



UNIVERSITÀ DEGLI STUDI DI MILANO
FACOLTÀ DI SCIENZE E TECNOLOGIE
Corso di Laurea in Informatica

**Black-box per flotte a noleggio:
monitoraggio/tracking veicoli,
segnalazione allarmi e profilazione
guida**

Relatore: Prof. Andrea Trentini

Correlatore: Dott. Alexjan Carraturo

Tesi di Laurea di: Alessandro Belfiore
Matr. 975826

Anno Accademico 2023/2024

Indice

Elenco delle figure	3
Elenco delle tabelle	4
1 Introduzione	5
1.1 Cos'è una scatola nera?	5
1.1.1 Utilità	5
1.1.2 Importanza	6
1.2 Obiettivi del progetto	7
2 Stato dell'arte	9
2.1 Tecnologie attuali di monitoraggio e tracking per veicoli	9
2.2 Gestione delle emergenze e sistemi di allarmi in uso	9
2.3 Soluzioni esistenti per la profilazione guidatori	9
3 Funzionalità del sistema	10
3.1 Monitoraggio	10
3.2 Tracking	10
3.3 Segnalazione di allarmi	11
3.3.1 Incidente	11
3.3.2 Assistenza	13
3.3.3 Rilevamento allarmi	13
3.4 Profilazione del comportamento del guidatore	16
4 Implementazione del sistema	20
4.1 Architettura	20
4.1.1 Componenti hardware utilizzati	20
4.1.2 Software e librerie utilizzate	20
4.2 Scelte progettuali	22
4.2.1 Filesystem	22
4.2.2 Sincronizzazione	24
4.3 Comunicazioni e trasmissione dati	24

4.3.1	CAN BUS (Control Area Network)	24
4.3.2	OBD scanner - esp32	27
4.3.3	esp32 - server remoto	27
5	Risultati ottenuti	28
5.1	Analisi dei risultati	28
5.2	Comportamenti non implementati	28
6	Conclusioni	30
6.1	Problemi aperti	30
6.2	Sviluppi futuri	30
6.2.1	Memoria	30
6.2.2	Profilazione conducente lato server remoto	32

Elenco delle figure

1.1	Esempio di scatola nera ¹	6
3.1	L'architettura del sistema: localizzazione GPS e modulo GSM ²	11
3.2	Schema situazione di emergenza dovuta ad un incidente ³	12
3.3	Diagramma di flusso per il monitoraggio dell'accelerometro ed eventuale invio SMS	13
3.4	Diagramma di flusso per il rilevamento di incidenti e invio SMS	14
3.5	Diagramma di flusso dei dati	17
3.6	Modello meccanico di un accelerometro a 3 assi	19
4.1	Schema di connessione dei vari componenti hardware della black box	21
4.2	Protocollo CAN BUS all'interno di un'autovettura	25
4.3	Pin dei cavi CAN sul connettore OBD-II	25
4.4	Formato dei pacchetti CAN standard	26
4.5	Esempio di scanner OBD-II utilizzato per il progetto	27
5.1	Modulo GSM da integrare	29
5.2	ESP32 SIM800L	29
6.1	Micro SD card	31
6.2	Esempio di classificazione binaria che stabilisce se una email è spam o no ⁴	33
6.3	K-means usato per la classificazione di 3 partizioni ⁵	35
6.4	Esempio di dendogramma ⁶	36
6.5	Esempio di dendogramma con tagli gerarchici ⁷	36
6.6	Esempio DBSCAN ⁸	37
6.7	Diagramma di flusso del sistema di profilazione conducente	37

Elenco delle tavelle

3.1	Descrizione degli assi dell'accelerometro utilizzati nell'ambito della profilazione del guidatore	18
3.2	Tabella di valutazione del conducente	19
4.1	Tabella "t1" dati relativi al monitoraggio del veicolo	22
4.2	Tabella "t2" tempo di campionamento	22
4.3	Tabella "t3" profilazione guida	23
4.4	Tabella "t4" allarmi critici	23
4.5	Tabella "t5" allarmi meno critici	23

Capitolo 1

Introduzione

Il settore assicurativo è stato uno dei primi a intuire le potenzialità di un tracciamento GPS in grado di registrare gli spostamenti di un determinato automobilista e di aiutarlo in caso di incidente segnalando ai soccorsi la posizione esatta dell'ac caduto, o in caso di furto segnalando alle forze dell'ordine la posizione dell'auto in tempo reale. Tutto questo è stato possibile grazie all'introduzione della cosiddetta "scatola nera" o black-box, un piccolo apparecchio da installare sull'auto di un assicurato, in grado di registrare tutti gli spostamenti di un'automobile, nonchè il comportamento alla guida del conducente. Può essere utilizzata dalla compagnia assicuratrice anche per profilare ulteriormente i clienti e offrire quindi delle polizze ulteriormente personalizzate e fatte su misura per il tipo di utilizzo [1].

1.1 Cos'è una scatola nera?

La black-box è un dispositivo elettronico utilizzato principalmente per monitorare e registrare vari parametri di funzionamento del veicolo e comportamento del conducente. Sebbene il termine sia spesso associato alle registrazioni di volo degli aerei, nel contesto automobilistico e assicurativo, la scatola nera ha una funzione leggermente diversa ma altrettanto cruciale. Ecco una descrizione dettagliata nei due paragrafi che seguono:

1.1.1 Utilità

- **Tracciamento GPS:** regista i movimenti del veicolo, inclusi i percorsi effettuati, le velocità raggiunte e le soste. Questo aiuta a localizzare il veicolo in caso di furto e a fornire assistenza immediata in caso di emergenza [2].

¹Preso da: <https://www.drivingtesttips.biz/wp-content/uploads/2016/01/car-insurance-black-box.jpg>



Figura 1.1: Esempio di scatola nera¹

- **Monitoraggio del comportamento di guida:** registra il comportamento del conducente, come accelerazioni brusche, frenate improvvise, velocità eccessive e svolte pericolose. Questi dati aiutano a valutare lo stile di guida e possono essere utilizzati per migliorare la sicurezza. [3]
- **Supporto in caso di incidenti:** fornisce dati cruciali sugli incidenti, come la velocità al momento dell'impatto e la posizione esatta dell'incidente. Queste informazioni possono essere molto importanti per le indagini post-incidente e per la richiesta di soccorsi. [4]

1.1.2 Importanza

L'importanza di una black box è data dai benefici che si possono trarre da essa:

- **Assistenza rapida:** in caso di incidente, la scatola nera può inviare automaticamente la posizione ai servizi di emergenza, riducendo i tempi di risposta.
- **Recupero dei veicoli rubati:** la localizzazione GPS permette di tracciare e recuperare rapidamente veicoli rubati.
- **Polizze assicurative personalizzate:** le compagnie assicuratrici possono utilizzare i dati registrati per profilare meglio i clienti e offrire polizze personalizzate basate sul comportamento di guida e sull'uso del veicolo.

-
- **Miglioramento della sicurezza:** i dati raccolti possono essere utilizzati per educare i conducenti su comportamenti di guida sicuri, contribuendo a ridurre il rischio di incidenti.

1.2 Obiettivi del progetto

Il principale obiettivo di questo progetto è sviluppare una black box avanzata per veicoli appartenenti ad una flotta di autonoleggio a lungo termine, che si distingua dalle soluzioni tradizionali per la sua capacità di diagnosticare sia il veicolo sia il comportamento del conducente. Le tradizionali black box non solo registrano i dati di viaggio e monitorano le condizioni del veicolo, ma svolgono anche funzioni fondamentali come il monitoraggio costante del veicolo, il tracciamento degli incidenti e la richiesta automatica di assistenza in caso di emergenza. La nostra black box, oltre a queste funzionalità di base, introduce due caratteristiche innovative che ne ampliano ulteriormente le potenzialità.

La **prima caratteristica distintiva** della nostra black box è la capacità di rilevare e segnalare tempestivamente sia allarmi critici che non, relativi a problemi meccanici del veicolo e comportamenti inappropriati del conducente. Questo sistema avanzato di diagnostica è in grado di identificare malfunzionamenti interni dell'auto vettura, come anomalie nel motore, nonché situazioni di pericolo come la presenza di gas nell'abitacolo. Inoltre, è capace di monitorare e segnalare comportamenti non leciti del conducente, la presenza di fumo nell'abitacolo potrebbe essere un parametro che deve essere rispettato dalla compagnia, guida sotto l'influenza di alcool, utilizzo del veicolo su strade non asfaltate, guida fuori dal territorio nazionale o trasporto di un rimorchio senza autorizzazione.

La **seconda caratteristica innovativa** è la generazione automatica di un punteggio di guida per il conducente. Questo voto è calcolato sulla base dei dati raccolti durante la guida, valutando il comportamento al volante del conducente. Il punteggio serve non solo come feedback per il conducente, aiutandolo a migliorare le proprie abitudini di guida, ma potrebbe essere utilizzato dalle compagnie assicurative per offrire polizze più personalizzate e vantaggiose, stimolando il conducente a mantenere uno stile di guida più prudente. È importante sottolineare che questa profilazione del conducente non richiede l'intervento di un esperto lato server remoto, poiché è la stessa black box a elaborare e generare automaticamente il punteggio di guida.

Il progetto mira a creare una black box che non solo registra e monitora i dati di viaggio e le condizioni del veicolo, ma offre anche un sistema di diagnostica integrato in grado di riconoscere comportamenti inappropriati del conducente ed in grado di valutarlo in base al suo stile di guida; migliorando la sicurezza e l'efficienza del veicolo e contribuendo ad una guida più responsabile e consapevole.

