems. Future research and development efforts should focus on enhancing sensor accuracy, improving interoperability, and implementing robust security measures to ensure the reliability and efficiency of these systems in real-world scenarios.

III. System Architecture

Hardware Components:

The Automated Vehicle Accident Alert System incorporates several hardware components to enable real-time monitoring of vehicle movements and facilitate prompt emergency response. The key hardware components include:

ESP32 Microcontroller: The ESP32 microcontroller serves as the central processing unit of the system. It is equipped with built-in Wi-Fi and Bluetooth capabilities, making it ideal for wireless communication and data transmission. The ESP32 processes sensor data collected from the accelerometer, gyroscope, and GPS module [3] and communicates with the AWS IoT Core for further analysis and alert generation.

MPU6050 Accelerometer, Gyroscope, and Temperature Sensor: The MPU6050 sensor is a combination of a three-axis accelerometer, a three-axis gyroscope, and a temperature sensor. It accurately measures the vehicle's acceleration, orientation, angular velocity, and temperature, providing comprehensive insights into its motion dynamics. The accelerometer detects sudden changes in velocity or impact forces associated with accidents, while the gyroscope measures rotational movements, aiding in accident detection and reconstruction. Additionally, the temperature sensor monitors the ambient temperature, which can provide valuable contextual information for accident analysis and environmental monitoring.

GY-GPS6MV2 GPS Module: The GY-GPS6MV2 GPS module provides precise location tracking and positioning data. It communicates with satellites to determine the vehicle's latitude, longitude, and altitude coordinates in real-time. The GPS module enables accurate geo-fencing and incident localization, allowing emergency responders to pinpoint the exact location of accidents and dispatch assistance efficiently.

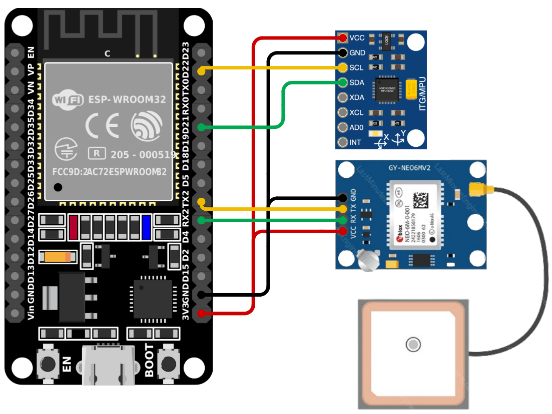

Figure 1- Hardware Setup

AWS IoT Services:

The Automated Vehicle Accident Alert System leverages various AWS IoT services to securely transmit, store, and process sensor data and facilitate instant alert generation. The integration with AWS services includes:

AWS IoT Core: AWS IoT Core acts as the central communication hub for the system, facilitating bidirectional communication between IoT devices and the cloud. It securely manages device connections, authenticates and authorizes device access, and enables seamless data exchange. The ESP32 microcontroller securely transmits sensor data to AWS IoT Core using MQTT protocol.

AWS Lambda: Lambda functions are used for serverless event-driven computing, enabling automatic and scalable execution of code in response to events. Lambda functions can be triggered by AWS IoT Core rules or SNS notifications to perform custom actions, such as data processing, analytics, or integration with external systems.

Amazon Simple Notification Service (SNS): SNS is utilized for instant alert generation and notification delivery. Upon detecting an accident, the AWS Lambda triggers a notification event, which is then routed to predefined recipients, including emergency services, family members, and relevant authorities, via SMS, email, or push notifications.

Amazon Timestream: Amazon Timestream is employed as a purpose-built time-series database service for storing sensor data and accident-related information. It offers scalability, high availability, and optimized storage for time-stamped data, making it suitable for real-time data ingestion and historical analysis. Sensor data, along with metadata such as timestamp and location, is logged into Amazon Timestream tables for efficient storage and retrieval.

Grafana: Grafana is employed for visualizing and analyzing sensor data through customizable dashboards and graphs. It allows users to gain insights into vehicle behavior, accident patterns, and response times. Grafana dashboards provide real-time monitoring of sensor readings, alert statuses, and historical trends, enabling stakeholders to make informed decisions and improvements.

