



Ricerca di Sistema elettrico

Report descrittivo delle attività di sviluppo informatico

M. Corazza, F. Karagulian, M. Lelli, C. Liberto, G. Valenti (ENEA)

REPORT DESCRITTIVO DELLE ATTIVITÀ DI SVILUPPO INFORMATICO

M. Corazza, F. Karagulian, M. Lelli, C. Liberto, G. Valenti (ENEA)

Aprile 2021

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dello Sviluppo Economico - ENEA

Piano Triennale di Realizzazione 2019-2021 - II annualità

Obiettivo: Tecnologie

Progetto: Tecnologie per la penetrazione efficiente del vettore elettrico negli usi finali

Work package: Mobilità

Linea di attività: 2.20 - Sviluppo di moduli di calcolo e simulazione della mobilità urbana e di protocolli di comunicazione IoT per la gestione real-time di flotte elettriche

Responsabile del Progetto: Claudia Meloni, ENEA

Responsabile del Work package: Maria Pia Valentini, ENEA

Indice

1	SOMMARIO	5
2	INTRODUZIONE	7
3	PROCESSAMENTO DELLE TRACCE GPS ED ESTRAZIONE DEI PATTERN DI MOBILITÀ	10
3.1	CONCETTI FUNZIONALI DEGLI FCD	10
3.1.1	Struttura dei file FCD	12
3.1.2	Preprocessamento delle collezioni FCD	14
3.1.3	Mappatura dei viaggi	17
3.1.4	Struttura del Database FCD	22
4	ELABORAZIONE DEI DATI DEL TRASPORTO PUBBLICO LOCALE	23
4.1	IL FORMATO GTFS (GENERAL TRANSIT FEED SPECIFICATION)	24
4.2	LO SCRIPT DI ESTRAZIONE DEI SERVIZI TPL	25
4.3	CALCOLO DELLE ALTITUDINI ALLE FERMATE DELLA RETE TPL	27
5	EMUCONS, MODULO PER IL CALCOLO DEI CONSUMI TTW E WTT PER VEICOLI PRIVATI E BUS	34
5.1	VEICOLI A COMBUSTIONE INTERNA: LA METODOLOGIA CORINAIR/COPERT	34
5.2	LE EMISSIONI DI CO ₂ NELLA FASE TTW	36
5.3	IL CASO DEGLI AUTOBUS	37
5.4	LE CURVE DI CONSUMO DEI VEICOLI ELETTRICI	38
5.5	LE COMPONENTI WTT	41
5.6	LA STRUTTURA DELLA FLOTTA	42
5.7	DESCRIZIONE DEL SOFTWARE EMUCONS	45
5.7.1	<i>carmain.py</i> :	45
5.7.2	<i>Busmain.py</i> :	46
5.7.2.1	Classe emucons	47
5.7.2.2	Readcopert	47
5.7.2.3	Selectenecons	47
5.7.2.4	Selectunique	47
5.7.2.5	Selectlines	47
5.7.2.6	Selectroutes	47
5.7.2.7	Writesheet	48
5.7.2.8	readENEAgdfs	48
5.7.2.9	Bus2line	48
5.7.2.10	Car2route	48
5.7.2.11	computecons	48
5.7.2.12	copFun	49
5.7.2.13	Convertcons	49
5.7.2.14	CO ₂	49
5.7.2.15	carcons	49
5.7.2.16	Wttene	50
5.7.2.17	Wttco2	50
5.8	ANALISI DEI RISULTATI	50
6	IL CALCOLO DEI CONSUMI PER I MODI DI TRASPORTO SU FERRO	58
7	STRUTTURA ED INTERFACCIA DELLA PIATTAFORMA SOFTWARE	61
7.1	DESCRIZIONE DELLA PIATTAFORMA E RAPPRESENTAZIONE DEI DATI	61
7.1.1	<i>Front-end</i>	62
7.1.2	<i>Back-end</i>	71
7.1.3	<i>Struttura file sorgenti</i>	72
7.1.3.1	Consumi_Bus (DB MySQL – lavorativo, prefestivo e festivo):	72
7.1.3.2	Stops.txt (GTFS data):	72
7.1.3.3	Hexagon_cover (file GEOJSON [26]):	72
7.1.3.4	Esa_attratti (file JSON [21]):	73
7.1.3.5	Esa_generati (file JSON):	73

7.1.3.6	Esa_parked (file JSON):	73
7.1.3.7	Hexagon FCD (file CSV):	74
7.1.3.8	Matched_routes (file GEOJSON):.....	74
7.1.3.9	Consumi_ore_giorno_mese (file CSV):	74
7.1.3.10	CO2 (file CSV):	74
8	CONCLUSIONI	75
9	RIFERIMENTI BIBLIOGRAFICI	76
10	ABBREVIAZIONI ED ACRONIMI	78

1 Sommario

Nel documento sono riportati i risultati ottenuti nel corso del secondo anno di attività, su un totale di tre anni, finalizzato all'integrazione e all'ampliamento dell'impianto modellistico della piattaforma software "EMU" sviluppata e sperimentata nell'area metropolitana di Roma nel corso del precedente periodo di ricerca relativo al PAR 2016-18.

La piattaforma "EMU" era stata ideata e sviluppata come strumento di ausilio allo studio di scenari di diffusione di flotte di veicoli elettrici nelle aree urbane per la stima e la visualizzazione su diverse scale spaziali e temporali delle potenziali fluttuazioni dei consumi elettrici e per la valutazione di soluzioni orientate allo sviluppo di una rete di ricarica pienamente adeguata alle esigenze di autonomia dei veicoli e di compatibilità con la capacità della rete elettrica.

La nuova versione della piattaforma software EMU è stata concepita con nuovi layer (o strati) informativi e un più esteso impianto modellistico capace di integrare, analizzare e interpretare open data e big data urbani riguardanti l'offerta, in termini di servizi e infrastrutture di trasporto, e la mobilità di veicoli passeggeri e merci.

L'obiettivo è di offrire un più ampio e articolato quadro conoscitivo sullo stato e sulla struttura della mobilità in termini di comportamento e abitudini dei viaggiatori, di accessibilità e fruizione degli spazi urbani, di offerta di servizi di trasporto collettivo, di consumi energetici per le diverse modalità di trasporto e di impatto sulle emissioni di gas responsabili del cambiamento climatico.

Il nuovo ambiente software consente di integrare, elaborare e visualizzare dati provenienti da fonti diverse per un più efficace governo della mobilità urbana in chiave sostenibile, basato sui criteri di riduzione dei consumi energetici e di decarbonizzazione del trasporto passeggeri e merci.

Il nuovo ambiente software è stato sviluppato mantenendo la tipica architettura Client-Server per essere accessibile via web da qualsiasi postazione collegata ad Internet senza la necessità di dover installare alcun software dedicato.

Durante la seconda annualità sono state svolte le seguenti attività di implementazione della piattaforma:

1. Sviluppo informatico delle funzioni lato Client e lato Server della nuova piattaforma;
2. Sviluppo del modello di simulazione della mobilità urbana passeggeri
3. Sviluppo e applicazione degli algoritmi di filtraggio, estrazione e analisi delle traiettorie di veicoli a partire da Floating Car Data;
4. Sviluppo e applicazione degli algoritmi di estrazione delle traiettorie dei mezzi di trasporto pubblico a partire dai dati GTFS;
5. Sviluppo e messa a punto degli algoritmi di calcolo dei consumi energetici Well-To-Tank e Tank-To-Wheel;

6. Sviluppo del modello di ottimizzazione della gestione operativa della distribuzione delle merci con veicoli elettrici

Nei seguenti paragrafi sono descritti i risultati relativi alle attività indicate nei punti 1, 3, 4 e 5. Le restanti attività, indicate nei punti 2 e 6, sono più diffusamente descritte nei deliverable specificatamente dedicati.

2 Introduzione

Le attività della presente ricerca sono finalizzate allo sviluppo di una piattaforma di supporto alle città per l'individuazione delle strategie ottimali di governo della mobilità orientate all'efficienza energetica e alla decarbonizzazione del settore del trasporto urbano oggi fortemente dipendente dall'uso di combustibili fossili e con alte potenzialità di risparmio energetico.

Base di partenza per l'attività di ricerca è la piattaforma eMU sviluppata e testata nel precedente periodo di ricerca assumendo come caso di studio l'area metropolitana di Roma [1].

eMU è stata ideata e realizzata per l'analisi e la valutazione di scenari legati alla diffusione di flotte di veicoli elettrici in contesti urbani e metropolitani con il proposito di fornire un nuovo strumento di ausilio alla pianificazione e progettazione degli interventi infrastrutturali e tecnologici atti a garantire una rapida ed efficiente transizione verso la mobilità elettrica. L'ambiente software di eMU, accessibile via WEB, consente di riprodurre e visualizzare su diverse scale spaziali e temporali le fluttuazioni della domanda di energia elettrica, corrispondenti a diverse ipotesi di diffusione dei veicoli elettrici, di configurazione dei punti di ricarica e di comportamento dei singoli individui nelle scelte di viaggio e di ricarica.

L'attività di ricerca in corso mira invece a potenziare e sviluppare ulteriormente la piattaforma eMU con nuovi layer informativi e un più esteso impianto modellistico capace di integrare, analizzare e interpretare open data e big data urbani riguardanti l'offerta di trasporto e la mobilità di persone e veicoli.

L'obiettivo è di offrire un più ampio e articolato quadro conoscitivo sullo stato e sulla struttura della mobilità in termini di comportamento e abitudini dei viaggiatori, di accessibilità e fruizione degli spazi urbani, di offerta di servizi di trasporto collettivo e, soprattutto, di consumi energetici per le diverse modalità di trasporto e di impatto sulle emissioni di gas climalteranti.

La ricerca mira in particolare a sviluppare un nuovo impianto modellistico capace di sfruttare sia i big data della mobilità, resi disponibili dalle moderne tecnologie di geo-localizzazione, sia gli open data provenienti da ambiti differenti al fine di trarne un più ampio e articolato quadro conoscitivo sulla struttura e dinamica della mobilità, sulla capacità attrattiva delle diverse zone, sulle prestazioni dei singoli modi di trasporto, sui consumi energetici finali e primari e sulle connesse emissioni di gas climalteranti del trasporto urbano. Il nuovo impianto modellistico consente di processare grandi quantità di tracce digitali di viaggi automobilisti, di elaborare gli open data in formato GTFS che descrivono percorsi, fermate e orari dei servizi di trasporto pubblico locale e i dati di domanda ed offerta degli operatori della distribuzione urbana delle merci, di simulare catene di spostamenti e percorsi su base settimanale, di calcolare i consumi energetici disaggregati per vettore energetico e modo di trasporto e di stimare le emissioni di gas serra considerando l'intero ciclo di vita dei vettori energetici utilizzati.

Un componente del nuovo impianto modellistico è rappresentato da un modello di concatenamento degli spostamenti in auto (Car Travel Chain Model) capace di riprodurre schemi di attività e sequenze di spostamenti e percorsi sul grafo stradale effettuati settimanalmente dalle auto private. La base utilizzata per la modellizzazione dei profili di mobilità delle auto private è costituita da consistenti dataset di tracce GPS registrate nell'area di studio presa in esame.

Un secondo modello, sviluppato dall'UDR DICEA della Sapienza Università di Roma, permette di elaborare i dati di domanda ed offerta degli operatori che effettuano consegne di merce ai destinatari finali, allo scopo di rendere più efficiente il processo logistico, riducendone i costi sostenuti dall'operatore e gli impatti energetici ed ambientali e di permettere una migliore gestione dei flussi di traffico veicolari e delle facilities pubbliche come piazzole di sosta per lo scarico delle merci e postazioni di ricarica di veicoli elettrici.

Ulteriori nuovi moduli riguardano il calcolo di consumi energetici ed emissioni climalteranti delle diverse componenti di trasporto. Il primo di questi è il modello di calcolo del consumo e delle emissioni climalteranti Tank-To-Wheel (TTW, letteralmente dal serbatoio alla ruota) associato ai singoli mezzi di trasporto, privato e pubblico su gomma, circolanti nell'area di applicazione. Si tratta di un modello di tipo disaggregato che stima il consumo finale di energia e le emissioni di gas serra per ogni veicolo in base alla categoria del veicolo stesso e ai dati di ogni singolo viaggio (percorrenza e velocità media) registrati dai veicoli sonda o simulati dal modello di generazione delle catene di spostamenti in auto su base giornaliera e settimanale.

Con analoghe funzionalità, ma diverso ambito di applicazione, è il modello sviluppato da UDR DICEA per il calcolo dei consumi elettrici del trasporto pubblico su ferro (Urban Rail Transport energy consumption model) a partire dai dati descrittivi del servizio offerto, codificati secondo lo standard GTFS, della tipologia e composizione dei treni, dei diagrammi del moto tra fermate, dell'andamento plano-altimetrico e del coefficiente di riempimento.

Infine, fa parte del nuovo impianto modellistico il modello per la stima dei consumi "Well-To-Tank" (WTT, letteralmente dal pozzo al serbatoio) in base ai processi di trasformazione e conversione energetica a monte del veicolo necessari per ottenere il vettore energetico (benzina, gasolio, CNG, GPL ed energia elettrica) per la propulsione dei veicoli. Questo modello è più ampiamente descritto nel Paragrafo 5 di questo documento.

Come più ampiamente descritto nei paragrafi successivi, l'impianto modellistico è stato inoltre integrato con procedure di processamento del dataset di tracce GPS per eliminare errori, incongruenze e valori anomali. A valle del processamento sono previste le procedure di estrazione e archiviazione di pattern e statistiche sulla mobilità veicolare pubblica e privata, sia passeggeri che merci, in termini di percorrenze, percorsi, tempi di viaggio, soste e catene di viaggio con prima origine ed ultima destinazione casa.

L'impianto modellistico comprende inoltre le procedure di lettura, integrazione ed archiviazione dei dati GTFS sul trasporto pubblico e dei layer geografici OpenStreetMap con la struttura topologica e funzionale della rete viaria e pedonale.

Il nuovo ambiente software è stato sviluppato mantenendo la tipica architettura Client-Server per essere accessibile via web da qualsiasi postazione collegata ad Internet senza la necessità di dover installare alcun software dedicato.

Come più ampiamente descritto nel seguito di questo documento, il software è organizzato in due parti: il front-end ed il back-end. Il front-end costituisce la parte software visibile all'utente che consente di interagire direttamente con l'ambiente di simulazione attraverso una rappresentazione visiva delle sue funzionalità (Figura 1: Schema funzionale della nuova piattaforma).

