Available online at www.garmian.edu.krd

Passer Journal of Basic and Applied Sciences

Passer

Passer 7 (Issue 2) (2025) 940-952

Original Article

VSLCloud: A LoRa-Enabled Cloud for Real-Time Alcohol Detection and Safer Driving

Twana Latif Mohammed [1], Sarko Salahalddin Ahmed [2], Mustafa Ibrahim Khaleel [3], Zryan Najat Rashid * [4]

[1] IT Department, Kurdistan Technical Institute, Sulaymaniyah 46001, Kurdistan Region, Iraq.
[2] Department of Network, Computer Science Institute, Sulaimani Polytechnic University, Sulaymaniyah 46001, Kurdistan Region, Iraq.
[3] Department of Computer, College of Science, University of Sulaimani, Sulaymaniyah 46001, Kurdistan Region, Iraq.
[4] Department of Computer Networks, Technical College of Informatics, Sulaimani Polytechnic University, Sulaymaniyah 46001, Kurdistan Region, Iraq.

Received 13 June 2025; revised 8 July 2025;
accepted 09 July 2025; available online 09 September 2025

DOI: 10.24271/PSR.2025. 530236.2187

ABSTRACT

VSLCloud offers an online, LoRa-compatible cloud-based intoxication detector, an innovative vehicular safety technology that can prevent driving while intoxicated. Complementing the traditional use of in-car sensors, the VSLCloud system embeds cloud connectivity with IoT-enabled sensors to continuously evaluate intoxication levels and send the results to central authorities. It has the capability, in extreme conditions, of calling for alerts or disabling the car by an incapacitated driver. This paper provides an original algorithm, VSLCloud, and compares the same with recognized machine learning-based techniques, i.e., Random Forest (RF), Support Vector Machine (SVM), and Deep Neural Networks (DNN). Test case-based experimental evaluation through ten diverse test scenarios confirms that the VSLCloud surpasses all these techniques by providing the maximum level of accuracy, the fastest response time, and the maximum level of reliability. For example, the VSLCloud achieved a maximum level of accuracy up to 96.2%, the lowest delay of 120–150 ms, and reliability over 99% in complex scenarios. These results prove the feasibility of the VSLCloud-based system of processing real-time information, fast enough for timely determination, and providing reliable performance through widely fluctuating conditions. The newly designed platform points towards enormous advances towards the development of smart, responsive, and scalable automobile-based safety solutions with great promise toward reducing the incidence of accidents involving intoxication.

Keywords: Alcohol Detection System, IoT Sensors, Cloud Computing, RF, SVM, DNN, LoRa.

1 Introduction

Contemporary traffic collisions are predominantly attributed to driving under the influence of Alcohol. Intoxicated drivers exhibit irregular behavior and are susceptible to impetuous decision-making, jeopardizing the safety of all road users, including themselves. Nigeria has enacted legislation prohibiting drivers from consuming Alcohol before or during the operation of a vehicle and has assigned law enforcement personnel to apprehend and prosecute offenders[1].

The primary cause of traffic accidents is human factors, such as weariness, exhaustion, and tension. These factors, in conjunction with drivers' infractions of traffic laws and driving behavior, primarily constitute the majority of accidents. The consumption of alcohol compromises a driver's judgment and reduces visual acuity, hence increasing the likelihood of a vehicular crash. Alcohol-impaired driving constitutes around forty out of a hundred of all vehicular accidents. Driving under the influence continues to be a widespread and deadly problem worldwide, with considerable repercussions for both society and the economy[1].

The MQ-3 is a device used to assess the quantity of Alcohol in individuals' blood or breath. Its purpose is to limit the number of accidents that occur when driving while under the influence of Alcohol. The apparatus can determine the quantity of ethanol present in the atmosphere through the utilization of an Arduino equipped with an alcohol sensor [2].

Alcohol consumption and vehicular operation remain a primary contributor to road accidents and fatalities globally; hence, the detection of Alcohol in drivers has been extensively researched throughout the years. To reduce the incidence of drunk driving and enhance road safety, many methods have been devised and implemented to measure alcohol levels in drivers. According to

* Corresponding author
E-mail address zryan.rashid@spu.edu.iq (Instructor).
Peer-reviewed under the responsibility of the University of Garmian.

the National Highway Traffic Safety Administration (NHTSA), 28% of all traffic fatalities in the United States in 2019 were attributable to drunk driving. This equates to one alcohol-related fatality approximately every 52 minutes. According to the European Commission, over 10,000 deaths occur annually in the European Union due to alcohol-related incidents[3].

The usage of vehicles has escalated significantly in contemporary society. Due to inadequate emergency services, heightened vehicular traffic has caused an increase in traffic crashes, resulting in several fatalities and property damage. The annual mortality rate results from delays in rescue operations. Nevertheless, automobiles are integrated with cutting-edge technology. The incidence of accidents will continue to increase daily, owing to delays in relaying information to the relevant authorities or in executing rescue operations[2].

Alcohol consumption is seen as a significant hazard to the lives, health, and safety of employees and bystanders. The detrimental consumption of Alcohol is a primary risk factor for public health and a significant cause of mortality globally. A World Health Organization report indicates that Alcohol ranks third among the risk factors for human health, with over 60 diseases and injuries linked to its consumption. Alcohol use disorder (alcohol addiction) is a severe condition with extensive implications for life and health[4].

1. 1 Objective

The primary objective of this study is to install alcohol sensors in the car. To achieve this, the research will focus on driver testing the alcohol level before starting the vehicle to evaluate its effectiveness in detecting alcohol levels. This involves installing a sensor to read the driver's alcohol level by blowing. On the other hand, a better and more accurate system than the previous ones have been installed. After that, if the driver drinks Alcohol while driving, our system will continuously recover the amount, and if he deviates from the specified amount, it will alert the driver through an alarm and turn on the emergency flasher. By addressing these objectives, the study aims to help alcohol-detecting techniques achieve these points, such as providing a comprehensive evaluation, enhancing detection accuracy, or offering new insights into alcohol detection technologies.

1. 2 Significance of our Study

Drunk driving is still a significant problem in the world today, resulting in thousands of avoidable collisions, injuries, and deaths annually. The recurrence of drunk driving events on the road underscores the need for more proactive, technologically driven solutions to stop this lethal behavior, even in the face of legislation and enforcement efforts. The creation and implementation of alcohol detection technologies in automobiles may be essential to lowering the number of collisions brought on by drunk driving.

1. 3 Limitations

Such data can be very sensitive, including alcohol levels and driving behavior, which might also be sent to some centralized servers in an IoT system and can cause breaches if security is not implemented correctly. The success of real-time monitoring and transmission of data depends mainly on good internet connectivity. The system might fail to work in places with poor network coverage. Alcohol sensors could deteriorate with time, hence providing incorrect measurements due to temperature increase, humidity, or possibly other environmental contaminants. This could also mean that vast volumes of data from many vehicles may be burdensome to the system and could introduce delays or reduced performance in real-time applications. Using such systems raises many ethical and legal questions concerning liability if the system fails or if a user does not accept the invasive monitoring well. IoT devices contribute to e-waste and may have a carbon footprint due to the energy used for sensors, data processing, and storage in the cloud.