System Architecture and Data Flow:

The overall system architecture of the Automated Vehicle Accident Alert System is depicted as follows:

Data Collection: The MPU6050 accelerometer and gyroscope continuously monitor the vehicle's motion and temperature, while the GY-GPS6MV2 GPS module tracks its location in real-time. Sensor data is collected by the ESP32 microcontroller, which preprocesses and packages the sensor data into JSON format, ensuring compatibility and efficiency for transmission to the AWS cloud infrastructure.

Data Transmission: The ESP32 securely transmits sensor data to the AWS IoT Core using encrypted MQTT connections over Wi-Fi. AWS IoT Core authenticates and authorizes the device connection and forwards the incoming data to downstream AWS services.

Real-time Analysis: Upon receiving sensor data, AWS IoT Core triggers rule-based actions to process and analyze the data in real-time. Custom Lambda functions may be invoked to perform data validation, and accident detection algorithms.

Alert Generation: If an accident is detected, AWS Lambda publishes a notification event to the Amazon SNS topic. SNS delivers instant alerts to predefined recipients via SMS, email, or push notifications, including emergency services and designated contacts.

Data Storage and Visualization: Sensor data, along with metadata, is stored in Amazon Timestream tables for historical analysis and forensic purposes. Grafana dashboards provide real-time visualization of sensor readings, alert statuses, and historical trends, enabling stakeholders to monitor system performance and make data-driven decisions.

IV. IMPLEMENTATION DETAILS:

Hardware Setup and Configuration:

The implementation of the Automated Vehicle Accident Alert System begins with the setup and configuration of the hardware components. This involves:

- Installing and configuring the ESP32 microcontroller, MPU6050 accelerometer and gyroscope, and GY-GPS6MV2 GPS module on the vehicle.
- Ensuring proper power supply and connectivity for all components.
- Download the project code from our GitHub repository.
- Initialize an MQTT client object in your code, providing it with the AWS IoT endpoint, port, and client ID
- Establish a connection to AWS IoT Core using the loaded certificates and keys.

Once the hardware components are set up and configured, the system is ready to collect, process, and transmit sensor data.

AWS Setup and Configuration:

- AWS Account Setup and IAM Configuration: An AWS account was created to facilitate the deployment of the IoT project. IAM roles and permissions were meticulously configured to ensure secure access to AWS services. Specifically, a role was created for IoT Core with permissions tailored to interact with various services including IoT Core, Lambda, SNS, Timestream, and others. Policies were attached to this role to grant necessary permissions.

