Vehicle Event Data Recorder and Driver Review System (VEDRAR) Using Machine Learning

Roshith K[1], Dr. C K Smitha[2]
[1]*Asst Professor, Dept of Electronics & Communication Engg, College of Engg Vadakara*
[2]*Associate Professor, Dept of Electronics & Communication Engg, College of Engg Vadakara*

***Abstract*—This paper is based on a developed (as model) system, VEDRAR-Vehicle event data recorder and driver review system aims to introduce a device that acquires and stores various data depending on different driving parameters for a period of drive with the aid of various sensors. The sensors placed in a required area of the vehicle are used to acquire real time vehicle data, stored in a local database XAMP interfaced using Aurdino. All the real time data collected by VEDRAR during each drive is uploaded to cloud. These data can be used in 3 ways 1) To classify the driver as excellent, good, moderate, bad and worse using machine learning algorithms which in turn used to provide review to the driver. 2) these data can be made use in case of an accident for the detection of its cause by Govt Authorities 3) used to see the status or the health of the various parts of the vehicle that whether the vehicle is safe for the ride or should go for the service.**

Keywords— Sensors, Aurdino

I. INTRODUCTION

According to latest statistics on causes of death by WHO, traffic injuries caused an estimated 1.25 million deaths every year i.e., 1 death/25 second. Apart from death, between 20-50 million people suffer from nonfatal injuries resulting in permanent disability. Only 28 countries, representing seven percent of the world's population have adequate laws that address all five risk factors (speed, drunk driving, helmets, seat-belts and child restraints). Without sustained action, road traffic deaths are predicted to be the 7th leading cause of death by 2020. Ninety percent of the road traffic deaths happen in low- and middle- income countries. This calls for the need of a low cost efficient safety device.

This paper is based on a developed (as model) system, VEDRAR-Vehicle event data recorder and driver review system aims to introduce a device that acquires and stores various data depending on different driving parameters for a period of drive with the aid of various sensors. VEDRAR is an electronic recording device which is used to store three characteristics of driving data namely speed, proximity and tilting angle of steering. Speed as it implies defined the current speed of the vehicle, proximity describes the nearness or closeness in space between the parent vehicle any nearby vehicle, and the tilt angle is the angle at which the steering must be turned for a comfortable drive. VEDRAR collect and store this data, the machine analyses the data with reference to the desired value and provide a review of the current movement of the vehicle if any variations are noticed so that the driver can take precaution an improve the condition. VEDRAR also upload all the vehicle stored data in a cloud so that as in any conventional machine mechanisms, These data can be made use in case of an accident for the detection of its cause, verify the condition of the vehicle before the ride.

A prototype model is prepared and implemented in a bike and the similar machine can be installed in four wheelers also. VEDRAR is an electronic device with hardware and software sections as discussed in next section.

II. MATERIALS

Vehicle event data recorder (VEDRAR) is a device that records the necessary driving data for a period of the drive with the aid of various sensors. The block diagram representation of the VEDRAR, is shown in figure (Fig-1) Hardware part includes various sensors Aurdino and Rasberry Pi. Software includes various machine learning algorithms for classification of the pattern of driving, various sensor data

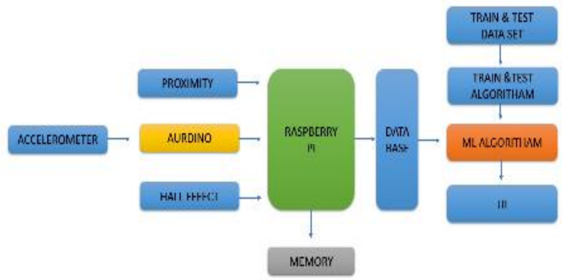

Fig-1 Block diagram representation

Ultrasonic (HC-SR04), accelerometer (HC-SR04) and Hall effect sesnors (3144) are used for measurement of proximity with the surroundings, speed of the vehicle and tilt angle of the vehicle steering. The data from the various sensors are fed into a raspberry-pi module and stored for local use and is uploaded to a cloud storage after every drive. The data loaded in the raspberry-pi is used to provide a local interface for the driver after every drive for self evaluation. The data send to the cloud can be retrieved at a distant location like the office of road transport authorities and process and analyze to review the driver's performance and evaluate his present status of driving skills. This categorization can be used by the road transportation authority as a criterion to pass the driving test. The local data stored in raspberry-pi can also be used to analyze driver in case of accidents.