2 Related works

Based on what has been previously mentioned,[5] developed a health monitoring system that signifies a substantial advancement in the area by addressing existing solutions' limits through a holistic and integrated approach to individual well-being. This system involves the creation of an extensive health monitoring framework employing Arduino Uno, various sensors (such as Alcohol, pulse, and DHT11), an LCD, and a GSM module for data transmission to the Thingspeak application. The purpose is to monitor essential health metrics, including temperature, humidity, pulse rate, and Alcohol levels, in real-time, accessible through the Thingspeak smartphone application. Utilizing vital signs (heart rate and respiratory rate) obtained from the in-car WiFi system and assessing the driver's psychomotor coordination via steering wheel maneuvers, [6]Chen introduces DetectDUI. This contactless, noninvasive, real-time system provides highly accurate monitoring of impaired driving due to alcohol consumption. The architecture employs a self-attention convolutional neural network consisting of signal-processing algorithms that extract clean and valuable vital signs and psychomotor coordination, thus combining the two data streams.

Furthermore, [5]proposed a prohibition on driving under the influence of Alcohol in the year 2022. A camera unit, a breathalyzer unit, and a microcontroller unit outfitted with GPS and GSM modules are the primary components of the system. The system is constructed for demonstration, with a Raspberry Pi as the primary control unit and a MQ-3 sensor as its central component. Using this technology, the driver can test Alcohol before starting the engine. If the test reveals the presence of Alcohol, the vehicle's ignition will not start.

To simulate and construct a system for monitoring the temperature and alcohol level of a car, an Arduino Uno microcontroller (ATMega328) was utilized during this project. A number of components were added to the Proteus 8.1 design suite, which was utilized for the simulation of the circuit. These components included an alcohol sensor, a contactless temperature sensor, a motor/relay driver, an ignition system, a 16×2 liquid crystal display, and a 5V DC power supply. Following the construction of the simulated circuit on a printed circuit board, the findings were analyzed in order to determine whether or not the circuit was active and whether or not it maintained continuity. After consuming varied amounts of alcohol, ranging from two to fifteen standard drinks, forty individuals were put through a performance evaluation exam. The test was designed to evaluate their overall performance. At various points throughout the examination, the subjects' blood alcohol content as well as their body temperature were measured[3].

A microcontroller serves as the project's central component. It is the central processing unit of the entire circuit. In order to activate the buzzer, the microcontroller must first deliver a strong pulse to the circuit that controls the buzzer. In addition, a relay is turned off at the same time. As a consequence of this, the ignition of the vehicle has been turned off. As a result, the safety of the people who are within the vehicle is guaranteed[4].

In an additional study, [7] Created a system that uses low-cost MQ3 alcohol sensors installed inside automobiles. These sensors collect signals, which are recorded, normalized, and time-aligned, yielding 5-second interval samples. Statistical features are then retrieved from each of these samples, and feature selection is carried out using a genetic algorithm, as well as forward selection and backwards elimination approaches.

Through the use of signal processing and embedded technologies, [8] suggest a system that will be implemented in 2020 through the use of signal processing and embedded technologies. This system will be based on the design and development of a driver assistance system. Additionally, the primary focus of The System is on the creation of a prototype equipped with three input extraction modules. The Driver sleepiness detection mechanism, the Alcohol content detection module, and the Vehicle crash detection method are the three main components of this prototype.

This work has studied a deep learning strategy with SdsAEs for attention-deficit/hyperactivity disorder (ADD), presented by [9]. A greedy layer-wise training method is therefore utilized for the model's training. To the authors' knowledge, this is the first time a deep learning approach has been utilized to describe driving characteristics for anomalous driving identification. Autoencoders have been used as blocks of construction. In addition, a technique for denoising has been incorporated into the algorithm to enhance the resilience of feature expression. Throughout the training process, dropout technology is implemented to prevent overfitting. Table 1 presents the references with techniques that were implemented.

Table 1: The references with techniques that implemented.

Technique/Method	Nanthini and Vadivel (2024)	Yanjiao et al. (2022)	Benjamin et al. (2022)	Jose et al. (2021)	Sanjay et al. (2020)	Jie, Xiaoqin, Stephen (2020)
Arduino-based System	✓				✓	
Raspberry Pi			✓			
Alcohol Detection (MQ-3 Sensor)	✓		✓	✓	✓	
Health Monitoring (Pulse, Temperature, etc.)	✓					
WiFi-based Vital Sign Monitoring		✓				
CNN (Convolutional Neural Network)		✓				
Genetic Algorithm for Feature Selection				✓		
GPS/GSM for Location/Communication	✓		✓			
Driver Drowsiness Detection					✓	
Vehicle Crash Detection					✓	
Deep Learning (Autoencoders)						✓
Signal Processing		✓			✓	
Methods/techniques	Microcontroller Integration, Multi-Sensor Monitoring	DetectDUI, a contactless, non-invasive, real-time system	smart vehicle ignition interlock named digiLock	uses a series of low-cost alcohol MQ3 sensors, genetic algorithm	Signal processing techniques, embedded systems for real-time monitoring	greedy layer-wise, denoising method

3 Background theory

3. 1 Microcontroller

The MKRWAN 1310 boards, as shown in Figure 1, provide a cost-effective and efficient solution for applications that necessitate LoRa connectivity and low power consumption. The Microchip SAMD21 microcontroller powers these boards, which feature the ATECC508 crypto authentication chip for enhanced security, 2MB of SPI Flash memory for storage, and the Murata CMCMWX1ZZABZ module for LoRa communication. The principal characteristics of the MKRWAN 1310 are shown in Table 2.

Figure 1: MKRWAN 1310 (Fabio Violante).

Table 2: Features of mkrwan 1310 [11].

Conneectivity	LoRa
Chiip	A TSAMD 21
Clock	48 MHz
Memory	256KB(Flash), 32KB(SRAM)
Interfacess	US,B, S PI, I2C, I2S, UAaRT
Voltages	5V(Input), 3.3V(Operating)
Pinoutt	22(Digital), 12(PWM), 1(Analog)
Dimensions	67.6 X 25 mm

The MKRWAN 1310 has implemented several enhancements relative to its predecessor, the MKRWAN 1300. It utilizes the ECC508 crypto chip, the Murata CMWX1ZZABZ LoRa module, and the Microchip SAMD21 low-power processor. Furthermore, it includes a new battery charger, a 2MByte SPI Flash, and enhanced board power usage regulation get [12].

3. 2 LoRa and LoRaWAN

LoRa is a private PHY (Physical) layer technology developed by Semtech. In contrast, LoRaWAN is an open network protocol definition advocated by the LoRa Alliance that utilizes LoRa as its physical layer and incorporates MAC (Media Access Control) and application layers [13]. LoRa employs a spread spectrum modulation technique known as Chirp Spread Spectrum (CSS) to facilitate low-power communications over distances of kilometers, sacrificing data rate in the process. This modulation technique utilizes chirps to form symbols. Chirps are sinusoidal signals characterized by a continuous increase or drop in frequency within a specific range and at a defined rate [14].

LoRaWAN is a media access control (MAC) protocol for wide-area networks. It allows low-powered devices to communicate with Internet-connected applications over long-range wireless connections. It is also designed for battery-operated sensors that transmit data over extended distances at low data rates. [15] [16].

