TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A MINOR PROJECT PROPOSAL

ON

EDR-BASED INTELLIGENT VEHICLE ACCIDENT PREDICTION WITH MULTISENSOR FUSION AND EDGE SUMMARIZATION

Submitted By:

Prenisha Upreti	(THA079BEI029)
Roshan Kr. Gupta	(THA079BEI034)
Suman Phuyal	(THA079BEI042)
Yugal Nyoupane	(THA079BEI048)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

Under the Supervision of

Er. Umesh Kanta Ghimire (UKG Sir)

December 2025

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

THAPATHALI CAMPUS

A MINOR PROJECT PROPOSAL

ON

EDR-BASED INTELLIGENT VEHICLE ACCIDENT PREDICTION WITH MULTISENSOR FUSION AND EDGE SUMMARIZATION

Submitted By:

Prenisha Upreti (THA079BEI029)

Roshan Kr. Gupta (THA079BEI034)

Suman Phuyal (THA079BEI042)

Yugal Nyoupane (THA079BEI048)

Submitted To:

Department of Electronics and Computer Engineering

Thapathali Campus

Kathmandu, Nepal

December 2025

ACKNOWLEDGEMENT

We would like to thank the Institute of Engineering, Tribhuvan University for providing us the opportunity to implement our knowledge and experience and showcase it into a project this semester.

We would like to express our gratitude to our department **Department of Electronics and Computer Engineering** for providing the opportunity and guiding us through every highs and lows. The department has been guiding and structuring the events to follow up for the minor project in ever so standard way giving us a clear roadmap to do our project.

We would also like to thank your supervisor; **Umesh Kanta Ghimire** for guiding, motivating and listening to the thoughts and creativity that run through our brains.

Prenisha Upreti	(THA079BEI029)
Roshan Kr. Gupta	(THA079BEI034)
Suman Phuyal	(THA079BEI042)
Yugal Nyoupane	(THA079BEI048)

ABSTRACT

"EDR(EventDataRecorder)BasedIntelligent Vehicle Accident Prediction with Multi-sensor Fusion and Edge Summarization" highlights the prevention mechanism and alert mechanism in the case of an accident. According to WHO "Approximately 1.19 million people die each year as a result of road traffic crashes." The casualties on the road are often misunderstood and blamed on drivers but the causes are often different. The real cause is never known. The miscommunication and misinformation of vehicle status, in pre-event and post-event of accident often lead into bad accident prevention. This project focuses primarily on prevention and post-event accident event data recorder and alerting via SOS in case of an accident. Multiple sensors like Flame Sensor, Gyroscope, GPS, Crash sensors are connected and embedded with a MultiSensor Fusion algorithm which gives accurate and comprehensive inferences than using some individual sensors with complex algorithms. In a microcontroller, by using the Multifusion algorithm, we integrate a waterfall model for the analytical viewpoint of accident. We use real time data. The real time data is event summarized for multiple instances and is fed into the classifier for either alerting in case of accident or warning in case of accident prevention. This project aims at solving the manual driven vehicle problem to alert the driver and also aims on being expandable for self-driven cars because accidents can happen to anyone,anything or anyway.

***Keywords*—**EDR, Multisensor Fusion, Edge Summarization

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

1. INTRODUCTION

“EDR (Event Data Recorder) Based Intelligent Vehicle Accident Prediction with Multisensor Fusion and Edge Summarization” is a project that focuses on pre-event and post-event analysis and predict the accident and log it. The MultiSensor Fusion gives more accurate data and we can use that data to feed into a real time frame. The edge processing eliminates the unnecessary flagged data that we do not need while analyzing and alerting the system. The traditional project lacked the fusion of two technologies in a single EDR, which just doesn’t make this a logger rather make it an intelligent predicting logger further explained in block diagram of methodology.

1.1. Background

Conventional vehicle EDR lacked the intelligent decision making capabilities. The project highlights an intelligent accident intelligent Vehicle Accident Prediction with Multisensor Fusion and Edge Summarization. The accuracy and timely emergency response results in a safety driven concern for the vehicles. The alert system that is classified accordingly makes the use case of this project more practical and more reliable compared to the traditional EDR. Multisensor fusion and edge summarization can improve overall reliability which is more practical in real-world deployment.