A. Hardware: .

Proximity sensor: A proximity sensor is an electronic sensor that can detect the presence of objects within its vicinity without any actual physical contact. The proximity sensor radiates or emits a beam of electromagnetic radiation, usually in the form of infrared light, and senses the reflection in order to determine the object's proximity or distance from the sensor. As shown above the HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively

Accelerometer: GY-61 DXL335 3-Axis Accelerometer Module is a three axis accelerometer sensor module based on ADXL335 integrated circuit. The ADXL335 is a triple axis accelerometer with extremely low noise and power consumption. The sensor has a full sensing range of +/-3g. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

Raspberry Pi: is a fully working 32-bit computer with a 1 GHz ARMv6 single core microprocessor (ARM1176), a VideoCore 4 GPU, and 512 MB of memory. The GPU is capable of driving a full HD display at 60 fps. This has 40 General Purpose Input and Output (GPIO) ports.

Hall Effect sensor: Hall effect sensors are used for proximity sensing, positioning, speed detection, and current sensing applications. Hall sensors are used to time the speed of wheels and shafts, such as for internal combustion engine ignition timing, tachometers and anti-lock braking systems. A Hall effect sensor is a device that is used to measure the magnitude of a magnetic field. In the pictured wheel with two equally spaced magnets, the voltage from the sensor will peak twice for each revolution. This arrangement is commonly used to regulate the speed of disk drives.

Arduino Uno: The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt battery. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform. The ATmega328 on the Arduino Uno comes preprogrammed with a boot loader that allows uploading new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. The Uno also differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

B. Software

Softwares used for data acquisition and classification are as follows

Arduino IDE, is used for Sensor Data Acquisition from Proximity sensor, Accelerometer and Hall effect sensor

Python is used for data preprocessing and classification.

Python (with libraries scikit-learn and Tensor Flow is used to train and run ML models that classify driver behavior as "good" or "bad". Raspberry Pi OS, Arduino microcontroller environment, is used for integrating all components into a functioning prototype.

III. METHOD

The flowchart shown in figure (Fig-20 illustrates the process of evaluating driver behavior using data from various sensors and a machine learning (ML) algorithm. Here's a step-by-step description of the flow:

1. Start: The process begins.
2. Sensor Data Collection: Three types of measurements are taken in parallel:
   - Distance Measurement using Proximity Sensor: Detects how close objects are to the vehicle.
   - Tilt Measurement using Accelerometer: Monitors the tilt or orientation of the vehicle.
   - Speed Measurement using Hall Effect Sensor: Captures the vehicle's speed.
3. Retrieve Data and Send to ML Algorithm: The sensor data is retrieved and sent to a machine learning algorithm for analysis.
4. Driver Behaviour Analysis: The ML algorithm evaluates the data and classifies the driver behavior as either good or bad.
5. Decision:
   - If the driver behavior is bad the feedback can be given to driver to improve driving behavior
6. Stop: The process ends.

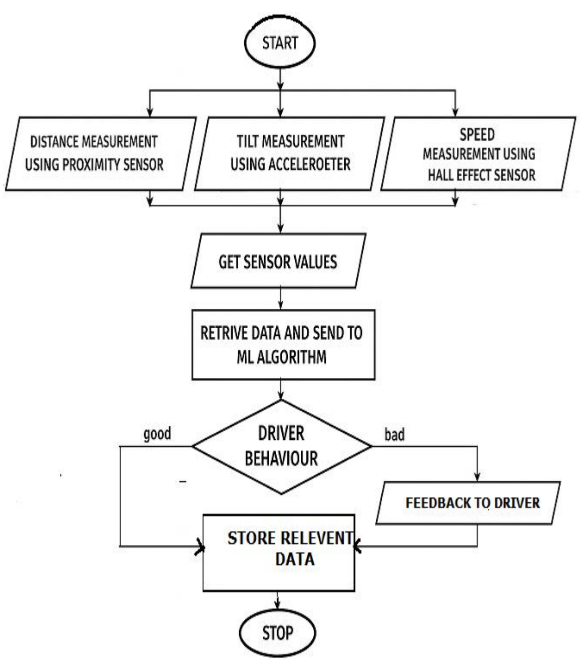

Fig -2 Flow chart of the process

The data is collected from the prototype setup and the data is preprocessed and analyzed using machine learning techniques the result obtained are described in the next section

IV. RESULT

The VEDRAR system was developed and deployed on a two-wheeler prototype for the acquisition and analysis of driving behavior data. The hardware component recorded parameters such as speed (right and left turn speeds), proximity (distances from surrounding objects), and steering tilt (inferred from pressure variations and speed imbalances during turns). The dataset was collected under real driving conditions and labeled based on driving quality—either Good or Bad—according to predefined thresholds and expert evaluation.