Capitolo 2

Stato dell'arte

- 2.1 Tecnologie attuali di monitoraggio e tracking per veicoli**
- 2.2 Gestione delle emergenze e sistemi di allarmi in uso**
- 2.3 Soluzioni esistenti per la profilazione guidatori**

Capitolo 3

Funzionalità del sistema

3.1 Monitoraggio

Il monitoraggio consiste nella raccolta dati di alcuni parametri provenienti dal lettore OBD e da alcuni sensori installati sulla black box. Il motivo per cui si è scelto di monitorare lo stato del veicolo è quello di ottenere una visione completa della sua salute. In particolare, i dati che vengono registrati sono l'accelerazione misurata grazie ad un accelerometro, utile per analizzare il comportamento dinamico del veicolo. Il sensore di alcool mi permette di rilevare la presenza di vapori alcolici nell'abitacolo, garantendo un ambiente sicuro; per assicurare una qualità dell'aria adeguata all'interno del veicolo, utilizzo un sensore specifico che rileva vari inquinanti, il modulo GPS fornisce coordinate precise, fondamentali sia per la tracciabilità del veicolo in caso di furto, sia per analizzare i percorsi abituali. Viene inoltre monitorata costantemente la temperatura del liquido di raffreddamento e dell'olio motore grazie allo scanner OBD. Inoltre, vengono registrati la coppia del motore per valutare le sue prestazioni e il voltaggio della batteria per prevenire problemi elettrici. Infine, il sensore del livello del carburante mi permette di tenere traccia dell'autonomia del veicolo.

3.2 Tracking

Questa funzionalità del sistema viene adoperata per localizzare i veicoli della flotta [5]. Il sistema di tracking è in grado di generare la posizione esatta del veicolo usando due parametri, latitudine e longitudine; i dati di geolocalizzazione, per ogni slot temporale di 5 minuti, vengono trasferiti al server remoto della centrale operative dell'autonoleggio; a posteriori potrebbero essere utilizzati per visualizzare i percorsi abituali del cliente o il percorso effettuato un preciso giorno. Il sistema di tracking è utile soprattutto in caso di furto del veicolo. L'antenna GPS e il

ricevitore GPS ricevono informazioni dal satellite nel formato NMEA (National Marine Electronics Association), queste vengono processate dall'esp32 e trasmesse al server remoto via GSM, il server una volta ricevuto i dati li memorizza in locale per analisi future.

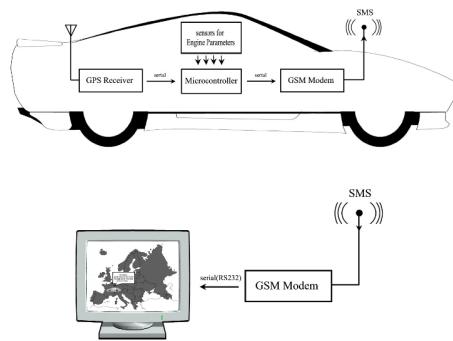


Figura 3.1: L'architettura del sistema: localizzazione GPS e modulo GSM¹

In molti casi il GPS potrebbe darci una risposta errata o non sufficientemente precisa, dovuta a molti fattori esterni, quali condizioni del tempo atmosferico e dell'ambiente che circonda il veicolo.

3.3 Segnalazione di allarmi

Il meccanismo di segnalazione allarmi prevede 3 diversi tipi di dinamiche che verranno presentate in dettaglio nei seguenti paragrafi.

3.3.1 Incidente

L'assistenza incidente tratta la dinamica relativa ad un incidente stradale, inviando le coordinate GPS ed un insieme di informazioni utili ad una stazione remota dell'autonoleggio, sarà compito della stazione avvisare i soccorsi che andranno poi a recarsi nel preciso punto di impatto.

Sostanzialmente il meccanismo si divide in due fasi: **rilevamento** e **notifica** vedi figura 3.3.

¹Preso da: <https://www.researchgate.net/profile/Mohammad-Alkhedher/publication/51989090/figure/fig5/AS:668830275760151@1536472972974/The-system-architecture-GPS-tracking-and-GSM-modules.ppm>

²Preso da: https://www.researchgate.net/publication/51989090_Hybrid_GPS-GSM_localization_of_automobile_tracking_system

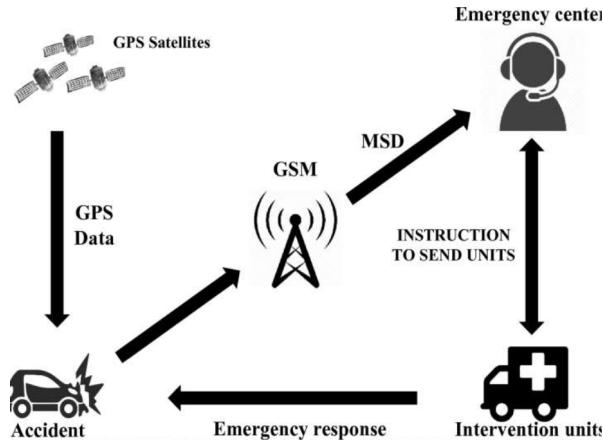


Figura 3.2: Schema situazione di emergenza dovuta ad un incidente²

I sensori maggiormente coinvolti sono l’accelerometro, il GPS e il dato relativo alla velocità del veicolo proveniente dalla rete can bus. Nella fase di campionamento, l’accelerometro continua ad estrarre informazioni relative alla forza G sperimentata dagli occupanti. Il GPS, nella fase di notifica, è utile per inviare le coordinate alla stazione remota. Infine la velocità del veicolo, coinvolta sempre in fase di notifica, viene mandata alla stazione remota e può essere utilizzata per future ricostruzioni/analisi dell’incidente. [6]

Per il rilevamento di una situazione di incidente posso utilizzare un accelerometro, in un evento del genere la black box subirà la stessa accelerazione degli occupanti del veicolo. La forza G che si ricava dall’accelerazione deve essere ragionevolmente alta per un incidente, per questo non vengono considerati incidenti se tutte le accelerazioni sono inferiori a 4G [4]. Infatti sulle auto gli airbag non vengono altro che azionati dal superamento di una certa soglia di accelerazione. Come secondo filtro viene utilizzato un sistema di conferma per evitare i falsi positivi, all’utente vengono concessi 30 secondi per annullare il rilevamento della situazione di incidente, terminato questo breve periodo, il messaggio verrà trasmetto al server remoto in automatico, in caso di annullamento si ritornerà ad una situazione di partenza

$$\begin{aligned} \text{Incidente} &= I \\ \text{Accident_threshold} &= 1 \end{aligned}$$

$$I = \begin{cases} 1 & \text{se } \left(\frac{\text{Accelerazione}}{4G} \right) \geq \text{Accident_threshold} \ \& \& \neg \text{userAbort} \\ 0 & \text{altrimenti} \end{cases}$$

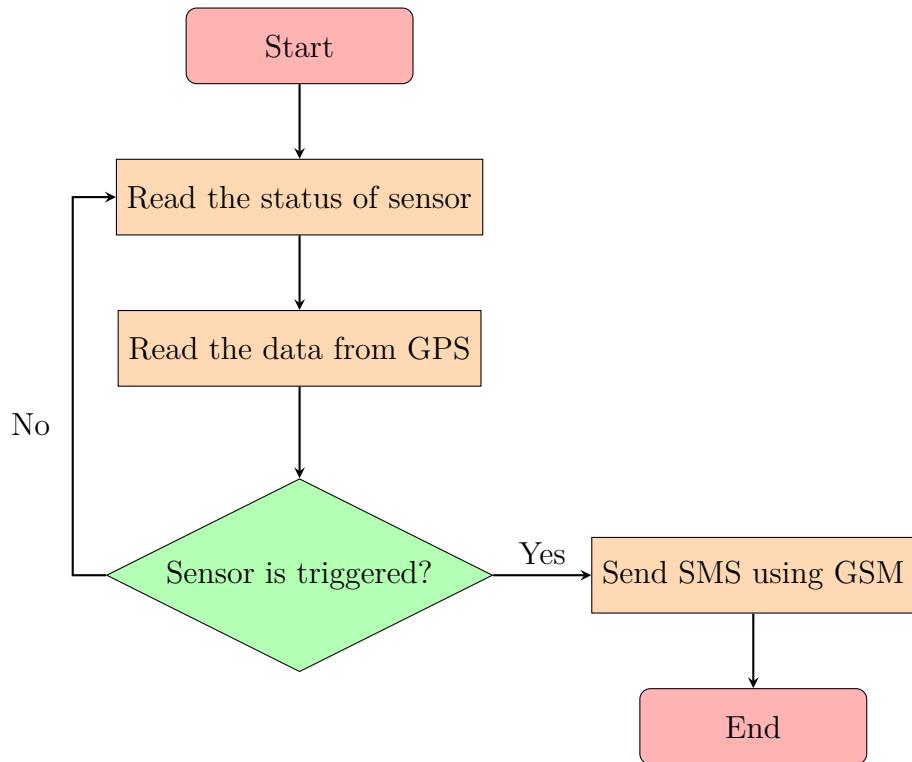


Figura 3.3: Diagramma di flusso per il monitoraggio dell’accelerometro ed eventuale invio SMS

A seguito dell’incidente le informazioni che possono essere trasmesse sono la **forza G** per capire quanto effettivamente è stato forte l’impatto, la **velocità del veicolo** per capire a quanto stava andando, le **coordinate GPS** per individuare il luogo in cui è avvenuto l’incidente e in che **momento** è avvenuto.

3.3.2 Assistenza

Il sistema include un pulsante che permette al conducente di comunicare direttamente con il server remoto (la centrale dell’autonoleggio), inviando un messaggio tramite GSM in caso di necessità.