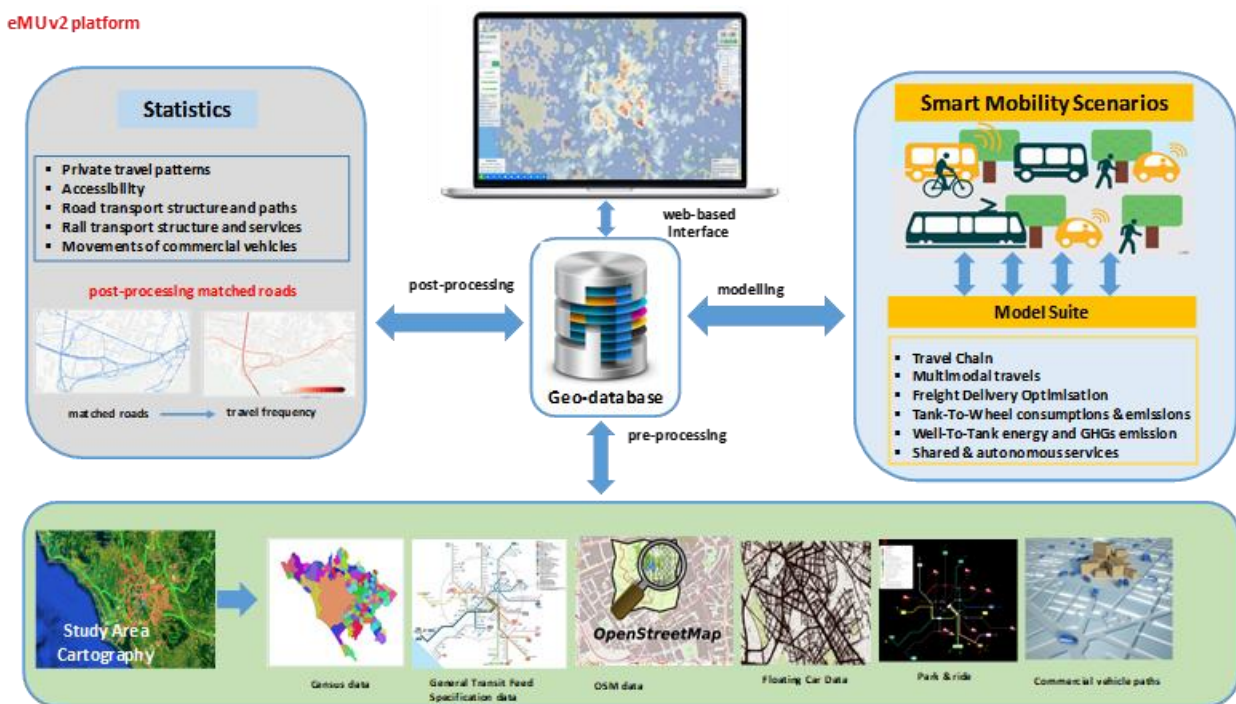


Figura 1: Schema funzionale della nuova piattaforma

In questo documento sono inoltre descritte le procedure software sviluppate per il processamento delle tracce GPS acquisite con tecniche Floating Car Data, l'estrazione di informazioni sulle scelte di viaggio, di percorso e sosta di un campione di veicoli e la stima della variabilità spaziale e temporale dei flussi di traffico.

Nel documento sono descritte anche le procedure software di estrazione ed integrazione dei dati open GTFS relativi alle corse, agli orari e alle fermate del trasporto pubblico locale, finalizzate specificatamente alla stima dei consumi energetici e delle relative emissioni di CO₂ per singola modalità di trasporto pubblico, su rotaia e su gomma.

3 Processamento delle tracce GPS ed estrazione dei pattern di mobilità

Il nuovo impianto modellistico si avvale di un insieme di procedure software per il filtraggio, la convalida e l'analisi delle tracce GPS.

Le procedure software svolgono inoltre la conversione e il caricamento dei dati e dei risultati in un geo-database relazionale dedicato, sviluppato in ambiente open source PostGIS-Postgresql. L'analisi delle tracce FCD include anche l'utilizzo di un algoritmo di map-matching per l'identificazione dei percorsi effettuati dai veicoli sulla rete stradale. Il processo di map-matching consente, in particolare, di associare le tracce GPS agli archi della rete digitalizzata e di studiare statisticamente le scelte di percorso e i tempi di percorrenza sugli archi attraversati, necessari anche alla calibrazione dei modelli di scelta dei percorsi.

Le procedure software sono state applicate a due distinti campioni di tracce FCD. Il primo campione di tracce GPS è stato acquisito dalla piattaforma telematica "OCTOTELEMATICS" che raccoglie in tempo reale dati trasmessi dai dispositivi connessi installati a bordo di veicoli. Il secondo campione include invece le tracce GPS raccolte dal sistema telematico VIASAT che acquisisce, sempre in tempo reale, dati trasmessi da sensori e dispositivi connessi.

Le stime effettuate sul campione "OCTOTELEMATICS" [2] evidenziano circa 170.000 auto residenti nella provincia di Roma corrispondenti a circa il 6,4% dell'intero parco auto circolante. Le tracce GPS sono riferite al mese di Maggio 2015. Le stime condotte sul campione "VIASAT" [3] rilevano circa 34.000 auto residenti, corrispondente a circa 1,3% del parco circolante nella provincia di Roma che nel 2015 ammontava a circa 2.665.000 auto (circa 615 auto ogni 1000 abitanti). Le tracce GPS VIASAT sono state registrate nel mese di Novembre 2019.

L'acquisizione della rete stradale digitalizzata è svolta tramite uno script in Python che estrae e converte in strato geografico le mappe digitali Open Street Map (OSM) disponibili su WEB. Lo script utilizza in particolare la libreria OSMnx per estrarre e convertire in modo completamente automatico la mappa digitale OSM di una specifica città in un grafo diretto con i relativi attributi geometrici e funzionali.

3.1 Concetti funzionali degli FCD

I Floating Car Data sono dati derivanti da veicoli sonda, che vengono raccolti dalle On Board Unit (OBU) installate, per lo più a scopi assicurativi, sui veicoli stradali. Questi costituiscono, quindi, dei "sensori mobili" nel flusso veicolare che non richiedono alcun tipo di installazione aggiuntiva lungo la sede stradale.

Le apparecchiature dei veicoli sonda contengono un localizzatore GPS (Global Positioning System), un accelerometro per la rilevazione dei crash ed un apparato di trasmissione/ricezione GSM/GPRS per lo scambio di dati fra il dispositivo e la Centrale Multiservizi.

L'affidabilità dei dati misurati dipende strettamente dalla qualità di ricezione del segnale GPS, che può risultare bassa quando il veicolo si mette in moto ed è richiesto più tempo per il calcolo della prima localizzazione (il "fixing"), in presenza di "canyon" urbani, quali vie strette o viali notevolmente alberati o nel caso di parcheggio all'interno di un garage. In condizioni ottimali di ricezione la precisione spaziale è dell'ordine dei 10 metri, mentre in condizioni di scarso segnale si può arrivare ad un errore di oltre 30 metri.

Naturalmente, esistono diversi sviluppi della tecnologia FCD che si differenziano per le prestazioni offerte: la precisione dei dati dipende sia dalla frequenza del campionamento e della trasmissione dei dati, dalla precisione del GPS e dal numero di veicoli sonda.

La differenza tra i dati FCD e quelli provenienti da sensori di traffico locali è che i primi misurano le caratteristiche del sistema traffico lungo l'infrastruttura, mentre i secondi in corrispondenza di una specifica sezione stradale per cui le diverse grandezze rilevate possono essere generalizzate nello spazio ipotizzando di assumere che il comportamento dei veicoli sia omogeneo e stazionario durante il periodo di osservazione e lungo il tratto stradale considerato. I dati rilevati dalle OBU presentano numerosi pregi, ai fini di un utilizzo per osservazioni ed analisi sulla mobilità. A differenza dei dispositivi fissi di monitoraggio, la tecnica FCD consente di ottenere informazioni più affidabili sui viaggi in termini di lunghezza dei percorsi e tempi di percorrenza, e di valutare i flussi origine-destinazione. La tecnica FCD è più economicamente sostenibile rispetto alle altre tecniche di monitoraggio: i dati sono acquisiti per altre finalità e, quindi, a costo marginale nullo ed inoltre i dati sono acquisiti in maniera regolare e protratta nel tempo, così da consentire analisi di tendenza.

La tecnica FCD fornisce il vantaggio di copertura dell'intera rete stradale: i dati sono diffusi sul territorio nazionale (ed extranazionale), anche se con intensità variabile; la tecnica FCD richiede inoltre minori oneri per la manutenzione.

E' da sottolineare tuttavia che nella fornitura standard, i dati provenienti dalle unità di bordo difettano di alcune informazioni importanti, come la tipologia e la destinazione d'uso del veicolo a cui sono riferiti o il percorso stradale utilizzato da un punto di rilevamento al successivo; inoltre, le rilevazioni non necessariamente garantiscono una significatività statistica sufficiente per poter estrapolare i risultati dell'analisi comportamentale dal campione all'universo.

Infine, essi riguardano solo un segmento della mobilità, quella realizzata attraverso veicoli stradali di proprietà privata di uso per lo più individuale, senza poter fornire indicazioni circa l'impiego delle altre modalità di trasporto o dei servizi pubblici.

La tecnologia FCD rappresenta una tecnica di acquisizione dei dati molto vantaggiosa ma occorre sempre valutare la rappresentatività delle informazioni acquisite dai veicoli sonda rispetto all'intera popolazione di veicoli circolanti.

Una delle principali limitazioni di questa tecnica è legata al tasso di penetrazione dei veicoli sonda nei volumi di traffico e alla frequenza di campionamento dei dati. Per questo motivo, quando si hanno tassi di penetrazione piuttosto bassi, occorre integrare le informazioni derivanti dalle OBU con i dati provenienti dalle varie tecnologie disponibili.

I dati provenienti dalle unità di bordo installate sui veicoli stradali, opportunamente integrati ed elaborati, rappresentano una risorsa preziosa per approfondire, dettagliare ed estendere la conoscenza della mobilità stradale privata nei suoi diversi aspetti, dai comportamenti individuali ed aggregati della domanda, ai livelli di congestione della rete e composizione dei flussi di traffico, sino agli impatti energetico-ambientali dell'esercizio dei veicoli stradali.

3.1.1 Struttura dei file FCD

Le procedure software implementate per l'archiviazione ed elaborazione delle tracce FCD sono riferite a due diverse collezioni di dati: OCTOTELEMATICS e VIASAT.

Octotelematics raccoglie questa tipologia di dato per fornire valore aggiunto ai servizi offerti dalle compagnie di assicurazione auto e dagli operatori di flotte veicolari. Il rilevamento Octotelematics riguarda le seguenti grandezze, associate ad un certo veicolo, identificato attraverso un codice ID univoco diverso dalla targa di immatricolazione:

1. La data di acquisizione dell'informazione (gg-mm-aa);
2. L'orario di acquisizione dell'informazione (hh:mm:ss);
3. Le coordinate spaziali del punto, in termini di latitudine e longitudine espresse in milionesimi di grado nel sistema WGS84;
4. La velocità istantanea del veicolo espressa in km/h;
5. La direzione di marcia del veicolo espressa in gradi;
6. La qualità del segnale GPS, espressa da un indice variabile tra 1 e 3 (assente, debole, buono);
7. Lo stato di moto del veicolo (0: accensione; 1: moto; 2: spegnimento);
8. la distanza in metri percorsa dalla traccia GPS immediatamente precedente.

L'unità di bordo memorizza tutte le informazioni appena descritte e, periodicamente, le trasmette ad un centro di elaborazione dati. L'OBUE memorizza le misurazioni GPS ogni 2 chilometri percorsi o, in alternativa, ogni 30 secondi quando il veicolo percorre un'autostrada o un tratto di viabilità urbana principale; ciò implica la necessità di una fase di processamento del dato per il matching rispetto alla rete stradale e la successiva individuazione dei percorsi.

In modo analogo, ma con alcune rilevanti diversità, la collezione FCD di Viasat include i seguenti attributi:

1. Identificativo anonimo del terminale Viasat (idterm)
2. Data e ora UTC della registrazione (timedate)

3. Latitudine del veicolo (in WGS84)
4. Longitudine del veicolo (in WGS84)
5. Velocità istantanea del veicolo in km/h (speed)
6. Direzione di marcia del veicolo (gradi sessagesimali, misurati dal Nord in senso orario: 0=Nord, 90=Est, 180=Sud, 270=Ovest) (direction).
7. Qualità del segnale GPS, espressa in decimi di HDOP, da un indice variabile compreso tra 1 (più accurato) e 150 (meno accurato) (grade).
8. Stato del motore del veicolo: 0= motore acceso, 1= motore spento (panel).
9. Tipo di veicolo sul quale è installato il terminale (car, fleet) (vehdtype).
10. La distanza in metri percorsa tra due tracce GPS consecutive (progressive).

Le registrazioni a bordo dei veicoli sono programmate ogni 30 secondi, dall'avvio del motore al suo arresto.

Questi attributi sono stati organizzati in una tabella di nome dataraw (Tabella 1: Tabella con attributi FCD) e quindi archiviata in un database PostgreSQL.

