



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

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1.1	Cos'è una scatola nera? . . . . .	3
1.1.1	Utilità . . . . .	3
1.1.2	Importanza . . . . .	3
1.2	Obiettivi del progetto . . . . .	3
<b>2</b>	<b>Stato dell'arte</b>	<b>4</b>
2.1	Tecnologie attuali di monitoraggio e tracking per veicoli . . . . .	4
2.2	Gestione delle emergenze e sistemi di allarmi in uso . . . . .	4
2.3	Soluzioni esistenti per la profilazione guidatori . . . . .	4
<b>3</b>	<b>Funzionalità del sistema</b>	<b>5</b>
3.1	Monitoraggio . . . . .	5
3.2	Tracking . . . . .	5
3.3	Segnalazione di allarmi . . . . .	6
3.3.1	Incidente . . . . .	6
3.3.2	Assistenza . . . . .	8
3.3.3	Rilevamento allarmi . . . . .	8
3.4	Profilazione del comportamento del guidatore . . . . .	11
<b>4</b>	<b>Implementazione del sistema</b>	<b>14</b>
4.1	Architettura . . . . .	14
4.1.1	Componenti hardware utilizzati . . . . .	14
4.1.2	Software e librerie utilizzate . . . . .	14
4.2	Scelte progettuali . . . . .	16
4.2.1	Filesystem . . . . .	16
4.2.2	Sincronizzazione . . . . .	18
4.3	Comunicazioni e trasmissione dati . . . . .	18
4.3.1	CAN BUS (Control Area Network) . . . . .	18
4.3.2	OBD scanner - esp32 . . . . .	21
4.3.3	esp32 - server remoto . . . . .	21

<b>5 Risultati ottenuti</b>	<b>22</b>
5.1 Analisi dei risultati . . . . .	22
5.2 Comportamenti non implementati . . . . .	22
<b>6 Conclusioni</b>	<b>24</b>
6.1 Problemi aperti . . . . .	24
6.2 Sviluppi futuri . . . . .	24
6.2.1 Memoria . . . . .	24
6.2.2 Profilazione conducente lato server remoto . . . . .	26

# Capitolo 1

## Introduzione

### 1.1 Cos'è una scatola nera?

#### 1.1.1 Utilità

#### 1.1.2 Importanza

### 1.2 Obiettivi del progetto

# **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, registro l'accelerazione utilizzando il sensore di accelerometria, 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. 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.

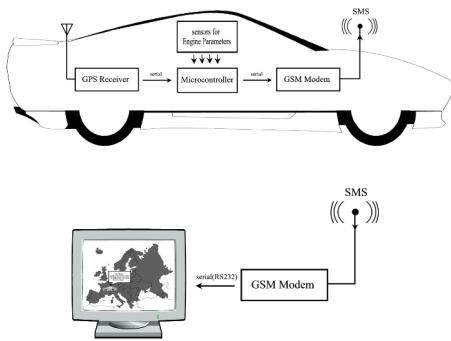


Figura 1: L'architettura del sistema: localizzazione GPS e modulo GSM<sup>1</sup>

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**

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

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

<sup>2</sup>Presato da: [https://www.researchgate.net/publication/51989090\\_Hybrid\\_GPS-GSM\\_localization\\_of\\_automobile\\_tracking\\_system](https://www.researchgate.net/publication/51989090_Hybrid_GPS-GSM_localization_of_automobile_tracking_system)

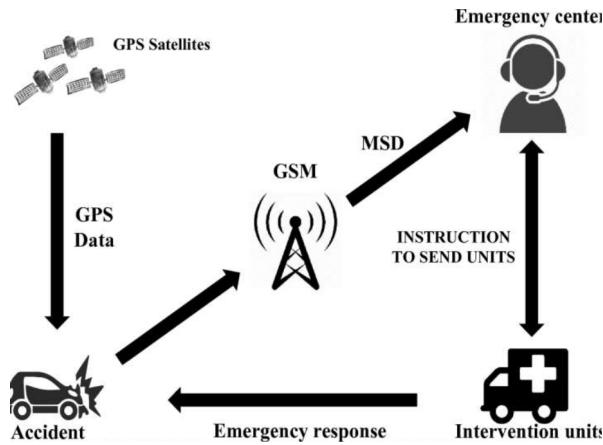


Figura 2: Schema situazione di emergenza dovuta ad un incidente<sup>2</sup>

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.