3.3.3 Rilevamento allarmi

Il rilevamento di allarmi consiste nell’avvertire il conducente e la centrale dell’autonoleggio di alcuni sistemi del veicolo non funzionanti e di alcuni comportamenti non corretti da parte del conducente. Per gestire tale situazione, la black box viene dotata di 3 led per indicare diversi tipi di situazioni e per suggerire al cliente il

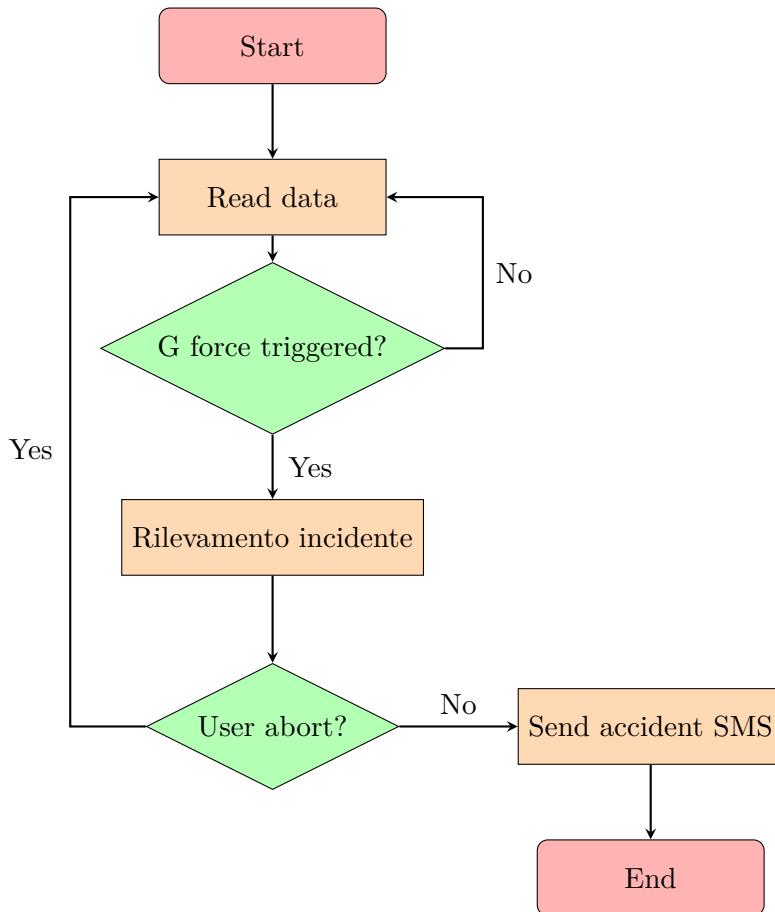


Figura 3.4: Diagramma di flusso per il rilevamento di incidenti e invio SMS

corretto comportamento da intraprendere nel caso dell'accensione di una spia (led) indicante una situazione di avvertimento.

- Il led **verde** indica che il veicolo **funziona correttamente** e il **comportamento** del conducente è **corretto**.
- Il led **giallo** indica un **avvertimento**, se vengono rilevati problemi di poca rilevanza o eseguite azioni non lecite viene azionato il led, dopo aver superato una certa soglia di tolleranza indicante il numero di volte in cui è stato rilevato il problema o eseguita l'azione non lecita, viene avvertita la centrale operativa. Il motto di questa spia è: se un problema di poca rilevanza o un'azione non lecita persistono, avverto la centrale.
- Il led **rosso** indica una situazione di **pericolo**, in questo caso la centrale operativa viene avvertita immediatamente.

Di seguito verranno proposte le dinamiche delle situazioni per le quali viene attivata una spia gialla piuttosto che una rossa.

Non critici (led giallo)

- Divieto di fumo all'interno dell'abitacolo per il mantenimento di un ambiente pulito e privo di odori sgradevoli per i successivi clienti.
- Rilevamento guida su terreni non adatti: fuoristrada o percorsi non asfaltati sono vietati al conducente per evitare danni al veicolo.
- Rilevamento guida su aree geografiche non percorribili: non è possibile recarsi ovunque con il veicolo della compagnia, ad esempio fuori da una certa nazione, o da un loro insieme.
- Rilevamento attività di trasporto/rimorchio: non è possibile trainare qualsiasi tipo di rimorchio con il veicolo a noleggio.

Critici (led rosso)

- Temperatura anomala olio motore: il motore non lavora nel suo range di temperatura e potrebbe guastarsi.
- Temperatura liquido refrigerante: una temperatura troppo alta potrebbe danneggiare componenti del motore, causando gravi guasti.
- Coppia motore anomala: mantenere la coppia motore entro limiti ottimali è essenziale per garantire prestazioni efficienti e sicure del veicolo, prevenendo l'usura prematura del motore e della trasmissione. Se la coppia è troppo bassa, si rischia una scarsa accelerazione; se troppo alta, si può verificare un eccessivo consumo di carburante e possibili danni meccanici.
- Monitoraggio della qualità dell'aria nell'abitacolo: utile per prevenire qualunque perdita di gas proveniente dai vari impianti dell'automobile, prevenendo così incendi.
- Rilevamento presenza di alcool: non è consentito al conducente mettersi alla guida in stato di ebrezza. Questa attività può essere monitorata attraverso un sensore che rileva la presenza di etanolo nell'ambiente. [7]
- Distanza di frenata anomala: analizzando dati di accelerazione, velocità e spazio percorso è possibile rilevare situazioni in cui la distanza di frenata è significativamente più lunga, ciò può indurre un problema al sistema frenante o all'aderenza dei pneumatici.

-
- Consumo anomalo di carburante: monitorando il consumo di carburante del veicolo e confrontandolo con i modelli di consumo attesi è possibile rilevare anomalie che potrebbero indicare problemi al motore.

3.4 Profilazione del comportamento del guidatore

Nel quadro del progetto dell’analisi della profilazione guida, sfruttiamo tecnologie di acquisizione dati per ottenere una comprensione dettagliata del comportamento del guidatore. A tale scopo, viene fatto uso di due principali fonti di informazioni: i **dati provenienti dal sistema diagnostico di bordo OBD-II** e quelli registrati da un **accelerometro**. Questi dispositivi forniscono una vasta gamma di dati in tempo reale, quali ad esempio velocità, accelerazione, decelerazione (frenata) e angoli di sterzata, consentendo di valutare numerosi aspetti relativi alla guida. Attraverso l’analisi approfondita di tali dati, siamo in grado di creare profili individuali dei guidatori e identificare modelli di comportamento distintivi che influenzano la sicurezza, l’efficienza e l’affidabilità del veicolo. Nel corso di questo paragrafo esploreremo come l’integrazione di queste tecnologie ci permetta di ottenere informazioni preziose per migliorare l’esperienza di guida e ottimizzare le operazioni di gestione della flotta.

I principali parametri che vengono catturati per l’analisi del guidatore sono: **rpm** ovvero i giri motore al minuto, la **velocità dell’auto**, il **carico del motore** e la **posizione della valvola farfalla**, provenienti dalla rete CAN-BUS presente all’interno del veicolo, **accelerazione**, **decelerazione** e **sterzate** provengono invece dall’accelerometro. Una volta che la black box riceve i valori dei parametri, essa è in grado di analizzarli e riconoscere situazioni riconducibili ad una guida impropria. Ogni 24h tali dati vengono adoperati per calcolare un punteggio di guida compreso tra 0 e 1. Questo punteggio potrebbe essere utilizzato dalla compagnia per varie finalità, come ad esempio fornire incentivi ai guidatori che adottano comportamenti di guida più sicuri. Ecco le principali fasi del sistema proposto 3.5:

ECU è l’unità centrale di controllo del veicolo dove sono collegati tutti i vari attuatori e sensori del veicolo: motore, aria condizionata, livello carburante... Lo scanner OBD-II trasmette i dati dalla ECU alla board via Bluetooth. La board

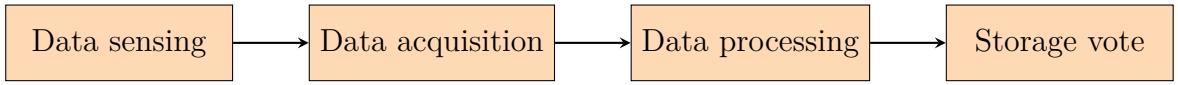


Figura 3.5: Diagramma di flusso dei dati

processa i dati ed è in grado di generare uno score del guidatore. Il punteggio [0,1] viene calcolato considerando:

- Le accelerazioni e decelerazioni brusche. [8]
- Le sterzate sia a sinistra che a destra brusche. [8]
- Il rapporto (relativo) di velocità del veicolo con i giri del motore che deve essere compreso tra i valori di [0.9,1.3] per una buona guida. [9]
- Il rapporto (relativo) tra la valvola a farfalla e i giri del motore che deve essere compreso tra i valori di [0.9,1.3] per una buona guida. [9]
- Il carico del motore che deve essere compreso tra il 20% e il 50% per una buona guida. [9]
- Gli eccessi di velocità data dalla differenza tra i limiti imposti dalla legge e quelli reali del veicolo.

```

1 // Implementazione task relativo alla profilazione guida
2 void profilazioneCallback() {
3     campionamento_Profilazione++;
4     sensors_event_t a, g, temp;
5     mpu.getEvent(&a, &g, &temp);
6
7     float x = a.acceleration.x;
8     float y = a.acceleration.y;
9
10    // Accelerazioni / decelerazioni brusche
11    if ((x / FORZA_G) > (ACCELERAZIONE_BRUSCA) || (x / FORZA_G)
12        ) < (DECELERAZIONE_BRUSCA)) {
13        nRilevamentiAccDecBrusche++;
14    }
15    // Sterzate sinistra e destra brusche
16    if ((y / FORZA_G) > (STERZATA_DX_BRUSCA) || (y / FORZA_G)
17        ) < (STERZATA_SX_BRUSCA)) {
18        nRilevamentiSterzateBrusche++;
19    }