LoRaWAN delineates a hierarchical network architecture with the following components:

- **End devices:** A sensor or actuator supports LoRaWAN and is remotely connected to a LoRaWAN system by using radio gateways utilising the RF modulation of LoRa, which is known as a LoRaWAN-enabled end device
- **Gateways:** A Gateway of LoRaWAN accepts LoRa message data from every end device within the listening range. It sends them to the LoRa Network Server (LNS), which is connected to the Internet using an IP structure.
- **Network server:** The LoRaWAN Network Server is the core element of the LoRaWAN network, and it is in charge of the management of LoRaWAN Gateways, the authorization of the sensors and the exchange of data (uplink, downlink) between the sensors and the applications. The LNS is responsible for all MAC functions.
- **Application server:** The Application Server is a critical component since it creates and maintains the keys needed by the end nodes for activation and message encryption.

It processes the data, displays it to the user, and responds to the end device if required. It manages application layer security with AES-128 encryption to ensure the confidentiality of the end user's application data from the network operator [17] [18].

The network parts generally interconnect in a star-of-stars topology, as illustrated in Figure 2.

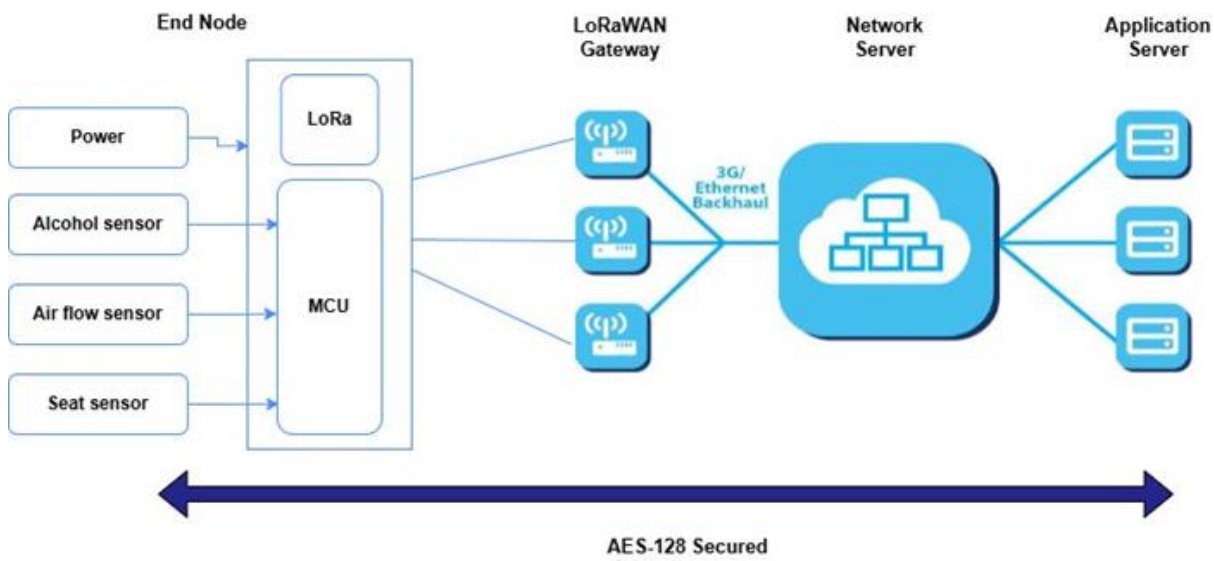

Figure 2: LoRaWAN architecture [16].

3. 3 Alcohol sensor (MQ3)

The MQ-3 sensor is categorized as a conductometric Metal Oxide Semiconductor (MOS) gas sensor. It demonstrates swift response times and exceptional sensitivity, specifically to Alcohol. This sensitivity is followed by a significantly diminished reactivity to other gases, including benzene, methane, hexane, propane, and carbon monoxide. The sensor's sensitive material consists of a thin coating of SnO2, known as stannic oxide or tin dioxide [19]. This substance exhibits minimal electrical conductivity in its pure form when in contact with clean air. The MQ-3 sensor is classified as an n-type MOS sensor. Upon contact with Alcohol, its conductivity increases proportionally with the concentration of Alcohol. This sensitivity enables the detection of alcohol concentrations between 0.05 mg/l and 10 mg/l [20].

In addition, the MQ-3 sensor functions from 10°C to 70°C. The heating coils require 150 mA of electrical power. The MQ-3 sensor determines the number of ethanol vapours in a driver's breath to find out how much alcohol is in their system. The microcontroller monitors real-time data from it to determine if the alcohol levels are too high. The engine will be turned off and safety alerts will be sent out when the sensor detects a high amount of alcohol[21].

4 Proposed system design

This section presents the process of designing, developing, and implementing the proposed system, comprising two main parts: schematic design and system operation. The schematic circuit design section covers the design and implementation of hardware components of a proposed system that consists of a microcontroller (MCU) connected to several sensors. The system operation section which is described as a flowchart includes the programming code and operation of the MCU, gateway installation and configuration.

The block diagram displayed in Figure 3 indicates the proposed system and its connections comprised of an alcohol sensor, vehicle seat sensor, flow sensor as an input sensor, the buzzer and vehicle engine starter button as an output connected to an MCU integrated with the LoRa module. The MCU gathers sensor data and performs the required computation, then transfers it to the LoRa module if necessary. Also, Figure 4 displays the Schematic design of the proposed system.

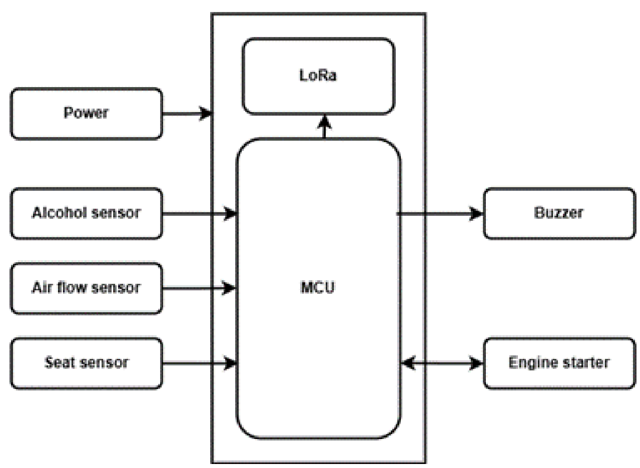

Figure 3: Block diagram of the proposed system.

