# An IoT Enabled Vehicular Decision Fusion Framework for Accident Detection and Classification

NIKHIL KUMAR [1], DEBOPAM ACHARYA [2], and DIVYA LOHANI [1]
[1] Department of Computer Science and Engineering, School of Engineering, Shiv Nadar University, Gautam Buddha Nagar, Uttar Pradesh, India - 201314
[2] School of Computing, DIT University, Dehradun, Uttarakhand, India - 248009

Increased number of vehicle-based road accidents is a key reason for the death and disability of people. Timely information on accidents can save lives. Current accident detection systems are either working towards increasing the accuracy of detection or the severity of the accident. Accurate information of an accident type can help the emergency medical services (EMS) to identify the most appropriate type of rescue and medical assistance to the victims. This work introduces a smartphone-based accident detection and classification (ADC) system that not only detects the accident but also classifies the type of accident as collision, rollover, or fall-off, using internal and external sensors. We have developed an end-to-end IoT system that exploits a multi-sensor data fusion framework to accurately classify the type of accident. The framework combines the decisions of three different classifiers based on Nave Bayes (NB), K-Nearest Neighbor (KNN), and Random Forest (RF) methods using stacking ensemble approach. Logistic Regression based stacking approach is found to be highly accurate in comparison to NB, KNN, and RF classifiers when they were used individually.

Keywords: Vehicle accident detection and classification, accident reporting, Internet of Things, sensor fusion, K- Nearest Neighbor, Nave Bayes, Random Forest, decision fusion, Stacking.

## 1. INTRODUCTION

Growth in population and per capita income has contributed to increased ownership of vehicles across the globe. The rise in vehicle numbers has resulted in a significant increase in traffic which in turn has contributed to an increase in fatalities and injuries due to road accidents. Over-speeding, rash driving, intoxication, driver's exhaustion, poor weather, poor road conditions, vehicle breakdown, and the presence of stray animals on the roads are among the causes of road accident deaths. More than 1.35 million people die and 50 million people suffer injuries every year because of accidents on the roads (WHO, 2018). Road fatalities are the eighth biggest cause of death, out of which about 74 percent were males and 26 percent females. More than 90 percent of all road deaths occur in low- and middle-income nations, despite possessing nearly 60 percent of the world's vehicles (WHO, 2019).

Prompt accident notification to appropriate emergency services such as the nearest police station, fire station, emergency medical services (EMS), etc. is necessary for providing aid to injured persons in a timely manner. In road accidents most human lives are lost due to delays in emergency medical help. As per the Golden Hour Principle (Lerner and Moscati, 2001), there is a strong chance that prompt medical help can prevent death during the golden hour. Golden Hour is defined as the time duration between traumatic injury and medical assistance. If the time from accident to first aid is reduced, then the probability of death becomes one third (Sánchez-Mangas et al., 2010). Although this essential automatic emergency notification mechanism is a standard accessory in certain newer and costlier cars, most cars of today are not equipped with these high-end automated warning systems.

Information and communication technologies have been used in the recent past to notify the EMS after an accident (Fogue et al., 2012). Most of the research work is limited to enhancing accident identification accuracy, assessing the severity of accidents, or reducing the post-accident rescue period (Ponte et al., 2016)(Chung and Recker, 2012). Failure to not classify the type of accident as a rollover, collision, or fall-off is another drawback of most of the existing accident detection systems. Merely recording the occurrence of an accident incident greatly restricts the capacity of EMS to provide the best kind of rescue and medical assistance to the victims. Therefore, an inexpensive and highly accurate accident detection and classification (ADC) system is required that can be retrofitted to any vehicle, and which automatically send the essential details of accidents to the EMS so that in time rescue action can be initiated.

*Contribution of this work:* Research work for developing methods for predicting, avoiding, and detecting road accidents is mainly focused on either improving accuracy or reducing rescue time after road accidents occur. In this work, we have proposed an end-to-end IoT architecture that can detect the accident, and send the emergency notification to the intended recipients to start the rescue operation on time. The research also focuses on finding the highly accurate ADC model. Our IoT architecture exploits the multi-sensor data fusion framework to accurately detect and classify the type of accident. This framework fuses the outputs (decisions) of Nave Bayes (NB), KNN (K-Nearest Neighbor), and RF (Random Forest) classifiers by using stacking ensemble, and concludes that logistic regression-based stacking ensemble outperforms these classifiers if they were used individually. This accident classification can sometimes be incredibly useful when planning the rescue operations. Our proposed solution is economical and effectual that can be retrofitted in any vehicle.

The rest of the paper is organized as follows. Major literature survey in the area of vehicle collision, rollover, and fall-off detection/notification, and multi-sensor fusion is described in Section 2. The architecture of our end-to-end IoT system and model variables explained in Section 3. The hardware setup and software setup is described in Section 4. Section 5 provides the description of possible accident scenarios, whereas experimental setup and data collection are explained in Section 6. Multi-sensor data fusion framework for the ADC system that incorporates four different ADC models, based on NB, KNN, RF classifiers, and Stacking ensemble technique, has been proposed in Section 7. Section 8 presents the performance evaluation of the proposed system, which is followed by concluding remarks in Section 9.

## 2. RELATED WORK

This section discusses research work related to the detection, localization, modeling, analysis, and reporting of road accidents. Sadeky et al. (2010) used real-time video clips from traffic control systems to create the histogram of flow gradients (HFG). The features of the HFG are used as inputs for designing the logistic regression-based model to predict the road accidents. In order to minimize the number of false alarms, Smolka and Skublewska-Paszkowska (2016) introduced a smartphone-based collision detection system that uses the magnetometer, accelerometer, and GPS modules. Several crash metrics were proposed by McIver et al. (1996). The measurements included the acceleration values, the squared acceleration, the amount of acceleration over a period of time, and the velocity curve shape. Kumar et al. (2019) have used two standard NHTSA datasets to train and test the Logistic Regression based learning model to predict the vehicle collision when the impactor hit the vehicle, and when the vehicle collides with a pole. Sada and Moriyama (1998) have suggested that a crash can be detected by adjusting the velocity threshold. The threshold adjustment was based on physical quantities included acceleration, jerk, displacement, and velocity.

A combination of different types of sensors including force sensor, accelerometer, and pressure sensor has been used by some systems (Dunwoody and Stern, 1998) for rollover sensing. Steiner et al. (1997) introduced a model for rollover detection using acceleration and yaw rate sensors.

Kumar et al. (2020b) have used the SVM (Support Vector Machine) based model to detect the fall-off of a vehicle from an altitude by using different parameters including the change in altitude, speed of the vehicle, and absolute linear acceleration with detection accuracy of 98.2 percent. Kumar et al. (2020) have utilized accelerometer, GPS and barometer sensors to classify the vehicle fall-off severity as mild, moderate, and severe. KNN (K-Nearest Neighbor) based supervised machine learning model has been used to classify the severity of the fall-off with an average F1-score of 0.95.

Some systems in the literature can notify the emergency services about a particular incident. Acharya et al. (2007) and Ibrahim et al. (2016) have developed systems that use data from the gyroscope and accelerometer to observe and report the roll-over events of the vehicles to the intended recipients. Aloul et al. (2015) devised an Android app to develops an accident detection system based on Hidden Markov Models (HMM) and Dynamic Time Warping (DTW) using data from the accelerometer. An SMS based accident notification, which contains the severity and location of the incident, is sent to the emergency services, family, and police. Bhatti et al. (2019) used a smartphone's built-in speed, position, pressure, sound, and gravity sensors to create a low-cost and compact solution, which can detect an accident incident and informs it to the local hospital. Fernandes et al. (2016) created an Android app that collects feedback from mobile sensors to identify crashes and rollovers, along with road risk alerts provided by nearby vehicles. Dar et al. (2019) have suggested an approach focused on fog computing to develop a low-cost smartphone sensors based system to reduce the delay in reporting an accident to the nearest hospital.

None of these mechanisms show the ability to classify the type of accident that is useful in determining the nature and amount of emergency services needed, and in particular the number of injured persons. Sensor fusion is used in different fields of engineering to enhance the accuracy of the systems by combining the data of different sensors. A Fuzzy expert system is used by the Damousis and Tzovaras (2008) to combine the most relevant features derived from the vehicle driver's eyelid activities and the system can predict the sleep effectively. Felisberto et al. (2014) has proposed a system that can use the fusion of different networked sensors to detect the accident of elderly people in the home environment. Moulik and Majumdar (2018) developed an IoT based system that uses the fusion of multiple ultrasonic sensors data to detect human accidental falls. The fuzzy inference system to implement that multi-sensor fusion is found to be sixteen percent more accurate than previous systems.

Sensor fusion has been used in different works for accident detection. However, for accident classification, to the best of our knowledge, no major work has been reported in the literature, especially for enhancing the accuracy. Chan (2002) has surveyed the methods used by many current crash detection systems. It also indicates multi-sensor data fusion (MSDF) can enhance the crash sensing capabilities, but the ability of MSDF had yet to be recognized. Zhang et al. (2018) tried to fuse data from multiple sources by integrating traffic measurements with social media tweets to track road incidents in real-time. According to the authors, when using the SVM as a classification model with 5-fold cross-validation, the integration of multiple data sources improves prediction accuracy. The CARSENSE program, which brings together twelve major industrial and research collaborators, proposed to create a sensor network to provide appropriate information about the vehicle environment to help drive at low speed in complicated urban conditions (Langheim, 2002). The article explains the program's key results in terms of improving individual sensors and fusing the information from these sensors into a fusion device.