1.2. Motivation

This project came into idea when there was a deadly bus accident in Trishuli where at least 54 people went missing after two passenger buses were swept away by a mudslide into the rain-swollen Trishuli River near Simaltal. Nepal police, Armed Police Force and Nepal Army personnel worked tirelessly to find the bus in the swelling river and continuous rainfall but the effort went into vain after the search had no result. Had the rescuers or finders had better technology, the last data recorded, the search would have been more predictive. The timely SOS would have been sent early. This is just an example of a larger problem that occurs in day to day life in a world moving so fast. There also are many instances where the drivers aren’t alerted, the state at which they are driving compared to a probabilistic scenario can be dangerous, if we find those pattern and pre-alert the driver , the accident case can be reduced and possibly save life of thousands travelling. This is why we chose "EDR-Based Intelligent Vehicle Accident Prediction with Multisensor Fusion and Edge Summarization"

1.3. Problem Statement

Existing systems for alerting or logging rely on single sensors or manual reporting, which makes them prone to false alarming and missed detection. The existing system depends on cloud based processing which in case of critical situations is very unrealistic and illogical because the delay in the immediate response required case often lead to major casualties. Safety, risk management and reliability are our main concerns overall. The lack of multisensor Fusion, no edge ML, poor logging are often not considered till date, even if those projects are never brought into a single module that can solve a larger problem scenario. By trying to integrate all the aforementioned problem's solutions into a single module we have tried to solve the existing problem.

1.4. Objectives

The objectives of this project are as follows:

1. To detect vehicle accidents or predict possible accident scenarios using Multi-Sensor Fusion and a Machine Learning model (Random Forest) implemented on a microcontroller using a TinyML approach, and to alert the driver or concerned authorities in real time.
2. To log possible accidental parameters or critical spikes in sensor data during an accident or a potential accident scenario for further analysis.

1.5. Scope and Limitations

1.5.1. Scope

The intelligence, detection and logging mechanism are the key features included in our project. We have used technologies like Multisensor Fusion, Edge summarization and Machine Learning model implemented on an ESP32 microcontroller using the TinyML approach. The system performs real-time data acquisition, local processing, alert generation and logging of critical sensor parameters during accident and pre-accident scenarios. The scope of this work is limited to small-scale prototype implementation and controlled testing conditions. The system does not include full-scale vehicle integration, cloud-based analytics, or legal emergency response systems.

1.5.2. Limitations

The project has a lot of scope, but due to time constraints and limited resources, some major components cannot be implemented in this phase. The addition of a camera module and accident detection through image classification using machine learning cannot be implemented yet. Limited resources also restrict us to using simple models rather than more complex ones at this stage of the project. In the future, with adequate time, computational power, and resources, advanced machine learning models along with camera-based image classification can be integrated to enhance accident detection accuracy.

2. LITERATURE REVIEW

Amin et al. [1] proposed an accident detection system that utilizes deceleration data from low-cost Micro Electro Mechanical Systems (MEMS) based Inertial Measurement Units (IMU). Their research demonstrates that the system can accurately detect collisions and maintain location tracking during GPS outages, successfully transmitting critical information to a base station.

Furthering the integration of connectivity, Kumar et al. [2] presented an IoT-based automotive accident detection and classification (ADC) system. By fusing smartphone-built-in sensors with connected external sensors, this system not only detects accidents but also classifies the type of collision. This detailed reporting improves the efficacy of emergency medical services (EMS), fire departments, and towing services by allowing for more precise resource planning.

Regarding data management, Yao and Atkins [3] introduced the "Smart Black Box" (SBB), which enhances traditional low-bandwidth logging with value-driven, high-bandwidth data capture. The SBB utilizes a deterministic Mealy machine to cache short-term data buffers based on similarity and information value. By formulating compression as a constrained multi-objective optimization problem, the system prioritizes high-value recordings and discard redundant data to optimize finite storage.

Predivtive modeling is addressed by Yang et al. [4], who employed a Random Forest algorithm to predict the severity of traffic accidents. By incorporating four key characteristics such as location, accident form, road information, and driving speed into thier model, they achieved high performance in data classification, resulting in a more efficient and accurate predictive framework.

This project proposes an integrated vehicle safety system that combines accident prediction and detection with a robust data logging framework for forensic investigation. By utilizing a Random Forest algorithm, the system can identify potential risks and detect vehicle accident in real-time, capturing critical pre-crash and post-crash telemetry. To ensure the accuracy of these observations, the system employs a linear Kalman filter to fuse data from the accelerometer, gyroscope, and GPS, effectively minimizing sensor noise and measurement errors.

Furthermore, the system addresses data management challenges through an event summarization technique. This method prioritizes the storage of relevant incident data, successfully reducing overall storage capacity requirements by half without compromising the integrity of the information needed for future investigations. Compromising the integrity of the information needed for future investigations.