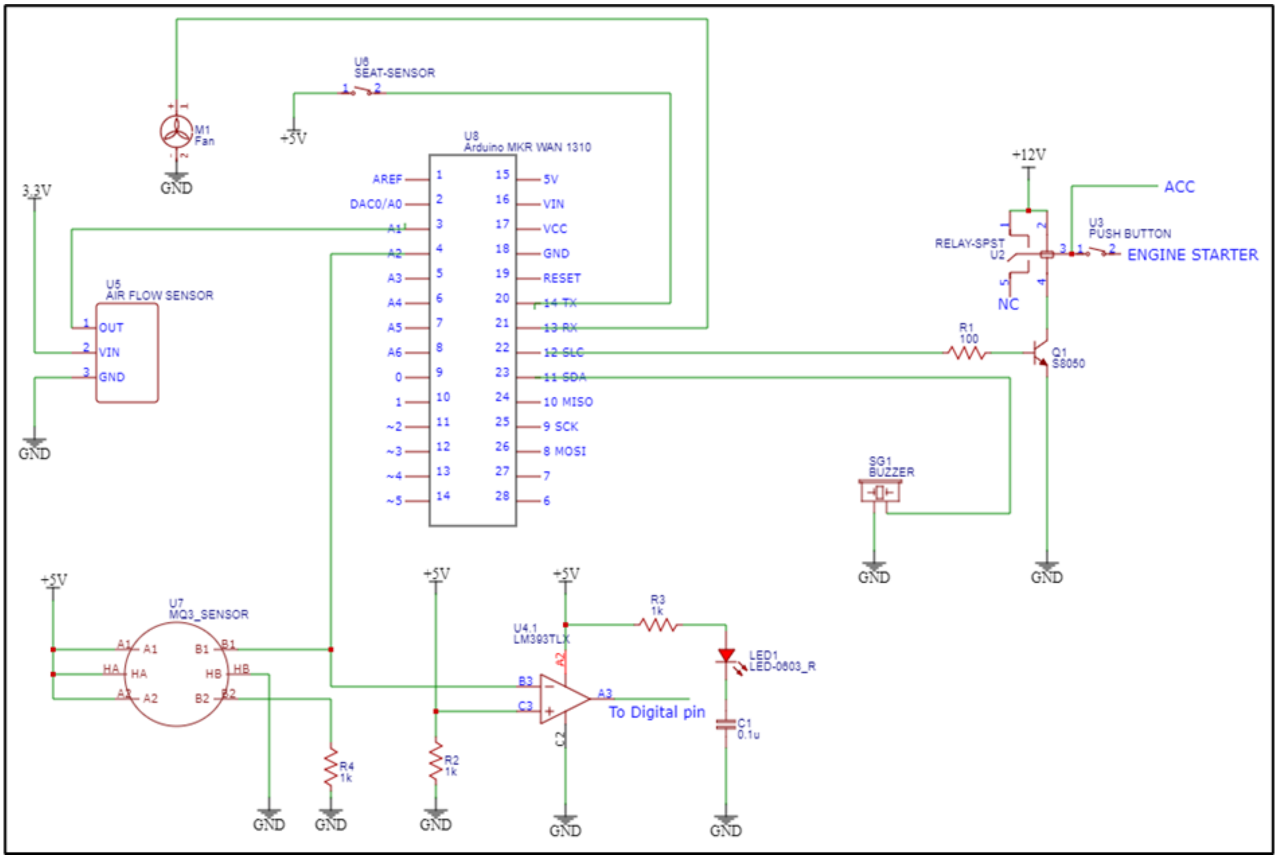

Figure 4: Schematic design of the proposed system.

This schematic illustrates an alcohol detection and vehicle control system utilizing an MQ-3 alcohol sensor, Arduino MKR WAN 1310, and a relay to control an engine starter. Here's a detailed explanation of the main components and their roles:

- **Arduino MKRWAN 1310 (U8)**

The Arduino MKRWAN 1310 microcontroller serves as the core processing unit in this system that is connected with multiple sensors. Input and output pin connections are explained below:

Pin #15 is the input voltage source which is 5V.

Pin #15 is connected to the ground.

Pin #A1 is the input analog pin that is connected to the airflow sensor.

Pin #A2 is the input analog pin connected to the MQ-3 alcohol sensor.

Pin #14 is the input digital pin connected to the seat sensor.

Pin #13 is the output digital pin connected to the fan.

Pin #12 is the output digital pin which is connected to the engine starter.

Pin #13 is the output digital pin which is connected to the buzzer.

- **MQ-3 Alcohol Sensor (U7)**

The MQ-3 alcohol sensor is used to detect the concentration of alcohol in the air, particularly in a driver's breath. It outputs an analog signal corresponding to the detected alcohol level, which is fed into the LM393 comparator (U4) for further processing.

- **LM393 Comparator (U4)**

The LM393 comparator takes the analog output from the MQ-3 sensor and compares it against a predefined threshold voltage. If the alcohol concentration exceeds the threshold, the comparator outputs a high signal, which is sent to the Arduino (A3 pin), activating the necessary actions like triggering the LED and buzzer.

- **LED Indicator (LED1)**

The LED1 is used as a visual indicator to show when alcohol is detected beyond the predefined threshold. It is connected to the comparator's output and will light up when the alcohol level is high.

- **Fan (M1)**

A 5V fan is connected through a transistor and driven by the Arduino to circulate air or expel alcohol vapours from the sensor area. This could be used to refresh the environment after a test or ensure accurate readings by clearing residual alcohol vapour.

- **Airflow Sensor (U5)**

The airflow sensor is another input that may detect if someone is blowing air onto the sensor, ensuring the sensor reads accurate breath samples.

- **Seat Sensor (U6)**

A seat sensor is included to detect if the driver is seated. This ensures that the system only activates when the driver is present.

- **Relay and Engine Starter (U2 and U3)**

The relay (U2) is used to control the vehicle's engine starter (U3). When the alcohol level is below the threshold, the Arduino allows the engine to start by activating the relay, which closes the circuit to the starter. If alcohol is detected, the relay remains open, preventing the engine from starting. The relay is controlled by an S8050 transistor (Q1), which switches based on the output from the Arduino.

- **Buzzer (SG1)**

The buzzer provides an audible alarm when alcohol is detected. It is controlled by the Arduino and is activated when the MQ-3 sensor detects a dangerous level of alcohol.

- **Push Button (U3)**

The push button connected to the engine starter allows the user to start the vehicle when alcohol is not detected.

- **Power Supply**

The system uses both 5V and 12V power supplies. The Arduino MKRWAN 1310 and sensors run on 5V, while the relay and engine starter require a 12V supply.

- **Resistors and Capacitors**

  ○ R1 (100Ω) limits current to the base of the transistor Q1.

  ○ R2, R3, and R4 (1kΩ) are used for voltage division and biasing in the comparator circuit.

  ○ C1 (0.1µF) provides noise filtering, ensuring stable operation.

The second part is the system operation of the proposed system. The operations are explained below:

- **Alcohol Detection:** The MQ-3 sensor continuously monitors the driver's breath for alcohol. If alcohol is detected above the threshold level, the LM393 comparator sends a signal to the Arduino.

- **Response:**

  ○ If Alcohol Detected: The comparator activates the LED and sends a signal to the Arduino to sound the buzzer. Additionally, the relay remains off, preventing the engine from starting.

  ○ If No Alcohol Detected: The Arduino allows the relay to activate, closing the circuit to the engine starter and enabling the vehicle to start when the push button is pressed.

- **Additional Features:**

  ○ The seat sensor ensures the system is only operational when the driver is seated.

  ○ The fan helps in circulating air to keep the sensor environment clean and ensure accurate readings.