```

```

18 // Rapporto v su giri motore
19 float fatt1 = ((kph / MAX_SPEED_CAR) / (rpm / MAX_RPM_CAR))
20     );
21 if ((fatt1 < RANGE_INF) || (fatt1 > RANGE_SUP)) {
22     nRilevamentiFatt1++;
23 }
24 // Rapporto posizione della valvola a farfalla su giri
25 // motore
26 float fatt2 = ((throttle / 100) / (rpm / MAX_RPM_CAR));
27 if ((fatt2 < RANGE_INF) || (fatt2 > RANGE_SUP)) {
28     nRilevamentiFatt2++;
29 }
30 // Carico motore
31 if ((load < LOAD_MIN) || (load > LOAD_MAX)) {
32     nRilevamentiCaricoMotoreOutOfRange++;
33 }
34 if (campionamento_Profilazione ==
35     MAX_CAMPIONAMENTI_PROFILAZIONE) {
36     updateCampionamentoProfilazione();
37 }
38 }

```

Per quanto riguarda il primo e il secondo punto è possibile utilizzare un accelerometro a 3 assi utile per rilevare le accelerazioni/decelerazioni e le sterzate improvvise:

Asse	Direzione	Azione/Evento
x	anteriore/posteriore	accelerazione/decelerazione (frenata)
y	sinistra/destra	sterzata
z	su/giù	strada dissestata

Tabella 3.1: Descrizione degli assi dell'accelerometro utilizzati nell'ambito della profilazione del guidatore

Il terzo e quarto punto sono invece calcolabili per via analitica raccogliendo dati provenienti dall'OBD-II; per quanto riguarda l'ultimo punto (carico del motore espresso in %) non si ha bisogno di alcuna manipolazione dei dati in quanto essi provengono direttamente dalla rete CAN-BUS.

Per valutare il comportamento del guidatore di seguito vengono mostrati i vari livelli di punteggio del guidatore:

²Presa da: <https://futuranet.it/wp-content/uploads/2023/01/Figura-4-6-768x789.jpg>

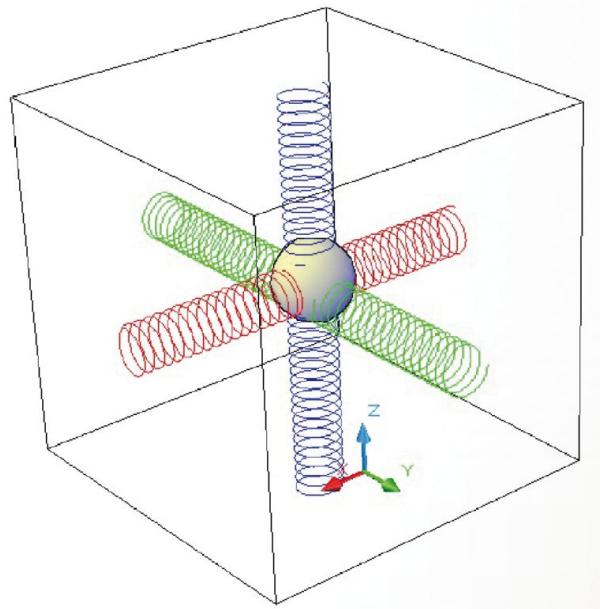


Figura 3.6: Modello meccanico di un accelerometro a 3 assi

Driver score	Driving behaviour
< 0.10	safe
0.1 - 0.19	modest
0.19 - 0.38	moderate risk
0.38 - 0.56	risk
0.56 - 0.75	reckless
> 0.75	critical

Tabella 3.2: Tabella di valutazione del conducente

Il punteggio del guidatore è inteso come la probabilità di rischio di procurare un incidente a seguito del viaggio osservato, pertanto sarà compreso tra 0 e 1.

Capitolo 4

Implementazione del sistema

4.1 Architettura

4.1.1 Componenti hardware utilizzati

Il microcontrollore principale utilizzato per l'elaborazione dei dati e la comunicazione con le periferiche è un **ESP-WROOM-32**, un chip di basso costo e bassa potenza con integrati i moduli WiFi e Bluetooth. Viene utilizzato un **Display LCD 16x2** utile per interagire con l'utente e avvisarlo in caso sia presente qualche problema al veicolo o viene rilevato qualche comportamento anomalo. **MQ-2** e **MQ-3** vengono adoperati come sensori per il rilevamento di gas e alcool, un **bottone** viene adoperato per la richiesta di assistenza alla centrale operativa, dei **led**, presenti in 3 colorazioni: rosso, giallo e verde, vengono adoperati come spie per richiamare, con un effetto visivo, all'attenzione dei possibili scenari che si potrebbero verificare. Un modulo **GPS GY-NEO-6M NEO6MV2** utile per le funzionalità di tracking e monitoraggio, un **accelerometro GY-521 MPU6050** utile per la profilazione del guidatore, o per segnalare un incidente e per concludere un modulo **RTC real-time-clock** utile per mantenere un orario anche quando il sistema non è sotto connessione WiFi e/o è spento.

4.1.2 Software e librerie utilizzate

Il codice è stato sviluppato sull'Arduino IDE, una piattaforma open-source che rende semplice scrivere codice e caricarlo direttamente nelle varie board scelte all'interno di una sezione apposita. Il linguaggio di programmazione utilizzato all'interno dell'IDE si chiama "Arduino programming language", la sintassi è molto simile ai linguaggi C/C++, ma i costrutti più complessi come puntatori o la programmazione orientata agli oggetti vengono messi in secondo piano [10]. Oltre a fornire i costrutti standard di un linguaggio imperativo, dato il contesto applicativo

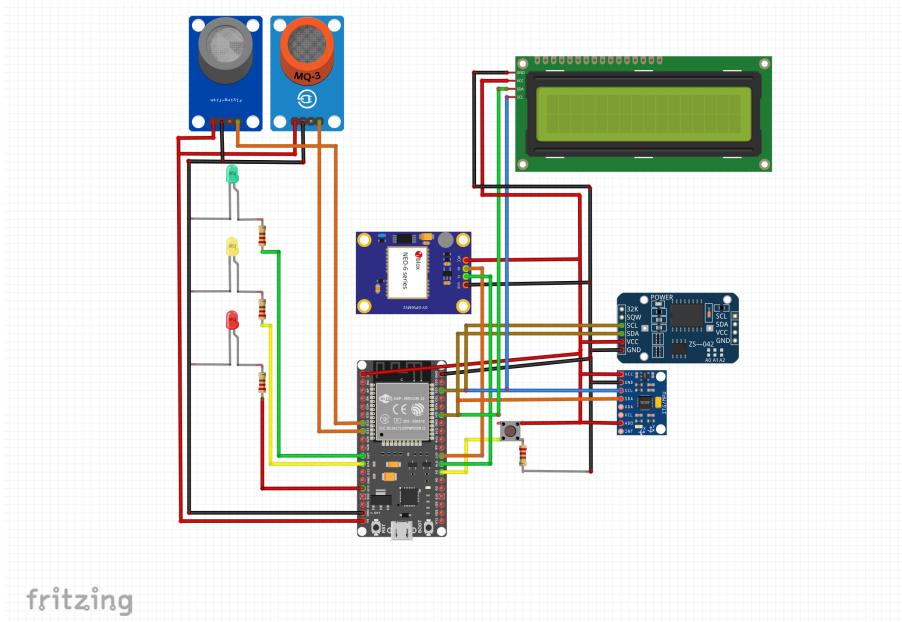


Figura 4.1: Schema di connessione dei vari componenti hardware della black box

dei sistemi di controllo (sensoristica e attuatoristica), offre funzionalità legate alla gestione dei segnali: I/O digitale e analogico, interrupt su eventi hardware e accesso al clock (inteso proprio come tempo trascorso dall'accensione).

Le librerie utilizzate per lo sviluppo del progetto si possono dividere in tre categorie:

- Le librerie dedicate all'interazione con i vari sensori, ad esempio **TinyGPSPlus.h** per il GPS, **Adafruit_MPU6050.h** per l'accelerometro, **LiquidCrystal_I2C.h** per il display, e **time.h**, **sntp.h**, **RTClib.h** per l'RTC.
- Le librerie dedicate alla gestione interna del filesystem, ad esempio **LittleFS.h** per la gestione del filesystem e **sqlite3.h** per la creazione di database e iterazione con essi.
- Le librerie dedicate alla comunicazione con i componenti esterni, ad esempio **"BluetoothSerial.h"** e **"ELMduino.h"** che consentono la comunicazione bluetooth tra ESP32 e lettore scanner OBD.