Several intelligent systems have been developed by different car manufacturing companies such as OnStar (Barabba et al., 2002) by General Motors, Ford SYNCs 911 Assist (Ghangurde, 2010) by Ford Motor Company, Safety Connect (Toyota Motor Corp, 2020) by Toyota Motor Corporation, BMW Assist (Englisch, 2002) by BMW, Suzuki Connect (Maruti-Suzuki, 2018) by Maruti-Suzuki India Ltd., etc. These systems are equipped with several state-of-the-art features but these systems are factory fitted, costly, and less focused on accident detection, classification, and notification.

This work introduced a new sensor fusion-based solution for detecting and classifying the accident events, which is not undertaken in any other current work. We have introduced a novel sensor fusion framework that implements sensor fusion (or data fusion) at different levels of processing to improve the accuracy of accident classification.

## 3. ARCHITECTURE OF THE SYSTEM

### 3.1 Architecture of our IoT System

We developed a smartphone-based IoT enabled architecture, as shown in Figure 1, to report automobile accident detection and classification, and to report the incident type and location to the relevant agencies for prompt and focused rescue operation. Sensors of Android smartphone and Sensordrone (Lohani and Acharya, 2017) are used to read the values of four different vehicle motion-related physical and environmental parameters viz. speed of the vehicle, absolute linear acceleration, change in altitude, and a maximum of pitch and roll, to address this research problem. Smartphone receives the Sensordrones data via Bluetooth link. If an accident occurs, the smartphone sends the relevant information such as accident type, location, vehicle ID, etc. to the IoT server using a 4G/LTE internet connection. IoT server sends the emergency notifications to the intended stakeholders such as family, and nearest police station, EMS, towing service, etc. using a multicast publish/subscribe system. Our architecture uses the fusion of sensors data at different levels of processing to increase the accuracy and reliability of the proposed ADC system. The detailed sensor fusion based workflow diagram of the system is described in section 7.

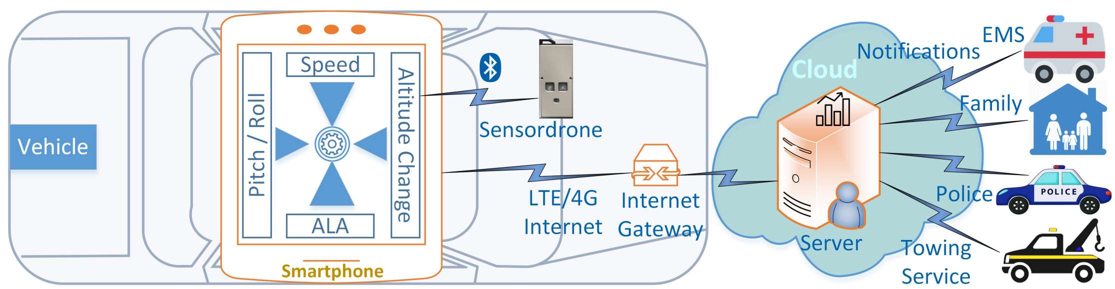


Figure 1. Architecture of proposed IoT system.

### 3.2 Model Variables

3.2.1 *Speed of the Vehicle.* It is very probable that if a vehicle encounters a significant accident, whether it is a rollover, a collision, or a fall-off, its velocity will eventually become almost zero. GPS of the smartphone is used to calculate the vehicles speed. GPS receives NMEA (National Marine Electronics Association) strings (NMEA, 2020) from satellites which contains information about the time, location, and speed of the device, etc. Following NMEA sentence showing the velocity 018.8 knots which is equal to 35 km/h.

*$GPRMC, 172232, A, 1831.333, N, 07734.321, E, 018.8, 054.4, 150320, 003.1, W*6A*

3.2.2 *Maximum of Roll and Pitch.* An object's rotation angles around the X-axis (longitudinal), Y-axis (lateral), and Z-axis (vertical) are recognized as roll, pitch, and yaw (azimuth), respectively (Figure 2(left)). Roll and pitch were measured by fusing the rotation information given by the three inertial sensors viz. accelerometer, gyroscope, and magnetometer of the smartphone using the complementary filter. Original rotational data is expressed in the matrix called rotational matrix. Euclidean space can be rotated $\theta$ anti-clockwise about the origin with the help of a rotation matrix. We find that a rollover occurred if either of the roll or the pitch is greater than 90. Taking the maximum of roll and pitch has reduced the dimensionality of the input vector without losing the sense of the variables, which decreases the training and testing time of ML algorithms.

No individual inertial sensor is sufficient to efficiently determine the orientation, due to their limitations and measurement drifts. The magnetometer sensor can accurately determine the azimuth but not pitch and roll, while the accelerometer sensor can accurately calculate the pitch and roll, but not azimuth. At higher frequency, short-term measurements of the magnetometer and accelerometer arrangement are not reliable due to some minor small external forces. For correction a low-pass filter is necessary because the combination of these two inertial sensors performs well at lower frequency. Gyroscope can measure the roll, pitch, and azimuth individually but it also suffers to strong drifts in measurements at low frequency in the long run. Gyroscope performs well at high frequency, so a high-pass filter is necessary to get more stable results.

Every sensor has its advantages and disadvantages, but the complementary fusion of these inertial sensors can reduce the measurement drifts substantially (Kumar et al., 2020a). A complementary filter (Kubelka and Reinstein, 2012), comprising of both low-pass filter and high-pass filter (Figure 2(right)), generates more precise results by adding the fractions of weight to the outputs of low pass and high pass filters. If $P_{gyro}$ is the pitch produced by the high-pass filter, and $P_{acc\_mag}$ is the pitch produced by the low-pass filter, then the resulting pitch by the complementary filter can be calculated as

$$Pitch = \alpha * P_{gyro} + (1 - \alpha) * P_{acc\_mag} \tag{1}$$

where $\alpha$ is the weight factor whose considered value is 0.98. Similarly, the roll and azimuth can be calculated.

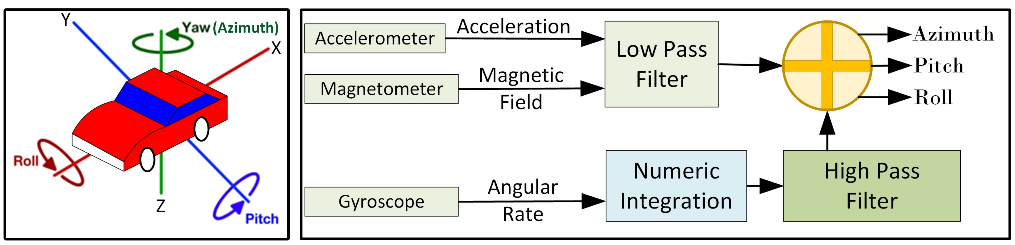


Figure 2(left). yaw, pitch and roll; Figure 2(right). complementary filter.

3.2.3 *Absolute Linear Acceleration (ALA).* When a moving vehicle drops from a height, the vehicle's orientation after the fall may not be the same as it was on the road. Due to this, deceleration may distribute in three gravitational axis X, Y, and Z, and the correct force of impact cannot be measured. Irrespective of the vehicles orientation, a resultant vector of distributed components of deceleration, called ALA (also known as Signal Magnitude Vector (Lee et al., 2018)), is calculated to handle this problem. ALA exhibits the characteristic of deceleration parallel to the direction of the collision. ALA is a positive value measured by the smartphones accelerometer in terms of $g$, where g $\approx$ 9.80665 m/s$^2$. If $DEC_X, DEC_Y$, and $DEC_Y$ are the decelerations along with the X-, Y-, and Z-axis then ALA can be calculated as

$$ALA = \sqrt{(DEC_X)^2 + (DEC_Y)^2 + (DEC_Z)^2} \tag{2}$$

In the literature, it is stated that if a vehicle hits a solid obstacle at a speed of more than 23 km/h, the deceleration strength will always cross 5$g$ (Kendall and Solomon, 2014). An accelerometer's normal signal generation rate may be more than 2000 Hz. Processing the data and recording the highest value of deceleration fluctuation, generated for 1ms to 2ms during the collision, at such a high frequency is very difficult, particularly when we need exact decisions with the help of limited resources. Although, this problem is addressed by Iyoda et al. (2016) by using a 10-ms moving average, but averaging technique can undermine the maximum value. To record the maximum deceleration at every axis, we have used a 10-ms moving maximum technique, which records the peak values of every 10-millisecond window. If $DEC\_X_{t1}$, $DEC\_X_{t2}$, $DEC\_X_{t3}$, , $DEC\_X_{tn}$ are $n$ readings of deceleration at X-axis during 10 milliseconds, the resultant highest deceleration

$DEC_X$ at X-axis will be

$$DEC_X = max(DEC\_X_{t1}, DEC\_X_{t2}, DEC\_X_{t3}, ..., DEC\_X_{tn}) \quad (3)$$

Resultant maximum decelerations at Y-axis and Z-axis can be observed similarly.