3. SYSTEM ARCHITECTURE AND METHODOLOGY

3.1. Proposed System Architecture

3.1.1. Proposed Sytem Block Diagram

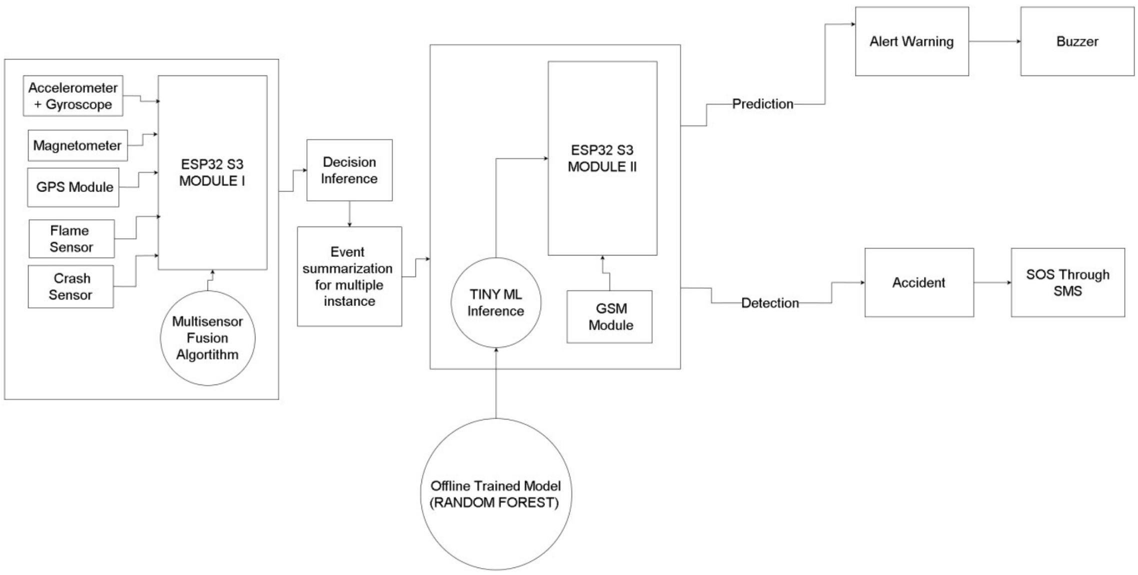

Figure 3.1.: *Block Diagram of Proposed System Architecture*

3.2. Working Principle

The proposed system uses a combination of multisensor fusion and TinyML based event classification (Random Forest) to monitor vehicles in pre-event and post-event analysis in real-time that responds to potential accidents and alerts in case of any time of accident. The system is built around two microcontrollers effectively separated for one being for multifusion and another for verifying the accident parameters via ML model for either prediction or alert.

The ESP32S3 Module I is responsible for acquisition of continuous data from multiple sensors. The multiple sensors includes accelerometer, gyroscope, magnetometer, a GPS module, a flame sensor, and a crash sensor respectively. These sensors collectively provide the information in a comprehensive manner regarding the motion, orientation, location and critical environment conditions of a vehicle at a given state. A sudden spike in a particular model is then forwarded into another module as by using decision inferences. Multisensor fusion algorithm - Kalman Filter is used, where the algorithm integrates reading from all sensors to produce a robust and reliable estimation of state, minimization of the noise and transient errors from the individual sensors accordingly. The decision inference mechanism aforementioned deviates and forwards the data only that has significant deviations from normal operating conditions in the system. The relevant features of the potential abnormal events are extracted and summarized to the raw sensor reading into meaningly indicators such as acceleration magnitude, orientation angles, sudden speed changes, or activation of crash and flame sensors, which form the input for machine learning classification.

The ESP32S2 Module II is where our model performs to classify , normal, warning or an accident event. We use a Random forest model which performs TinyMl inference in our module. In the case of a normal event, the system continues monitoring without intervention. If the event is classified as a warning, the system triggers a local alert, activating a buzzer to notify the driver and allowing time to observe the situation before taking further action. When an accident is detected, the system immediately activates a GSM module to send an SOS message to predefined emergency contacts, ensuring timely assistance. A brief observation window is included to confirm persistent abnormal conditions before sending the SOS message, reducing the likelihood of false alarms.

Overall, the system combines real-time sensor monitoring, robust data fusion, and lightweight machine learning to provide a reliable, fast, and energy-efficient solution for intelligent vehicle accident detection. By integrating both preventive warnings and emergency communication, the system enhances vehicle safety while maintaining low computational requirements suitable for microcontroller-based embedded platforms.

3.3. Kalman Filter Algorithm

The Kalman Filter is an optimal recursive estimation algorithm that simplifies the complex computational process of data fusion by introducing assumptions regarding system dynamics, state evolution, and the statistical properties of noise and estimation errors [5]. It provides an efficient solution to the linear quadratic estimation problem for systems affected by stochastic disturbances.

The Kalman Filter formulates estimates of the internal state of a linear dynamic system by processing a sequence of noisy measurements. The system is assumed to be disturbed by zero-mean Gaussian white noise, and the filter recursively updates the state estimates as new measurements become available [6]. Owing to its recursive structure, the Kalman Filter is well suited for real-time applications such as sensor fusion, navigation, and vehicle safety systems.