To evaluate the effectiveness of the system in reviewing driving behavior, machine learning models were trained on the recorded dataset. Five classification algorithms were tested: Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN).

A. Model Performance

Model	Accuracy
Logistic Regression	100%
Decision Tree	100%
Random Forest	100%
K-Nearest Neighbors	100%
Support Vector Machine	90%

The results demonstrate excellent classification performance for most of the models, with Logistic Regression, Decision Tree, Random Forest, and KNN achieving 100% accuracy, correctly identifying both Good and Bad driving sessions in the test set. SVM performed slightly lower at 90% accuracy, which may be due to its sensitivity to parameter tuning and data scaling, especially given the relatively small dataset.

B. Confusion Matrix Analysis

Confusion matrices reveal that the top-performing models had zero false positives and false negatives, highlighting the robustness of the collected sensor data and the effectiveness of feature selection. This is critical in a real-time driver review system, where incorrect feedback could lead to driver frustration or ignored warnings.

V. DISCUSSION

The high classification accuracy indicates that the chosen features—turning speed, proximity distances, and pressure (related to tilt or steering correction)—are strongly correlated with driving behavior quality. These parameters serve as reliable indicators for the VEDRAR system to analyze and classify driver performance effectively.

The use of machine learning in VEDRAR transforms it from a passive data logger to an active review and feedback system. By evaluating sensor data in real time or post-ride, the system can:

- Alert drivers during driving.
- Provide a summarized report of each ride session.
- Store all data in a cloud platform for post-accident analysis, insurance assessments, or regulatory checks.

While the prototype was implemented on a two-wheeler, the model and algorithms are generic and can be scaled for four-wheelers or commercial fleets, where more complex driving environments exist. This paves the way for its potential adoption in intelligent transportation systems and automotive safety standards.

Limitations and Future Work

Though initial results are promising, the dataset used is limited in size. Further data collection over varied terrain, traffic, and rider profiles would enhance model generalizability. Additionally, integrating GPS, accelerometer data, and real-time alerts will make VEDRAR more comprehensive.

This setup can be modified to a smart driving test system that uses sensor data and ML to assess the behavior of the driver, in which driving test passes or fails based on behavior analysis.

VI CONCLUSION

The idea of the black box is thought-provoking since it is such a tiny machine in a huge vehicle like aircraft that reveals the technical cause of an accident occurred to the vehicle. Through this project, we tried to implement a similar method in a typical automobile. As an initiation implemented our VEDRAR in a bike. The similar machine can be installed in four wheelers too. VEDR collected and stored all sort of driving data that can be accessed during any case of an accident, for detecting the actual reason behind the calamity. Thus VEDRAR helps the authority for official confirmation on the cause of the accident. In the second part, VEDRAR provides a classification on the driving status to the authority from machine reading. So classification application is an advanced property of VEDR.

In the future, a progressive application can be attached to the precent system which store contact numbers in it. This application is a precaution taken for the sake of victims is an accident scene. The driver can incorporate his/her friends or family members number to VEDRAR in advance So, in case of an accident, the authority in charge can inform about the some to one of the numbers saved in the device. This enables a helping band for the faster take care of the victim. Thus in future VEDRAR can be manufactured with lifesaving application so as to help society by providing extra security to citizens life.

REFERENCE

[1] A. Kbassem, R. Jabr, G. Salamouni, and Z. K. Maalouf, “Vehicle black box system,” in 2008 2nd Annual IEEE Systems Conference, April 2008, pp. 1–6.

[2] M. J. Prasad, S. Arundathi, N. Anil, , and B. S. Kariyappa, “Automobile black box system for accident analysis,” in 2014 International Conference on Advances in Electronics Computers and Communications, Oct 2014, pp. 1–5.

[3] A. Vilaca, P. Cunha, and A. L. Ferreira, “Systematic literature review on driving behavior,” in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Oct 2017, pp. 1–8.

[4] S. Yan, Y. Teng, J. S. Smith, and B. Zhang, “Driver behavior recognition based on deep convolutional neural networks,” in 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Aug 2016, pp. 636–641.

[5] Zhang et al. (2018) Driver Behavior Recognition via Interwoven Deep CNN with Multi-stream Inputs"
IEEE Acess 2020 10.1109/ACCESS.2020.3032344

[6] Makinist et al. (2016) , “Estimating Driving Behavior by a Smartphone", *IEEE Intelligent Vehicles Symposium (IV)* ,June 3–7, 2012, Pages: 234–239, 10.1109/IVS.2012.6232298

[7] Haluk Eren, Semiha Makinist, Erhan Akin, Alper Yilmaz, " Estimating Driving Behavior by a Smartphone", IEEE Intelligent Vehicles Symposium 2012