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. 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 trasmesso al server remoto in automatico, in caso di annullamento si ritornerà ad una situazione di partenza

Incidente =  $I$   
Accident\_threshold = 1

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

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**

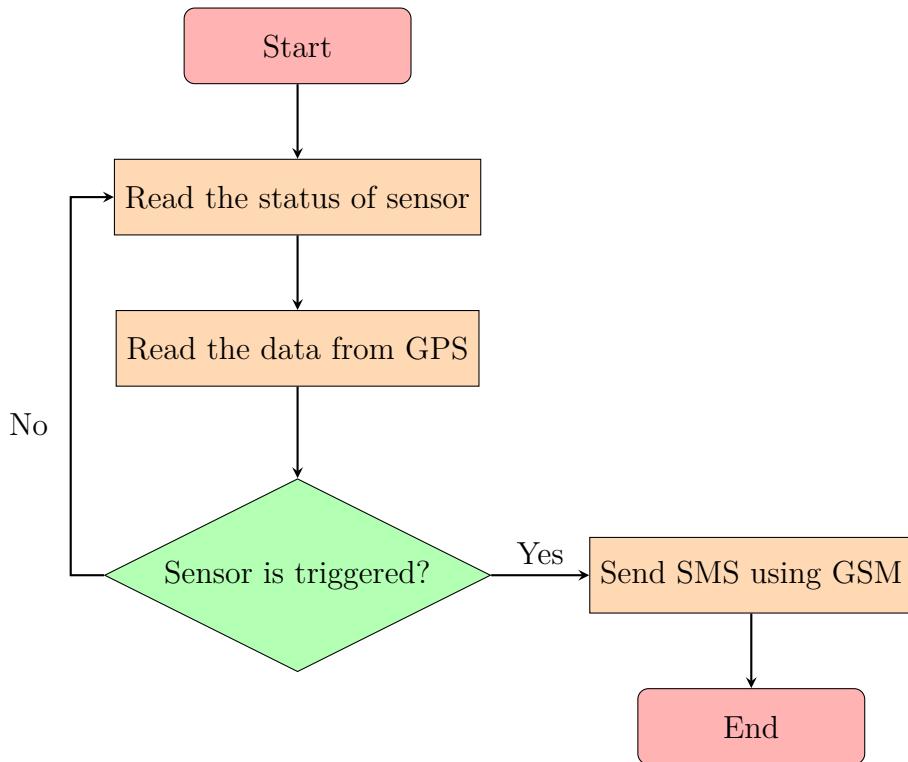


Figura 3: Diagramma di flusso per il monitoraggio dell'accelerometro ed eventuale invio SMS

**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 prevede l'utilizzo di un bottone per comunicare direttamente con il server remoto, inviando un opportuno messaggio via GSM, di nessità da parte del conducente.

### 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 le situazioni relative all'ultima dinamica, la black box viene dotata di 3 led per indicare diversi tipi di situazioni e per suggerire al cliente il corretto comportamento da intraprendere nel caso dell'accensione di una spia (led) indicante una situazione di avvertimento.