The algorithm follows a predictor–corrector structure and can be divided into two distinct phases:

- Time Update (Prediction)
- Measurement Update (Correction)

3.3.1. Assumptions

The Kalman Filter operates under the following assumptions [5]:

1. The system dynamics are linear and described by the state transition equation:

$$\mathbf{s}_t = \mathbf{A}\mathbf{s}_{t-1} + \mathbf{B}\mathbf{u}_t + \mathbf{w}_t \tag{3.1}$$

   where $\mathbf{s}_t$ represents the system state vector, $\mathbf{u}_t$ denotes the control input, $\mathbf{A}$ is the state transition matrix, $\mathbf{B}$ is the control input matrix, and $\mathbf{w}_t$ is zero-mean Gaussian process noise with covariance $\mathbf{Q}$.
2. The measurement model is linear and defined as:

$$\mathbf{z}_t = \mathbf{H}\mathbf{s}_t + \mathbf{v}_t \tag{3.2}$$

   where $\mathbf{z}_t$ is the measurement vector, $\mathbf{H}$ is the observation matrix, and $\mathbf{v}_t$ is zero-mean Gaussian measurement noise with covariance $\mathbf{R}$.
3. The process and measurement noises are mutually independent and uncorrelated over time.

3.3.2. Time Update (Prediction Phase)

In the time update phase, the Kalman Filter predicts the state of the system at the next time step using the system model [7].

State Prediction

$$\hat{\mathbf{s}}_{t|t-1} = \mathbf{A}\hat{\mathbf{s}}_{t-1|t-1} + \mathbf{B}\mathbf{u}_t \tag{3.3}$$

Covariance Prediction

$$\mathbf{P}_{t|t-1} = \mathbf{A}\mathbf{P}_{t-1|t-1}\mathbf{A}^T + \mathbf{Q} \tag{3.4}$$

After the prediction step, the prior estimate of the state is represented by a Gaussian distribution characterized by the mean $\hat{\mathbf{s}}_{t|t-1}$ and covariance $\mathbf{P}_{t|t-1}$.

3.3.3. Measurement Update (Correction Phase)

The measurement update phase corrects the predicted state using the current measurement, minimizing the estimation error covariance [5].

Kalman Gain

$$\mathbf{K}_t = \mathbf{P}_{t|t-1}\mathbf{H}^T \left(\mathbf{H}\mathbf{P}_{t|t-1}\mathbf{H}^T + \mathbf{R}\right)^{-1} \tag{3.5}$$

Innovation (Measurement Residual)

$$\mathbf{y}_t = \mathbf{z}_t - \mathbf{H}\hat{\mathbf{s}}_{t|t-1} \tag{3.6}$$

State Update

$$\hat{\mathbf{s}}_{t|t} = \hat{\mathbf{s}}_{t|t-1} + \mathbf{K}_t\mathbf{y}_t \tag{3.7}$$

Covariance Update

$$\mathbf{P}_{t|t} = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\,\mathbf{P}_{t|t-1} \tag{3.8}$$

The updated state estimate and covariance represent the optimal linear unbiased estimate of the system state under the given assumptions, completing one iteration of the Kalman filtering process [5].

3.4. Random Forest Machine Learning Model

A random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(x, \Theta_k), k = 1, \ldots\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x [8].

The common element in all of these procedures is that for the kth tree, a random vector Θ_k is generated, independent of the past random vectors $\Theta_1, \ldots, \Theta_{k-1}$ but with the same distribution; and a tree is grown using the training set and Θ_k, resulting in a classifier $h(x, \Theta_k)$ where x is an input vector. For instance, in bagging the random vector Θ is generated as the counts in N boxes resulting from N darts thrown at random at the boxes, where N is the number of examples in the training set. In random split selection, Θ consists of a number of independent random integers between 1 and K. The nature and dimensionality of Θ

depends on its use in tree construction. After a large number of trees is generated, they vote for the most popular class. These procedures are called random forests [9].

3.4.1. Random Forest Convergence

Given an ensemble of classifiers $h_1(x), h_2(x), \ldots, h_K(x)$, and with the training set drawn at random from the distribution of the random vector (Y, X), define the margin function as:

$$\mathrm{mg}(X, Y) = \mathrm{avg}_k I(h_k(X) = Y) - \max_{j \neq Y} \mathrm{avg}_k I(h_k(X) = j), \tag{3.9}$$

where $I(\cdot)$ is the indicator function. The margin measures the extent to which the average number of votes at (X, Y) for the correct class exceeds the average vote for any other class. The larger the margin, the more confidence in the classification. The generalization error is given by:

$$PE^* = P_{X,Y}(\mathrm{mg}(X, Y) < 0), \tag{3.10}$$

where the subscripts X, Y indicate that the probability is over the X, Y space.