4.2 Scelte progettuali

4.2.1 Filesystem

Nel filesystem dell'esp32 viene creato un database (dati.db) all'interno del quale sono presenti 5 tabelle, utili per la gestione di tutti gli eventi previsti dalla black-box. Per interagire con il db viene utilizzato SQLite, una libreria scritta in linguaggio C che implementa un motore di database SQL piccolo, veloce, autonomo, ad alta affidabilità e completo. SQLite è il motore di database più utilizzato al mondo. SQLite è integrato in tutti i telefoni cellulari e nella maggior parte dei computer ed è integrato in innumerevoli altre applicazioni utilizzate ogni giorno [11]. SQLite è particolarmente adatto per l'uso in telefoni cellulari, set-top box, televisori, console di gioco, fotocamere, orologi, elettrodomestici da cucina, termostati, automobili, macchine utensili, aeroplani, sensori remoti, droni, dispositivi medici e robot: **"internet of things"**. Di seguito le tabelle di memorizzazione dati dell'intero sistema con qualche esempio di dato memorizzato:

nomeSensore	valore	tipoDato	unitàMisura	timestamp	synchronised	Priority
TEXT	TEXT	TEXT	TEXT	INTEGER (seconds since epoch)	INTEGER (boolean)	INTEGER
"Liquido raffreddamento"	"98"	"INTEGER"	"°C"	1715786497	0	3
"Voltaggio batteria"	"12.3"	"FLOAT"	"Volt (V)"	1715269497	1	8

Tabella 4.1: Tabella "t1" dati relativi al monitoraggio del veicolo

Tabella t1 Il campo **timestamp** indica il numero di secondi trascorsi dall'epoch (01/01/1970).

nomeSensore	tempoStorage	variabile	fattIncr	soglia	tMin	tMax
TEXT	INTEGER	INTEGER	INTEGER	INTEGER	INTEGER	INTEGER
"Olio motore temp"	"45"	"1"	"15"	0.25	45	600
"Coordinate GPS"	"30"	"0"	NULL	NULL	NULL	NULL

Tabella 4.2: Tabella "t2" tempo di campionamento

Tabella t2 Il campo **tempoStorage** indica ogni quanto memorizzare il valore, **variabile** indica se il tempo di storage può cambiare, **fattIncr** ci dice di quanto aumentare o diminuire il tempoStorage, **soglia** delimita se i dati campionati sono omogenei, ovvero minori di essa, oppure eterogenei, maggiori di essa (fai approfondimento statistico), infine **tMin** e **tMax** sono i corrispettivi upper e lower bound per il tempo di storage.

Logica del tempo di campionamento: se dopo una serie di letture (n ad esempio) i dati hanno una bassa deviazione standard, ovvero sono molto simili

tra di loro, memorizzo i dati del veicolo con una frequenza minore, aumentando il tempo di storage di un certo fattore, presente nella tabella di campionamento (**fattIncr**); se la deviazione standard è alta, significa che i dati sono abbastanza differenti tra di loro e li monitoro con una frequenza maggiore. Naturalmente vengono posti dei lower e upper bound per il tempo di memorizzazione entro i quali il tempo stesso può variare.

data	voto	lastStorage	synchronised
TEXT	TEXT	INTEGER	INTEGER
”2024-03-10 12:00:00”	”0.99”	1715786497	1
”2024-07-07 17:56:20”	”0.53”	1815786497	0

Tabella 4.3: Tabella ”t3” profilazione guida

Tabella t3 Il campo **lastStorage** indica il momento in cui è stato memorizzato il voto inteso come numero di secondi trascorsi dall’epoc.

problema	data	synchronised
TEXT	TEXT	INTEGER
”Alcool”	”2024-07-07 17:56:20”	1
”Gas nell’abitacolo”	”2024-07-24 14:40:34”	0

Tabella 4.4: Tabella ”t4” allarmi critici

Tabella t4

problema	rilevamenti	soglia	daMandare	synchronised
TEXT	INTEGER	INTEGER	INTEGER	INTEGER
”Fumo”	3	3	1	0
”Fuori range GPS”	0	5	0	0

Tabella 4.5: Tabella ”t5” allarmi meno critici

Tabella t5 Il campo **soglia** indica il numero massimo di rilevamenti per ciascun tipo di problema, se il sistema black-box continua a rilevare il problema e la soglia viene superata, allora viene aggiornato il campo **daMandare**, indicante che l’informazione deve arrivare al server remoto.

4.2.2 Sincronizzazione

La black-box è progettata per registrare dati in assenza di connessione WiFi, appena viene rilevata la presenza di una rete, il WiFi di casa ad esempio, essi vengono sincronizzati con il server remoto e vengono aggiornati i campi **synchronised** delle tabelle in modo tale da non inviare al server remoto dati duplicati e per poter gestire al meglio l'uso dello spazio di memoria all'interno del filesystem, eliminando i dati già sincronizzati. L'unica tabella che non presenta il campo synchronised è quella relativa al tempo di campionamento, utilizzata come tabella di configurazione per la gestione dei tempi di storage relativi ai dati di monitoraggio

4.3 Comunicazioni e trasmissione dati

4.3.1 CAN BUS (Control Area Network)

I protocolli di bus governano il **trasferimento dei pacchetti** lungo la rete del veicolo. Varie sottoreti e centinaia di sensori comunicano su questi sistemi a bus, inviando messaggi che controllano il comportamento del veicolo e le informazioni che la rete "conosce" in ogni dato momento. Il bus CAN si trova in una posizione standard su tutti i veicoli, cioè sul connettore OBD-II. I pacchetti che viaggiano sul bus di un veicolo, invece, non sono standardizzati. Le comunicazioni critiche per il veicolo, come la gestione dei giri del motore (RPM) e dei freni, viaggiano su linee di bus ad alta velocità, mentre quelle non critiche, come il blocco delle porte e il controllo A/C, viaggiano su linee di bus a media o bassa velocità.

Protocollo CAN BUS CAN (Control Area Network) è un protocollo semplice utilizzato nel settore manifatturiero e automobilistico. I veicoli più recenti sono pieni di piccoli sistemi embedded e di unità di controllo elettronico (ECU) che comunicano tra di loro utilizzando il protocollo CAN. Questo protocollo è stato uno standard delle vetture e dei piccoli autocarri statunitensi dal 1996, ma è stato reso obbligatorio solo nel 2008 (nel 2001 per i veicoli europei) [12]. CAN utilizza due cavi a doppino intrecciato: **CAN high** e **CAN low** e usa il **segnale differenziale**, il che significa che, quando arriva un segnale, CAN aumenta la tensione su una linea e riduce quella dell'altra in misura uguale. I sensori e le ECU hanno un ricetrasmettitore che verifica che entrambi i segnali siano innescati; se non lo sono, rifiuta i pacchetti come rumore.

⁰Presa da: <https://farelettronica.it/web/app/uploads/2022/04/02-controller-area-network-ecu-evoque-adam-ali-small.png>

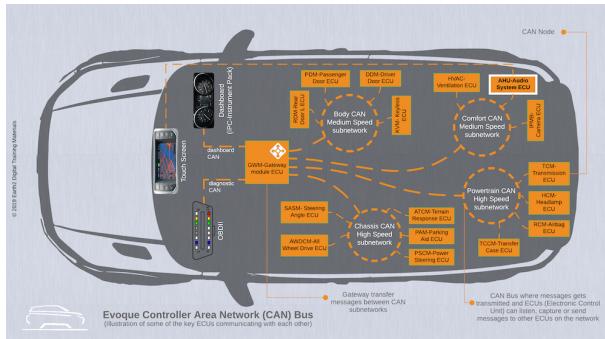


Figura 4.2: Protocollo CAN BUS all'interno di un'auto vettura

Il connettore OBD Molti veicoli sono dotati di un connettore **OBD-II**, detto anche connettore del collegamento diagnostico (DLC, Diagnostic Link Connector), che comunica con la rete interna del veicolo. Solitamente questo connettore si trova sotto il piantone dello sterzo o nascosto da qualche parte nel cruscotto in una posizione comunque relativamente accessibile. La tensione a riposo dei cavi CAN è di 2.5V, quando arriva un segnale, sommerà o sottrarrà 1V (3.5V o 1.5V). I fili CAN attraversano il veicolo e collegano le ECU e altri sensori e sono sempre in coppia. Le connessioni CAN high e CAN low sono presenti sui pin 6 e 14 del connettore OBD-II, come illustrato nella figura sottostante 4.3. [12]

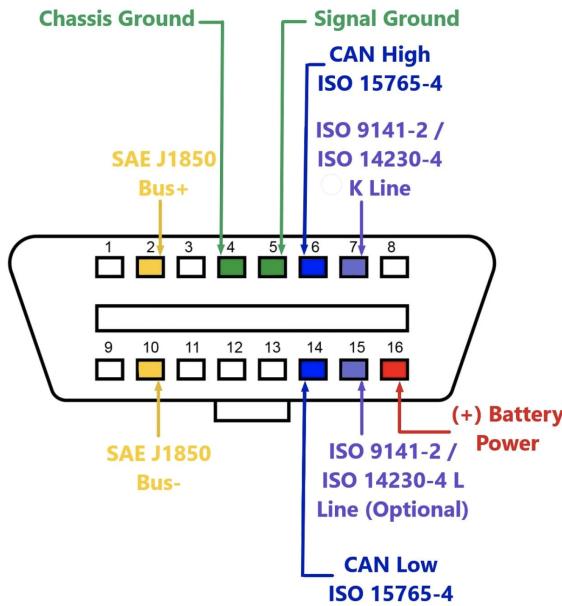


Figura 4.3: Pin dei cavi CAN sul connettore OBD-II

Struttura dei pacchetti del bus CAN Vi sono due tipi di pacchetti CAN: **standard** ed **estesi**. I pacchetti sono come quelli standard, semplicemente hanno uno spazio maggiore per conservare ID. **Pacchetti standard:** Ogni pacchetto di bus CAN contiene quattro elementi fondamentali:

- **L'ID di arbitraggio** (arbitration ID) è un messaggio broadcast che indica l'ID del dispositivo che cerca di comunicare, anche se ogni dispositivo può inviare più ID di arbitraggio. Se sul bus vengono inviati due pacchetti CAN contemporaneamente, vince quello con l'ID di arbitraggio più basso.
- **IDE**, estensione dell'identificatore (Identifier Extension) Questo bit è sempre 0 per i pacchetti CAN standard.
- **DLC**, codice della lunghezza dei dati (Data Length Code) Questa è la dimensione dei dati che varia da 0 a 8 byte.
- **Dati** Questi sono i dati in senso stretto. Le dimensioni massime dei dati veicolati da un pacchetto standard del bus CAN è di 8 byte, ma alcuni sistemi forzano le dimensioni sempre a 8 byte, "imbottendo" il pacchetto.