Tabella 1: Tabella con attributi FCD

idterm	idtrajectory	lat	lon	timedate	speed	grade	panel	vehdtype	segment	progressive
5435399	99752935	41.56	12.63	2019-10-29 15:37:35	0	8	1	1	0	0
5435399	99752935	41.57	12.63	2019-10-29 15:38:35	3	6	1	1	1	0
5435399	99752935	41.58	12.63	2019-10-29 15:40:35	9	8	1	1	2	200
5435399	99752935	41.59	12.64	2019-10-29 15:41:35	5	8	1	1	3	200
5435399	99752935	41.59	12.64	2019-10-29 15:47:35	2	7	1	1	4	300
5435399	99752935	41.59	12.64	2019-10-29 15:47:55	7	8	1	1	5	300
5914297	60819796	41.60	12.65	2019-10-29 18:33:48	0	8	1	1	0	0
5914297	60819796	41.61	12.65	2019-10-29 18:34:48	18	7	1	1	1	0
5914297	60819796	41.62	12.65	2019-10-29 18:38:48	40	7	1	1	2	400
5914297	60819796	41.63	12.65	2019-10-29 18:39:48	29	6	1	1	3	700
5914297	60819796	41.64	12.65	2019-10-29 18:40:48	41	6	1	1	4	1200
5914297	60819796	41.64	12.65	2019-10-29 18:41:48	37	7	1	1	5	1800

La figura seguente riporta l'area di campionamento dei dati FCD che interessa un'area di circa 700 km quadrati attorno la città di Roma:

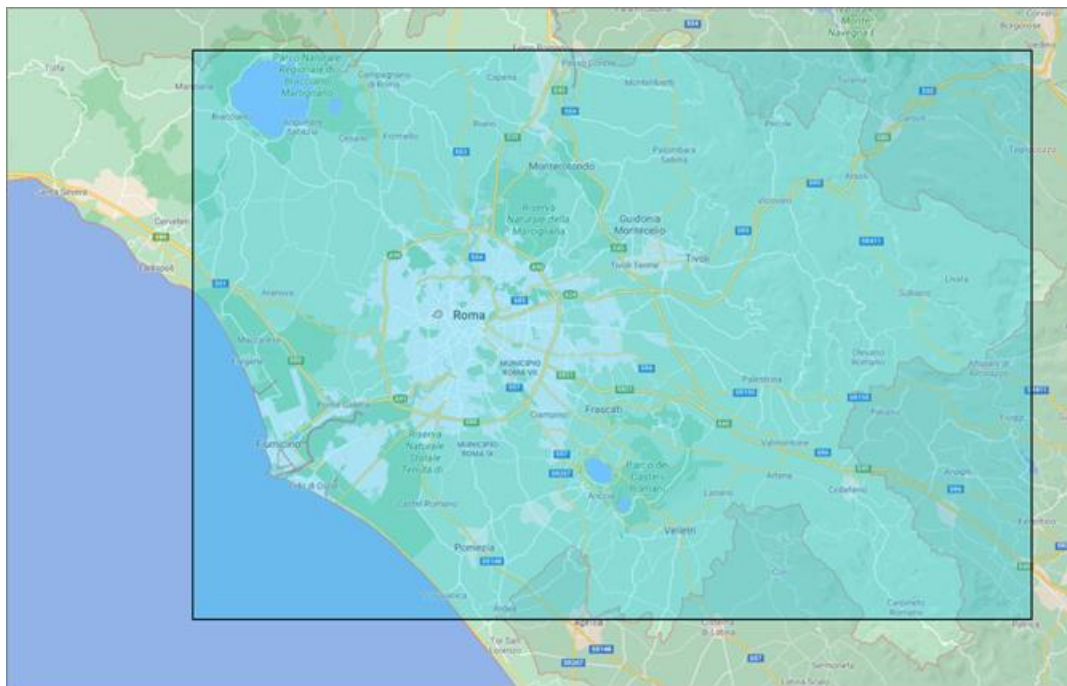


Figura 2: Area di estrazione dati Viasat nella provincia di Roma

3.1.2 Preprocessamento delle collezioni FCD

Gli FCD possono essere affetti da errori di misura e/o da malfunzionamenti dell’unità di bordo. Per questo motivo è stata necessaria una loro verifica e validazione prima di poterli utilizzare come input per le successive analisi. Un’elaborazione iniziale è stata effettuata per identificare e rimuovere gli outliers e per costruire un dataset organizzato per viaggi (idtrajectory) realizzati da ogni veicolo (idterm) a cui sono state associate le sequenze di tracce GPS registrate dall’unità di bordo durante gli spostamenti.

L’operazione di filtraggio e correzione dei dati è stata articolata in tre fasi:

1. Rimozione di registrazioni con valori ripetuti di latitudine e di longitudine.
2. Identificazione delle registrazioni di inizio di un nuovo viaggio ed assegnazione di un codice identificativo (idtrajectory). Per l’identificazione di un nuovo viaggio è stata considerata una soglia di 10 minuti tra due registrazioni successive.
3. Assegnazione della sequenza temporale di ogni traccia GPS (segment).

È stata quindi costruita una tabella di nome routecheck simile a quella dataraw con i dati FCD grezzi ma con l’assenza di outliers e con un campo riportante l’identificativo del viaggio e della sequenza temporale di ogni traccia e quindi archiviata nel database.

Successivamente sono state sviluppate delle procedure per individuare i malfunzionamenti dei dispositivi di bordo e codificare le diverse tipologie di errore o anomalie riscontrate.

Gli errori, sono stati classificati considerando i seguenti parametri:

- a) Qualità del segnale (grade). È stato scelto il valore soglia di 15. Le tracce GPS con un grade uguale od inferiore a 15 sono stati considerati più accurati rispetto alle tracce con un grade superiore a 15.
- b) Distanza euclidea maggiore della distanza percorsa. È stato verificato se la distanza euclidea calcolata tra due posizioni GPS temporalmente consecutive è maggiore della distanza progressiva chilometrica riportata con il dato FCD.
- c) Intervallo di tempo tra due tracce consecutive > 10 minuti. È stato verificato se la localizzazione temporale della traccia GPS corrisponde alla fine di un viaggio, che per convenzione è stato definito considerando un tempo di sosta tra due tracce consecutive della durata di 10 minuti.
- d) Velocità istantanea superiore a 250 km/h. Tutte le tracce che hanno riportato una velocità superiore a 250km/h sono state classificate come degli outliers.
- e) Progressiva chilometrica tra due tracce uguale a zero. È stato verificato se la distanza chilometrica percorsa da due tracce consecutive subisce un incremento diverso da zero.
- f) Stato del motore rimane invariato tra fine e inizio viaggio: sempre acceso (panel = 1, on-on) o sempre spento (panel = 0, off-off). Oppure se tra la fine di un viaggio e l'inizio del viaggio successivo lo stato del motore cambia. Estrazione dei viaggi e statistiche.

Per ogni sequenza di tracce GPS attribuita ad un viaggio (idtrajectory) e per ogni veicolo (idterm) è stata creata una tabella di nome route (Tabella 2: Tabella con attributi di viaggio) dove sono state quantificate le seguenti informazioni relative ad ogni singolo viaggio:

1. Distanza in metri percorsa lungo ogni singolo viaggio (tripdistance_m)
2. Tempo e data di origine del viaggio (timedate_o)
3. Tempo di sosta tra due viaggi successivi in secondi (breaktime_s)
4. Durata del viaggio in secondi (triptime_s)

Tabella 2: Tabella con attributi di viaggio

idtrajectory	idterm	tripdistance_m	timedate_o	breaktime_s	triptime_s	vehetype
82058056	3116923	8100	2019-09-30 07:15:39	37241	846	1
83013204	3116923	9500	2019-09-30 17:50:26	46453	1692	1
15705471	3116923	7900	2019-10-01 07:12:51	49627	862	1
29937667	3116923	10400	2019-10-01 21:14:20	34709	1128	1

Lo schema della tabella route consente quindi di archiviare tutti i viaggi effettuati da ciascun veicolo del campione FCD.

Dalla tabella route è inoltre possibile estrarre una serie di statistiche sulla mobilità e sull'uso dei singoli veicoli nell'area di studio:

1. Figura 3: Distribuzione della distanza percorsa durante tutti i viaggi effettuati nella provincia di Roma
2. Figura 4: Distribuzione dei tempi di sosta tra i viaggi effettuati nella provincia di Roma
3. Figura 5: Distribuzione dei tempi di viaggio effettuati nella provincia di Roma
4. Figura 6: Distribuzione delle velocità istantanee degli autoveicoli circolanti nella provincia di Roma



Figura 3: Distribuzione della distanza percorsa durante tutti i viaggi effettuati nella provincia di Roma

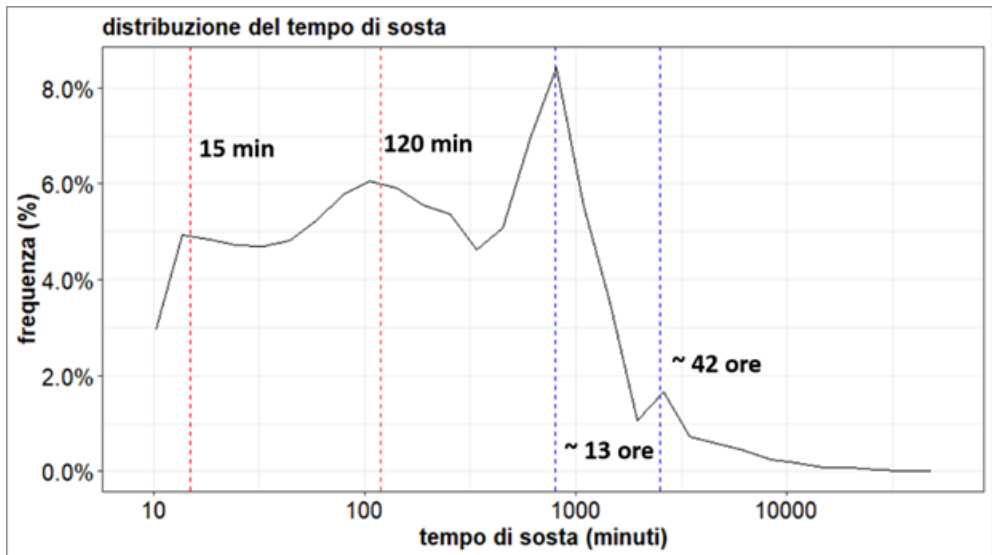


Figura 4: Distribuzione dei tempi di sosta tra i viaggi effettuati nella provincia di Roma

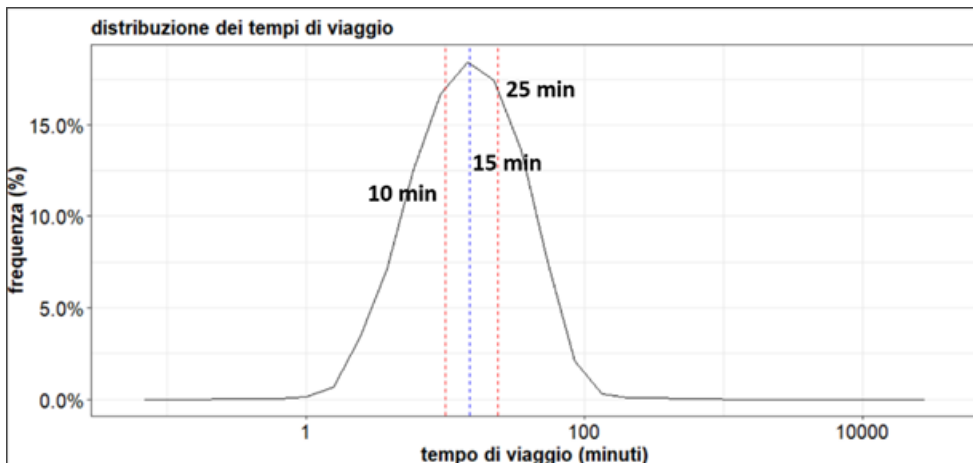


Figura 5: Distribuzione dei tempi di viaggio effettuati nella provincia di Roma

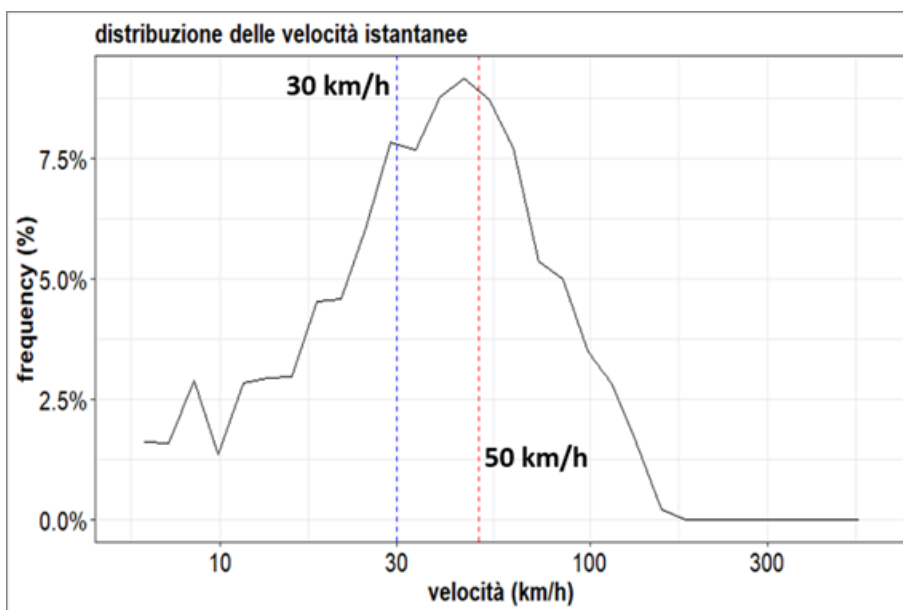


Figura 6: Distribuzione delle velocità istantanee degli autoveicoli circolanti nella provincia di Roma

3.1.3 Mappatura dei viaggi

L'identificazione delle traiettorie percorse dalle tracce FCD lungo gli archi della rete stradale di Roma è possibile attraverso l'utilizzo di un algoritmo di map-matching sviluppato e implementato dal laboratorio "Sistemi e Tecnologie per la Mobilità Sostenibile" (STMS) dell'ENEA [4].

L'algoritmo associa una sequenza di tracce GPS a delle traiettorie reali sulla rete. La metodologia è basata su un processo Markoviano che usa le distanze realmente percorse come input, la qualità del segnale GPS e la direzione della traiettoria di viaggio.

Per questo proposito, è stata esaminata la struttura topologica e geometrica della rete stradale ed è stata considerato il vincolo spaziale determinato dalla distanza tra due tracce consecutive. L'algoritmo di map-matching assegna quindi il nodo più probabile che si trova in prossimità di una traccia GPS.

Questa probabilità tiene in considerazione l'errore della traccia GPS (tipicamente 4-5 metri), e la distanza tra due nodi consecutivi che si trovino in prossimità di due tracce. Per ogni traccia l'algoritmo individua una possibile sequenza di archi che distano geometricamente meno di una certa distanza dalla traccia presa in esame. Ad ogni traccia è associato un buffer che identifica gli archi più probabili ed anche i nodi più probabili (Figura 7: rappresentazione grafica di tracce GPS (g_k) all'interno di una rete). Le tracce GPS sono indicate con g_k mentre i nodi sono indicati con X_i .