In random forests, $h_k(X) = h(X, \Theta_k)$. For a large number of trees, it follows from the Strong Law of Large Numbers and the tree structure that:

Theorem 1. As the number of trees increases, for almost all sequences $\Theta_1, \ldots,$ PE^* converges to

$$P_{X,Y}\Big(P_\Theta(h(X, \Theta) = Y) - \max_{j \neq Y} P_\Theta(h(X, \Theta) = j) < 0\Big). \tag{3.11}$$

This result explains why random forests do not overfit as more trees are added, but produce a limiting value of the generalization error [8].

3.4.2. Key Features of Random Forests

- Its accuracy is as good as AdaBoost and sometimes better [8].
- It is relatively robust to outliers and noise [8].
- It is faster than bagging or boosting [8].
- Provides useful internal estimates of error, strength, correlation, and variable importance [8].
- Simple and easily parallelized [9].

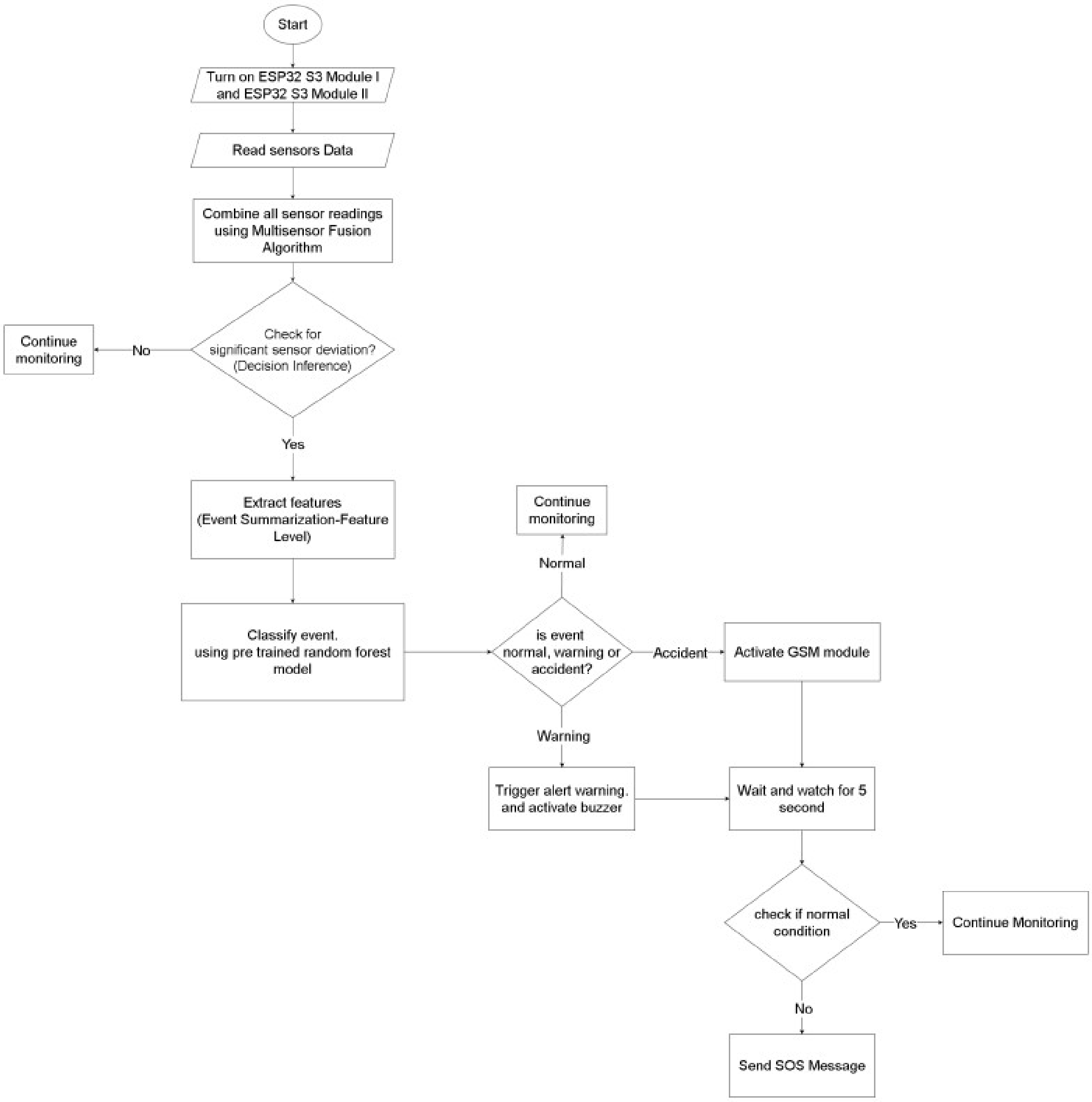

Figure 3.2.: *Flowchart of Proposed System Architecture*

4. IMPLEMENTATION DETAILS

4.1. Hardware Implementation

Accelerometer and Gyroscope Sensors

An Accelerometer (ACC) and Gyroscope (GY) sensors together form an Inertial Measurement Unit (IMU) used to measure linear acceleration and angular velocity of the vehicle. These sensors continuously monitor dynamic motions such as acceleration, braking, skidding, tilting, sharp turns, rollover, and collision forces.

Reasons to Use Accelerometer and Gyroscope

- Measures both linear and rotational motion of the vehicle.
- Enables accurate detection of sudden impacts and abnormal movements.
- Provides continuous, low-noise motion data suitable for real-time processing.
- Low power consumption and minimal computational overhead.
- Compact, cost-effective, and easily integrable with microcontrollers.
- Highly compatible with rule-based and ML-based accident detection systems.

The combined motion data from the accelerometer and gyroscope allows effective feature extraction for detecting unsafe driving behavior, loss of vehicle stability, and crash events, making the IMU a key component in accident prediction and vehicle safety applications.

Suitable Accelerometer and Gyroscope Sensor: MPU6050

The MPU6050 is a 6-axis MEMS IMU that integrates a 3-axis accelerometer and a 3-axis gyroscope along with an internal Digital Motion Processor (DMP). It outputs motion data digitally via the I^2C interface. By combining acceleration and rotational data, it effectively detects sudden motion changes and crash occurrences.