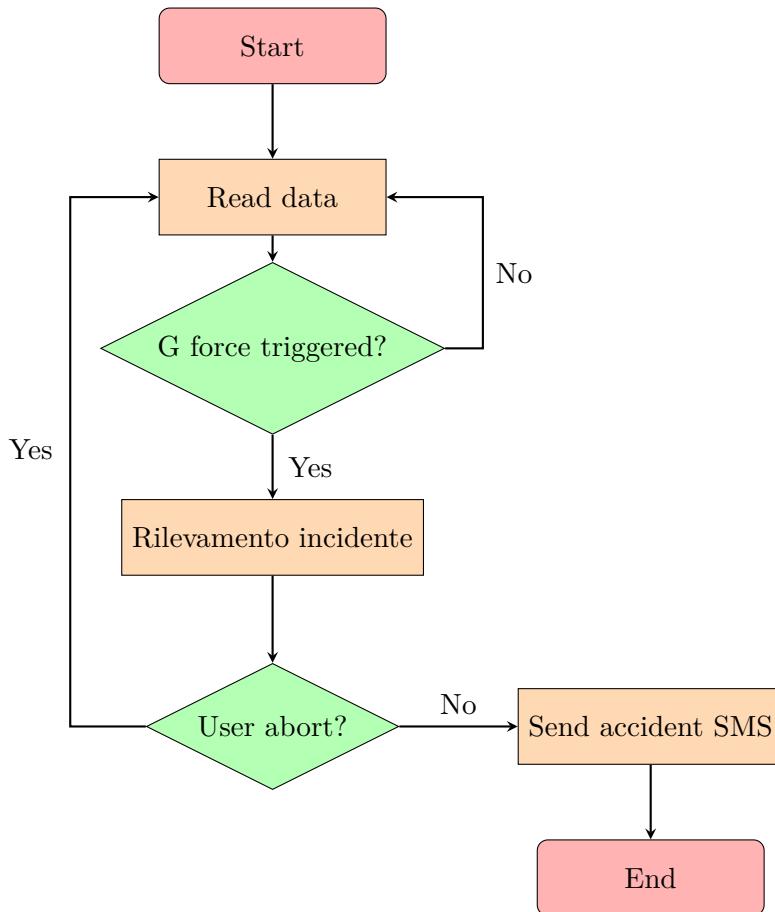


Figura 4: Diagramma di flusso per il rilevamento di incidenti e invio SMS

- 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, avverte 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

- 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

- 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 e prevenire 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.
- 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 **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:

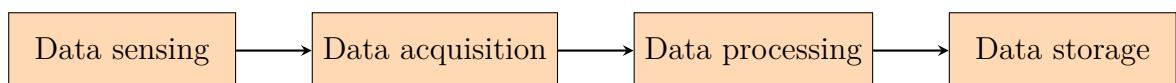


Figura 5: Diagramma di flusso dei dati

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 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.

- Le sterzate sia a sinistra che a destra brusche.
- 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.
- 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.
- Il carico del motore che deve essere compreso tra il 20% e il 50% per una buona guida
- Gli eccessi di velocità data dalla differenza tra i limiti imposti dalla legge e quelli reali del veicolo

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

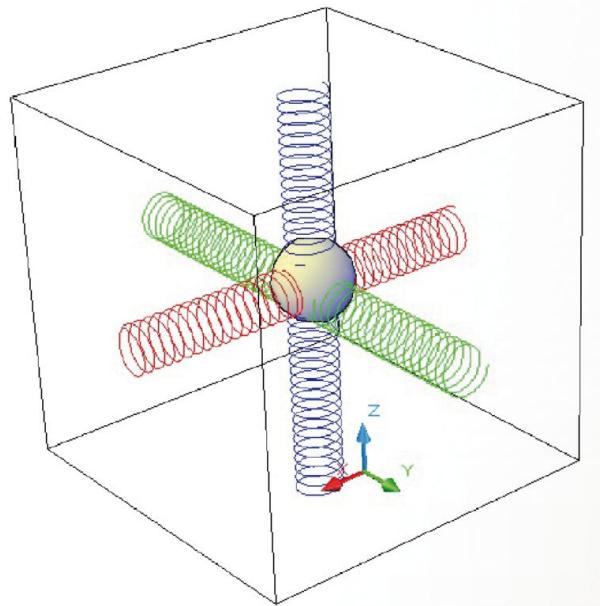


Figura 6: Modello meccanico di un accelerometro a 3 assi

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:

<b>Driver score</b>	<b>Driving behaviour</b>
< 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 1: Driver Evaluation Table

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.