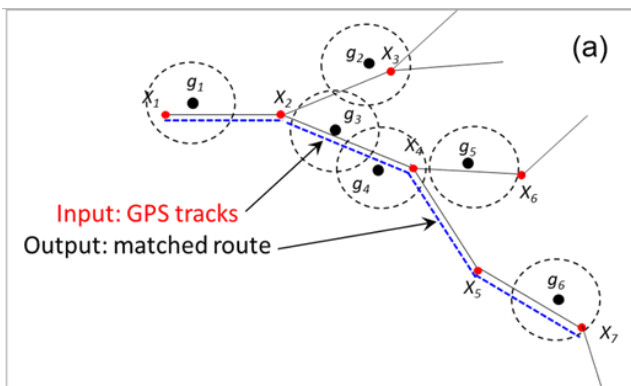


Figura 7: rappresentazione grafica di tracce GPS (g_k) all'interno di una rete

In una seconda fase viene individuata la sequenza di nodi in base alle informazioni ottenute dal grafo e di cui la lunghezza dei cammini minimi e le geometrie spaziali di archi e nodi. Inoltre, l'ordine dei nodi permette di determinare la direzione di marcia di ogni traccia.

Quindi, una volta identificati i nodi adiacenti ad ogni traccia, è possibile individuare la transizione più probabile tra nodi adiacenti. Se tra due tracce ci sono archi e nodi ma non sono nelle immediate prossimità di una traccia, è stato ipotizzato che il veicolo abbia scelto il cammino più breve.

Quindi il percorso simulato è ottenuto dalla sequenza di archi tra i nodi consecutivi più probabili. Il processo metodologico è illustrato in Figura 8 (Figura 8: rappresentazione della Catena di Markov).

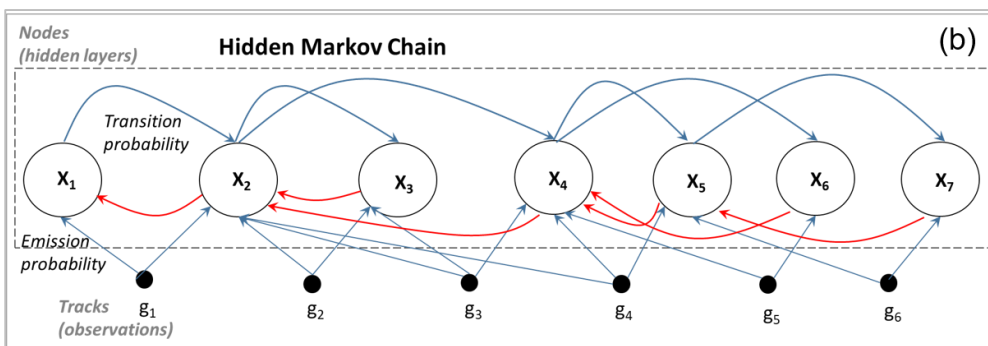


Figura 8: rappresentazione della Catena di Markov

Per ogni traccia g_k è stata definita una probabilità di associazione (emission probability o probabilità di emissione) ai nodi adiacenti come risultato dell'intersezione dei buffers con gli archi vicini. La probabilità di emissione è rappresentata generalmente da una distribuzione di tipo normale secondo l'equazione (1):

(1)

$$p_E(x_i|g_k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d_{ik}^2}{2\sigma^2}}$$

dove:

$d_{ki} = \{g_k/r_i\}$ è la distanza tra la traccia e l'arco adiacente appartenente al grafo, σ è la deviazione standard delle distanze tra una traccia e i nodi adiacenti (x_i, x_{i+1}) individuati tra quelli presenti in un buffer centrato sulla traccia. Per ogni nodo x_i , è quindi possibile ottenere la traccia GPS più vicina attraverso il calcolo della emission probability.

Un'altra grandezza definita come la probabilità di Transizione (Transition Probability) è intesa come la probabilità che un veicolo viaggi da un nodo x_i a quello vicino $x_j = x_{i+1}$ seguendo un determinato percorso. Alle transizioni tra un nodo e l'altro lungo un determinato arco, viene quindi assegnata una probabilità di transizione che considera la differenza tra la distanza Euclidea $d(x_i, x_{i+1})$ e la distanza del percorso più breve $D(x_i, x_{i+1})$ tra due nodi consecutivi. La transition probability viene quindi definita nell'equazione (2):

(2)

$$p_T(x_i|x_{i+1}) \sim e^{-[d(x_i, x_{i+1}) - D(x_i, x_{i+1})]}$$

dove $d(x_i, x_{i+1})$ è stato calcolato utilizzando l'algoritmo di [5].

Attraverso una serie iterativa di cicli, vengono determinati i nodi con i valori massimi di emission probability e di transition probability. I nodi più probabili vengono quindi ordinati in una lista secondo la loro sequenza temporale. Come illustrato in Figura 7 il collegamento dei nodi più probabili indica il percorso definitivo ricostruito dal map-matching. In questa fase viene anche calcolata la velocità media del veicolo tra i due nodi più probabili e la distanza percorsa.

Come output, l'algoritmo di map-matching fornisce una serie di risultati per ciascun veicolo (idterm) e per ciascun viaggio (idtrajectory) tra cui: la sequenza degli archi stradali attraversati identificati dai nodi alle sue estremità u e v , la data e l'ora di riferimento (timedate), la lunghezza dell'arco (length) e la velocità media (mean_speed) (Tabella 3: Esempio di Output del Map-Matching).

Tabella 3: Esempio di Output del Map-Matching

idterm	idtrajectory	timedate	U	v	sequenza	mean_speed	length
2752542	108495646	2019-10-09 04:17:56	1864769418	1864769280	0	66	363405
2752542	108495646	2019-10-09 04:17:56	1864769280	322288972	1	66	579575
2752542	108495646	2019-10-09 04:17:56	322288972	908324173	2	66	196902
2752542	108495646	2019-10-09 04:17:56	908324173	908324254	3	66	83335
2752542	108495646	2019-10-09 04:18:50	908324254	908324258	4	24	799012
2752542	108495646	2019-10-09 04:18:50	908324258	908324254	5	24	799012
2752542	108495646	2019-10-09 04:18:50	908324254	1645048080	6	24	281921
2752542	108495646	2019-10-09 04:18:50	1645048080	908324170	7	24	134674
2752542	108495646	2019-10-09 04:18:50	908324170	908324227	8	24	87819
2752542	108495646	2019-10-09 04:18:50	908324227	908324241	9	24	449225

L’accuratezza dell’algoritmo di map-matching, ossia la percentuale di viaggi mappati sulla rete stradale rispetto al totale di tutti i viaggi, è di circa l’85%.

L’imprecisione dell’algoritmo deriva dal fatto che non è sempre possibile determinare con certezza gli archi stradali attraversati dal veicolo a partire dalla sequenza di tracce GPS registrate. Ciò è dovuto principalmente ad imprecisioni del rilevamento GPS, da malfunzionamenti dell’unità di bordo e, in minima parte, da errori topologici presenti nel grafo digitalizzato.

L’algoritmo di map-matching è molto dispendioso dal punto di vista computazionale poiché elabora le tracce puntuali di ogni singolo veicolo su tutta la rete stradale. Quindi, per mappare tutte le traiettorie registrate durante il mese di ottobre 2019 è stato implementato un modello calcolo parallelo che consiste nell’elaborazione simultanea dei dati FCD riferiti ad ogni singolo veicolo. Questo ha consentito l’elaborazione di circa 2.5 milioni di viaggi durante un tempo di calcolo di circa 20 giorni.

Ogni percorso mappato è formato da una sequenza di archi che viene archiviata nel database in una tabella di nome mapmatching. La Figura 9 mostra la mappatura di tutti i dati FCD sul grafo stradale relativi ad un giorno del mese di ottobre 2019 per i veicoli privati e la Figura 10 per i veicoli commerciali. Come si può osservare, i percorsi effettuati dai veicoli privati sono maggiormente concentrati sulle arterie principali che permettono l’accesso all’interno della città di Roma, sul Grande Raccordo Anulare, le autostrade e le strade provinciali (Figura 9: Esempio di mappatura dei veicoli Viasat sulla rete stradale della provincia di Roma durante il Mercoledì 9 Ottobre 2019 per le automobili). D’altra parte, i percorsi effettuati dai veicoli commerciali sono maggiormente concentrati sul Grande Raccordo Anulare e sulle autostrade (Figura 10:

Esempio di mappatura dei veicoli Viasat sulla rete stradale della provincia di Roma durante il Mercoledì 9 Ottobre 2019 per i veicoli commerciali).

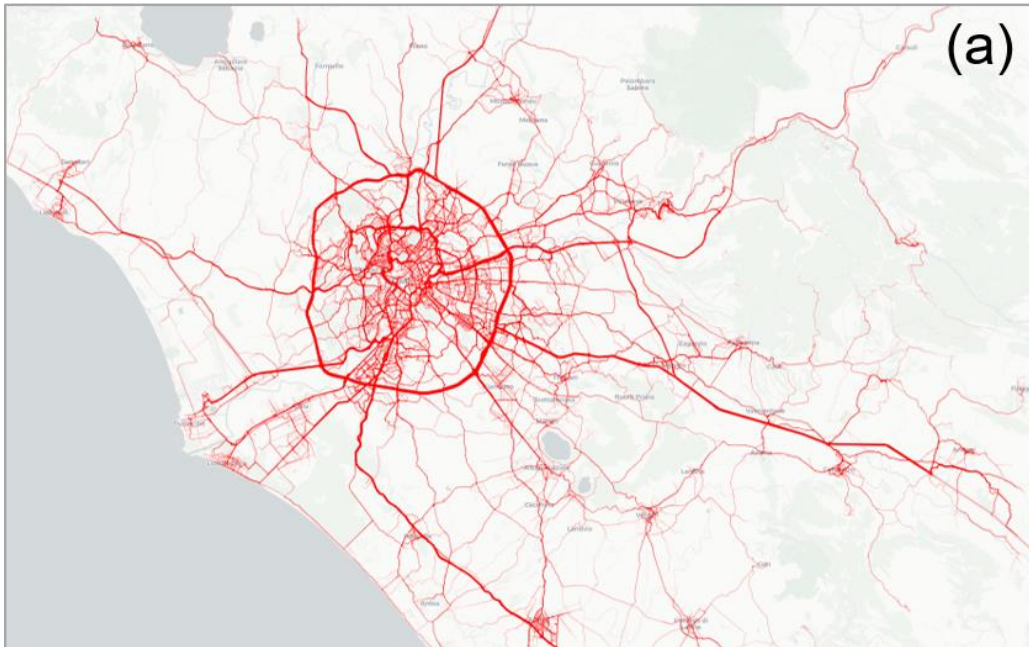


Figura 9: Esempio di mappatura dei veicoli Viasat sulla rete stradale della provincia di Roma durante il Mercoledì 9 Ottobre 2019 per le automobili

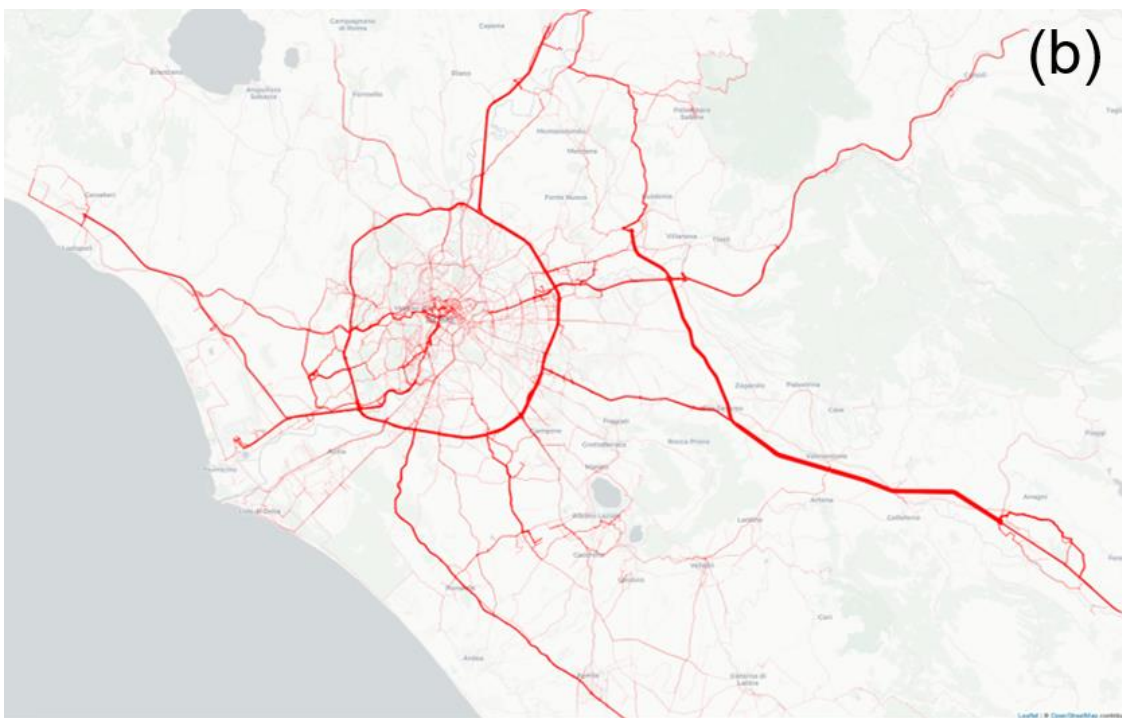


Figura 10: Esempio di mappatura dei veicoli Viasat sulla rete stradale della provincia di Roma durante il Mercoledì 9 Ottobre 2019 per i veicoli commerciali

3.1.4 Struttura del Database FCD

Come riportato di sopra, tutti i dati di input e di output generati dai diversi algoritmi per l'elaborazione dei viaggi e della mappatura delle loro traiettorie sulla rete stradale di Roma sono stati organizzati in tabelle e archiviati in un database utilizzando l'ambiente PostgreSQL.

La Figura 11 riporta la struttura del database utilizzato per identificare i viaggi, le loro traiettorie ed il modello di map-matching.

La tabella di input è quella contenente i dati FCD grezzi (dataraw), ossia l'insieme delle tracce GPS con i relativi attributi. Questa tabella è relazionata ad un'altra tabella di nome obu attraverso l'identificativo unico di ogni veicolo. La tabella obu, riporta alcuni metadati del veicolo come marca, modello, e portata (quando disponibile).