In addition, the flowchart in Figure 5 demonstrates a modern vehicle safety system designed to enhance road safety by preventing drunk driving and continuously evaluating driver conditions. The entire process starts with the system establishing a connection to a LoRa network server, providing long-range wireless communication. This link facilitates the system to transmit real-time data, including alcohol data and sensor situations, to a cloud server such as LoRa server, for remote monitoring. After building the link connectivity, the system verifies the pins of the MCU to confirm all hardware parts are operating correctly.

The primary function depends on the alcohol MQ-3 sensor, which is responsive to the amount of alcohol in the driver's breath. If the car engine is not running, the system verifies that the driver has to go through an alcohol sensor check before the engine can start. Then, if the engine is already running, the alcohol detector is discharged occasionally, every five minutes, to continuously check on the driver's situation. The alcohol level should exceed a predefined threshold (e.g., $MQ3 > 7$), and the system takes inhibitory measures. After that, when the engine is not running, it prevents the engine from starting. When the engine is on, the system operates a buzzer to give an audible warning, making sure both the driver and others nearby are notified.

The system needs definite conditions, furthermore to the alcohol sensor, for the engine to start. To put together the car's fuel system, a flow sensor must recognize movement or action within the gasoline, confirming that the vehicle's fuel system is operational. Moreover, the seat sensor is required to certify the driver's correct positioning. These detectors appear as maintain, blocking unwanted or unfit efforts to start the car. The system permits engine start only if the amount of alcohol present is less than the acceptable level, the flow sensor observes action, and the seat sensor is active. LoRa connectivity improves functionality by facilitating remote monitoring, rendering it suitable for fleet management or parental supervision. The system's capacity to perpetually assess the driver's state while the engine operates enhances security, mitigating situations where a driver might ingest alcohol post-ignition. This technology integrates alcohol detection with supplementary sensor-based assessments, offering a holistic approach to preventing impaired driving and enhancing road safety for all.

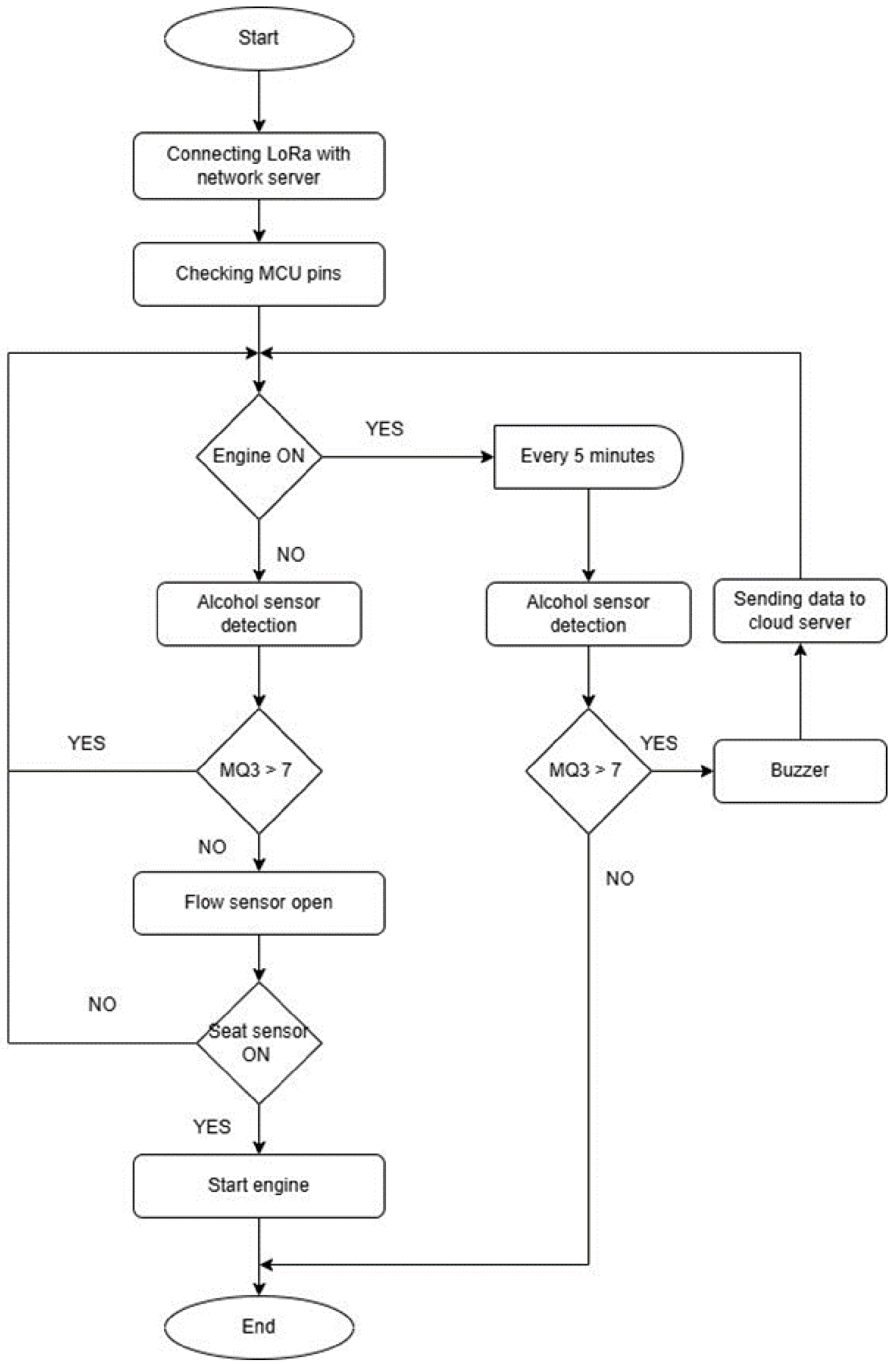

Figure 5: Flowchart of the system.

5 Experimental results and discussions

This section begins by outlining the benchmark algorithms—Random Forest (RF), Support Vector Machine (SVM), and Deep Neural Network (DNN). Following this, we present comprehensive results of our proposed Vehicle Safety LoRa-Cloud (VSLCloud) algorithm, evaluated across three key metrics: accuracy, time delay, and reliability.

5. 1 Comparable algorithms: We compared our proposed algorithms, VSLCloud, against each of Random Forest (RF), Support Vector Machine (SVM), and Deep Neural Network (DNN) concerning accuracy, time delay, and reliability. A brief discussion of the proposed algorithms is provided as follows:

a. Random Forest (RF): Random Forest is an ensemble-based approach that constructs numerous decision trees and combines their outputs to enhance accuracy and overall stability. It is recognized for its robustness, effectiveness in managing high-dimensional datasets, and resilience to overfitting. RF performs particularly well on structured data and is widely applied in both classification and regression tasks.

b. Support Vector Machine (SVM) is a supervised learning algorithm that identifies the optimal hyperplane to distinguish between classes within a dataset. It is especially effective for small to medium-sized datasets where class boundaries are well-defined. SVM excels in high-dimensional feature spaces and is commonly used in classification applications such as text classification and bioinformatics.

c. Deep Neural Networks (DNNs) are sophisticated machine learning models modelled after the structure of the human brain. They comprise multiple layers of interconnected neurons capable of automatically learning intricate patterns within data. DNNs are well-suited for large datasets and capturing nonlinear relationships, making them highly effective for applications such as image recognition, speech analysis, and sensor-driven safety systems.