3.2.4 *Change in Altitude.* Change in the altitude is the most responsible factor that dramatically changes during the fall, which helps the model to predict the vehicles fall-off more accurately. There is no specific instrument that can calculate altitude exclusively (because GPS cannot measure the altitude shift at the same coordinates), but atmospheric pressure $P$ can be used to calculate it using the following formula.

$$Altitude = 44330.77 * (1 - (\frac{P}{P_0})^{0.190263}) \quad (4)$$

where $P_0$ ($\approx$ 101.325 kilo-pascals) is atmospheric pressure at sea level. We have used the barometric altimeter of the Sensordrone to monitor the atmospheric pressure $P$.

## 4. HARDWARE AND SOFTWARE SETUP OF THE SYSTEM

### 4.1 Hardware Setup

A full hardware setup is shown in Figure 3, which consists of a *SAMSUNG Galaxy S8* Android smartphone, a *Sensordrone*, and a 1:12 scaled radio-controlled (RC) toy car.

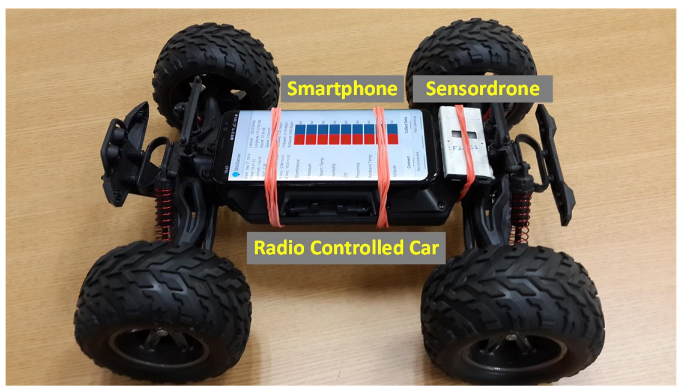


Figure 3. Hardware Setup.

In addition to external sensors, a 9-axis inertial sensor (range: 179 for pitch and roll; and 16$g$ for deceleration) and GPS sensor of the *Samsung Galaxy S8* Android smartphone has been used. Inertial sensors are used to measure the roll, pitch, and deceleration and a GPS sensor is used to calculate the vehicles speed and accidents location. Although for our proposed system any Android smartphone can be used because now a day almost every Android smartphone has inertial and GPS sensors, but *S8*s 4 GB RAM and 1.9 GHz processor makes it a little favorable computing device for the experiment purpose.

For measuring the altitude of the vehicle, *Sensordrone*s barometer is used. *Sensordrone* (Lohani and Acharya, 2017) is a portable programmable device that contains seven different inherent sensors to measure different parameters of the environment, such as light, temperature, pressure, humidity, CO, proximity, etc. It can be connected with any computing device using Bluetooth.

It is not viable to iteratively get experimental data samples using a real vehicle as it is a very expensive and life-threatening process. So, we have used GPTOYSs Foxx S911 RC Car (GP TOYS, 2020), a 1:12 scaled radio-controlled (RC) toy car, to emulate the real-life accident scenario. This toy car is made for off-road gaming and it has a maximum speed of 53 km/h, operating range of more than 80 meters, 2.4 GHz 4-channel transmitter frequency, 45°turning angles, dimension (L= 31.0 cm, W=26.5 cm, H=15.0 cm), and 1.078 kg weight.

## 4.2 Software Setup

In this work, two Android applications for sending and receiving emergency alerts have been developed. To capture and process the data coming from different sensors, *SNUSense* app is installed in the smartphone of accident victim (or car driver).

As shown in Figure 4(left), *SNUSense* is developed to capture and process different parameters from smartphone sensors (such as rotation angles, vehicle speed, ALA, location, date/time), and from *Sensordrone* sensors (such as pressure, temperature, altitude, humidity, CO, illuminance, etc.). Although, *SNUSense* is a multi-sensing application which aggregates several environmental parameters, in this work it is tuned to process only four parameters viz., speed, ALA, altitude change, and the maximum of roll and pitch to detect and classify the accident type and notify it to the *Google Firebase IoT Server* (Google Developers, 2020) using 4G/LTE internet link. It can record all parameters in a CSV file in the smartphones memory.

*SNUAlertApp* is developed to receive emergency alerts about accidents as shown in Figure 4(right). It need to be installed in smartphones of all rescue stakeholders. An alert notification with alarming beep noise appears in the rescuers smartphone with some important information about the accident and the victim such as drivers ID, accident location, severity, etc. A single click on the notification shows the accident location on *Google Map* with a tracker icon on it, which help the rescuers to locate the accident site easily.

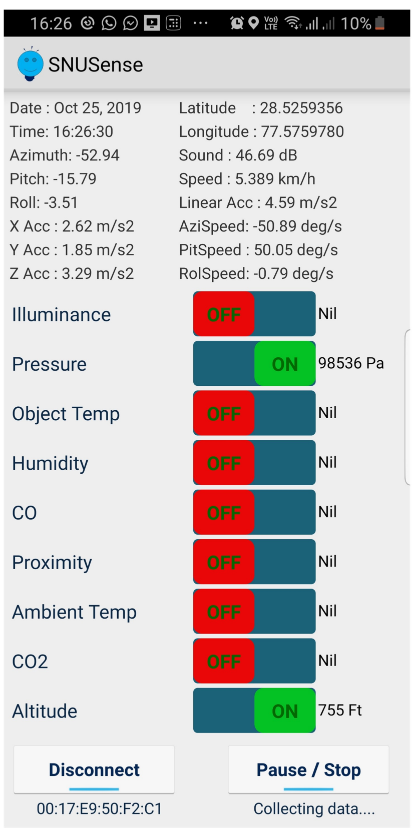


Figure 4(left). SNUSense application;

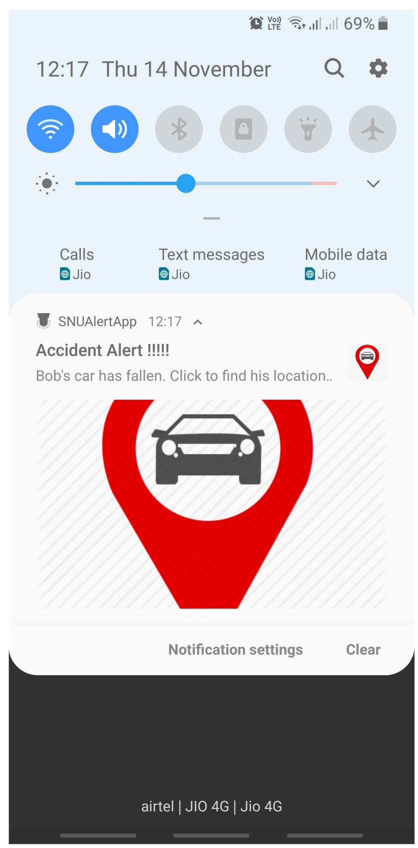


Figure 4(right). SNUAlertApp application.

## 5. ACCIDENT SCENARIOS

This work focuses on the detection and classification of four separate vehicle accident scenarios, (a) collision, (b) fall-off, (c) rollover and (d) no-accident, based on the four variables of (a) speed, (b) ALA, (c) altitude change, and (d) maximum of pitch and roll. Table I shows the thresholds of four model variables for each accident scenario. It is very intuitive that after every type of accident speed becomes zero eventually. The speed threshold of 2 km/h is considered because of the GPS sensors delay in measurement updates. The ALA threshold is considered to 5g for all types of collisions to avoid false positives. Also, in the literature, it is mentioned that if a vehicle impacts with a non-moving obstacle (whether on the road or with the earth after falling from an altitude) with a minimum speed of 23 km/h (i.e. 21 feet/s), then deceleration always marks $5g$ (Kendall and Solomon, 2014). It can also be observed from the following standard Equation (5) of velocity that when an object (or vehicle) falls from more than 8 feet, it attains a final velocity of 22 feet/s in less than 0.8 seconds, if the initial velocity is considered as zero.

$$final\_velocity = initial\_velocity + g * t \tag{5}$$

Here $g$ is the gravitational acceleration and $t$ is the time. Some sensors such as GPS and barometer show some lag in reading timings of parameters that we have considered while implementing the models.

Table I: ACCIDENT DETECTION AND CLASSIFICATION CRITERIA

| **Accident Type** | **Speed (2 km/h)** | **ALA (5g)** | **Altitude Change (8 feet)** | **Max of Pitch and Roll (90°)** |
|---|---|---|---|---|
| Collision | <2km/h | >5g | <8feet | Any Value |
| Rollover | <2km/h | Any Value | Any Value | >=90° |
| Fall-off | <2km/h | >5g | >8feet | Any Value |
| No Accident | Conditions other than above mentioned. | | | |

## 6. EXPERIMENTAL SETUP AND DATA COLLECTION

### 6.1 Experimental Setup

To carry out collision, rollover, and fall-off experiments and to mimic the real-life accident scenarios, Sensordrone and the Android smartphone is attached close to the CG (center of gravity) of the RC car to obtain the unbiased readings. We used a plastic road barricade as an obstacle to crash the RC car and collected the collision data (Figure 5(right)). To conduct the fall-off experiments, RC car is dropped multiple times on a wooden Tennis court from a 15 feet elevated running track in Indoor Sports Complex of *Shiv Nadar University* (Figure 5(left)). We used a 4 cm thick board of plywood to perform the rollover tests. Board was tilted 30° with the flat land surface (Figure 5(center)). The speed of the car during all the experiments was 35 km/h.