Di seguito si riporta una descrizione sintetica del contenuto informativo di ogni tabella:

- a) La tabella obu archivia le informazioni fornite dai dati FCD di VIASAT per ogni veicolo identificato anonimamente con un numero idterm. Il tipo di veicolo devicetype indica se il veicolo è un'automobile (car) od appartiene ad una flotta commerciale (fleet).
 - b) La tabella routecheck associa la sequenza di viaggi (idtrajectory) ad ogni veicolo secondo il suo identificativo unico (idterm) oltre eventuali anomalie riscontrate nei dati di input.
 - c) La tabella route riporta i parametri che caratterizzano ogni viaggio (idtrajectory) identificato nella tabella routecheck quali: distanza percorsa durante il viaggio, tempo di viaggio, tempo di sosta, coordinate di origine e destinazione e geometria della traiettoria del viaggio.
 - d) La tabella mapmatching riporta tutti i nodi (u, v) che risultano dall'assegnazione degli archi per ogni sequenza di dati FCD corrispondenti ad ogni viaggio (idtrajectory) e veicolo (idterm). Questa tabella contiene quindi tutti gli output dell'algoritmo di map-matching.
 - e) La tabella edges contiene tutti i parametri del grafo stradale ottenuti da OpenStreetMap quali: i codici identificativi degli estremi (nodi) degli archi u, e v, la lunghezza di ogni arco, la tipologia di strada (highway), il nome della strada ec
- Di seguito si riporta una descrizione sintetica del contenuto informativo di ogni tabella:
- f) La tabella obu archivia le informazioni fornite dai dati FCD di VIASAT per ogni veicolo identificato anonimamente con un numero idterm. Il tipo di veicolo devicetype indica se il veicolo è un'automobile (car) od appartiene ad una flotta commerciale (fleet).
 - g) La tabella routecheck associa la sequenza di viaggi (idtrajectory) ad ogni veicolo secondo il suo identificativo unico (idterm) oltre eventuali anomalie riscontrate nei dati di input.

- h) La tabella route riporta i parametri che caratterizzano ogni viaggio (idtrajectory) identificato nella tabella routecheck quali: distanza percorsa durante il viaggio, tempo di viaggio, tempo di sosta, coordinate di origine e destinazione e geometria della traiettoria del viaggio.
- i) La tabella mapmatching riporta tutti i nodi (u, v) che risultano dall'assegnazione degli archi per ogni sequenza di dati FCD corrispondenti ad ogni viaggio (idtrajectory) e veicolo (idterm). Questa tabella contiene quindi tutti gli output dell'algorithmo di map-matching.
- j) Labella edges contiene tutti i parametri del grafo stradale ottenuti da OpenStreetMap quali: i codici identificativi degli estremi (nodi) degli archi u, e v, la lunghezza di ogni arco, la tipologia di strada (highway), il nome della strada etc.

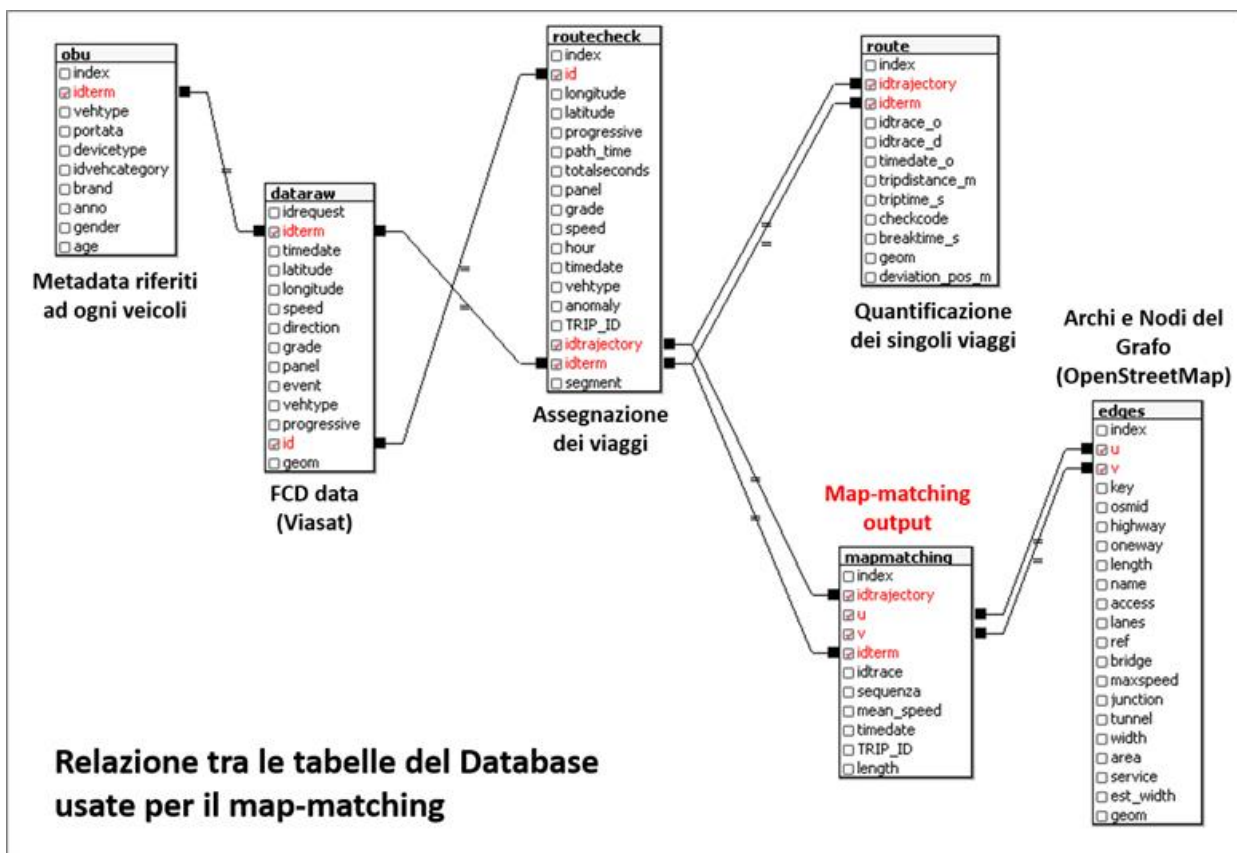


Figura 11: Struttura del database FCD

4 Elaborazione dei dati del Trasporto Pubblico Locale

Nel corso del secondo anno di attività sono state sviluppate e applicate al caso di studio di Roma le procedure di estrazione ed integrazione dei dati, disponibili nel formato open data "GTFS", sul programma di esercizio del servizio di trasporto pubblico locale. Il GTFS è un formato standard nato per definire orari e informazioni geografiche relativi al servizio erogato dalle aziende di trasporto pubblico locale.

L'output finale della procedura comprende tutti i dati necessari al calcolo delle prestazioni energetiche e delle emissioni di gas climalteranti per singola modalità di trasporto pubblico, su rotaia e su gomma.

L'output finale comprende inoltre i dati per la rappresentazione su layer cartografici di linee, fermate e frequenze del servizio.

4.1 Il formato GTFS (*General Transit Feed Specification*)

I dati di trasporto per la mobilità pubblica, indicati come GTFS, sono degli open data ben conosciuti da tutte le agenzie di trasporti internazionali. È grazie ai dati GTFS che queste agenzie possono comunicare con il ben noto motore di ricerca Google Maps. Le specifiche dettagliate dei dati GTFS si possono trovare al seguente link: <https://developers.google.com/transit/gtfs/reference/>.

I dati GTFS si possono scaricare da siti web sulla mobilità pubblica di molte città e/o provincie. Di solito i dati GTFS sono forniti in formato compresso (.zip files) il cui contenuto racchiude una raccolta di files in formato testo.

Una tipica collezione di dati GTFS è descritta dai seguenti file [1]:

- a) agency.txt, con i dati codificati sull'agenzia di trasporto;
- b) stops.txt, con i dati sulle paline segnalanti le fermate per la salita e la discesa dei passeggeri;
- c) routes.txt, con la descrizione delle singole linee;
- d) trips.txt, con la descrizione del singolo percorso, fermata per fermata, di ogni linea;
- e) stop_times.txt, con gli orari di passaggio alle singole fermate per ogni singolo trip;
- f) calendar_dates.txt, con la descrizione del periodo e dei giorni della settimana in cui sono attive le corse presenti nel file trips.txt;
- g) shapes.txt, con la descrizione del percorso su strada effettuato da ogni singolo trip inteso come sequenza di coppie di coordinate geografiche nel formato WGS84.
- h) feed_info.txt, con informazioni aggiuntive sulla validità dell'intero pacchetto di file.

Una collezione di file GTFS può includere, opzionalmente, anche i seguenti file:

- a) fare_attributes.txt, con le informazioni sulle tariffe per singole linee
- b) fare_rules.txt, con le regole per l'applicazione delle tariffe
- c) frequencies.txt, con una rappresentazione compressa del servizio a orario fisso
- d) transfers.txt, con le regole per il trasbordo tra linee
- e) pathways.txt, con le informazioni sui percorsi all'interno delle stazioni

Questo archivio può essere considerato a tutti gli effetti come un insieme di tabelle collegate tra loro da campi comuni (keys) (Figura 12: Relazione tra files presenti nell'archivio GTFS). Nel suo complesso, l'archivio completo raccoglie diversi aspetti degli orari del trasporto pubblico e la posizione geografica dei punti di fermata dei veicoli. Se disponibili, anche le informazioni delle tariffe (costi) inerenti ai percorsi tra fermate sono incluse nell'archivio dei dati GTFS.

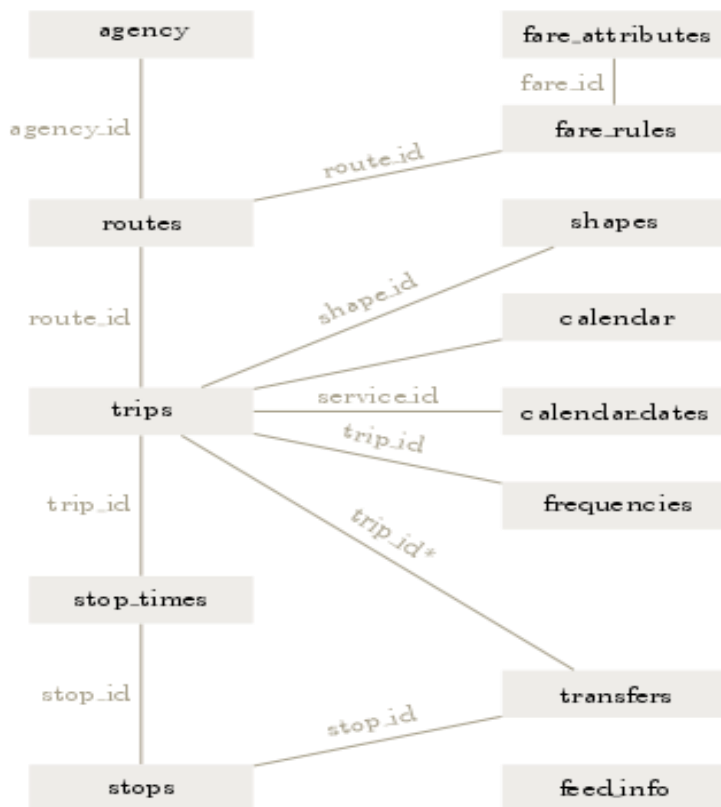


Figura 12: Relazione tra files presenti nell'archivio GTFS

4.2 Lo script di estrazione dei servizi TPL

Il software sviluppato per la lettura e l'estrazione dei dati del servizio TPL è articolato nei seguenti passi sequenziali:

1. Lettura del file Calendar_dates.txt ed estrazione dei servizi attivi per la giornata di riferimento
2. Lettura dei file "routes.txt" ed "trips.txt" ed estrazione dei dati del servizio TPL per singola linea e corsa
3. Lettura del file "stop_times.txt" ed estrazione delle singole tratte tra fermate adiacenti di ogni singola corsa
4. Lettura del file "stop.txt", estrazione delle informazioni di fermata e calcolo delle altimetrie
5. Lettura del file shape.txt, creazione delle linestring tra fermate adiacenti per ciascuna corsa e calcolo delle distanze tra fermate adiacenti
6. Generazione dei file con i risultati aggregati per singola tratta delle corse programmate nella giornata di riferimento.

In particolare, al passo 1 sono estratti i codici dei servizi programmati (service_id) di TPL nella giornata di riferimento. Il codice "service_id" viene poi utilizzato al passo 2 per l'estrazione dei codici di linea "route_id" e di corsa "trip_id" riportati rispettivamente nei file routes.txt e trips.txt. Al passo 2, insieme al codice corsa

“trip_id”, viene estratto anche codice “shape_id”, associato alla corsa, utilizzato per individuare il tracciato definito nel file shape_txt.

Al passo 3 è prevista l'estrazione delle singole tratte del percorso programmato per ciascuna corsa “trip_id”. Le tratte sono individuate tra fermate adiacenti a partire dalla fermata capolinea di partenza. Le informazioni estratte in questo passo sono il codice identificativo e il numero progressivo della fermata, l'orario previsto di arrivo e partenza e la lunghezza della tratta in metri se disponibile.

Al passo 4, le singole tratte di ogni corsa sono completate con le informazioni contenute nel file stop.txt; in particolare il nome della fermata, la coppia di coordinate geografiche nel formato WGS84 e l'altimetria sul livello del mare, input necessario alla stima delle pendenze stradali e, successivamente, al calcolo dei consumi di carburante e delle emissioni di CO₂. Per il calcolo dell'altimetria è stata sviluppata una specifica procedura software più dettagliatamente descritta nella sezione seguente.

Al passo 5, viene eseguita l'operazione di estrazione delle “linestring” con le informazioni geografiche dei percorsi tra fermate adiacenti. Una singola “linestring” rappresenta una sequenza di punti e i segmenti lineari che li connettono. Le informazioni geografiche delle “linestring” sono utilizzate per calcolare le percorrenze in metri tra fermate adiacenti. Per l'estrazione delle singole “linestring” tra fermate adiacenti di una corsa è stata sviluppata una procedura di segmentazione della polyline rappresentativa dell'intero percorso della corsa fornita nel file shape.txt. Per la segmentazione delle tratte si individua per ogni fermata il vertice della polyline più vicino.

Infine al passo 6 viene eseguita l'operazione di generazione dei file per il TPL su gomma (autobus) e su ferro (metro, tram e treni locali) con tutte le informazioni aggregate per singola tratta e corsa programmata.