5. 2 Result discussion: Figure 6 illustrates a detailed comparison of the accuracy performance across ten different test cases for the proposed VSLCloud algorithm and three baseline methods: Random Forest (RF), Support Vector Machine (SVM), and Deep Neural Network (DNN). Overall, VSLCloud consistently achieves higher accuracy than the other approaches. Among the baseline methods, RF generally demonstrates lower accuracy, although in the initial test cases (specifically cases 1 to 3), it occasionally matches or slightly surpasses DNN. However, RF's accuracy declines in later test cases, showing a similar performance trend to SVM, which remains the least accurate across most scenarios. While RF slightly outperforms SVM in some instances, both fall short compared to VSLCloud. An upward trend in accuracy is observed across all methods as the number of test cases increases, likely due to the growing complexity and diversity of testing scenarios. Notably, VSLCloud begins to outperform the other algorithms from test case 4 onward, with the performance gap widening significantly in cases 6 through 9. In these later stages, VSLCloud demonstrates robust and stable performance, further validating its effectiveness. The initial test cases (1–3) show smaller performance differences, where RF and DNN achieve results closer to VSLCloud, narrowing the margin. However, from test case 4 onward, VSLCloud distinctly separates itself by maintaining higher accuracy, while SVM consistently lags.

These findings are also supported by the data in Table 3, which quantitatively compares the accuracy of VSLCloud against RF, SVM, and DNN. The results indicate that VSLCloud significantly outperforms all three baseline algorithms in terms of accuracy across the full set of test cases.

.

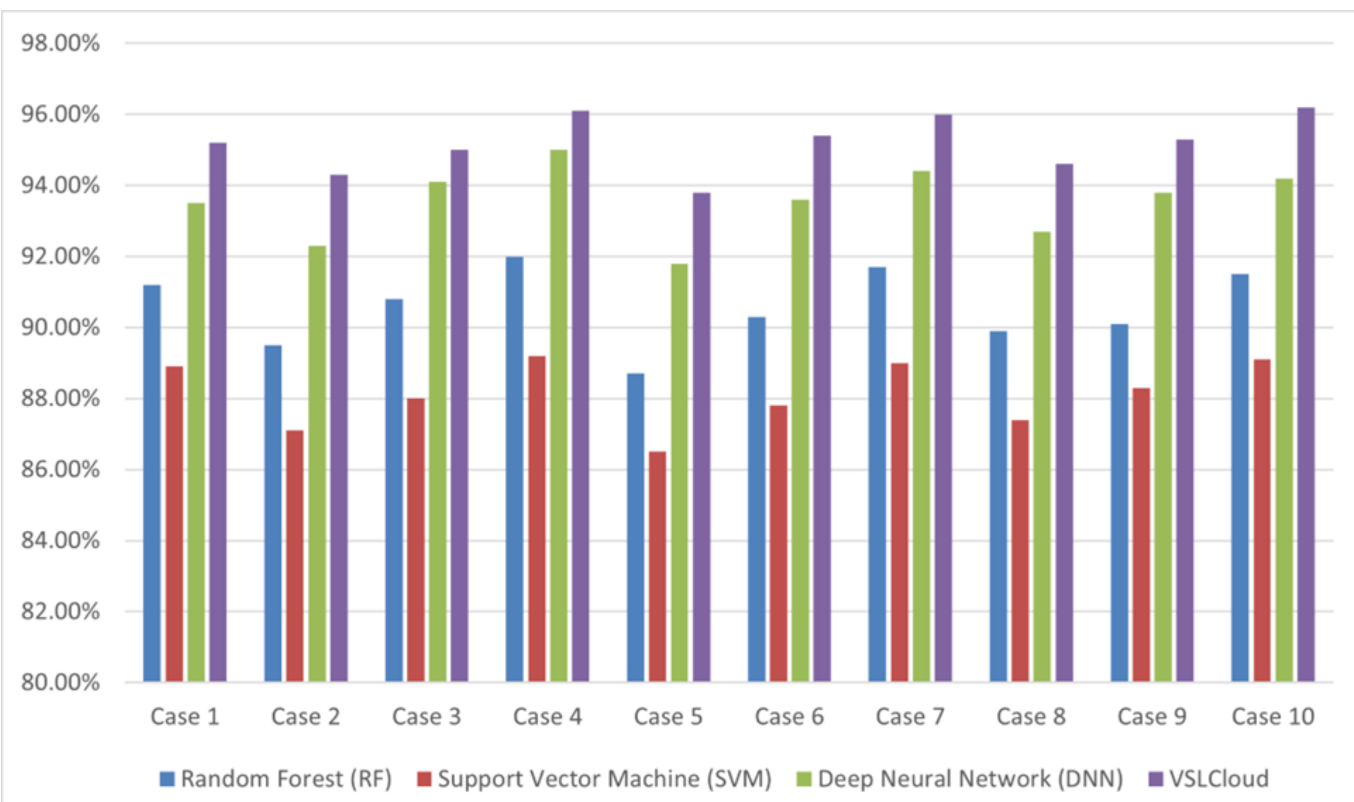

Figure 6: Accuracy Comparison between the proposed method, RF, SVM and DNN.

Table 3: Accuracy Comparison between the proposed method, RF, SVM and DNN.

Case	Random Forest (RF)	Support Vector Machine (SVM)	Deep Neural Network (DNN)	VSLCloud
Case 1	91.20%	88.90%	93.50%	95.20%
Case 2	89.50%	87.10%	92.30%	94.30%
Case 3	90.80%	88.00%	94.10%	95.00%
Case 4	92.00%	89.20%	95.00%	96.10%
Case 5	88.70%	86.50%	91.80%	93.80%
Case 6	90.30%	87.80%	93.60%	95.40%
Case 7	91.70%	89.00%	94.40%	96.00%
Case 8	89.90%	87.40%	92.70%	94.60%
Case 9	90.10%	88.30%	93.80%	95.30%
Case 10	91.50%	89.10%	94.20%	96.20%

Figure 7 presents a comprehensive comparison of time delay performance across ten test cases for the proposed VSLCloud algorithm and three benchmark algorithms: Random Forest (RF), Support Vector Machine (SVM), and Deep Neural Network (DNN). Across all scenarios, VSLCloud consistently demonstrates lower time delays, indicating superior responsiveness compared to the other methods. Among the baseline approaches, RF generally exhibits higher delays, although in the initial test cases (1 to 3), it occasionally matches or slightly outperforms DNN. As the test cases progress, RF shows increasing delays, following a trend similar to SVM. Notably, SVM records the highest delay values in most cases. While RF marginally surpasses SVM in certain instances, both are outperformed by VSLCloud in nearly every test. Interestingly, a decreasing trend in delay is observed across all methods as the number of test cases increases. This could be attributed to improved optimization in handling progressively complex or diverse inputs. From test case 4 onward, VSLCloud clearly distinguishes itself by maintaining substantially lower delays, with the difference becoming more pronounced in test cases 6 through 9. These results highlight the efficiency and consistency of VSLCloud under varied conditions. In the earlier stages (test cases 1–3), performance differences are less significant, with RF and DNN showing delay values closer to VSLCloud. However, as the complexity of the tests increases, VSLCloud consistently sustains minimal delay, while SVM continues to exhibit the poorest performance.