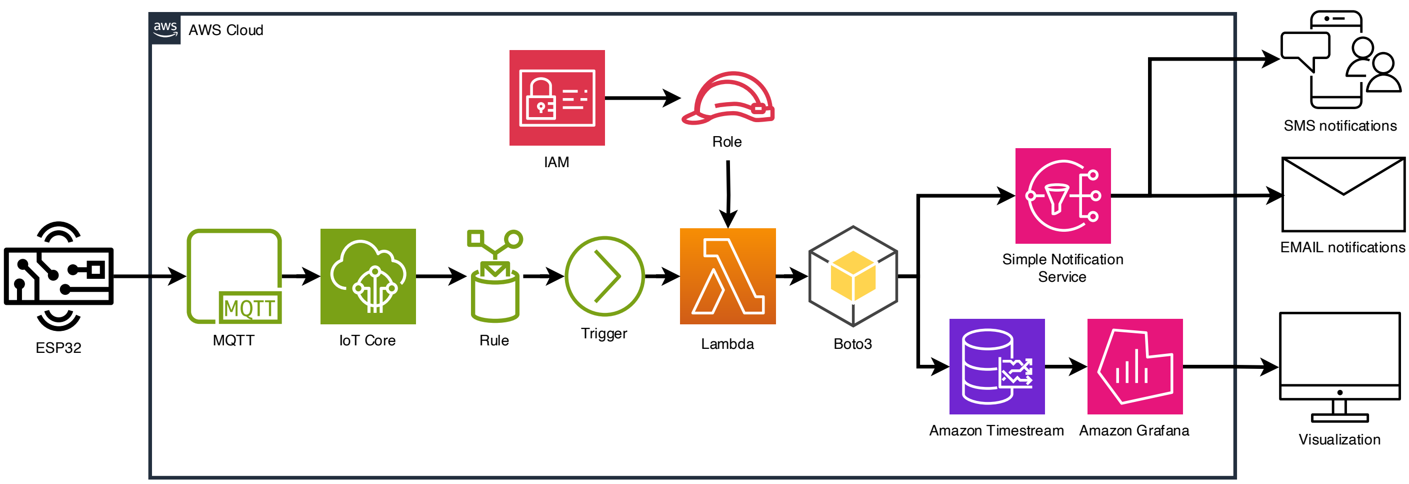

Figure 2 AWS Architecture

- AWS IoT Core Setup: Within AWS IoT Core, a representation of the vehicle was established as an IoT Thing. To facilitate communication, a policy was generated and attached to the thing. This policy allowed the thing to both publish and subscribe to MQTT topics, enabling seamless data exchange.
- AWS IoT Core Rules: IoT Core rules were implemented as triggers for Lambda functions, utilizing raw MQTT message data. These rules act as essential intermediaries, ensuring seamless transmission of the data to designated Lambda functions.
- Amazon SNS Setup: An Amazon SNS topic was created dedicated to accident alerts and notifications. Subscription endpoints were configured to notify designated recipients, including emergency services and relevant authorities, through various channels such as SMS, email, or push notifications.
- Amazon Timestream Setup: Amazon Timestream was chosen as the database to store time-series sensor data. Tables were created within Timestream to efficiently store and retrieve sensor readings along with associated timestamps, ensuring data integrity and accessibility for analysis.
- Amazon Managed Grafana Setup: Grafana was deployed using the Amazon Managed Grafana service, providing a robust platform for real-time visualization of sensor data. Data sources were configured to connect seamlessly to Amazon Timestream, enabling dynamic visualization of sensor data.
- Lambda Function Development: The Lambda function effectively parses incoming sensor data, extracting relevant information such as acceleration and temperature, and stores it in AWS Timestream for further analysis.
- Lambda Function Development: Through careful logic implementation, the function detects critical events like accidents or fires based on sensor readings, promptly triggering notifications via AWS SNS to alert designated recipients, facilitating swift response and mitigation actions.
- Note: You can find the Lambda function code in our GitHub repository.


```json
{
  "acceleration_x": 10.33337498,
  "acceleration_y": -0.174776718,
  "acceleration_z": -0.622492433,
  "temperature": 25.42411804,
  "latitude": 31.259058,
  "longitude": 34.79650497,
  "timestamp": "2024/3/11 08:52:13.00"
}
```


Figure 3 MQTT Message

V. SECURITY MEASURES:

ESP32 Security Measures:

Utilization of "secrets.h" for storing sensitive information like WiFi credentials and AWS IoT details separately, reducing the risk of inadvertent exposure.

Implementation of MQTT over TLS for encrypted data transmission, along with X.509 authentication mechanisms for device authentication, ensuring secure communication with AWS IoT Core.

AWS Services Security Measures:

Integration of AWS Identity and Access Management (IAM) for robust security measures across the AWS IoT ecosystem.

Fine-grained access control through IAM policies, enabling granular permissions management and enforcing the principle of least privilege.

Enforcement of IAM policies to restrict access to sensitive resources and ensure a heightened overall security posture.

These security measures collectively fortify the IoT project's infrastructure, safeguarding sensitive data and interactions both at the device and cloud levels. By implementing secure storage practices, encryption protocols, and access controls, the system mitigates risks associated with unauthorized access and data breaches, ensuring the confidentiality, integrity, and availability of its resources.