A titolo di esempio nella Tabella 4 sono riportati i risultati della procedura implementata per il caso di una corsa del servizio bus di Roma.

Nella tabella sono riportate le 38 tratte della corsa “2495-10” della linea “628” che collega i due capolinea “Baronio” e “Volpi/Farnesina”. Per ogni tratta tra due fermate consecutive sono riportati gli orari di partenza dalla fermata a monte e di arrivo alla fermata a valle, il numero progressivo della tratta, il tempo, in secondi, e la distanza percorsa in metri tra le due fermate, la “linestring” con le informazioni geografiche del percorso tra le due fermate e le quote altimetriche delle fermate espressi in metri sopra il livello del mare.

Tabella 4: Esempio di output della procedura di elaborazione e integrazione della collezione GTFS

route	route	trip_id	from_stop_name	Departure	to_stop_name	Arrival	progress	timeinsec	dist_metr	geom	fromstop_id_elevation	tostop_id_elevation
bus	628	2495-10	Baronio	12:03:00	De Cesare	12:04:02	1	62	242	[[41.872483, 12.520062], [41.872755, 44.3	44.3	43.4
bus	628	2495-10	De Cesare	12:04:02	Villani	12:05:06	2	64	241	[[41.87163, 12.519872], [41.871442, ;	43.4	37.7
bus	628	2495-10	Villani	12:05:06	Platina	12:05:56	3	50	186	[[41.870285, 12.51793], [41.870182, ;	37.7	35
bus	628	2495-10	Platina	12:05:56	Coppi A.	12:06:40	4	44	164	[[41.86941, 12.516908], [41.869407, ;	35	33.2
bus	628	2495-10	Coppi A.	12:06:40	Macedonia	12:07:28	5	48	181	[[41.870037, 12.515408], [41.869862, ;	33.2	31.4
bus	628	2495-10	Macedonia	12:07:28	Macedonia/Gregorovii	12:08:22	6	54	208	[[41.870177, 12.513712], [41.870455, ;	31.4	36.4
bus	628	2495-10	Macedonia/Gregorovii	12:08:22	Numanzia/Latina	12:09:24	7	62	234	[[41.87193, 12.51281], [41.872128, 12	36.4	39
bus	628	2495-10	Numanzia/Latina	12:09:24	Satrico/Zama	12:11:28	8	124	468	[[41.87389, 12.511837], [41.874298, ;	39	36.9
bus	628	2495-10	Satrico/Zama	12:11:28	Satrico/Collazia	12:12:16	9	48	181	[[41.876387, 12.509177], [41.876353, ;	36.9	38.1
bus	628	2495-10	Satrico/Collazia	12:12:16	Epiro	12:13:50	10	94	355	[[41.877237, 12.507322], [41.877442, ;	38.1	40
bus	628	2495-10	Epiro	12:13:50	Pannonia/De Mattias	12:15:12	11	82	310	[[41.878858, 12.504048], [41.87885, ;	40	31.9
bus	628	2495-10	Pannonia/De Mattias	12:15:12	Pannonia/Tracia	12:15:58	12	46	178	[[41.880215, 12.50116], [41.880322, ;	31.9	28.1
bus	628	2495-10	Pannonia/Tracia	12:15:58	Terme Caracalla/Valle	12:18:54	13	176	667	[[41.881477, 12.499858], [41.881505, ;	28.1	24.2
bus	628	2495-10	Terme Caracalla/Valle	12:18:54	Terme Caracalla/Porte	12:21:00	14	126	473	[[41.881003, 12.494872], [41.88125, ;	24.2	23.9
bus	628	2495-10	Terme Caracalla/Porte	12:21:00	Cerchi/Porta Capena	12:22:10	15	70	266	[[41.883578, 12.490418], [41.883613, ;	23.9	20.7
bus	628	2495-10	Cerchi/Porta Capena	12:22:10	Cerchi/Bocca Della Ve	12:24:48	16	158	600	[[41.885355, 12.488473], [41.886477, ;	20.7	17.1
bus	628	2495-10	Cerchi/Bocca Della Ve	12:24:48	Petroselli	12:25:52	17	64	243	[[41.888575, 12.48268], [41.888763, ;	17.1	16.1
bus	628	2495-10	Petroselli	12:25:52	Ara Coeli/P.za Venezia	12:27:44	18	112	422	[[41.890152, 12.480963], [41.890422, ;	16.1	17.1
bus	628	2495-10	Ara Coeli/P.za Venezia	12:27:44	Plebiscito	12:30:06	19	142	533	[[41.893538, 12.481805], [41.893697, ;	17.1	20
bus	628	2495-10	Plebiscito	12:30:06	Argentina	12:31:40	20	94	359	[[41.896467, 12.480985], [41.896302, ;	20	16.2
bus	628	2495-10	Argentina	12:31:40	Rinascimento	12:33:00	21	80	302	[[41.896122, 12.476855], [41.896112, ;	16.2	16
bus	628	2495-10	Rinascimento	12:33:00	Senato	12:34:28	22	88	334	[[41.897277, 12.474218], [41.897753, ;	16	16
bus	628	2495-10	Senato	12:34:28	Zanardelli	12:35:30	23	62	232	[[41.900265, 12.473913], [41.900438, ;	16	17
bus	628	2495-10	Zanardelli	12:35:30	Lgt Marzio	12:36:48	24	78	298	[[41.901383, 12.472187], [41.901842, ;	17	16.5
bus	628	2495-10	Lgt Marzio	12:36:48	Lgt Marzio/Porto Di Ri	12:37:30	25	42	160	[[41.90301, 12.473882], [41.903305, ;	16.5	16
bus	628	2495-10	Lgt Marzio/Porto Di Ri	12:37:30	Passeggiata Ripetta	12:40:00	26	150	562	[[41.904257, 12.474908], [41.904337, ;	16	16
bus	628	2495-10	Passeggiata Ripetta	12:40:00	Lgt Delle Navi	12:43:46	27	226	860	[[41.90908, 12.474142], [41.909145, ;	16	17.6
bus	628	2495-10	Lgt Delle Navi	12:43:46	Mazzini/Monte Grappa	12:46:28	28	162	612	[[41.916482, 12.471657], [41.916535, ;	17.6	18
bus	628	2495-10	Mazzini/Monte Grappa	12:46:28	P.za Mazzini	12:47:50	29	82	307	[[41.917912, 12.467548], [41.917752, ;	18	18.1
bus	628	2495-10	P.za Mazzini	12:47:50	Oslavia/Mazzini	12:48:32	30	42	158	[[41.916538, 12.464322], [41.916285, ;	18.1	18
bus	628	2495-10	Oslavia/Mazzini	12:48:32	Oslavia/Bainsizza	12:49:48	31	76	290	[[41.916727, 12.463127], [41.91733, ;	18	17
bus	628	2495-10	Oslavia/Bainsizza	12:49:48	V.le Angelico/Marescia	12:52:44	32	176	664	[[41.919088, 12.461657], [41.91921, ;	17	17
bus	628	2495-10	V.le Angelico/Marescia	12:52:44	Lgt Cadorna/M.llo Giar	12:54:30	33	106	406	[[41.921223, 12.45882], [41.921695, ;	17	17
bus	628	2495-10	Lgt Cadorna/M.llo Giar	12:54:30	Lgt Cadorna/Ostello G	12:55:44	34	74	278	[[41.924852, 12.458722], [41.92577, ;	17	17.2
bus	628	2495-10	Lgt Cadorna/Ostello Gi	12:55:44	De Bosis/Stadio Tennis	12:57:30	35	106	402	[[41.927337, 12.458288], [41.927345, ;	17.2	17.1
bus	628	2495-10	De Bosis/Stadio Tennis	12:57:30	Lgt M.llo Diaz/De Bosis	12:58:36	36	66	248	[[41.93083, 12.459165], [41.930885, ;	17.1	17
bus	628	2495-10	Lgt M.llo Diaz/De Bosis	12:58:36	Boselli	13:01:18	37	162	611	[[41.932692, 12.460705], [41.932753, ;	17	19.4
bus	628	2495-10	Boselli	13:01:18	Volpi/Farnesina	13:05:00	38	222	837	[[41.93463, 12.458642], [41.934698, ;	19.4	40.6

4.3 Calcolo delle altitudini alle fermate della rete TPL

Per stimare i consumi associati al sistema del trasporto pubblico è necessario conoscere, fra gli altri dati di partenza, anche le pendenze dei singoli tratti dei percorsi che effettuano i mezzi durante le loro corse. Sfortunatamente, questo dato non è disponibile e occorre ricavarlo per via indiretta.

Un buon punto di partenza è costituito dai valori dell'altitudine delle singole fermate che puntellano i percorsi: in questo caso, le pendenze dei singoli tratti si possono calcolare in prima approssimazione facendo il rapporto fra la differenza di altitudine e la distanza stradale delle due fermate consecutive.

Purtroppo il database delle fermate ("Stops.txt") contiene le informazioni sulle coordinate geografiche (latitudine e longitudine) ma non quelle relative alle altitudini: pertanto, è stato necessario ricavare queste ultime da un'altra fonte e inserirle nell'archivio.

Una possibile fonte di dati per le altitudini è costituita dai modelli digitali del terreno (*Digital Terrain Model*, *DTM*). Il *DTM* è la rappresentazione digitale dell'orografia della superficie terrestre e consiste in sostanza di una matrice numerica (in termini informatici: mappa *raster*) stesa metaforicamente sulla regione rappresentata, in cui ciascun elemento di matrice (in termini digitali: *pixel*) contiene il dato di altitudine del punto geografico sottostante.

A volte, in alternativa al *DTM* si usano i modelli digitali di elevazione (*Digital Elevation Model*, *DEM*), i quali memorizzano l'elevazione sopra lo zero — assegnato normalmente al livello del mare — comprensiva dell'altezza degli eventuali manufatti umani: ad esempio, l'elevazione di un punto geografico corrispondente al tetto di un grattacielo sarà data dalla somma dell'altitudine del sito in cui il grattacielo è

costruito più l'altezza di quest'ultimo. Ovviamente, nei punti in cui le costruzioni sono assenti elevazione e altitudine del terreno sono coincidenti.

Un parametro importante di un DTM è la sua *risoluzione piana* (altrimenti detta *risoluzione spaziale*), ovvero la distanza metrica fra due pixel adiacenti del reticolato geografico rappresentato dalla matrice: più spinta è la risoluzione, più accurata risulterà la rappresentazione del terreno. Così, un DTM con risoluzione a 20 metri è largamente adeguato alle comuni applicazioni geografiche. Per converso, una risoluzione spinta è impegnativa in termini di risorse sia hardware per memorizzare i dati e sia software per la sua elaborazione; pertanto, sono disponibili anche DTM a risoluzioni meno spinte (75 metri e anche più) per la regione italiana (ma anche per l'intero globo) che richiedono meno risorse e nondimeno si rivelano utili per la gran parte delle applicazioni. Infine, l'adeguatezza di una risoluzione dipende pure dal gradiente orografico del terreno: paesaggi "dolci" sono ben rappresentabili con risoluzioni più lasche, mentre pendii ripidi richiedono risoluzioni più spinte.

Affinché un DTM sia utile, occorre pure che le mappe siano georeferenziate, ovvero sia nota la legge che associa ad ogni pixel, la cui posizione all'interno della mappa raster è individuata dal numero di *riga* e da quello di *colonna*, le coordinate del punto geografico che vi corrisponde. In questo modo infatti, conoscendo le coordinate geografiche del sito d'interesse — nel nostro caso: la posizione di una fermata — è possibile risalire al corrispondente pixel della mappa ed estrarre il valore d'altitudine ivi memorizzato.

Il passaggio sin qui descritto ha trascurato in realtà una complicazione in più in quanto la georeferenziazione di un DTM è effettuata normalmente ricorrendo a uno dei sistemi di riferimento cartografici *metrici* associati a quello *geografico* di latitudine e longitudine prescelto: per l'Italia, ad esempio, associabili al sistema geodetico geografico WGS84¹, sono disponibili i due riferimenti metrici WGS84-UTM32 e WGS84-UTM33. Il ricorso alla proiezione su mappa basata sui riferimenti metrici è dettato dalla necessità di preservare più fedelmente le distanze in piano fra i punti geografici e i corrispondenti pixel in mappa.

Stando così le cose, l'attribuzione dell'altitudine ad una fermata si compone dei seguenti passi:

1. lettura delle coordinate geografiche (*lat, long*) della fermata dall'archivio;
2. traduzione delle coordinate geografiche (*lat, long*) nelle corrispondenti metriche (*Nord, Est*) in cui è rappresentata la mappa raster del DTM;
3. conversione delle coordinate metriche (*Nord, Est*) in quelle di pixel (*riga, colonna*);
4. estrazione del dato di altitudine memorizzato nel pixel così individuato;
5. inserimento del valore estratto dalla mappa nell'archivio delle fermate (in una nuova colonna aggiuntavi allo scopo).

¹ WGS84 sta per *World Geodetic System 1984*.

Ciascuno dei passaggi elencati sopra comporta dei dettagli procedurali che vengono spiegati qui di seguito.

In primo luogo, la fonte utilizzata in questo lavoro è il DEM75 messo gratuitamente a disposizione dall'ISPRA – Istituto Superiore per la Protezione e la Ricerca Ambientale, attraverso la propria piattaforma on-line SINAnet - Sistema Informativo Nazionale Ambientale (SINAnet) [8].

Come indica il nome, la risoluzione di questo strumento è pari a 75 metri, sufficiente a localizzare con precisione le singole fermate; nondimeno la precisione verrà ulteriormente migliorata con un metodo d'interpolazione geografica di cui si dirà dopo. Sebbene la fonte sia un DEM e non un DTM, la posizione delle fermate garantisce che i dati estratti siano effettivamente le "altitudini" del terreno non le "elevazioni".

Il DEM75 così scaricato dall'ISPRA è stato importato in un'applicazione GIS e quindi da questa esportato in formato Esri ASCII raster format (estensione .asc, [9]), che è idoneo per essere trattato con le procedure di estrazione sviluppate "in domestico", ovvero predisponendo un programma ad hoc in un linguaggio di programmazione evoluto (PYTHON).

Infatti, il problema di estrarre le altitudini per diecimila punti geografici – le fermate – e inserirle in un archivio tabellare è troppo specifico perché possa essere implementato agevolmente ricorrendo alle funzionalità standard già in dotazione dei GIS. Il DEM così ottenuto consta di 17381 righe per 13463 colonne e contiene altitudini del territorio della regione italiana (Figura 13: La regione geografica rappresentata nel DEM75 (altitudini) di ISPRA) con la risoluzione lineare di 75 metri/pixel.