---

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

# 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. (metti riferimento al libro del prof). Oltre a fornire i costrutti standard di un linguaggio

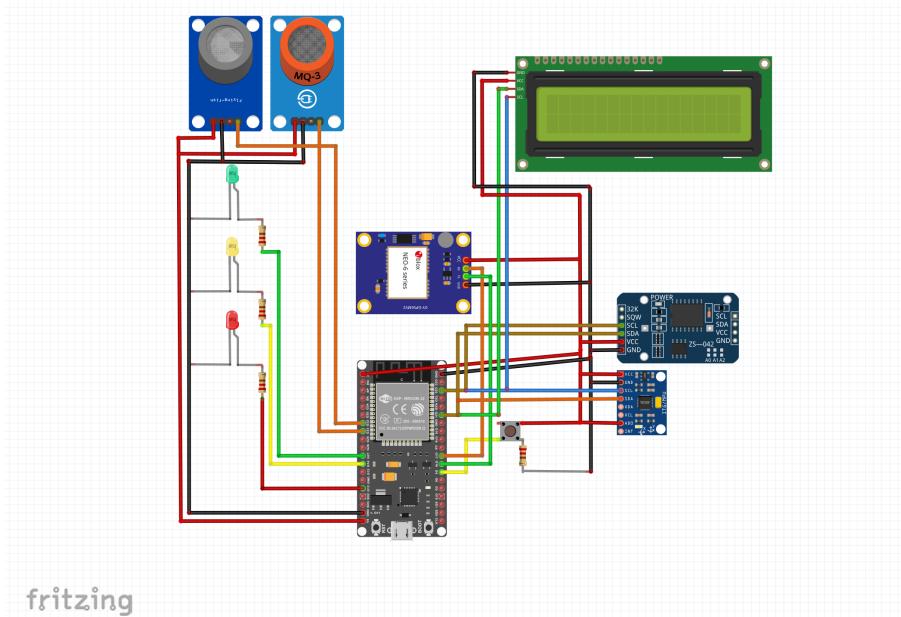


Figura 7: Schema di connessione dei vari componenti hardware della black box

imperativo, dato il contesto applicativo 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 per la blackbox. 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. 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:

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 2: 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

Tabella 3: 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** delitita 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.

<b>data</b>	<b>voto</b>	<b>lastStorage</b>	<b>synchronised</b>
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: 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.

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

Tabella 5: Tabella t4 allarmi critici

#### Tabella t4

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

Tabella 6: 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 **syncronised** 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). 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.