### 6.2 Data Collection

Sudden drop of speed, and the rise of ALA is evidence of the RC car's collision event in Figure 6(left). Graph in Figure 6(center) is representing an RC car's fall-off scenario, in which a gradual drop in speed, and rise in altitude-change indicates the duration of fall, and a sudden rise in ALA indicates the collision on the tennis court after the fall. One second moving window is used to monitor the altitude change since the dropping object will reach the final velocity of more than 30 km/h in one second, which is high enough to produce a deceleration of more than $5g$. In Figure 6(right), the sudden rise in the maximum value of pitch and roll is accurately predicting the rollover incident. The red dotted line in each graph indicates the incident when an accident occurred.

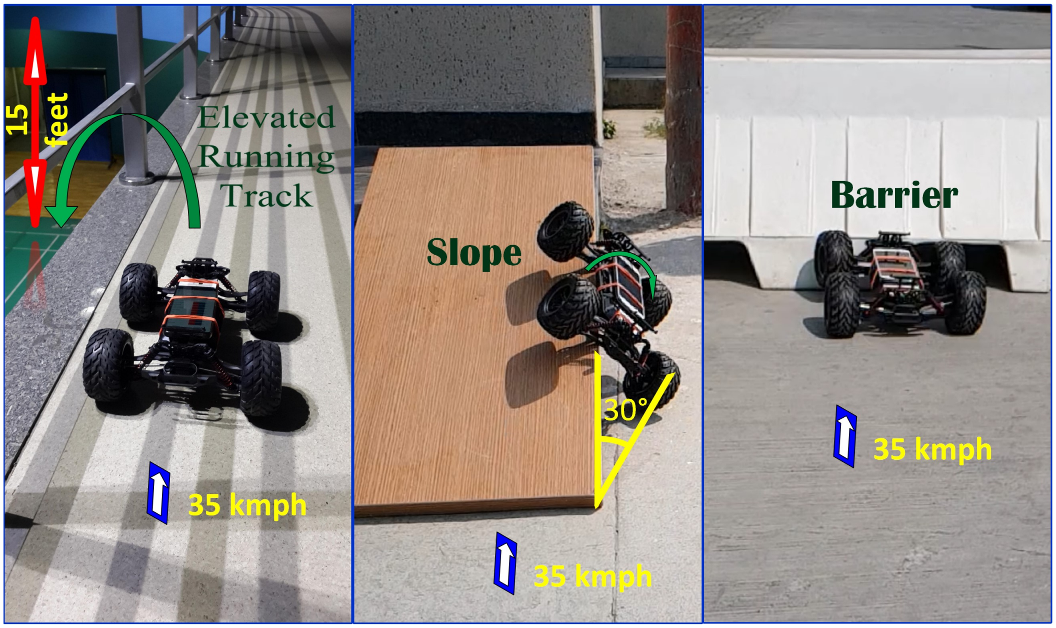


Figure 5(left). Fall-off setup application; Figure 5(center). Rollover setup; Figure 5(right). Collision setup.

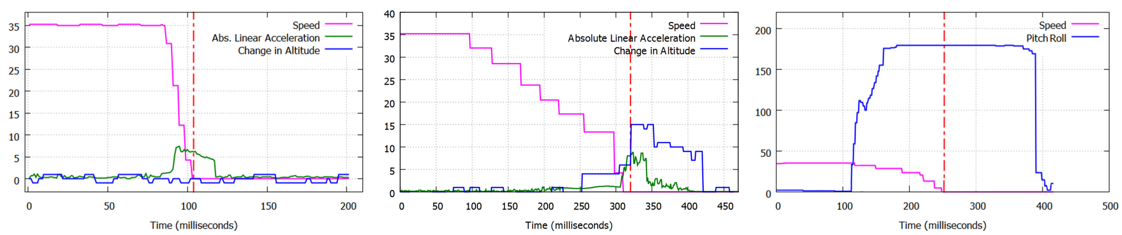


Figure 6(left). Collision; Figure 6(center). Fall-off; Figure 6(right). Rollover.

## 7. MULTI-SENSOR DATA FUSION FRAMEWORK FOR ACCIDENT DETECTION AND CLASSIFICATION (WORKFLOW DIAGRAM)

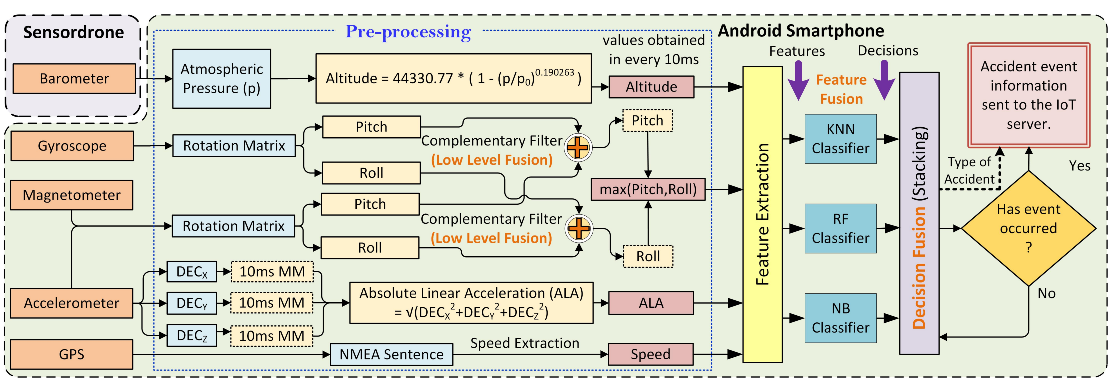


Figure 7. Workflow diagram of proposed system.

Sensor fusion is the synthesis of different sensors' data, which provides better results than a single sensor. In this work, a multi-sensor data fusion framework is proposed whose workflow diagram is shown in Figure 7. Our framework is based on a refined categorization of three-level modeling, where sensor fusion can be applied at low-level, intermediate-level, and high-level of processing (Elmenreich, 2002). Low-level fusion incorporates multiple raw data sources to produce new data (features) which is intended to be more insightful than the inputs. Intermediate fusion combines various features to make a decision. In this work, we have used KNN, NB, and RF classifiers individually to implement the intermediate fusion. In order to achieve the high accuracy, high-level fusion combines the decisions of several classifiers. Decision fusion methods include ensemble methods (e.g., voting, stacking, etc.), fuzzy-logic and statistical methods. We are using Stacking ensemble approach to fuse the decisions of KNN-, NB-, and RF-based classifiers.

## 7.1 Raw-Data Fusion (Low Level Fusion)

From raw data collection to the model variables (features) extraction, all aspects of the pre-processing part of the framework are already explained in detail in section 3.2. Almost at every stage of pre-processing, different kinds of raw data have been fused to increase the accuracy of the resultant data. This includes (a) getting the ALA from the accelerometer (Cooperative Fusion), (b) applying the complementary filter to stabilize the drifts in pitch and roll values provided by the three inertial sensors (Complementary Fusion), and (c) recording the maximum value of pitch and roll (Competitive Fusion) (Elmenreich, 2002).

## 7.2 Feature Fusion (Intermediate Level Data Fusion)

In classification problems, to make the correct decisions, several features need to be combined in some way. There are different approaches such as statistical methods, probabilistic models, machine learning (ML) models, etc., which can be used to make a classification, decision, or inference. In this work, we are analyzing the performance of three ML models viz., KNN, RF, and NB, so that we can compare their accuracy to each other, and to a meta-classifier that fuses their output in the next level of fusion.

7.2.1 *K-Nearest Neighbor (KNN) based ADC Model.* K-Nearest Neighbor (KNN) is a simple algorithm that predicts each observation based on its "similarities and differences" with other observations. It is a memory-based algorithm, which can't be represented by a closed model. This means training samples are needed at run time and predictions are made directly from the relationships of the samples. This makes the KNN method much quicker for small datasets and more flexible than other algorithms such as Linear Regression, SVM, etc., which need training. K-NN learns through analogy, as a comparison is performed between a single test tuple and a collection of identical training tuples (Adeniyi et al., 2016).

The test tuples classification is based on similarities with the class of nearest k neighbors, where k represents the number of neighbors required for class determination (Steenwijk et al., 2013). Pairwise distance between observations is used to determine the similarity or dissimilarity between observations. Euclidean distance and Manhattan distance both are the most common metrics to measure the distance between two observations. In this model, we have used the Euclidean distance metric that measures straight line distance between two samples. In this research, KNN technique is used to model the case of vehicle accident detection and classification, based on four input variables (attributes), (a) AltitudeChange, (b) MaxOfPitchAndRoll, (c) Speed, and (d) ALA, and one output class variable - AccidentType. The possible outcomes of the proposed model are: (a) Collision, (b) Rollover, (c) Fall-off, and (d) No-Accident. If $x_r(x_{r1}, x_{r2}, ..., x_{rM})$ is training tuple and $x_s(x_{s1}, x_{s2}, ..., x_{sM})$ is the testing tuple, then for all features, Euclidean distance, $d(x_r, x_s)$, between them is calculated by

$$d(x_r, x_s) = \sqrt{\sum_{j=1}^{M} (x_{rj} - x_{sj})^2} \tag{6}$$

where $M$ is the number of attributes. In our model, the value of $M$ is four as the number of training attributes are four. The classifier's decision is based on the test tuple being at the closest distance to the trained tuples. If $d_1, d_2, ..., d_k$ are the Euclidean distances of KNN training tuples of different classes from a testing tuple $x_s$, then the class of $x_s$ would be the same as the class of testing tuple whose Euclidean distance from $x_s$ is $min\{d_1, d_2, ..., d_k\}$.