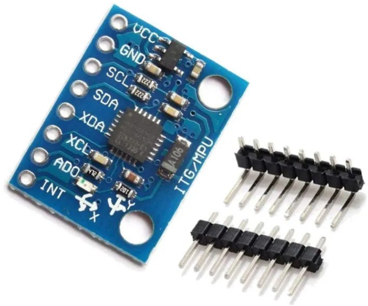

Figure 4.1.: *MPU6050 Accelerometer and Gyroscope Module*

Magnetometer Sensor

A Magnetometer Sensor (MS) is a sensor that measures the strength and direction of a magnetic field, typically the Earth's magnetic field, to determine vehicle orientation and heading.

Reasons to Use Magnetometer

- Provides absolute heading reference.
- Improves long-term stability of motion estimation.
- Low power consumption.
- High-resolution digital output.
- Cost-effective and ML-friendly.

HMC5883L MEMS-Based Magnetometer

The HMC5883L is a 3-axis MEMS magnetometer that uses magnetoresistive sensing elements to measure magnetic field components along the X, Y, and Z axes and provides digital output through the I^2C interface.

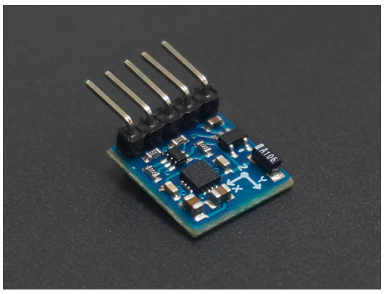

Figure 4.2.: *HMC5883L Magnetometer Module*

GPS Sensor

A Global Positioning System (GPS) sensor determines the absolute geographical position of a vehicle by receiving signals from multiple satellites. It provides latitude, longitude, speed, altitude, and timestamp information essential for vehicle tracking and accident localization.

Reasons to Use GPS Module

- Provides accurate real-time latitude and longitude.
- Measures vehicle speed and heading.
- Enables precise accident location reporting.
- Low power consumption for continuous tracking.
- Stable and reliable data output for ML processing.

Figure 4.3.: *GPS NEO-6M Module*

Flame Sensor

A Flame Sensor (FS) detects the presence of fire by sensing infrared radiation emitted during combustion. It is critical for identifying fire hazards following vehicle collisions.

Reasons to Use Flame Sensor

- Enables early fire detection.
- Fast response time.
- Simple microcontroller interface.
- Low power consumption and cost.
- Suitable for rule-based and ML-based decision systems.

Figure 4.4.: *Flame Sensor Module*

Crash Sensor

A Crash Sensor (CS), also known as an impact sensor, detects sudden mechanical shocks or collision forces acting on the vehicle.

Reasons to Use Crash Sensor

- Immediate collision detection.
- High reliability in impact sensing.
- Very low processing requirement.
- Fast response for emergency systems.
- Cost-effective and durable.
- Strong support for decision inference.