Figura 13: La regione geografica rappresentata nel DEM75 (altitudini) di ISPRA

DEM75 è georeferenziato nel sistema di riferimento WGS84-UTM32²; in questo riferimento l'unità di misura è il metro mentre gli assi cartesiani sono così individuati:

- a) asse orizzontale: il parallelo di latitudine 0° (equatore), orientato da Ovest verso Est.
- b) asse verticale (meridiano centrale): il meridiano di latitudine 9° Est da Greenwich, orientato da Sud verso Nord. Ai punti posti sul meridiano centrale è imposto un falso Est = 500 000, per cui le località geografiche con Est < 500 000 hanno longitudine inferiore a 9°, quelle con Est > 500 000 hanno longitudine superiore a 9°³.

Nel riferimento WGS84-UTM32 le coordinate del pixel (della riga più) in basso e (della colonna più) a sinistra del DEM75 di ISPRA sono:

- a) Nord = 3923965
- b) Est = 310593

² Occorre precisare che il riferimento WGS84 e quelli che da questo derivano non sono ammessi ufficialmente in Italia. Però è il più noto e usato in tutto il mondo, Italia compresa, e costituisce pressoché uno standard d'uso nei sistemi GIS. Comunque, WGS84 può essere assimilato ai soli effetti pratici (e tenendo ben presente la distinzione sia teorica e sia di modalità realizzativa) a quello nazionale ufficiale denominato ETRS89(ETRF89) [9].

³ L'intera regione geografica italiana è compresa longitudinalmente nell'intervallo 6°÷19° est (rispetto a Greenwich) e latitudinalmente nell'intervallo 35°÷48° nord.

Pertanto le coordinate metriche del pixel posto diametralmente in alto a destra si calcolano come segue:

a) Nord = $3923965 + 75 * (17381 - 1) = 5227465$

b) Est = $310593 + 75 * (13463 - 1) = 1009960$

Nelle espressioni precedenti, 75 è il fattore di proporzionalità (passo) fra le coordinate metriche e la numerazione delle righe e delle colonne; il segno + davanti a 75 in entrambe le espressioni è giustificato dal fatto che la coordinata Nord è crescente dal basso verso l'alto e quella Est da sinistra verso destra.

Come accennato sopra, la posizione dei singoli pixel in una mappa raster è individuata mediante i numeri di riga e di colonna. In base a una convenzione possibile, fra le più utilizzate, la prima riga della mappa è quella in alto ed è contrassegnata con 1, la prima colonna è quella di sinistra e anch'essa è contrassegnata con 1; pertanto nel DEM75 il pixel $(r,c)=(1,1)$ è quello in alto a sinistra, mentre $(r,c)=(17381,13463)$ corrisponde al vertice in basso a destra⁴

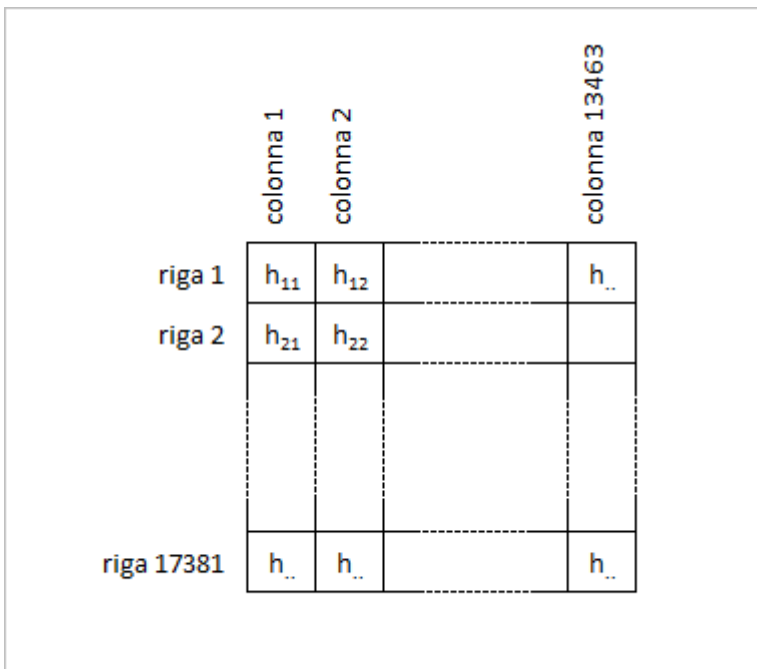


Figura 14: Numerazione delle righe e delle colonne adottata per la mappa raster

Sempre in base alla convenzione così stabilita, le coordinate metriche WGS84-UTM32 di ciascun pixel sono date dalle espressioni:

$$\begin{cases} \text{Nord} = N_1 - \Delta * (r - 1) = 5227465 - 75 * (r - 1) \\ \text{Est} = E_1 + \Delta * (c - 1) = 310593 + 75 * (c - 1) \end{cases} \quad (3)$$

In cui N_1 e E_1 sono le coordinate metriche del pixel $(r,c)=(1,1)$ preso come origine e Δ è il fattore di conversione lineare metri/pixel:

⁴ Una convenzione alternativa – qui NON seguita – è quella che al vertice in alto a sinistra fa corrispondere i valori $(r,c) = (0,0)$ mentre a quello in basso a destra i valori $(r,c) = (17381-1,13463-1) = (17380,13462)$.

$$\Delta = 75 \text{ m/px}$$

Infine, il segno meno (-) nella prima delle due Equazioni (3) tiene conto del fatto che la numerazione delle righe procede in verso opposto a quello della coordinata metrica Nord. Dalle Equazioni (3), per inversione, scriviamo le relazioni per calcolare, partendo dalle coordinate metriche di un sito geografico, la posizione (r,c) del pixel corrispondente nella mappa:

$$\begin{cases} r = \frac{5227465 - Nord}{75} + 1 \\ c = \frac{Est - 310593}{75} + 1 \end{cases} \quad (4)$$

A questo punto l'intera procedura è pressoché delineata: dall'archivio delle fermate si estraggono le coordinate geografiche $(lat, long)$ di ciascuna, si trasformano nelle corrispondenti coordinate metriche $(Nord, Est)^5$ e, mediante le Equazioni (4) si risale alla posizione (r,c) del pixel nel DEM75. Tale pixel è in corrispondenza della fermata stessa: si estrae allora dalla mappa il dato di altitudine e lo si aggiunge finalmente nell'archivio delle fermate.

Resta da esaminare un ultimo dettaglio: la mappa raster rappresenta in realtà un campionamento del terreno secondo un reticolato regolare, e non è detto che la posizione di una fermata corrisponda esattamente a un nodo (pixel) di tale reticolato; anzi, in generale, può distare, nel nostro caso, fino a $\frac{75}{2}\sqrt{2} = 53 \text{ metri}$ dal nodo più vicino campionato. Ciò si traduce nel fatto, che a un'assegnata coppia arbitraria di coordinate metriche $(Nord, Est)$, non sempre -anzi quasi mai - le equazioni (4) vi fanno corrispondere dei valori numerici interi di r e c . In questo caso, occorre effettuare un'interpolazione fra i valori di altitudine dei nodi (pixel) che sono vicini al punto (r,c) . I metodi interpolativi sono numerosi ma, poiché i dati di altitudine del DEM75 sono disposti secondo un reticolato metrico regolare, si può ricorrere al metodo che fa uso dei dati di altitudine dei quattro pixel posti ai vertici del quadrato in cui ricade il punto da calcolare (Figura 15: Interpolazione con quattro punti noti di un reticolo regolare).

⁵ Gli algoritmi di trasformazione dalle coordinate geografiche a quelle metriche e viceversa, per qualunque sistema di coordinate geodetico, compreso il WGS84, sono disponibili in letteratura. Un inventario completo è in [29]. Per la proiezione UTM (Universale trasversa di Mercatore), adottata nei riferimenti metrici WGS84-UTM, sono disponibili anche le formule di Gauss-Boaga, utilizzabili (e utilizzate) per il territorio italiano [30]

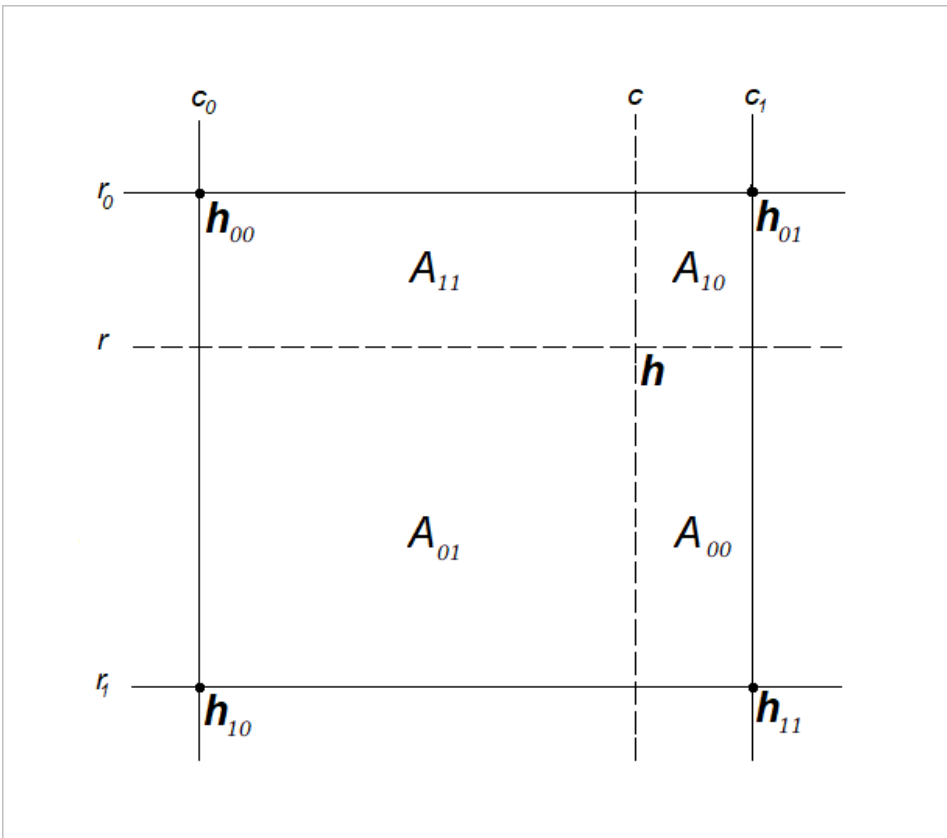


Figura 15: Interpolazione con quattro punti noti di un reticolo regolare

Detti pertanto (r, c) i valori risultanti dal calcolo (4), introduciamo i seguenti simboli relativi ai 4 dati che si ottengono combinando i punti $i=(0,1)$ con $j=(0,1)$ (la funzione $\text{int}()$ restituisce per troncamento la parte intera di un numero scritto in forma decimale):

1. $r_0 = \text{int}(r)$
2. $r_1 = r_0 + 1$
3. $c_0 = \text{int}(c)$
4. $c_1 = c_0 + 1$

Indicate con A_{ij} le quattro aree mostrate in Figura 15, la formula d'interpolazione usata è la seguente:

$$h(r, c) = \frac{h_{00}A_{00} + h_{01}A_{01} + h_{10}A_{10} + h_{11}A_{11}}{A_{00} + A_{01} + A_{10} + A_{11}} \quad (5)$$

In base ad essa, le quattro altitudini h_{ij} sono "pesate" secondo le aree A_{ij} .

Dalla Figura 15 si vede anche che ciascun peso

$$p_{ij} = \frac{A_{ij}}{\sum_{i=0}^1 \sum_{j=0}^1 A_{ij}}$$

è tanto più grande in relazione agli altri tre quanto più il punto (r_i, c_j) è vicino al punto (r, c) . Al limite, se (r, c) coincidesse con uno dei punti (r_i, c_j) , il peso p_{ij} di quest'ultimo varrebbe 1 e i restanti tre si azzererebbero.

L'Equazione (5) può essere posta in una forma più elegante. Se poniamo:

$$a) \quad y=r-r_0$$

$$b) \quad x=c-c_0$$

allora avremo:

$$1. \quad A_{00}=(1-x)(1-y)$$

$$2. \quad A_{01}=(1-x)y$$

$$3. \quad A_{10}=x(1-y)$$

$$4. \quad A_{11}=xy$$

e in definitiva:

$$h(r, c) = \frac{h_{00}(1-x)(1-y) + h_{01}y(1-x) + h_{10}x(1-y) + h_{11}xy}{1} \quad (6)$$

essendo banalmente $A_{00} + A_{01} + A_{10} + A_{11} = 1$

Riarrangiando l'Equazione (6) si perviene alla formula definitiva:

$$h(r, c) = h_{00} + x(h_{10} - h_{00}) + y(h_{01} - h_{00}) + xy(h_{00} + h_{11} - h_{01} - h_{10})$$

dalla quale si evince la circostanza che l'interpolazione qui proposta non è altro che la funzione quadratica nelle coordinate spaziali x e y in cui, dei sei termini in totale, sono nulli solo i due quadrati x^2 e y^2 (mentre è presente il termine triangolare xy).

5 Emucons, Modulo per il calcolo dei consumi TTW e WTT per veicoli privati e bus

Emucons contiene due programmi principali, `carmain.py` e `busmain.py` che permettono rispettivamente il calcolo dei consumi e delle emissioni di CO₂ sia TTW che WTT per autoveicoli privati e autobus del Trasporto Pubblico Locale. L'energia necessaria per i veicoli (Tank To Wheel – TTW) è ottenuta a partire dalle curve di consumo adottate in COPERT, le cui caratteristiche principali sono descritte nel prossimo paragrafo.

5.1 Veicoli a combustione interna: la metodologia CORINAIR/COPERT

I valori medi consumo ed emissione specifici legati alle necessità di trazione dei veicoli sono fortemente dipendenti dall'ambito territoriale in cui il veicolo effettua il servizio e dalla tipologia di servizio.

In particolare, a parità di caratteristiche del veicolo, i principali fattori territoriali che influenzano i valori specifici di consumo energetico e delle emissioni in atmosfera di un veicolo sono:

- a) il profilo delle velocità istantanee lungo il tragitto;
- b) le pendenze stradali affrontate lungo il percorso;
- c) in particolare nel caso di mezzi pesanti e di autobus il carico trasportato.