7.2.2 *Random Forest (RF) based ADC Model.* Random Forest is a supervised machine learning algorithm for regression and classification, which is proposed by Breiman (2001) and Biau (2012). Random Forest is an ensemble technique (Breiman et al., 1984) used in several classification areas (Seera and Lim, 2014) that works with the idea of the grouping of multiple weak learners to create a strong learner. CART (Classification and Regression Trees) technique (Genuer et al., 2008) is used by the RF to develop a bagging technique (Liaw and Wiener, 2002) based on multiple decision trees. The purpose of the CART technique is to learn the best possible classification and relationship between dependent variables and independent variables. For RF, each tree chooses a subset of the data set randomly to construct an independent decision tree. Every randomly chosen subset divided repeatedly from the root node to the child node until this division reaches the leaf node without pruning. Every tree individually classifies the features and the target variable, and decides for the final class. Every tree provides a class, and based on the majority voting, RF decides the final class.

In our data set we have four independent variables (a) AltitudeChange, (b) MaxOfPitchAndRoll, (c) Speed, and (d) ALA, and one dependent variable - AccidentType. If R is the number of RF trees, our RF algorithm can be summarized as follows:
***Step 1:*** *For each tree $T_r$ of R, repeat the following steps 2 to 7:*
***Step 2:*** *Draw Z as the bootstrap sample of size N from training data set X*
***Step 3:*** *Generate an RF tree $T_r$ using Z by recursively repeat steps 4 to 6 for each node until the minimum node size one is reached.*
***Step 4:*** *Randomly select m (=2) variables (where $1 <= m <= \lfloor\sqrt{4}\rfloor$) from M (=4) variables*
***Step 5:*** *Choose the best variable among m randomly selected variables to split the node based on the minimum Gini Index (Han et al., 2016)*
***Step 6:*** *Split the node into two child nodes*
***Step 7:*** *The final output class would be the ensemble of all RF trees i.e. if $C_r(x)$ is the predicted class of $r^{th}$ RF tree and x is the random vector then final output class C of complete Random Forest would be*

$$C = majority\_vote\{C_r(x)\}_1^r \tag{7}$$

7.2.3 *Nave Bayes (NB) based ADC Model.* Another NB-based classification technique is employed for ADC. The NB approach is highly scalable, and the number of predictors scales linearly (Šingliar and Hauskrecht, 2010). Because of its computational simplicity and its ability to be trained very quickly (Yang et al., 2015), it is preferred over other classification techniques. As it is not sensitive to irrelevant features, it is resilient to noisy data. This approach is based on the assumption that the predictor variables are autonomous, i.e. the existence of certain features does not affect a specific feature present in a class. The Nave Bayes model includes multiple predictor variables (AltitudeChange, MaxOfPitchAndRoll, Speed, and ALA) and a target variable (AccidentType) as model output. Let $Y$ be the state of target variable (or class), and vector $X = (x_1, x_2, ..., x_n)$ is the state of $n$ predictor variables. We need to measure the likelihood of $Y$ given $X$ to estimate the value of $Y$ based on $X$. The conditional probability $Y \mid X$ is calculated as

$$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)} \tag{8}$$

where, $p(Y)$ and $p(X)$ are directly derived constants from the data. To get the value of $X \mid Y$, it is factorized as

$$p(X|Y) = p(x_1, x_2, \cdots, x_n|Y) = \prod_{i=1}^{n} p(x_i|Y) \tag{9}$$

From equations (8) and (9), we get

$$p(Y|X) = \frac{p(Y)}{p(X)} \prod_{i=1}^{n} p(x_i|Y) \tag{10}$$

where, parameters $p(Y), p(X)$, and $p(x_i \mid Y)$ of our classification model are directly learned from the training data. Hence, the conditional distribution of $Y \mid X$ is calculated from the equation above. The value of target variable $Y \mid X$ is the classification output and $Y$ with the highest probability will be the state of the model.

### 7.3 Decision Fusion (High Level Data Fusion)

Several highly accurate revolutionary classifiers for real-life data classification have been developed in recent years. However, the accuracy of many of these classifiers depends on the application in hand. The presumption of most classifiers that real-life data has a strong similarity to training data may not be valid in real-life implementation. Lack of suitable training data is a significant reason for the poor performance of most of the classifiers.

Due to the good performance achieved in many tasks such as classification or regression problems, ensemble models have become more important over the last few years (Ren et al., 2016). Ensemble learning (Dietterich, 2000)(Mendes-Moreira et al., 2012) is an approach that integrates various models of learning to boost the results obtained from each particular model. The combination (fusion) of several classifiers outputs (decisions) may reduce the probability of choosing an under-performing classifier. The errors made by one classifier, when classifying instances, are usually averaged out by another classifier's right classification, so that the cumulative classification accuracy is improved. Bagging, boosting, and stacking are the most commonly used and well-known ensemble methods.

Bagging (Bühlmann et al., 2002)(Breiman, 1996a), also called bootstrap aggregating, is an ensemble method that combines multiple decision trees predictions to minimize variance in predictions, thereby eliminate over-fitting by model averaging, and is resilient to outliers and noisy data. This eventually contributes to greater predictive accuracy.

Boosting (Freund and Schapire, 1997)(Schapire, 1990) is another ensemble method that provides sequential learning of the predictors. The first predictor is learned throughout the entire data set, while the following are learned through the training set based on the results of the previous one.

In this paper, the stacking approach (Breiman, 1996b) has been used because bagging and boosting uses the same type of weak learners. On the other side, in stacking, base classifiers should be of different type to allow a final impartial decision. These stacked ensembles appear to outperform all of the individual base learners (e.g., a single Random Forest or Gradient Boosting Method) and have proved to be an asymptotically optimal learning system (Van Der Laan et al., 2007).

7.3.1 *Stacking based ADC model.* Stacking has a structure of two levels: level-0 classifiers (base-classifiers) and level-1 classifier (meta-classifier) (Wolpert, 1992). The base-classifiers are trained with the training data set, and their predictions are produced. Then the meta-classifier is trained to map the outputs of the base-classifiers to the actual class label with the meta-data. The meta-data could be $\{(y_j^1, y_j^2, ..., y_j^n), y_j\}$, where $y_j^n$ denotes the prediction provided on the $j^{th}$ instances by the $n^{th}$ base-level classifier, and $y_j$ is the actual class name. The qualified base-classifiers will give their predictions individually during the process of classifying a new data instance, and those predictions will be used as the feedback to meta-classifier to generate the final decision. Let $Y$ denote the target variable, $X$ denotes the input space, and $y_j^1, y_j^2, ..., y_j^n$ denote the predictions learned from n-machine learning algorithms. A linear ensemble model constructs a predictive function for an interval target $j$,

$$b(\hat{y_j}) = w_1 * y_j^1 + w_2 * y_j^2 + ... + w_n * y_j^n \tag{11}$$

where $w_1, w_2, ..., w_n$ are the model weights and $\hat{y_j}$ is the predicted class. An easy way to define these weights is to set them all equal to *1/n* so that each variable contributes to the final ensemble equally. Additionally, you can assign greater weight to the models that you think would work better. While assigning weights by hand can often be fair, using a learning algorithm to estimate them can usually boost the final ensemble efficiency. Logistic regression is a widely used approach for final model stacking, due to its computational efficiency and model interpretability. The model weights ($w_n$) are determined in a regression model by solving the following least-squares problem:

$$min \sum_{j=1}^{n} (y_j - (w_1 * y_j^1 + w_2 * y_j^2 + ... + w_n * y_j^n))^2 \tag{12}$$

In this work, in each run of our stacking ensemble, K-Nearest Neighbor, Random Forest, and Nave Bayes learners are the base-learners with equal model weights and Logistic Regression learner is the meta-learner.

## 8. PERFORMANCE EVALUATION

This segment evaluates the performance of different ADC models based on K-Nearest Neighbor (KNN), Nave Bayes (NB), Random Forest (RF), and Logistic Regression based Stacking Ensemble Model (Stacking). The results were obtained through the *SNUSense* databases 1050 mixed experimental observations of collision, rollover, fall-off, and no-accident events. 10-fold cross-validation is used to validate (regularize) each model by explicitly breaking the data set before the training and testing process, so that all models can use the same data splits. K-fold cross-validation is among the techniques used to assess a machine learning model's efficacy; this is also a re-sampling method used to validate a model when we have a small data set.

Scatter plots of each model are shown in Figure 8, which are easily understandable with the color-coding scheme used. In these scatter plots, colored dots are depicting true positives (TP), and cross signs are depicting the false positives (FP) and false negatives (FN) results of accident events.

Confusion matrix in Figure 9 is showing the number of correctly and incorrectly classified observations for each model for detailed analysis. The performance of proposed classification models is evaluated through accuracy, precision, recall, and F1-score metrics with the help of the confusion matrix (Powers, 2011).