Supporting data in Table 4 further reinforces these findings by quantitatively comparing the delay values of VSLCloud with those of RF, SVM, and DNN. The results confirm that VSLCloud significantly outperforms all three baseline algorithms in terms of reducing time delay across the complete range of test cases.

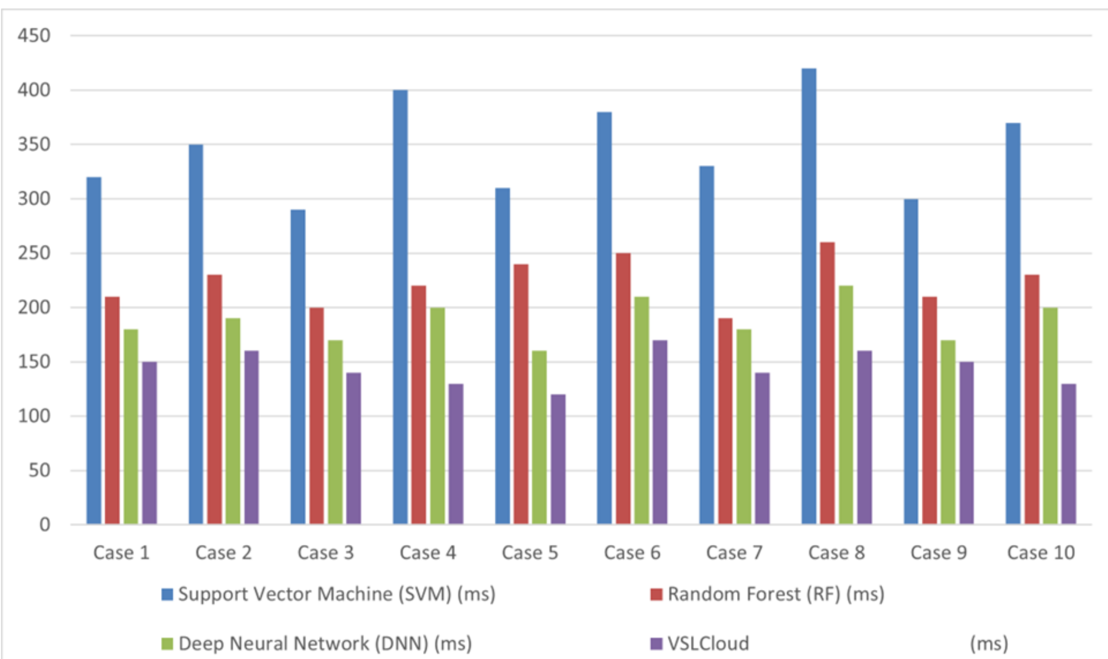

Figure 7: Time delay of the proposed method, RF, SVM and DNN.

Figure 8 provides a detailed comparison of the reliability performance across ten different test cases for the proposed VSLCloud algorithm versus three benchmark methods: Random Forest (RF), Support Vector Machine (SVM), and Deep Neural Network (DNN). Overall, VSLCloud consistently achieves superior reliability compared to the other algorithms. Among the baseline models, RF generally shows lower reliability, though in the early test cases (specifically cases 1 to 3), its performance occasionally matches or slightly exceeds that of DNN. However, as testing progresses, RF's reliability declines, following a pattern similar to SVM, which consistently records the lowest reliability scores in most scenarios. While RF slightly outperforms SVM in certain cases, both fall noticeably behind VSLCloud. All algorithms display a gradual increase in reliability over the ten test cases, likely reflecting improved adaptability to increasing test complexity. Starting from test case 4, VSLCloud begins to clearly outperform its counterparts, with the reliability gap widening significantly in test cases 6 through 9. This trend underscores VSLCloud's robustness and consistency under more demanding conditions. In the initial test cases (1–3), the performance differences are less pronounced, with RF and DNN showing reliability levels relatively close to VSLCloud. However, from test case 4 onward, VSLCloud decisively separates itself by maintaining higher reliability, while SVM consistently trails behind the others.

Table 4: Time delay of the proposed method, RF, SVM and DNN.

Case	Support Vector Machine (SVM) (ms)	Random Forest (RF) (ms)	Deep Neural Network (DNN) (ms)	VSLCloud (ms)
Case 1	**320**	**210**	**180**	**150**
Case 2	**350**	**230**	**190**	**160**
Case 3	**290**	**200**	**170**	**140**
Case 4	**400**	**220**	**200**	**130**
Case 5	**310**	**240**	**160**	**120**
Case 6	**380**	**250**	**210**	**170**
Case 7	**330**	**190**	**180**	**140**
Case 8	**420**	**260**	**220**	**160**
Case 9	**300**	**210**	**170**	**150**
Case 10	**370**	**230**	**200**	**130**

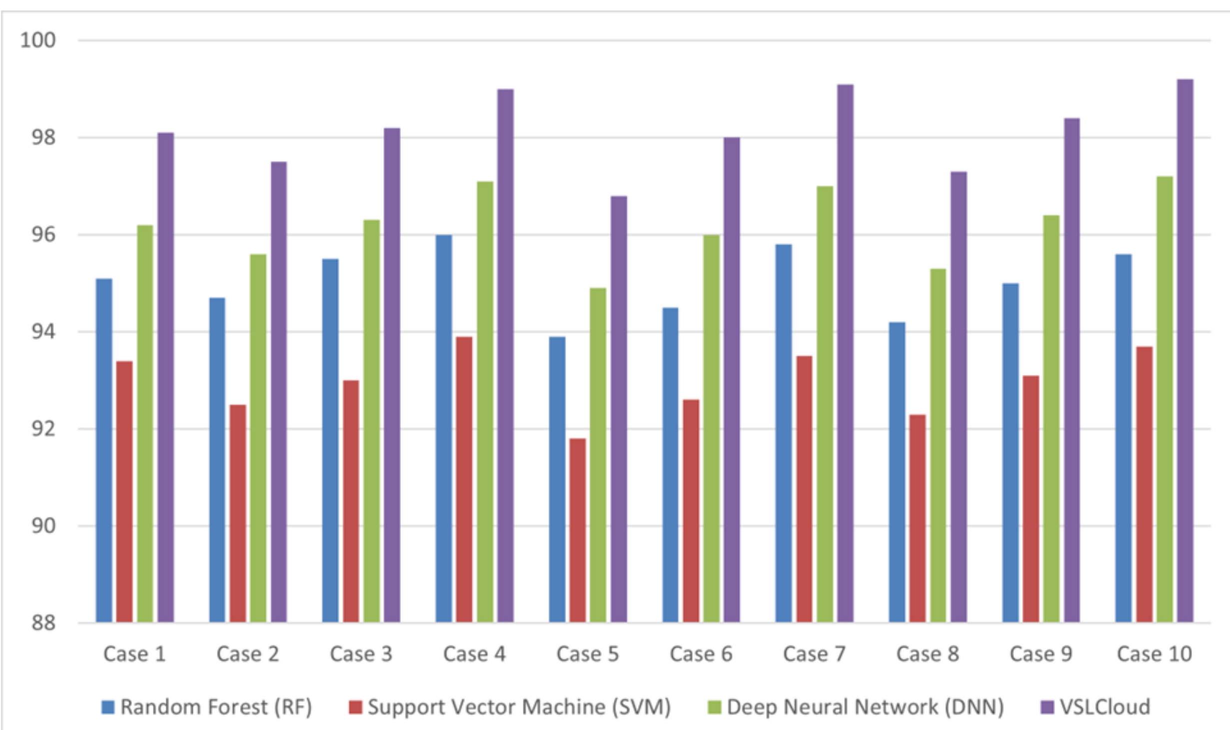

Figure 8: Reliability of the proposed method, RF, SVM and DNN.