---

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

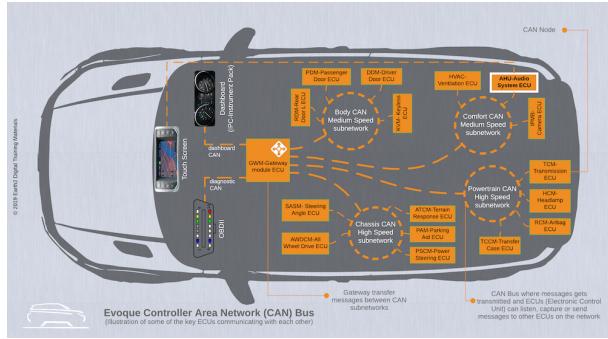


Figura 8: Protocollo CAN BUS all'interno di un'automobile

**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 (metti riferimento).

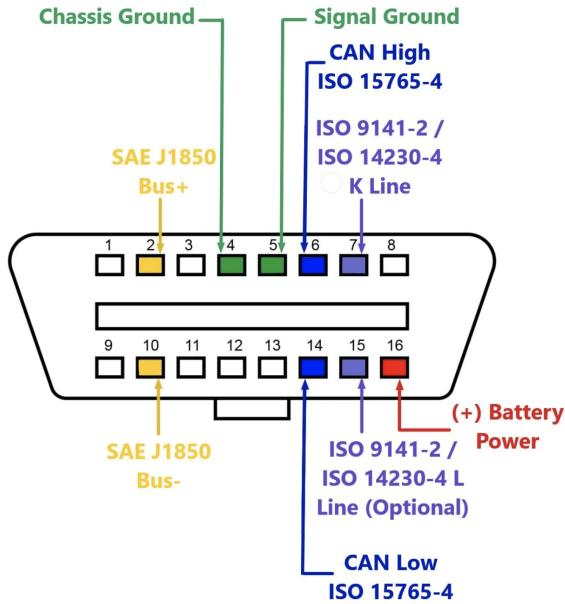


Figura 9: Pin dei cavi CAN sul connettore OBD-II

**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.

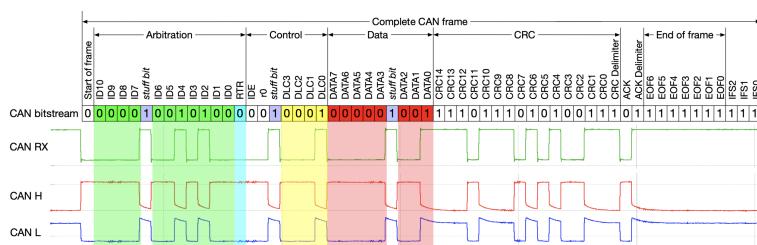


Figura 10: Formato dei pacchetti CAN standard

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

<sup>0</sup>Presada: [https://www.flexihub.com/images/upload/flexihub/articles/diagnostics/pinout\\_explanation.jpg](https://www.flexihub.com/images/upload/flexihub/articles/diagnostics/pinout_explanation.jpg)

<sup>6</sup>Presada: <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 11: 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.

<sup>0</sup>Presa da: [https://m.media-amazon.com/images/I/51crnV3RTbL.\\_AC\\_SX679\\_.jpg](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 12: Modulo GSM da integrare

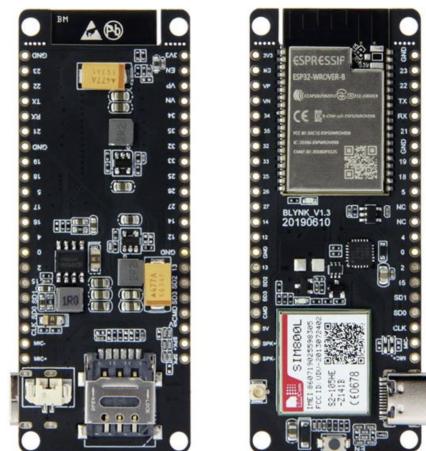


Figura 13: ESP32 SIM800L

---

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

<sup>0</sup>Presa 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.

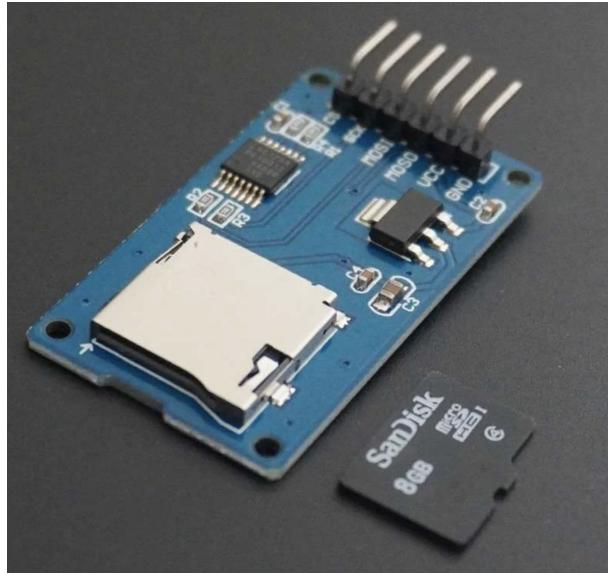


Figura 14: Micro SD card

**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:

---

<sup>0</sup>Preso 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.