*Accuracy* is the most obvious performance metric for a classification model which is simply a ratio of correctly predicted observations and total observations. Accuracy is a good measure but only when the number of TP and true negatives (TN) are almost same.

$$Accuracy = \frac{total\ of\ true\ positive\ observations}{total\ of\ actual\ positive\ observations} \tag{13}$$

*Precision* is the ratio of total TP observations and all positive observations. More precision indicates a less FP rate.

$$Precision = \frac{total\ of\ true\ positive\ observations}{total\ of\ positively\ predicted\ observations} \tag{14}$$

*Recall* is the ratio of total TP observations and a total of actual positive observations.

$$Recall = \frac{total\ of\ true\ positive\ observations}{total\ of\ actual\ positive\ observations} \tag{15}$$

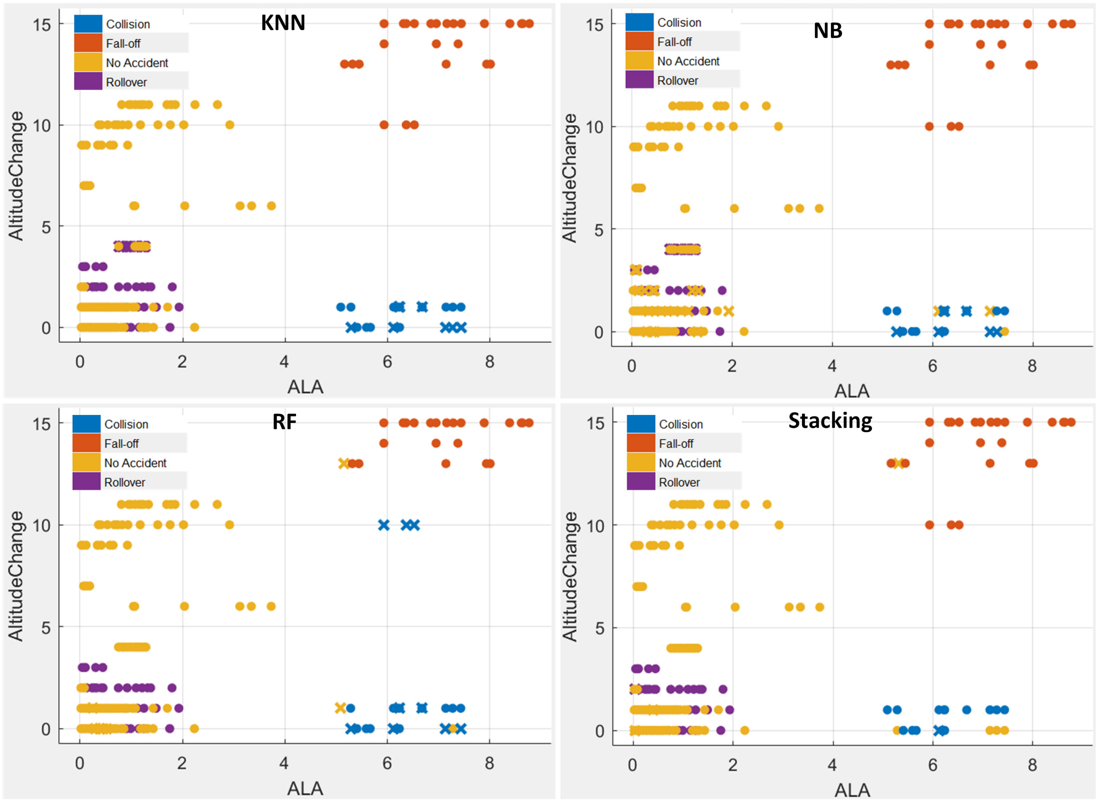


Figure 8. Scatter plots of KNN(top-left), NB(top-right), RF(bottom-left) and Stacking(bottom-right) based ADC models.

*F1-score* is the harmonic mean of precision and recall. F1-score is generally more useful than accuracy, particularly if the class distribution is uneven.

$$F1 - score = 2 * (\frac{Precision * Recall}{Precision + Recall}) \tag{16}$$

The classification efficiency of the KNN based classification technique is presented in Table II. The KNN based classification model has a mean of precision, recall, and F1-score of 0.87, 0.98, and 0.92, respectively.

Table II: DETECTION AND CLASSIFICATION PERFORMANCE OF KNN-BASED ADC MODEL

| Accident Type | TP | FP | TN | FN | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|---|---|
| Collision | 18 | 11 | 1021 | 0 | 99.95% | 0.62 | 1.00 | 0.77 |
| Fall-off | 45 | 0 | 1005 | 0 | 100.00% | 1.00 | 1.00 | 1.00 |
| No Accident | 726 | 0 | 280 | 44 | 95.81% | 1.00 | 0.94 | 0.97 |
| Rollover | 217 | 33 | 800 | 0 | 96.86% | 0.87 | 1.00 | 0.93 |

An average F1-score of 0.92 for the four types of accident incidents suggests high accuracy of the proposed KNN-based ADC model. The model performs best with F1-score of 1.00 for fall-off accident detection, followed by a no-accident, rollover, and fall-off cases with F1-score of 0.97, 0.93, and 0.77, respectively. The overall accuracy of the model is observed 95.8 percent with 1006 TP observations.

Table III is providing a description of proposed NB-based ADC models performance. The NB-based model has an average precision, recall, and F1-score of 0.89, 0.94, and 0.91, respectively.

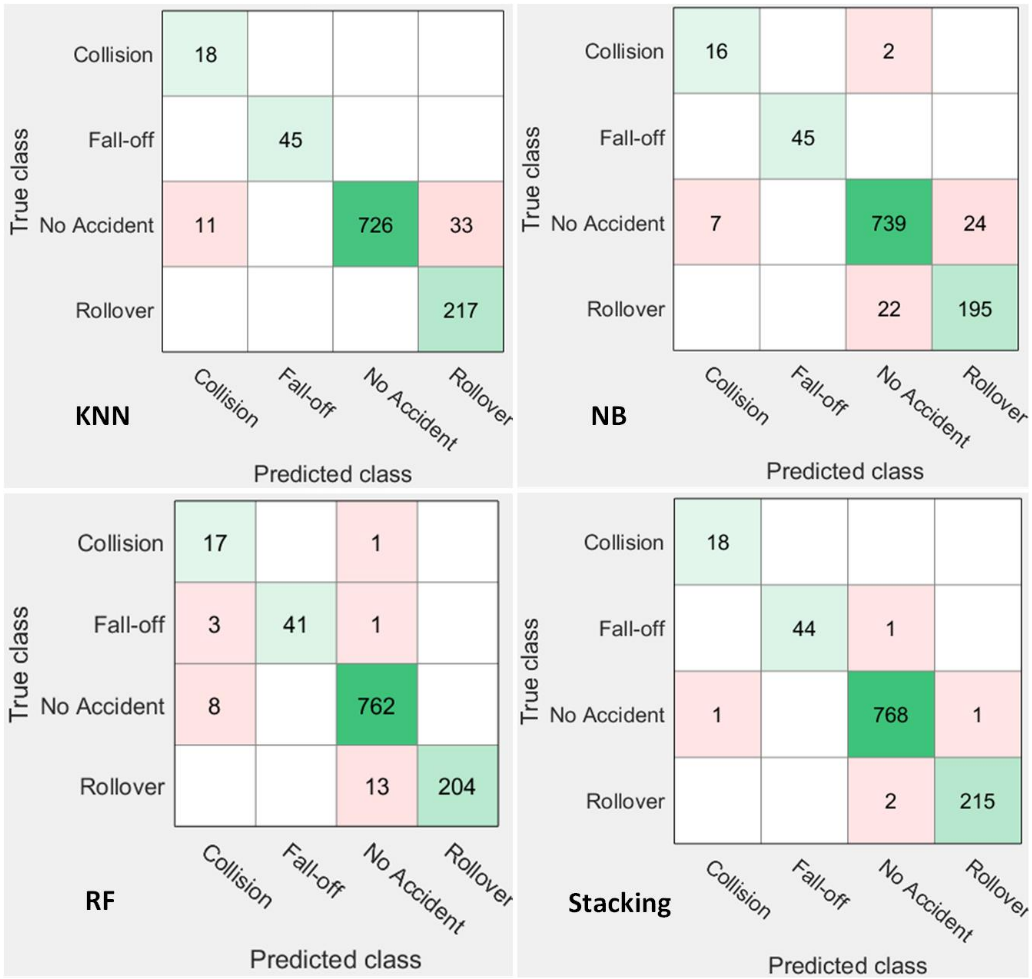


Figure 9. Confusion matrix of KNN(top-left), NB(top-right), RF(bottom-left), and Stacking(bottom-right) based ADC models.

The model performs better in fall-off accident events with an F1-score of 1.00 followed by no-accident, rollover, and collision with F1-scores of 0.96, 0.89, and 0.78 respectively. With a total of 995 TP observations, the accuracy of NB-based model is found to be 94.76 percent.