Figure 4.5.: *Crash Sensor Module*

GSM Module

A Global System for Mobile Communications (GSM) module enables embedded systems to send and receive data over cellular networks using a SIM card. It allows communication with predefined contacts such as emergency services and family members.

GSM Module (SIM800L)

The GSM module communicates with the controller via UART using AT commands. Once a SIM card is inserted, the module registers with the nearest cellular base station. During emergency events such as crashes or fire detection, the module automatically transmits SMS alerts to predefined mobile numbers.

Reasons to Use GSM Module

- Enables long-distance communication without internet dependency.
- Real-time emergency alert transmission via SMS.
- Reliable operation in rural and remote areas.
- Simple microcontroller integration.
- Low power consumption.
- Cost-effective and widely supported.

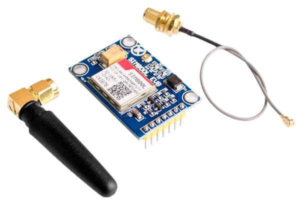

Figure 4.6.: *GSM SIM800L Module*

ESP32-S3 Microcontroller

The Espressif System on Chip (ESP32-S3) is a high-performance, low-power microcontroller developed by Espressif Systems for advanced embedded and IoT applications. It integrates wireless connectivity and AI acceleration features, making it suitable for real-time sensing and edge intelligence.

ESP32-S3 Module

The ESP32-S3 is based on a dual-core Xtensa® 32-bit LX7 processor operating up to 240 MHz. It includes built-in Wi-Fi and Bluetooth Low Energy (BLE 5.0), along with rich peripheral support such as SPI, I^2C, UART, ADC, PWM, and GPIOs. Support for external flash and PSRAM enables efficient handling of sensor data and ML models.

Reasons to Use ESP32-S3

- High processing capability.
- Optimized for machine learning inference.
- Supports multiple sensors simultaneously.
- Low power operation.

How This Is Compatible with Machine Learning

The ESP32-S3 is specifically designed for edge machine learning applications. Its LX7 processor supports vector instructions that accelerate mathematical operations used in ML inference. Frameworks such as TensorFlow Lite for Microcontrollers and ESP-DSP enable deployment of trained models directly on the device. In this project, the ESP32-S3 performs real-time sensor fusion, feature extraction, and decision inference locally, reducing latency and eliminating cloud dependency. This makes it ideal for safety-critical applications such as crash and fire detection.

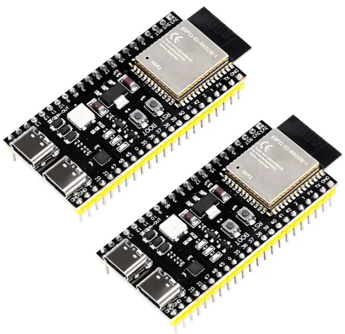

Figure 4.7.: *ESP32-S3 Module*

5. EXPECTED OUTCOMES

1. **Accident Prediction & Crash Detection**
   The system is expected to analyze multisensory fused data including acceleration, angular velocity, heading direction, fire detection, and crash impact signals to compute a real-time accident risk-score. Based on this risk score, the system will predict the potential accident scenarios and detect crash occurrences under impact conditions.
2. **EDR - Event Data Recorder**
   The Event Data Recorder (EDR) module is expected to continuously record and store structured sensor and event data in real time. When an accident occurs, the EDR preserves critical event information including sensor readings and system decisions. The purpose of this module is to provide reliable event data for post-accident analysis and investigation.
3. **Alert and SOS Signals**
   The system is expected to trigger real-time alerts or warning signals such as buzzer activation, to notify the driver upon detecting high risk-score scenarios for accidents. In the event of a confirmed accident, the system generates SOS signals and transmits them to nearby helplines via the GSM module.

A. PROJECT BUDGET

The total budget required for the successful completion of the project is estimated to be between **NPR 8000 - NPR 11000**. This budget covers various aspects of the project, including hardware purchases, component materials, software services, and miscellaneous expenses.

A.1. Bill of Materials (BOM)

This section details all components that will be integrated into the final product/system. These are the materials that constitute the deliverable hardware (if any).

Table A.1.: *Project Budget*

SN	Components	Qty	Unit Cost (NPR)	Total (NPR)
1	ESP32 S3	2	1500	3000
2	GSM800L	1	1000	1000
3	GPS	1	800	800
4	MPU6050	1	700	700
5	HMC5883L	1	450	450
6	Flame Sensor	1	320	320
7	Crash Sensor	1	100	100
8	SIM Card	1	100	100
9	Bread Board	1	450	450
10	Resistor	1	150	150
11	Buzzer	1	100	100
12	Wire	1	200	200
13	Power Supply	1	400	400
			Total	**7770**

B. PROJECT TIMELINE

The project will be executed over a period of approximately 6 months, divided into several key phases. Each phase includes specific tasks and milestones to ensure steady progress toward project completion.