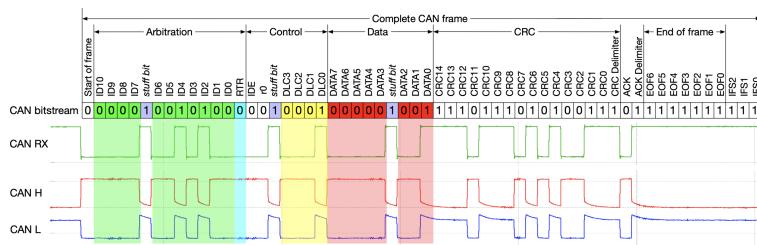


Figura 4.4: Formato dei pacchetti CAN standard

Poiché i pacchetti del bus CAN sono di tipo broadcast, tutti i controller sulla stessa rete vedono tutti i pacchetti. I pacchetti non portano con sè informazioni sul controller che li abbia inviati. Poiché qualsiasi dispositivo può vedere e trasmettere pacchetti.

Pacchetti estesi: I pacchetti estesi sono come quelli standard, tranne che possono essere concatenati fra loro per creare ID più lunghi. I pacchetti estesi sono fatti in modo da adattarsi alla formattazione CAN standard per mantenere la compatibilità a ritroso. Così se un sensore non ha il supporto per i pacchetti estesi, non si guasterà se un altro dispositivo trasmette pacchetti CAN estesi sulla

⁰Presi da: https://www.flexihub.com/images/upload/flexihub/articles/diagnostics/pinout_explanation.jpg

⁰Presi da: <https://upload.wikimedia.org/wikipedia/commons/5/54/CAN-bus-frame-with-stuff-bit-and-correct-CRC.png>

stessa rete. I pacchetti standard e quelli estesi si differenziano anche per l'uso dei flag. Se esaminati i pacchetti estesi in un dump di rete, vedrete che, a differenza di quelli standard, usano un flag **SSR** (Substitute remote request) anzichè **RTR** (Remote transmission request), con SSR impostato al valore 1. Avranno anche il campo IDE impostato al valore 1 ed i pacchetti avranno identificatori a 18 bit, che è la seconda parte dell'identificatore standard a 11 bit.

4.3.2 OBD scanner - esp32

Per leggere i messaggi presenti all'interno della rete CAN BUS è necessario di un lettore OBD-II in grado di connettersi con il corrispettivo connettore OBD presente all'interno del veicolo e di trasmettere dati via **Bluetooth** ad un apparecchio esterno, nel nostro caso l'ESP32. Per leggere e decifrare i messaggi provenienti dalla rete CAN, è stata utilizzata una libreria ELMduino; utile per interfacciare facilmente l'ESP32 con uno scanner OBD-II. Tramite questa libreria è possibile interrogare tutti i PID supportati da OBD-II per raccogliere un'ampia varietà di dati, ad esempio velocità, giri al minuto, temperatura del motore...



Figura 4.5: Esempio di scanner OBD-II utilizzato per il progetto

4.3.3 esp32 - server remoto

Quando la black box rileva la presenza di una connessione Internet, si collega ad un server remoto, il quale è sempre in ascolto, per trasmettere dati relativi al monitoraggio, alla segnalazione di allarmi e sulla profilazione del conducente. Nel progetto viene adoperato Netcat, un programma open source a riga di comando di comunicazione remota, utilizzabile sia col protocollo TCP sia col protocollo UDP. Lato server è necessario mettersi in ascolto per ricevere dati al momento di sincronizzazione, dopodichè è possibile redirigere l'output su di un file utilizzato appositamente per la raccolta dati. [13]

⁰Presa da: https://m.media-amazon.com/images/I/51crnV3RTbL._AC_SX679_.jpg

Capitolo 5

Risultati ottenuti

5.1 Analisi dei risultati

5.2 Comportamenti non implementati

Alcuni comportamenti relativi alla segnalazione di allarmi critici e non sono stati implementati. Questi riguardano la gestione degli allarmi relativi al **consumo anomalo di carburante** e alla **distanza di frenata anomala** per quanto riguarda gli allarmi critici, il **rilevamento della presenza di fumo** per quanto riguarda invece gli allarmi meno critici. Quest'ultima funzionalità è facilmente implementabile con l'installazione di un sensore apposito che rileva la presenza di fumo nell'ambiente che lo circonda; l'implementazione delle prime due funzionalità è stata volutamente non implementata data la complessa natura dei due problemi, che richiedono uno studio complesso e approfondito. Questi aspetti rimangono importanti per sviluppi futuri e rappresentano punti chiave per migliorare ulteriormente il sistema.

Un altro componente non installato nel sistema attuale è il **modulo GSM con relativa antenna**, che permette la comunicazione con il server remoto della centrale operativa indipendentemente dalla disponibilità di una rete WiFi. L'installazione del modulo GSM può essere facilmente eseguito seguendo due approcci:

- **Modulo separato:** un modulo GSM con relativa antenna può essere facilmente aggiunto al sistema esistente. Questi moduli sono ampiamente disponibili e possono essere integrati con l'ESP32 attraverso interfacce standard (metti riferimento alla figura sotto).
- **Modulo integrato:** esistono versioni dell'ESP32 che includono un modulo GSM integrato, semplificando ulteriormente l'implementazione di questa

funzionalità senza la necessità di componenti aggiuntivi (metti riferimento alla figura sotto).



Figura 5.1: Modulo GSM da integrare



Figura 5.2: ESP32 SIM800L

⁰Presa da: <https://i.ebayimg.com/images/g/~dEAAOSwmwdhZk07/s-11200.jpg>

⁰Presa da: <https://futuranet.it/wp-content/uploads/2021/02/1606-HR1761.jpg>

Capitolo 6

Conclusioni

6.1 Problemi aperti

Problema di decremento dello spazio di storage Nel sistema black box sono presenti due task di pulizia della memoria (fai riferimento a testo precedente), uno di routine che pulisce i record già sincronizzati ed uno che pulisce i record sincronizzati e non se viene superata una soglia massima di occupazione del filesystem (90%) che elimina la metà dei record di monitoraggio più datati. Il problema riscontrato risale a quest'ultimo task. Lo spazio di storage man mano che vengono memorizzati record aumenta in modo corretto, tuttavia a seguito delle operazioni di pulizia dei record effettuate dal task di routine che rimuove dalla tabella di monitoraggio i record già sincronizzati, lo spazio di storage non si decrementa come previsto. Tale problema persiste anche dopo aver tentato di utilizzare un particolare comando **VACUUM**, che solitamente dovrebbe riorganizzare il database e liberare lo spazio inutilizzato. Il secondo task di pulizia, che riguarda il monitoraggio della capacità di memoria del filesystem e l'eliminazione dei record meno recenti, non è stato implementato nel codice.

6.2 Sviluppi futuri

6.2.1 Memoria

Negli sviluppi futuri del progetto, un'area di miglioramento fondamentale è l'integrazione di una memoria esterna con la board. Questa aggiunta apporterà significativi vantaggi operativi e funzionali, suddivisibili in due principali categorie: l'aumento dello spazio di storage per il monitoraggio del veicolo e la capacità di caricare e utilizzare mappe stradali per verificare il rispetto dei limiti di velocità.

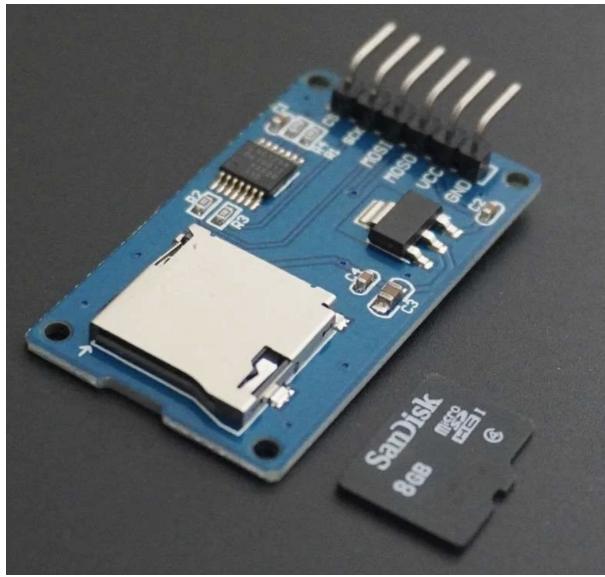


Figura 6.1: Micro SD card

Maggiore spazio di storage per il monitoraggio del veicolo Uno dei limiti attuali dell'ESP32 è la sua capacità di storage relativamente limitata, che può rappresentare un ostacolo nel contesto del monitoraggio continuo dei dati del veicolo. L'integrazione di una memoria esterna, come una scheda microSD o modulo di memoria flash, può aumentare notevolmente la quantità di dati che possono essere registrati e memorizzati. Questo incremento di capacità di storage consente di:

- **Monitoraggio continuo:** registrare dati continuamente per periodi più estesi senza preoccuparsi di esaurire lo spazio disponibile.
- **Sincronizzazione rara con il server remoto:** ridurre la necessità di sincronizzare frequentemente i dati raccolti con un server centrale. I dati possono essere raccolti e conservati localmente per lunghi periodi e sincronizzati solo quando una rete WiFi stabile è disponibile.

Caricamento di mappe per il rispetto dei limiti di velocità Un'altra funzionalità utile per la profilazione che potrebbe essere implementata con l'aggiunta di una memoria esterna è la possibilità di caricare e gestire mappe stradali. Questa capacità può essere adoperata per monitorare il rispetto dei limiti di velocità del veicolo in diverse aree geografiche. I vantaggi includono:

⁰Preso da: <https://www.electronicwings.com/storage/PlatformSection/TopicContent/452/description/MicroSD.jpg>

-
- **Verifica del rispetto dei limiti di velocità:** Utilizzando mappe che includono informazioni sui limiti di velocità in zone urbane, extraurbane, e autostradali, il sistema può monitorare in tempo reale se il veicolo rispetta i limiti imposti in ogni tratto stradale.
 - **Sicurezza stradale:** aumentare la sicurezza stradale, informando il conducente in caso di superamento dei limiti di velocità.