**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 vorremo 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. Di seguito un esempio di classificazione binaria:

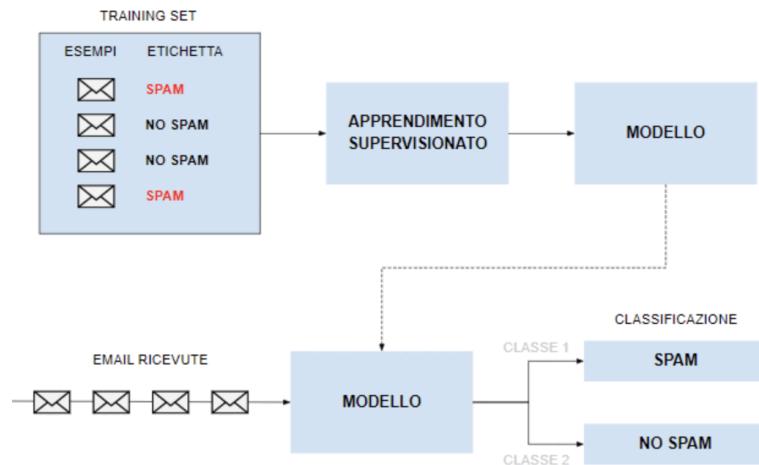


Figura 15: Esempio di classificazione binaria che stabilisce se una email è spam o no<sup>2</sup>

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

---

<sup>2</sup>Presa 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.

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)

---

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

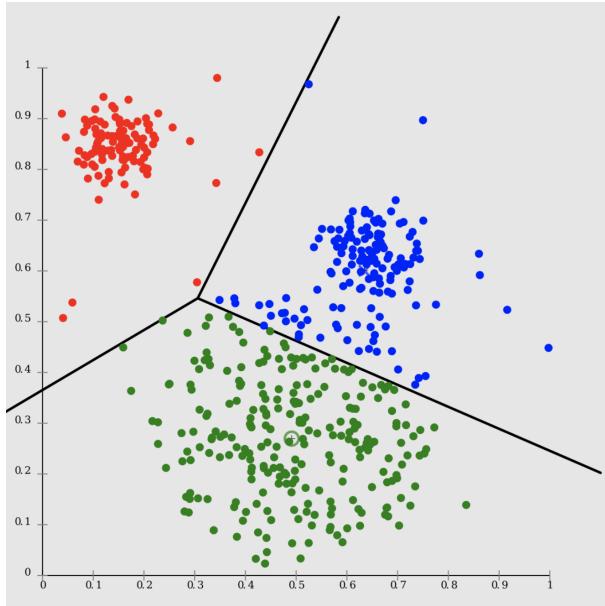


Figura 16: K-means usato per la classificazione di 3 partizioni<sup>3</sup>

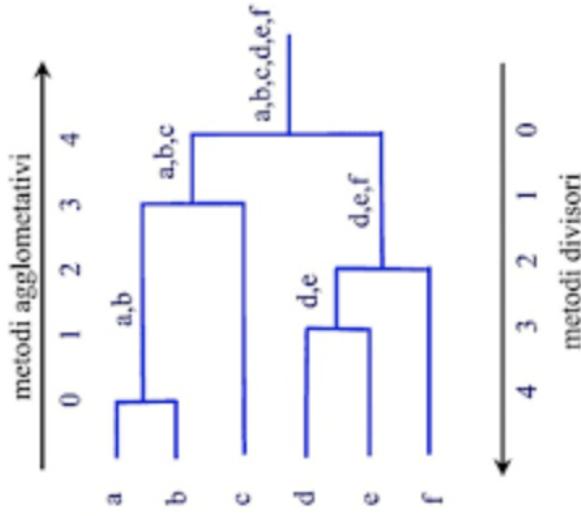
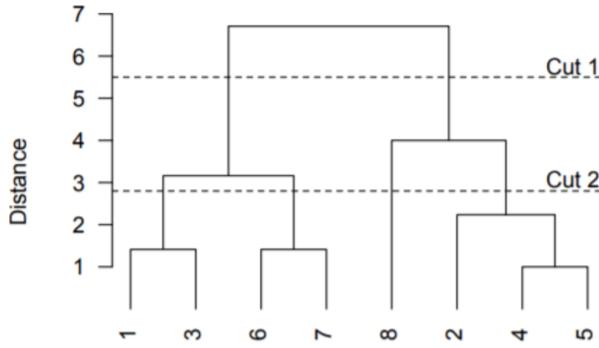
Una gerarchia di cluster può essere interpretata come un albero binario standard in cui la radice rappresenta tutti gli insiemi di oggetti dati da clusterizzare che formano il livello più alto della gerarchia (livello 0). Ad ogni livello, i nodi che sono il sottoinsieme dell'intero dataset corrispondono 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.