VI. *EXPERIMENTAL VALIDATION:*

Our experimental validation of the Automated Vehicle Accident Alert System involved thorough testing on an RC car platform to simulate real-world scenarios. Utilizing an RC car (Figure 4) allowed us to replicate vehicle movements and potential accident scenarios in a controlled environment, facilitating comprehensive evaluation of the system's functionality and performance.

Figure 4 RC Car Experiment

We conducted a series of scenario-based tests to assess the system's ability to detect abnormal events such as sudden accelerations, decelerations, and collisions, triggering instant alerts and notifications as appropriate. The integration of sensors and IoT devices on the RC car enabled real-time monitoring of vehicle dynamics, ensuring timely detection and reporting of accidents.

Our experiments on the RC car platform provided valuable insights into system behavior and performance under various conditions.

Figures 5 and 6 illustrate the results of our testing: Figure 5 depicts the Grafana Dashboard, showcasing real-time visualization of sensor data, while Figure 6 showcases an Email Notification generated by the system in response to an accident event, providing crucial details such as the accident location, timestamp, and severity level.

These visual representations validate the effectiveness of our system in detecting and responding to accident events promptly, demonstrating its suitability for deployment in actual vehicular environments.

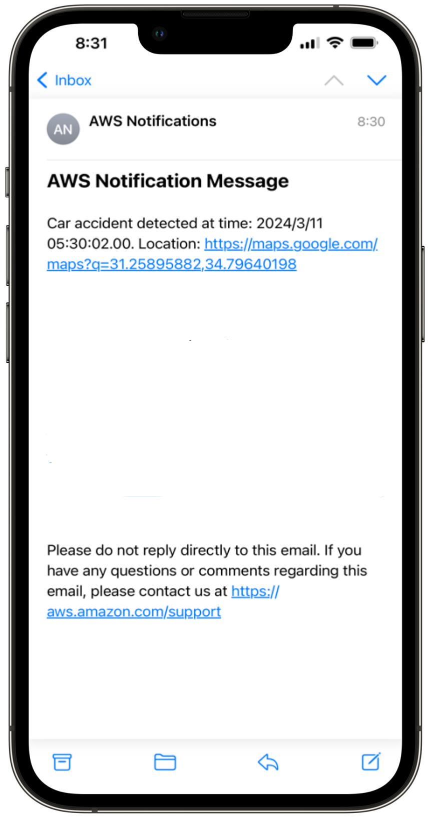

Figure 6 Email Notification

Figure 5 Grafana Dashboard

VII. RELATED WORK:

In the field of traffic accident detection and rescue systems, our project aligns with many significant studies, each offering distinct perspectives and insights. One such study introduces an IoT-based Accident Detection System [2].

The paper outlines an IoT-based accident detection system comprising a Raspberry Pi single-board computer, an accelerometer (ADXL345), and a GPS module. This system autonomously identifies vehicular accidents by monitoring abrupt changes in acceleration using the accelerometer and retrieves location data through the GPS module.

Upon accident detection, the system transmits relevant data, including severity level and GPS coordinates, to a cloud server via the ThingSpeak API. Subsequently, the server disseminates accident notifications to emergency services and pre-defined contacts via Twilio for text messaging.

Furthermore, the system continuously tracks the vehicle's location at regular intervals post-accident until manual intervention by authorities.

Our system closely resembles the one described above; however, we utilize an ESP32 controller and the Amazon cloud infrastructure. Additionally, we incorporate temperature sensors to detect fire incidents, a feature not addressed in their article.

VIII. CONCLUSION:

The IoT Vehicle Accident Detector project represents a significant advancement in the realm of vehicular safety and emergency response systems. Through the integration of sensor technologies, cloud-based communication platforms, and security measures, our system offers a proactive approach to accident detection and response.

By addressing the challenges associated with delayed emergency response in vehicle accidents, our project aims to minimize response times and improve outcomes for accident victims. The real-time monitoring capabilities provided by the system enable swift detection of accidents and prompt alert generation to designated emergency services and contacts, facilitating rapid intervention and assistance.

Furthermore, our experimental validation on an RC car platform demonstrated the effectiveness of our system in detecting abnormal events such as sudden accelerations, decelerations, and collisions, thus validating its suitability for deployment in actual vehicular environments.