Table III: DETECTION AND CLASSIFICATION PERFORMANCE OF NB-BASED ADC MODEL

| Accident Type | TP | FP | TN | FN | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|---|---|
| Collision | 16 | 7 | 1025 | 2 | 99.14% | 0.70 | 0.89 | 0.78 |
| Fall-off | 45 | 0 | 1005 | 0 | 100.00% | 1.00 | 1.00 | 1.00 |
| No Accident | 739 | 24 | 256 | 31 | 94.76% | 0.97 | 0.96 | 0.96 |
| Rollover | 195 | 24 | 811 | 22 | 95.62% | 0.89 | 0.90 | 0.89 |

Table IV explains the efficiency of ADC model based on the RF. After the no-accident events with F1-score of 0.99, the model performs well for rollover events with an F1-score of 0.97, which is followed by fall-off and collision events with F1-scores of 0.95 and 0.74, respectively. The RF model's average precision, recall, and F1-score were observed as 0.90, 0.96, and 0.91, respectively. The accuracy of the RF-based model is found to be 97.52 percent with 1024 TP observations.

Table IV: DETECTION AND CLASSIFICATION PERFORMANCE OF RF-BASED ADC MODEL

| Accident Type | TP | FP | TN | FN | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|---|---|
| Collision | 17 | 11 | 1021 | 1 | 98.86% | 0.61 | 0.98 | 0.74 |
| Fall-off | 41 | 0 | 1005 | 4 | 99.62% | 1.00 | 0.91 | 0.95 |
| No Accident | 762 | 15 | 265 | 8 | 97.81% | 0.98 | 0.99 | 0.99 |
| Rollover | 204 | 0 | 833 | 13 | 98.76% | 1.00 | 0.94 | 0.97 |

Although, NB-, KNN-, and RF-based classification models perform well, one can notice that in comparison to these methods, the stacking ensemble method achieved a lower false-positive rate with 1045 TP observation. Decision fusion of NB, KNN, and RF models output by logistic regression-based meta classifier is found to be highly accurate with an accuracy of 99.52 percent.

Table V shows that the mean of precision, recall, and F1-score of the model are 0.99, 0.99, and 0.99, respectively. After the no-accident class with F1-score of 1.0, the model performs well with rollover and fall-off events with F1-score of 0.99 for each, which is followed by collision with an F1-score of 0.97.

Table V: DETECTION AND CLASSIFICATION PERFORMANCE OF STACKING-BASED ADC MODEL

| Accident Type | TP | FP | TN | FN | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|---|---|---|
| Collision | 18 | 1 | 1031 | 0 | 99.90% | 0.95 | 1.00 | 0.97 |
| Fall-off | 44 | 0 | 1005 | 1 | 99.90% | 1.00 | 0.98 | 0.99 |
| No Accident | 768 | 3 | 277 | 2 | 99.52% | 1.00 | 1.00 | 1.00 |
| Rollover | 215 | 1 | 832 | 2 | 99.71% | 1.00 | 0.99 | 0.99 |

The accuracy of KNN, RF, NB and Stacking based ADC models are $95.8\% \pm 1.5$, $97.52\% \pm 2.0$, $94.76\% \pm 3.0$, and $99.52\% \pm 0.48$, respectively. By analyzing the results in the preceding text and error bar graph of Figure 10, we can state that decision fusion by stacking ensemble outperforms the individual NB, KNN, and RF-based classifiers for our accident detection and classification problem.

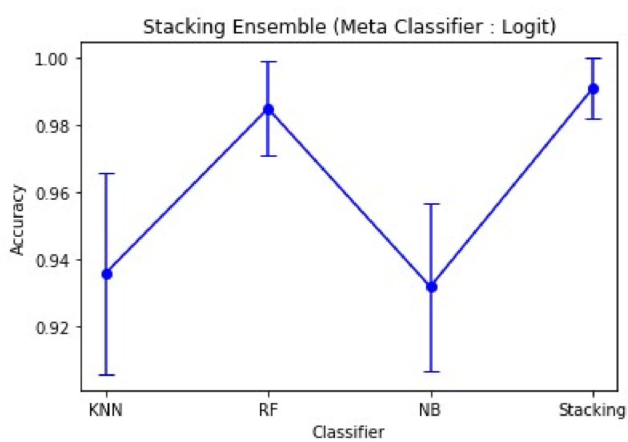


Figure 10. Error bar graph of all approaches.

## 9. CONCLUSION

This work proposed a novel smartphone-based end-to-end IoT system to detect, classify, and report vehicle accidents. The proposed system uses several sensor fusion schemes to enhance the accuracy of the classification. Four input features i.e. altitude change, maximum of pitch and roll, speed and ALA, are used to accurately classify the accident events as collision, fall-off, rollover, and no-accident. Android smartphone's GPS and 9-axis inertial sensors, and Sensordrones atmospheric barometer sensor are used to monitor the values of input parameters. For initial classification, three base classifiers namely NB, KNN, and RF have been used and their performance is compared using the accuracy, precision, recall, and F1-score metrics. To further increase the classification accuracy of our ADC system we have fused the output of three base classifiers namely NB, KNN, and RF, by a meta classifier. Logistic regression based stacking ensemble method is used as a meta classifier. When individually analyzed, the RF-based model is found to be highly accurate with an average F1-score of 0.97 than NB and KNN based models, however, stacking, with three base classifiers, outmatched RF-based models with 0.99 average F1-score. Our ADC system is economical in comparison to factory fitted systems and can be retrofitted easily in existing vehicles.

## References


Acharya, D., Kumar, V., and Gaddis, G. M. 2007. A mobile system for detecting and notifying vehicle rollover events. In *15th International Conference on Advanced Computing and Communications (ADCOM 2007)*. IEEE, Guwahati, India, pp.268–275.

Adeniyi, D. A., Wei, Z., and Yongquan, Y. 2016. Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. *Applied Computing and Informatics Vol.12,* No.1, pp.90–108.

Aloul, F., Zualkernan, I., Abu-Salma, R., Al-Ali, H., and Al-Merri, M. 2015. IBump: Smartphone application to detect car accidents. *Computers and Electrical Engineering Vol.43*, pp.66–75.

Barabba, V., Huber, C., Cooke, F., Pudar, N., Smith, J., and Paich, M. 2002. A multimethod approach for creating new business models: The general motors OnStar project. *Interfaces Vol.32,* No.1, pp.20–34.

Bhatti, F., Shah, M. A., Maple, C., and Ul Islam, S. 2019. A novel internet of things-enabled accident detection and reporting system for smart city environments. *Sensors (Switzerland) Vol.19,* No.9, p.2071.

Biau, G. 2012. Analysis of a random forests model. *The Journal of Machine Learning Research Vol.13,* No.1, pp.1063–1095.

Breiman, L. 1996a. Bagging predictors. *Machine Learning Vol.24,* No.2, pp.123–140.

Breiman, L. 1996b. Stacked regressions. *Machine Learning Vol.24,* No.1, pp.49–64.

Breiman, L. 2001. Random forests. *Machine Learning Vol.45,* No.1, pp.5–32.

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. 1984. *Classification and regression trees.* CRC press.

Bühlmann, P., Yu, B., et al. 2002. Analyzing bagging. *The Annals of Statistics Vol.30,* No.4, pp.927–961.

Chan, C. Y. 2002. A treatise on crash sensing for automotive air bag systems. *IEEE/ASME Transactions on Mechatronics Vol.7,* No.2, pp.220–234.

Chung, Y. and Recker, W. W. 2012. A Methodological Approach for Estimating Temporal and Spatial Extent of Delays Caused by Freeway Accidents. *IEEE Transactions on Intelligent Transportation Systems Vol.13,* No.3, pp.1454–1461.

Damousis, I. G. and Tzovaras, D. 2008. Fuzzy fusion of eyelid activity indicators for hypovigilance-related accident prediction. *IEEE Transactions on Intelligent Transportation Systems Vol.9,* No.3, pp.491–500.

Dar, B. K., Shah, M. A., Islam, S. U., Maple, C., Mussadiq, S., and Khan, S. 2019. Delay-Aware Accident Detection and Response System Using Fog Computing. *IEEE Access Vol.7*, pp.70975–70985.

Dietterich, T. G. 2000. Ensemble methods in machine learning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, pp.1–15.

Dunwoody, A. B. and Stern, D. S. 1998. System and method for the detection of vehicle rollover conditions. US Patent 5,825,284.

Elmenreich, W. 2002. An introduction to sensor fusion. *Vienna University of Technology, Austria Vol.502*, pp.1–28.

Englisch, A. 2002. BMW ASSIST-Telematics for Safety and Convinience. In *9th World Congress on Intelligent Transport Systems ITS America*, ITS Japan, Ed. ERTICO (Intelligent Transport Systems and Services-Europe).

Felisberto, F., Fdez-Riverola, F., and Pereira, A. 2014. A ubiquitous and low-cost solution for movement monitoring and accident detection based on sensor fusion. *Sensors (Switzerland) Vol.14,* No.5, pp.8961–8983.

Fernandes, B., Alam, M., Gomes, V., Ferreira, J., and Oliveira, A. 2016. Automatic accident detection with multi-modal alert system implementation for ITS. *Vehicular Communications Vol.3*, pp.1–11.

Fogue, M., Garrido, P., Martinez, F. J., Cano, J. C., Calafate, C. T., and Manzoni, P. 2012. Automatic accident detection: Assistance through communication technologies and vehicles. *IEEE Vehicular Technology Magazine Vol.7,* No.3, pp.90–100.

Freund, Y. and Schapire, R. E. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences Vol.1,* No.55, pp.119–139.