**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

---

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

<sup>5</sup>Preso 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>

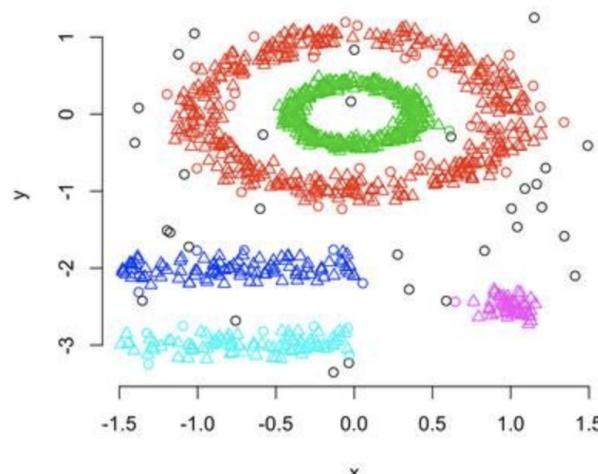
Figura 17: Esempio di dendogramma<sup>4</sup>Figura 18: Esempio di dendogramma con tagli gerarchici<sup>5</sup>

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 generare uno score del guidatore e di giudicarlo in base a questo, la possibilità di

---

<sup>6</sup>Presa 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 19: Esempio DBSCAN<sup>6</sup>

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.

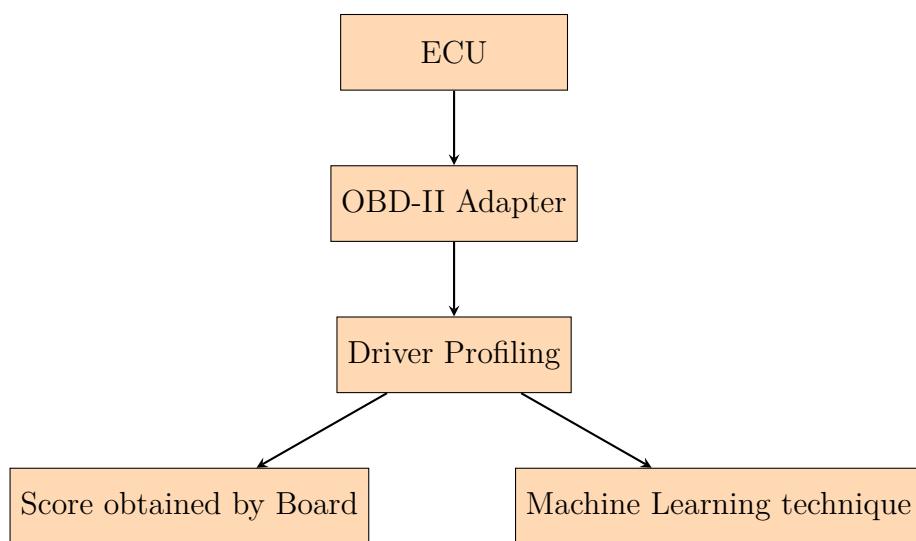


Figura 20: Diagramma di flusso del sistema di profilazione conducente