6.2.2 Profilazione conducente lato server remoto

Per concludere il nostro studio possiamo affiancare alla capacità della board di valutazione del guidatore, tecniche di Machine Learning relative sia all'apprendimento supervisionato che non supervisionato, consentendoci di esaminare più approfonditamente la natura della nostra flotta. In questo contesto, ci concentreremo su due principali tecniche di Machine Learning: la **classificazione** e il **clustering**. La classificazione ci permette di assegnare ai guidatori delle etichette, come ad esempio 'prudente' piuttosto che 'negligente', in base ai loro comportamenti al volante, mentre il clustering ci consente di raggruppare i guidatori in base a similitudini nel comportamento di guida. Esploremo queste tecniche nel dettaglio, evidenziando le loro applicazioni specifiche e i benefici che possono apportare all'analisi e alla gestione dell'autonoleggio. [14]

Machine Learning nell'analisi dei guidatori: applicazioni nel nostro contesto Una volta processati i dati relativi all'analisi di profilazione del guidatore, essi possono essere inviati ad un server remoto. Dopodichè è possibile usare varie tecniche di machine learning di apprendimento supervisionato ad esempio la **classificazione** o di apprendimento non supervisionato ad esempio il **clustering**, per assegnare ai clienti dell'autonoleggio una certa etichetta di classificazione o raggrupparli in cluster in modo tale da poter osservare un generale comportamento della flotta.

Apprendimento supervisionato La **classificazione** è una tecnica di apprendimento supervisionato, l'obiettivo è quello di predire la classe di appartenenza di un'istanza di input in un insieme discreto di classi; ad esempio, classificare se una mail è spam o non, se un paziente è soggetto a diabete o meno e molti altri esempi. Nel nostro caso di studio vorremmo poter classificare i clienti come guidatori prudenti o negligenti in base ai valori di valutazione studiati durante l'analisi di profilazione fatta precedentemente. Questa tecnica di ML prevede la suddivisione dei dati in due insiemi distinti: un insieme di addestramento e un insieme di test. L'insieme di addestramento viene adoperato per addestrare il modello, mentre

l’insieme di test viene utilizzato per valutarne le prestazioni su dati non visti. Per addestrare un modello di apprendimento supervisionato, vengono utilizzati diversi algoritmi che si differenziano per tipo di problema e per tipo di dati. Alcuni esempi di algoritmi comuni includono Support Vector Machine (SVM), Random Forest, Decision Tree, Neural Networks e Regressione Lineare. Dopo l’addestramento, il modello viene valutato utilizzando l’insieme di test. Una volta soddisfatti delle prestazioni del modello, esso può essere pubblicato, implementato in un’applicazione o integrato in un sistema più ampio per fare previsioni su nuovi dati [15]. Di seguito un esempio di classificazione binaria:

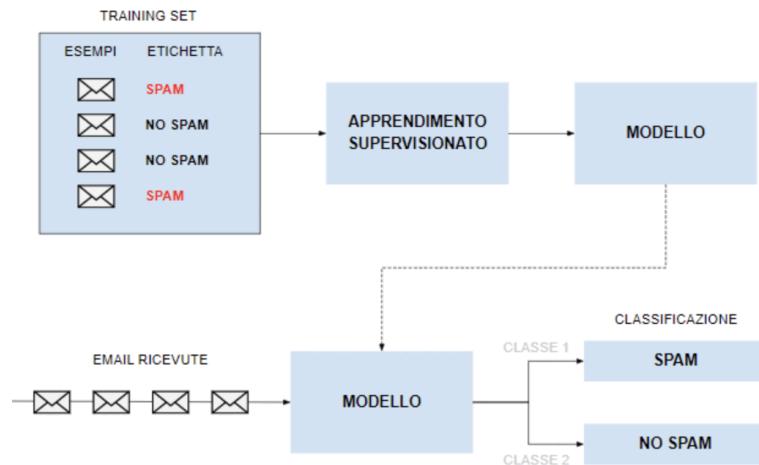


Figura 6.2: Esempio di classificazione binaria che stabilisce se una email è spam o no²

Se nel nostro studio si volesse classificare il guidatore in più classi, prudente, moderato, negligente ad esempio, con lo scopo di stipulare un contratto assicurativo in base allo stile di guida, esiste la classificazione multiclasse dove sono presenti più di due classi possibili tra cui scegliere e dove ogni istanza da classificare può appartenere ad una sola classe tra le possibili.

Apprendimento non supervisionato Il **clustering** è una tecnica di apprendimento non supervisionato il cui scopo è quello di raggruppare un insieme di oggetti in cluster; gli oggetti dovranno avere un’alta similarità intra-cluster e una bassa inter-cluster. È una tecnica molto complessa dato che i vari algoritmi imparano osservando i dati e non tramite degli esempi. Grazie a questa tecnica è possibile visualizzare il comportamento dei guidatori della flotta. L’ideale utopico

²Presa da: <https://www.andreaminini.com/data/andreaminini/apprendimento-supervisionato-esempio-1.gif>

sarebbe quello di avere cluster il cui valore medio che li rappresenta contenga valori di valutazione (quelli citati nella sezione profilazione) ragionevoli o che non si discostino troppo dai valori attesi del guidatore prudente [16].

Di seguito vengono suggerite alcune tecniche che potrebbero essere utilizzate per clusterizzare i propri clienti:

- Clustering partizionale
- Clustering gerarchico
- Clustering basato sulla densità

Clustering partizionale Il clustering partizionale è la tecnica più comunemente utilizzata tra gli algoritmi di clusterizzazione. Questi algoritmi lavorano per minimizzare un determinato criterio di clustering, spostando iterativamente i punti dati tra i cluster fino a raggiungere una partizione ottimale. Questo approccio suddivide gli oggetti in K partizioni, ognuna delle quali rappresenta un cluster. La suddivisione avviene in base ad una specifica funzione obiettivo. I cluster vengono formati per ottimizzare un criterio di partizionamento definito, come una funzione di dissimilarità basata sulla distanza, in modo che gli oggetti all'interno di un cluster siano considerati "simili", mentre quelli in cluster diversi siano considerati "dissimili". I metodi di clustering partizionale sono particolarmente adatti per applicazioni in cui è richiesto un numero predefinito di cluster, alcuni esempi di algoritmi di clustering includono K-means, PAM (Partition Around Medoids) e Clara.

Clustering gerarchico Gli algoritmi di clustering gerarchico dividono o uniscono un dataset in una sequenza di partizioni nidificate. La gerarchia delle partizioni nidificate può essere agglomerativa (bottom-up) o divisiva (top-down). Nel metodo agglomerativo, il clustering inizia con ogni singolo oggetto in un singolo cluster e continua a raggruppare le coppie di cluster più vicine fino a quando tutti gli oggetti sono insieme in un unico cluster. Il clustering gerarchico divisivo, d'altra parte, inizia con tutti gli oggetti in un singolo cluster e continua a dividere i cluster più grandi in cluster più piccoli fino a quando tutti gli oggetti sono separati in cluster unitari. Entrambi i metodi gerarchici mostrano il modo naturale di rappresentare i cluster, chiamato dendogramma. Esempi di questi algoritmi sono Rock, Birch (Balance Iterative Reducing and Clustering using Hierarchies), Cure (Cluster Using Representatives)

³Preso da: <https://upload.wikimedia.org/wikipedia/commons/thumb/e/e5/KMeans-Gaussian-data.svg/1280px-KMeans-Gaussian-data.svg.png>

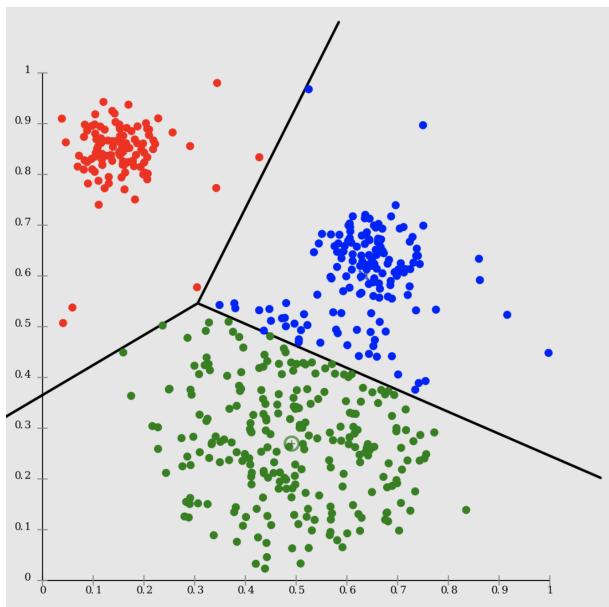


Figura 6.3: K-means usato per la classificazione di 3 partizioni³

Una gerarchia di cluster può essere interpretata come un albero binario standard in cui la radice rappresenta tutti gli insiemi di oggetti dati da clusterizzare che formano il livello più alto della gerarchia (livello 0). Ad ogni livello, i nodi che sono il sottoinsieme dell'intero dataset corrispondono al cluster. Gli elementi in ciascuno di questi cluster possono essere determinati attraverso l'attraversamento dell'albero dal nodo del cluster corrente alla base singleton, rappresentata dalle foglie dell'albero. Questa gerarchia di cluster è chiamata dendogramma. Il vantaggio fondamentale di avere un metodo di clustering gerarchico è che consente di tagliare la gerarchia al livello desiderato e questa caratteristica lo rende diversamente diverso dagli altri algoritmi di clustering. Esistono diversi algoritmi di clustering gerarchico agglomerativo che utilizzano misure di similarità diverse e quindi, basati su questo, sono presenti differenti algoritmi di clustering agglomerativo: linkage singolo, linkage completo, linkage medio di gruppo, linkage del centroide, criterio di Ward.