B.1. Gantt Chart

This is the estimated timeline for the project, detailing the key phases and milestones. The project is structured into several stages, each with specific tasks and deliverables.

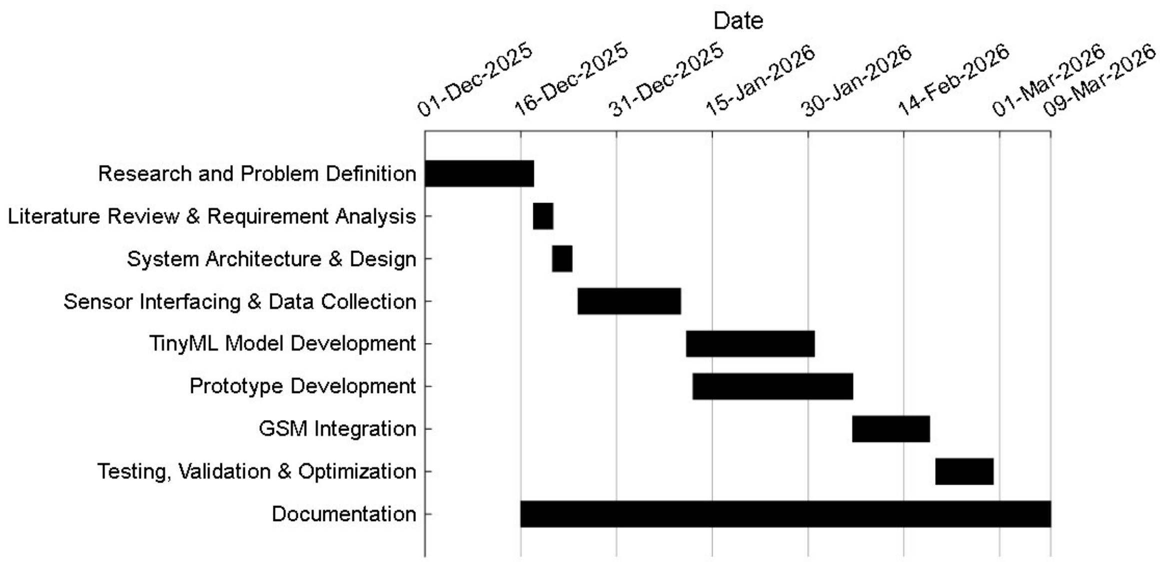

Figure B.1.: *Project Gantt Chart*

C. FEASIBILITY

This project is proposed to be implemented using hardware components such as the ESP32-S3 microcontroller, accelerometer, gyroscope, magnetometer, flame sensor, crash sensor, ultrasonic sensor, GPS module, and GSM module. These components are readily available in the market and are cost-effective, making the system economically feasible for real-world deployment. The software tools used in this project are free and open-source. Programming of the ESP32-S3 is carried out using the Arduino IDE and ESP-IDF. For data analysis and machine learning model development, Python-based libraries such as NumPy and Pandas are used. These tools support sensor data processing, feature extraction, and model training during the learning phase of the project. The dataset required for training and validating the prototype is planned to be obtained from open-source platforms such as Kaggle, along with real-time data collected from the sensors during testing. This combination helps in improving model reliability and understanding real-world sensor behavior. The project requires a SIM card and a GSM-enabled mobile network to function effectively. The system is installed in a vehicle with a SIM card inserted into the GSM module, enabling communication over the mobile network. This allows the ESP32-S3 to send alerts, SOS messages, and location information to predefined contacts or emergency services during accident or fire detection events.

REFERENCES

[1] M. S. Amin *et al.*, "Low cost gps/imu integrated accident detection and location system," *Indian Journal of Science and Technology*, vol. 9, no. 10, pp. 1–9, 2016.

[2] N. Kumar, D. Acharya, and D. Lohani, "An iot-based vehicle accident detection and classification system using sensor fusion," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 869–880, 2021.

[3] Y. Yao and E. Atkins, "The smart black box: A value-driven high-bandwidth automotive event data recorder," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1484–1496, 2021.

[4] J. Yang, S. Han, and Y. Chen, "Prediction of traffic accident severity based on random forest," *Journal of Advanced Transportation*, vol. 2023, p. Article ID 7641472, 2023.

[5] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[6] C. Montella, "The kalman filter and related algorithms: A literature review," University Research Report, Tech. Rep., 2011.

[7] H. Durrant-Whyte and T. C. Henderson, *Multisensor Data Fusion*. Springer, 2006.

[8] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[9] T.Rajendran. (2025) Vehicle accident detection using random forest. Accessed: 2025-12-25. [Online]. Available: https://www.scribd.com/document/793483984/20