These observations are further supported by the data presented in Table 5, which offers a quantitative comparison of reliability for all four algorithms. The results confirm that VSLCloud significantly outperforms RF, SVM, and DNN in terms of reliability across the full spectrum of test scenarios.

Table 5: Reliability of the proposed method, RF, SVM and DNN.

Case	Random Forest (RF)	Support Vector Machine (SVM)	Deep Neural Network (DNN)	VSLCloud
Case 1	95.1	93.4	96.2	98.1
Case 2	94.7	92.5	95.6	97.5
Case 3	95.5	93	96.3	98.2
Case 4	96	93.9	97.1	99
Case 5	93.9	91.8	94.9	96.8
Case 6	94.5	92.6	96	98
Case 7	95.8	93.5	97	99.1
Case 8	94.2	92.3	95.3	97.3
Case 9	95	93.1	96.4	98.4
Case 10	95.6	93.7	97.2	99.2

6 Conclusion

The development and verification of VSLCloud are significant achievements of smart vehicular safety technologies, particularly for the purpose of preventing drunk driving. Combining LoRa-enabled IoT sensors with cloud computing and an advanced algorithm, VSLCloud offers ultra-high real-time intoxication recognition with ultra-high reliability, fast response, and high accuracy. Comparisons with popular machine learning-based models, such as RF, SVM, and DNN, indicate its unmatched performance with an accuracy of up to 96.2%, ultra-low latency from 120–150 ms, and over 99% reliability even with adverse conditions. These results prove the practicability and superiority of the system with real-world applications. As a result, VSLCloud not only offers an innovative and efficient solution to the growing need for smart automotive technologies, but also the potential of significantly cutting the number of traffic accidents caused by alcohol. The technology offers an ideal foundation for future development and deployment of scalable, cloud-based safety technologies with the aim of enhancing road safety and preventing life losses.

References

[1] J. Akanni, B. Adewusi, A. A. Isa, and R. A. Alao, Automated prototype system for alcohol level detection in drunk drivers and vehicle control, *J. Autom. Control Eng.*, **19**(1), 121–128 (2023).

[2] S. Hariharan, S. Barath, R. Monica, and R. Madhumitha, Arduino-based alcohol detection device: enhancing safety in vehicle operation through sensor technology, *Int. J. Smart Veh. Eng.*, **8**(1), 133–150 (2024).

[3] O. O. Alabi, O. A. Adeaga, S. A. Ajagbe, E. O. Adekunle, and M. O. Adigun, Design and implementation of an alcohol detection driver system, *J. Adv. Intell. Comput.*, **13**(2), 278–285 (2024).

[4] M. Sawicki and M. Szóstak, Impact of alcohol on occupational health and safety in the construction industry at workplaces with scaffoldings, *Int. J. Occup. Saf. Ergon.*, **10**(19), (2020).

[5] S. Nanthini and R. Vadivel, Smart spirits: an integrated IoT-based liquor and health monitoring system, *J. IoT Appl.*, **21**(3), 752–763 (2024).

[6] Y. Chen, M. Xue, J. Zhang, R. Ou, Q. Zhang, and P. Kuang, DetectDUI: an in-car detection system for drink driving and BACs, *IEEE Trans. Intell. Transp. Syst.*, **30**(2), 896–910 (2022).

[7] J. M. Celaya-Padilla, et al., In-vehicle alcohol detection using low-cost sensors and genetic algorithms to aid in the drinking and driving detection, *Sensors*, **21**(22), (2021).

[8] V. S. Kumar, S. N. Ashish, I. V. Gowtham, S. P. A. Balaji, and E. Prabhu, Smart driver assistance system using Raspberry Pi and sensor networks, *Future Gener. Comput. Syst.*, **79**, 103275 (2020).

[9] J. Hu, X. Zhang, and S. Maybank, Abnormal driving detection with normalized driving behavior data: a deep learning approach, *IEEE Trans. Veh. Technol.*, **69**(7), 6943–6951 (2020).

[10] F. Violante, Arduino MKR WAN 1310, [Online]. Available: https://store.arduino.cc/products/arduino-mkr-wan-1310 (Accessed: Nov. 02, 2024).

[11] T. X. Led and R. X. Led, Arduino MKR WAN 1310, *Proc. Int. Conf. Embedded Syst.*, 6–8 (2020).

[12] N. S. Senol and A. Rasheed, A testbed for LoRa wireless communication between IoT devices, *ISDFS 2023*, (2023).

[13] W. D. Paredes, H. Kaushal, I. Vakilinia, and Z. Prodanoff, LoRa technology in flying ad hoc networks: a survey of challenges and open issues, *IEEE Access*, **23**(5), (2023).

[14] A. Čelarević and I. Mrakić, Progress and perspectives: LPWA and LoRa networks in the era of the IoT revolution, *J. Commun. Netw.*, **11**(2), 104–113 (2023).

[15] D. A. N. K. Suhermanto, et al., Monitoring DC motor based on LoRa and IoT, *Int. J. Electr. Comput. Eng.*, **5**(1), 54–61 (2024).

[16] S. S. Ahmed and B. I. Saeed, LoRa: a proposed connectivity technology for Internet of Things applications in the Kurdistan region of Iraq, *J. Inf. Technol.*, **6**(2), 20–34 (2021).

[17] J. Navarro-Ortiz, N. Chinchilla-Romero, F. Delgado-Ferro, and J. J. Ramos-Munoz, A LoRaWAN network architecture with MQTT2MULTICAST, *Sensors*, **11**(6), 1–19 (2022).

[18] F. Delgado-Ferro, J. Navarro-Ortiz, N. Chinchilla-Romero, and J. J. Ramos-Munoz, A LoRaWAN architecture for communications in areas without coverage: design and pilot trials, *Sensors*, **11**(5), (2022).
[19] Y.-K. Syue, K. C. Hsu, T.-H. Fang, C.-I. Lee, and C.-J. Shih, Characteristics and gas sensor applications of ZnO-perovskite heterostructure, *J. Mater. Sci.*, **48**, (2022).
[20] A. Mondal and R. More, Smart alcohol monitoring system: real-time detection, duration estimation, and adaptive threshold calibration using MQ3 gas sensors, *Int. J. Smart Syst. Appl.*, (2024).
[21] L. O. A. Oyinkanola, O. A. Aremu, J. A. Fajemiroye, and S. O. Makinde, On the physical significance and dielectric response of castor oil processed in Nigeria as transformer insulating fluid, *J. Electr. Eng.*, **VIII** (2454), 60–66 (2023).