Genuer, R., Poggi, J.-M., and Tuleau, C. 2008. Random forests: some methodological insights. Ph.D. thesis, INRIA, France.

Ghangurde, M. 2010. Ford SYNC and microsoft windows embedded automotive make digital lifestyle a reality on the road. In *SAE Technical Papers*.

Google Developers. 2020. Google Firebase. Retrieved from https://firebase.google.com/.

GP TOYS. 2020. Foxx S911 RC Truck. Retrieved from https://g-p.hk/gptoys-foxx-s911.html.

Han, H., Guo, X., and Yu, H. 2016. Variable selection using Mean Decrease Accuracy and Mean Decrease Gini based on Random Forest. In *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*. IEEE, pp.219–224.

Ibrahim, H. A., Aly, A. K., and Far, B. H. 2016. A system for vehicle collision and rollover detection. In *Canadian Conference on Electrical and Computer Engineering*. IEEE, pp.1–6.

Iyoda, M., Trisdale, T., Sherony, R., Mikat, D., and Rose, W. 2016. Event Data Recorder (EDR) developed by Toyota Motor Corporation. *SAE International Journal of Transportation Safety Vol.4,* No.1, pp.187–201.

Kendall, J. and Solomon, K. 2014. Airbag deployment criteria: Institute of risk & safety analyses. *The Forensic Examiner*.

Kubelka, V. and Reinstein, M. 2012. Complementary filtering approach to orientation estimation using inertial sensors only. In *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, pp.599–605.

Kumar, N., Barthwal, A., and Acharya, D. 2019. Modeling vehicle collision events using internet of things. In *2019 IEEE 16th India Council International Conference (INDICON) - Symposium Proceedings*. IEEE, Rajkot, India, pp.1–4.

Kumar, N., Barthwal, A., Acharya, D., and Lohani, D. 2020a. Modeling IoT Enabled Automotive System for Accident Detection and Classification. In *IEEE Sensors Applications Symposium (SAS)*. IEEE, Kuala Lumpur, Malaysia. In Press.

Kumar, N., Barthwal, A., Acharya, D., and Lohani, D. 2020b. Modeling Vehicle Fall

Detection Event Using Internet of Things. In *Advances in Signal Processing and Intelligent Recognition Systems*, S. M. Thampi, R. M. Hegde, S. Krishnan, J. Mukhopadhyay, V. Chaudhary, O. Marques, S. Piramuthu, and J. M. Corchado, Eds. Springer Singapore, Singapore, pp.220–233.

Kumar, N., Barthwal, A., Lohani, D., and Acharya, D. 2020. Vehicle Fall Severity Modeling using IoT and K-Nearest Neighbor Algorithm. In *2020 International Conference on COMmunication Systems and NETworkS, COMSNETS 2020.* IEEE, Bangalore, India, pp.105–109.

Langheim, J. 2002. Environment Sensing for Advanced Driver Assistance CARSENSE. In *Advanced Microsystems for Automotive Applications Yearbook 2002.* Springer, Berlin, Heidelberg, pp.96–105.

Lee, Y., Yeh, H., Kim, K. H., and Choi, O. 2018. A real-time fall detection system based on the acceleration sensor of smartphone. *International Journal of Engineering Business Management Vol.10*, pp.1–8.

Lerner, E. B. and Moscati, R. M. 2001. The Golden Hour: Scientific Fact or Medical "Urban Legend"? *Academic Emergency Medicine Vol.8,* No.7, pp.758–760.

Liaw, A. and Wiener, M. 2002. Classification and regression trees by random forest. *R News Vol.2,* No.3, pp.18–22.

Lohani, D. and Acharya, D. 2017. Real time in-vehicle air quality monitoring using mobile sensing. In *2016 IEEE Annual India Conference (INDICON).* IEEE, pp.1–6.

Maruti-Suzuki. 2018. SUZUKI CONNECT: Advanced Telematics Solution. Retrieved from https://www.marutisuzuki.com/corporate/technology/suzuki-connect/.

McIver, G., Carlin, M., Bormann, J., Muckley, R., and McCurdy, R. 1996. Method and apparatus for sensing a vehicle crash condition using velocity enhanced acceleration crash metrics. US Patent 5,587,906.

Mendes-Moreira, J., Soares, C., Jorge, A. M., and Sousa, J. F. D. 2012. Ensemble approaches for regression: A survey. *ACM Computing Surveys (CSUR) Vol.45,* No.1, pp.1–40.

Moulik, S. and Majumdar, S. 2018. FallSense: An Automatic Fall Detection and Alarm Generation System in IoT-Enabled Environment. *IEEE Sensors Journal Vol.19,* No.19, pp.8452–8459.

NMEA. 2020. National Marine Electronics Association. Retrieved from https://www.nmea.org/.

Ponte, G., Ryan, G. A., and Anderson, R. W. 2016. An estimate of the effectiveness of an in-vehicle automatic collision notification system in reducing road crash fatalities in South Australia. *Traffic Injury Prevention Vol.17,* No.3, pp.258–263.

Powers, D. M. 2011. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation. *Journal of Machine Learning Technology*.

Ren, Y., Zhang, L., and Suganthan, P. N. 2016. Ensemble classification and regression-recent developments, applications and future directions. *IEEE Computational intelligence magazine Vol.11,* No.1, pp.41–53.

Sada, H. and Moriyama, H. 1998. Crash Sensor. US Patent 5,777,225.

Sadeky, S., Al-Hamadi, A., Michaelis, B., and Sayed, U. 2010. Real-time automatic traffic accident recognition using HFG. In *Proceedings - International Conference on Pattern Recognition.* IEEE, pp.3348–3351.

Sánchez-Mangas, R., García-Ferrrer, A., De Juan, A., and Arroyo, A. M. 2010. The probability of death in road traffic accidents. How important is a quick medical response? *Accident Analysis and Prevention Vol.42,* No.4, pp.1048–1056.

Schapire, R. E. 1990. The Strength of Weak Learnability. *Machine Learning Vol.5,* No.2, pp.197–227.

Seera, M. and Lim, C. P. 2014. A hybrid intelligent system for medical data classification. *Expert Systems with Applications Vol.41,* No.5, pp.2239–2249.

Šingliar, T. and Hauskrecht, M. 2010. Learning to detect incidents from noisily labeled data. *Machine Learning Vol.79,* No.3, pp.335–354.

Smolka, J. and Skublewska-Paszkowska, M. 2016. A method for collision detection using mobile devices. In *Proceedings - 2016 9th International Conference on Human System Interactions, HSI 2016*. IEEE, pp.126–132.

Steenwijk, M., Pouwels, P., Daams, M., Van Dalen, J., Caan, M., Richard, E., Barkhof, F., and Vrenken, H. 2013. Accurate white matter lesion segmentation by k nearest neighbor classification with tissue type priors (kNN-TTPs). *NeuroImage: Clinical Vol.3*, pp.462–469.

Steiner, P., Weidel, P., Küblbeck, H., Steurer, H., Hora, P., and Zechmair, D. 1997. Roll over detection. In *SAE Technical Papers*.

Toyota Motor Corp. 2020. SAFETY CONNECT. Retrieved from https://www.toyota.com/connected-services/safetyconnect/.

Van Der Laan, M. J., Polley, E. C., and Hubbard, A. E. 2007. Super learner. *Statistical Applications in Genetics and Molecular Biology*.

WHO. 2018. Global status report on road safety 2018: summary. Tech. rep., World Health Organization, Geneva.

WHO. 2019. World health statistics 2019: monitoring health for the SDGs, sustainable development goals. Tech. rep., World Health Organization, Geneva.

Wolpert, D. H. 1992. Stacked generalization. *Neural Networks Vol.5,* No.2, pp.241–259.

Yang, B., Lei, Y., and Yan, B. 2015. Distributed Multi-Human Location Algorithm Using Naive Bayes Classifier for a Binary Pyroelectric Infrared Sensor Tracking System. *IEEE Sensors Journal Vol.16,* No.1, pp.216–223.

Zhang, Z., He, Q., Gao, J., and Ni, M. 2018. A deep learning approach for detecting traffic accidents from social media data. *Transportation research part C: emerging technologies Vol.86*, pp.580–596.

**Mr. Nikhil Kumar** is a PhD student of Computer Science and Engineering at Shiv Nadar University, India. His research interests are context aware computing, wireless sensor network, vehicular accident detection and classification modeling, and Internet of Things.

**Dr. Debopam Acharya** is currently Professor and Dean of School of Computing at DIT University, Dehradun, India. His research interests are in the areas of Internet of Things, Mobile Sensing and Smart Environment. He has worked in the Location Based Services group at Garmin Inc., Olathe, Kansas, USA and tenure track faculty at Georgia Southern University, Statesboro, GA, USA. His research has been funded by several public and private sector organisations like BlackBerry, Dell, and the Ministry of Electronics and IT, Govt. of India. Dr. Acharya earned his PhD from the University of Missouri-Kansas City, Missouri, USA.

**Dr. Divya Lohani** is serving as an Assistant Professor in the Department of Computer Science and Engineering at Shiv Nadar University, India. She received her PhD from Indian Institute of Information Technology, Allahabad, India. Her research interests include Internet of Things, sensor networks, sensor data analytics, context aware computing, mobile computing and cybersecurity. She is a member of the IEEE.