The project introduces an innovative IoT-based vehicle accident detection system designed to enhance emergency response efficiency. This system incorporates advanced sensor technologies, including the MPU6050 accelerometer and gyroscope, the GY-GPS6MV2 GPS module, and temperature sensors. These sensors are strategically integrated into the vehicle to continuously monitor its dynamics in real-time, aiming to promptly and accurately detect accidents.

In essence, the IoT Vehicle Accident Detector project contributes to the ongoing efforts to enhance road safety and emergency response efficiency. By leveraging IoT technologies and innovative approaches, we aim to make a meaningful difference in mitigating the consequences of vehicular accidents and ultimately, saving lives on our roads [7-9].

IX. FUTURE WORK:

The field of vehicle accident detection and emergency response presents numerous opportunities for further development and enhancement. One potential direction for advancement is the integration of additional treatment options for injured individuals within the system. For instance, incorporating pulse monitoring and medical data collection, as demonstrated in article [2], could enable more comprehensive medical assessments and interventions at the accident scene.

Furthermore, exploring the possibility of implementing a voice-operated system to assess the condition of the wounded could further improve the efficiency of emergency response efforts. Such a system could allow injured individuals to communicate their needs and symptoms verbally, facilitating more accurate triage and treatment prioritization by emergency responders.

Moreover, the scope of applications for in-car technologies extends beyond accident detection and response. There is considerable potential in collecting various driver behavior data using sensors that assess driving quality. This data can be invaluable for car rental and insurance companies in assessing driver risk profiles and adjusting insurance premiums accordingly. By leveraging sensor data to monitor factors such as speed, acceleration, braking patterns, and adherence to traffic laws, these companies can better evaluate driver behavior and make informed decisions regarding insurance coverage and rates.

In summary, the field of vehicle accident detection and emergency response offers numerous avenues for innovation and expansion. By incorporating additional treatment options, exploring voice-operated systems, and leveraging in-car technologies for data collection, researchers and practitioners can further enhance the safety and well-being of individuals on the road while improving the efficiency of emergency response systems.

GITHUB PROJECT REPOSITORY:

https://github.com/mikileib/IoT-Car-accident-detector

REFERENCES:

[1] World Health Organization. (2018). "Global status report on road safety 2018." [Online]. Available: https://www.who.int/publications/i/item/9789241565684

[2] Arif Shaik, Natalie Bowen, Jennifer Bole ,Gary Kunzi, Daniel Bruce, Ahmed Abdelgawad, Kumar Yelamarthi, “ Smart Car: An IoT Based Accident Detection System “,College of Science and Engineering, Central Michigan University, Mt Pleasant, MI. Press, 2018.

[3] “Guide to NEO-6M GPS Module with Arduino“ https://randomnerdtutorials.com/guide-to-neo-6m-gps-module-with-arduino/

[4] “Connecting ESP32 to Amazon AWS IoT Core using MQTT“ https://how2electronics.com/connecting-esp32-to-amazon-aws-iot-core-using-mqtt/

[5] Amazon Web Services. (2024). AWS IoT Developer Guide. https://docs.aws.amazon.com/pdfs/iot/latest/developerguide/iot-dg.pdf

[6] Amazon Web Services. (2021). Visualizing Data in Amazon Timestream using Grafana [Video]. https://www.youtube.com/watch?v=pilkz645cs4

[7] “Faster emergency assistance: Bosch introduces automatic emergency calls for motorcycles” https://www.bosch-presse.de/pressportal/de/en/faster-emergency-assistance-bosch-introduces-automatic-emergency-calls-for-motorcycles-211520.html

[8] “Pair in peril in Nanoose Bay didn’t know it, but phone sent distress signal” https://vancouversun.com/news/local-news/nanoose-bay-crash-phone-sent-distress-signal

[9] “CRASH DETECTION SYSTEM RAISES ALARM AS BMW FLIPS ONTO ROOF” https://www.islandecho.co.uk/crash-detection-system-raises-alarms-as-bmw-flips-onto-roof/