⁴Presa da: <https://www.developersmaggioli.it/wp-content/uploads/2019/06/images.png>

⁵Presa da: <https://www.researchgate.net/profile/Emanuele-Lo-Biundo/publication/344452675/figure/fig7/AS:942183599325184@1601645483382/Figura-23-Esempio-di-clustering-gerarchico-su-8-osservazioni.ppm>

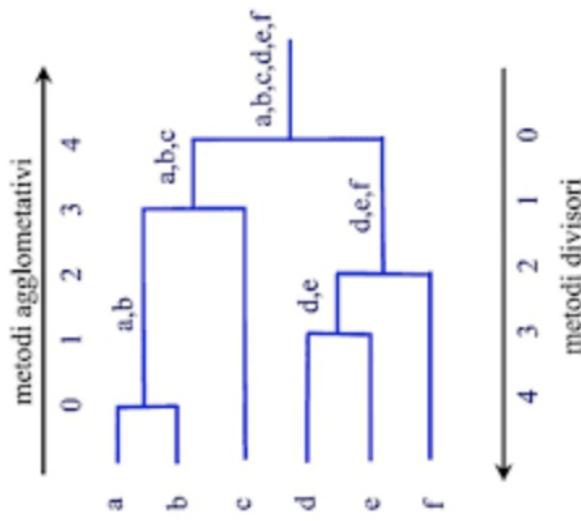


Figura 6.4: Esempio di dendogramma⁴

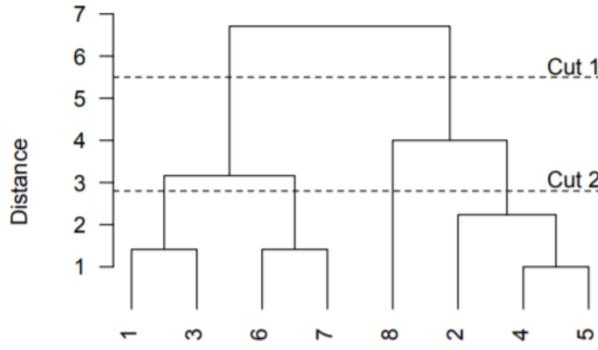


Figura 6.5: Esempio di dendogramma con tagli gerarchici⁵

Clustering basato sulla densità Gli algoritmi di clustering basati sulla densità sono ideati per la creazione di cluster di forma arbitraria. In questo approccio, un cluster di forma arbitraria è considerato come una regione in cui la densità degli oggetti supera una soglia. Nel clustering density-based, il raggruppamento avviene analizzando l'intorno di ogni punto nello spazio. In particolare, viene considerata la densità di punti in un intorno di raggio fissato. DBSCAN e SSN RDBC sono algoritmi tipici di questo tipo.

Per concludere lo schema (metti riferimento immagine) indica che con la black box è possibile sfruttare la capacità valutativa della board, la quale è in grado di

⁶Presi da: <https://www.researchgate.net/profile/Emanuele-Lo-Biundo/publication/344452675/figure/fig7/AS:942183599325184@1601645483382/Figura-23-Esempio-di-clustering-gerarchico-su-8-osservazioni.ppm>

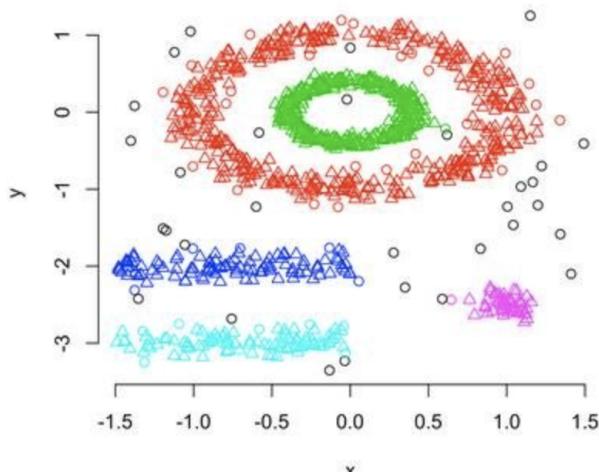


Figura 6.6: Esempio DBSCAN⁶

generare uno score del guidatore e di giudicarlo in base a questo, la possibilità di classificarlo utilizzando tecniche di machine learning o la possibilità di utilizzare entrambe le tecniche per avere un quadro completo relativo al suo stile di guida.

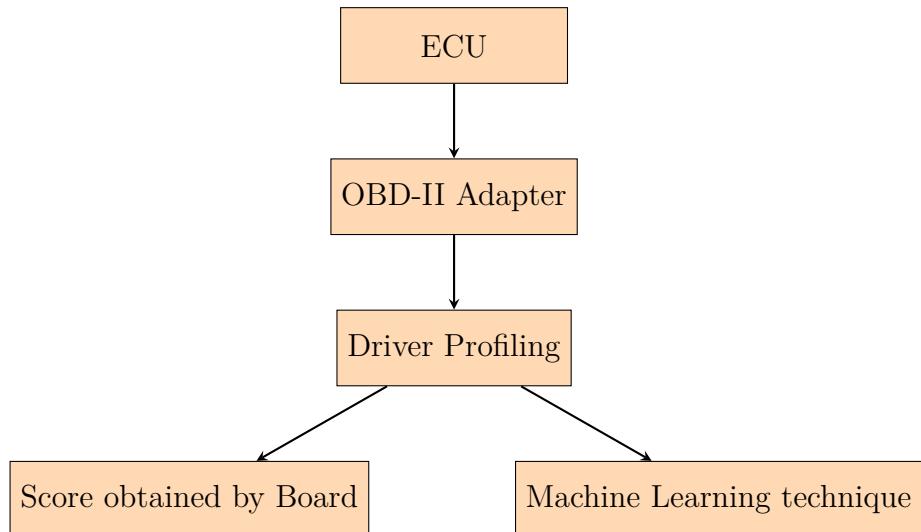


Figura 6.7: Diagramma di flusso del sistema di profilazione conducente

Bibliografia

- [1] Mattia Fracassi. «Assicurazioni R.C.A.: Un Modello per l'Uso della Scatola Nera». Tesi di laurea mag. Università Ca' Foscari Venezia, 2013. URL: <http://dspace.unive.it/bitstream/handle/10579/3606/815923-1169047.pdf?sequence=2>.
- [2] June Myint Mo Khin e Nyein Nyein Oo. «Real-time vehicle tracking system using Arduino, GPS, GSM and web-based technologies». In: *International Journal of Science and Engineering Applications* 7.11, 433-436 (2018).
- [3] Galih Hermawan e Emir Husni. «Acquisition, modeling, and evaluating method of driving behavior based on OBD-II: A literature survey». In: *IOP Conference Series: Materials Science and Engineering*. Vol. 879. 1. IOP Publishing. 2020, p. 012030.
- [4] Chris Thompson et al. «Using smartphones to detect car accidents and provide situational awareness to emergency responders». In: *Mobile Wireless Middleware, Operating Systems, and Applications: Third International Conference, Mobilware 2010, Chicago, IL, USA, June 30-July 2, 2010. Revised Selected Papers 3*. Springer. 2010, pp. 29–42.
- [5] Mohammad A Al-Khedher. «Hybrid GPS-GSM localization of automobile tracking system». In: *arXiv preprint arXiv:1201.2630* (2012).
- [6] Suraj Pratap Shubham, Madan Kumar, Sarika Jain et al. «A survey on IoT based automatic road accident detection». In: *2021 5th international conference on intelligent computing and control systems (ICICCS)*. IEEE. 2021, pp. 1–7.
- [7] Azam Ahmar e Kshitij Singh. «Road Accident Prevention Using Alcohol Detector And Accelerometer Module. No. 5047». In: *EasyChair* (2021).
- [8] Mohamed Fazeen et al. «Safe driving using mobile phones». In: *IEEE Transactions on Intelligent Transportation Systems* 13.3 (2012), pp. 1462–1468.

-
- [9] Shi-Huang Chen, Jeng-Shyang Pan, Kaixuan Lu et al. «Driving behavior analysis based on vehicle OBD information and adaboost algorithms». In: *Proceedings of the international multiconference of engineers and computer scientists*. Vol. 1. 2015, pp. 18–20.
 - [10] Andrea Trentini Alexjan Carraturo. *Sistemi Embedded: teoria e pratica*. Leditizioni LediPublishing, 2023.
 - [11] SQLite. *When to Use SQLite*. 2024. URL: <https://www.sqlite.org/whentouse.html> (visitato il giorno 28/07/2024).
 - [12] Craig Smith. *Il manuale dell'hacker di automobili. Guida per il penetration tester*. Edizioni LSWR, 2016.
 - [13] NetCat. *TCP Server and TCP Client Commands*. 2022. URL: <https://www.varonis.com/blog/netcat-commands> (visitato il giorno 29/07/2024).
 - [14] Wikipedia. *Apprendimento automatico — Wikipedia, The Free Encyclopedia*. <http://it.wikipedia.org/w/index.php?title=Apprendimento%20automatico&oldid=138273572>. [Online; accessed 29-July-2024]. 2024.
 - [15] Iqbal Muhammad e Zhu Yan. «SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY.» In: *ICTACT Journal on Soft Computing* 5.3 (2015).
 - [16] Shraddha K Popat e M Emmanuel. «Review and comparative study of clustering techniques». In: *International journal of computer science and information technologies* 5.1 (2014), pp. 805–812.