In un approccio ottimale la determinazione del consumo energetico e delle emissioni in atmosfera richiederebbe la conoscenza di dati puntuali ed istantanei. Tuttavia, nel corso degli anni, specie ai fini della predisposizione degli inventari nazionali delle emissioni in atmosfera richiesti dalla normativa internazionale nell'ambito della Convenzione sull'Inquinamento Atmosferico Transfrontaliero a lunga distanza (CLRTAP), sono state messe a punto delle metodologie che semplificano la stima utilizzando dati medi ed aggregati.

In Europa, la metodologia di riferimento è CORINAIR che, nel caso del trasporto stradale, si declina attraverso il tool di calcolo COPERT, un software sviluppato sotto il coordinamento dell'Agenzia Europea per l'Ambiente (EEA) dalla società EMISIA, e finanziato dall'Unione Europea, per il calcolo delle emissioni di inquinanti e dei consumi del parco veicolare. E' un software aperto a tutti ed è utilizzato da molti paesi europei, inclusa l'Italia, per realizzare i *repository* ufficiali di emissione di inquinanti per il settore trasporti. Documentazione dettagliata è disponibile sul sito dell'EMISIA [10].

Nell'ambito di CORINAIR vengono considerati tutti i più importanti inquinanti, gas serra e specie tossiche, tra cui precursori dell'ozono, anidride carbonica, metano, protossido di azoto, sostanze acide, particolato (black carbon e organic carbon). Sono inoltre considerate sostanze cancerogene, diossine e metalli pesanti e i consumi energetici per trazione. La metodologia utilizzata è coerente con le linee guida IPCC per il calcolo dei gas serra ed è parte delle metodologie incluse nelle raccomandazioni EMEP/EEA per il calcolo delle emissioni di inquinanti in atmosfera.

Le emissioni ed i consumi totali sono il risultato della somma delle emissioni "a caldo", relative al motore alla temperatura normale di funzionamento, e di quelli durante il periodo di transizione a partire dall'accensione del motore fino al momento in cui la temperatura di crociera viene raggiunta (emissioni "a freddo").

Il calcolo di consumi ed emissioni a caldo si basa su curve che descrivono l'andamento dei valori specifici (per km percorso) in funzione della velocità media di spostamento.

Le curve sono in generale specializzate per tipologia di veicolo e, nel caso di alcune tipologie, sono parametrizzate in relazione al carico del veicolo ed alla pendenza media del percorso. Nelle più recenti versioni di COPERT tutte le funzioni precedentemente utilizzate per il calcolo delle emissioni e dei consumi

sono state unificate in un'unica funzione generale, di volta in volta semplificata ponendo gli opportuni coefficienti pari a zero. Nel caso del consumo per trazione E la formula viene espressa come segue:

$$E_{k,p,m}[MJ/km] = \frac{\alpha_{k,p,m} \cdot v^2 + \beta_{k,p,m} \cdot v + \gamma_{k,p,m} + \frac{\delta_{k,p,m}}{v}}{\varepsilon_{k,p,m} \cdot v^2 + \epsilon_{k,p,m} \cdot v + \eta_{k,p,m}} \quad (7)$$

dove v rappresenta la velocità media, k la tipologia di veicolo, p la pendenza media e m il carico. Una struttura funzionale analoga è disponibile per ogni inquinante o gas serra considerato.

Nel caso degli autobus, la pendenza p e il carico m sono espressi in classi: rispettivamente $< 0.2\%$, $[0.2\% \div 0.4\%)$, $[0.4\% \div 0.6\%)$, $> 0.6\%$ e 0% , 50% , 100% . Queste curve rappresentano il punto di partenza per il calcolo dell'energia spesa e delle emissioni prodotte in diverse situazioni di utilizzo del veicolo.

COPERT diversifica le curve del consumo e delle emissioni specifici per più di 450 tipi di veicoli con motore a combustione interna, tra cui veicoli privati e autobus per servizio di trasporto pubblico locale, limitatamente alle alimentazioni a gasolio e gas compresso.

I coefficienti legati alle diverse tipologie di veicoli e alle diverse classi di carico e di percorso sono il risultato di simulazioni e attività di misura sperimentale su motori, su banco a rulli e in condizioni reali di circolazione portate avanti in numerosi progetti di ricerca inclusi nei programmi di cooperazione europea COST 319 e COST 346 (european COOperation in the field of Scientific and Technical Research) e nei progetti ARTEMIS [11] e PARTICULATE del 5° programma quadro della Commissione Europea.

In particolare il progetto ARTEMIS ha permesso di definire gli standard relativi a gran parte delle metodologie di calcolo e dei formati che costituiscono l'attuale struttura del database HBEFA (Handbook emission Factors for Road Transport – ampiamente utilizzato in COPERT), supportato da un numero elevato di centri di ricerca e agenzie nazionali europee e mantenuto dal centro di ricerca svizzero INFRAS (www.infras.ch) [12].

5.2 Le emissioni di CO₂ nella fase TTW

Le emissioni totali di Anidride Carbonica sono date da tre contributi differenti: la combustione di carburante, la combustione dei lubrificanti e l'aggiunta di additivi contenenti carbonio negli scarichi.

La CO₂ emessa dalla combustione di carburante m dalla tecnologia k viene derivata dal consumo del carburante attraverso il bilancio di massa del carbonio in esso contenuto attraverso la seguente relazione (8):

$$E_{CO_2; k,m} = 44.011 \cdot \frac{FC_{k,m}}{12.011 + 1.008 \cdot r_{H:C,m} + 16.0000 \cdot r_{O:C,m}} \quad (8)$$

dove $FC_{k,m}$ è il consumo di carburante e $r_{H:C}$ e $r_{O:C}$ sono rispettivamente i rapporti fra idrogeno e carbonio e ossigeno e carbonio per il combustibile considerato.

Nel caso del software Emucons, il consumo è calcolato in unità di energia e non di massa, per cui la formula 2 è stata convertita nella seguente formula (9):

$$CO_2 = a \cdot E \quad (9)$$

Dove i valori di a sono elencati in Tabella 5 per i diversi carburanti, mentre E rappresenta l'energia spesa per trazione.

Tabella 5: Coefficienti per il calcolo delle emissioni di CO₂ a partire dall'energia consumata

Tecnologia	a [g/MJ]
Diesel	73.648
Benzina	72.91
CNG/LNG	57.618
LPG	65.59
Elettrico/Idrogeno	0

5.3 Il caso degli autobus

I fattori di emissione degli autobus sono ricavati a partire dai dati HBEFA [12], disponibili per tutte le categorie EURO 1, ..., 6 e precedenti, e per le due sottocategorie EGR/SCR in EURO 5.

Per quanto riguarda l'alimentazione a metano, COPERT considera solamente gli autobus urbani alimentati a CNG, e non anche quelli alimentati a LNG; inoltre non dispone di curve specializzate per classi di pendenza o di carico.

A partire dalla formula generale (Equazione (7)), in Tabella 6, a titolo di esempio, sono riportati i parametri relativi ai veicoli diesel Euro 6 con peso tra 15 e 18 Tonnellate per le diverse classi di pendenza e le tre classi di carico per il calcolo del consumo di energia:

Tabella 6: - Coefficienti COPERT per il consumo di energia per trazione per diverse configurazioni della pendenza e del carico per gli autobus urbani con massa tra 15 e 18 Tonnellate a tecnologia Diesel

Pendenza	Carico	Vmin	Vmax	α	β	γ	δ	ϵ	ζ	η
-0.06	0.00	5.0	85.0	-0.003	0.230	-0.971	1.94	0.002	-0.007	0.02
-0.06	0.50	5.0	85.0	-0.003	0.239	-1.075	2.26	0.002	-0.010	0.03
-0.06	1.00	5.0	85.0	-0.003	0.306	-1.501	3.17	0.003	-0.016	0.05
-0.04	0.00	5.0	85.0	-0.007	0.846	-1.822	2.75	0.005	0.014	0.00
-0.04	0.50	5.0	85.0	-0.005	0.541	-1.461	2.01	0.003	0.006	0.00
-0.04	1.00	5.0	85.0	-0.005	0.495	-1.428	1.84	0.003	0.004	0.00
-0.02	0.00	5.0	85.0	0.000	-0.041	2.795	2.65	0.000	0.007	0.15
-0.02	0.50	5.0	85.0	-0.001	1.150	-2.564	3.60	0.005	0.019	0.00

Pendenza	Carico	Vmin	Vmax	α	β	γ	δ	ϵ	ζ	η
-0.02	1.00	5.0	85.0	0.000	-0.047	3.530	0.63	0.000	0.009	0.13
0.00	0.00	5.0	85.0	0.000	-0.014	2.658	3.26	0.000	0.005	0.13
0.00	0.50	5.0	85.0	0.000	-0.015	5.339	4.33	0.000	0.010	0.21
0.00	1.00	5.0	85.0	0.040	1.904	-3.764	5.30	0.007	0.027	0.00
0.02	0.00	5.0	85.0	0.000	0.043	3.745	5.81	0.000	0.008	0.15
0.02	0.50	5.0	85.0	-0.001	0.095	5.152	6.43	0.000	0.011	0.17
0.02	1.00	5.0	85.0	0.264	3.359	-6.546	8.94	0.017	0.024	0.00
0.04	0.00	5.0	85.0	0.001	0.060	4.381	8.71	0.000	0.006	0.17
0.04	0.50	5.0	85.0	0.009	0.120	8.588	14.86	0.000	0.007	0.29
0.04	1.00	5.0	70.0	0.011	1.541	15.608	11.37	0.000	0.059	0.30
0.06	0.00	5.0	80.0	0.008	0.030	7.109	14.84	0.000	0.003	0.25
0.06	0.50	5.0	65.0	0.246	2.877	-3.054	-11.04	0.009	0.060	-0.19
0.06	1.00	5.0	55.0	-0.007	2.730	17.933	11.82	0.000	0.078	0.27

Le curve di consumo corrispondenti sono mostrate nella seguente Figura 16.

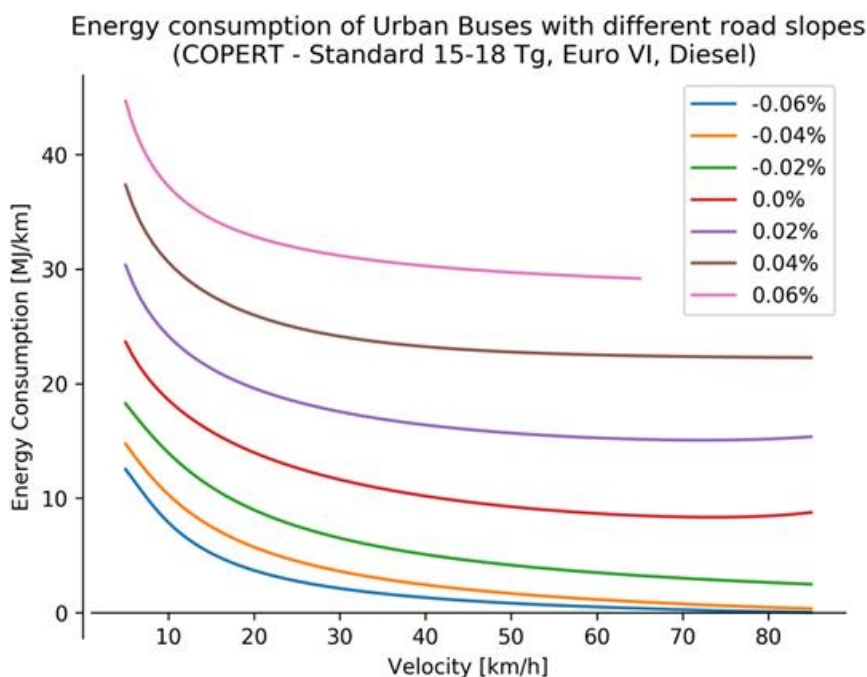


Figura 16: Curve di consumo espresse in energia (MJ/km) per il range di velocità considerato da COPERT, per gli autobus urbani con tecnologia Diesel e massa tra 15 e 18 Tonnellate

5.4 Le curve di consumo dei veicoli elettrici

La metodologia CORINAIR/COPERT non copre le tecnologie veicolari a trazione puramente elettrica. Le flotte considerate in questo studio comprendono tuttavia, nel caso degli autobus, anche veicoli elettrici, per i quali COPERT non permette di calcolare i consumi di trazione. Per colmare questa lacuna, nel corso dell'ultimo periodo di svolgimento dell'Accordo di Collaborazione MISE – ENEA sulla Ricerca di Sistema

Elettrico nel triennio 2015-2018, all'interno della Scheda Progettuale Mobilità sono state realizzate campagne di misura sul campo, grazie alla collaborazione della CTM di Cagliari, con la quale si è potuto procedere ad alcune acquisizioni all'interno di un Protocollo di collaborazione fra ENEA ed alcune Aziende associate ad ASSTRA, che dispongono di veicoli ad alimentazione elettrica.

Le misure sono state effettuate durante la stagione estiva del 2018, approfittando del fatto che in estate i filobus da 12 metri in servizio sulla linea 5, di due distinti modelli con caratteristiche molto simili fra loro, effettuano percorso prolungato sino ai lidi marini di Cagliari utilizzando la modalità della autonoma marcia in batteria per diversi chilometri.

Quando in modalità marcia autonoma, i filobus di Cagliari sono assimilabili a bus elettrici a batteria; il peso del pantografo e della piccola batteria montata per effettuare il percorso fuori linea aerea, nel complesso, è confrontabile con quello di una batteria di maggiori dimensioni, quale quella usualmente montata su autobus elettrici configurati per effettuare un servizio giornaliero senza ricariche intermedie durante le soste al capolinea. Tenuto conto di questa combinazione, i valori di consumo analizzati possono essere riferiti ad un autobus elettrico a batteria da 12 m con ricarica notturna al deposito.

L'attività di ricerca è stata principalmente mirata ad estrapolare delle curve di consumo per trazione (al netto del consumo per condizionamento abitacolo, analizzato separatamente) in funzione della velocità commerciale media di tratta e dei parametri di carico e pendenza.

Di seguito sono riportati due esempi delle curve di consumo ottenute (Figura 17 e Figura 18). La struttura di queste curve ne permette un utilizzo diretto con la stessa metodologia COPERT, in modo da semplificare il calcolo dei consumi di trazione per tutta la flotta considerata.

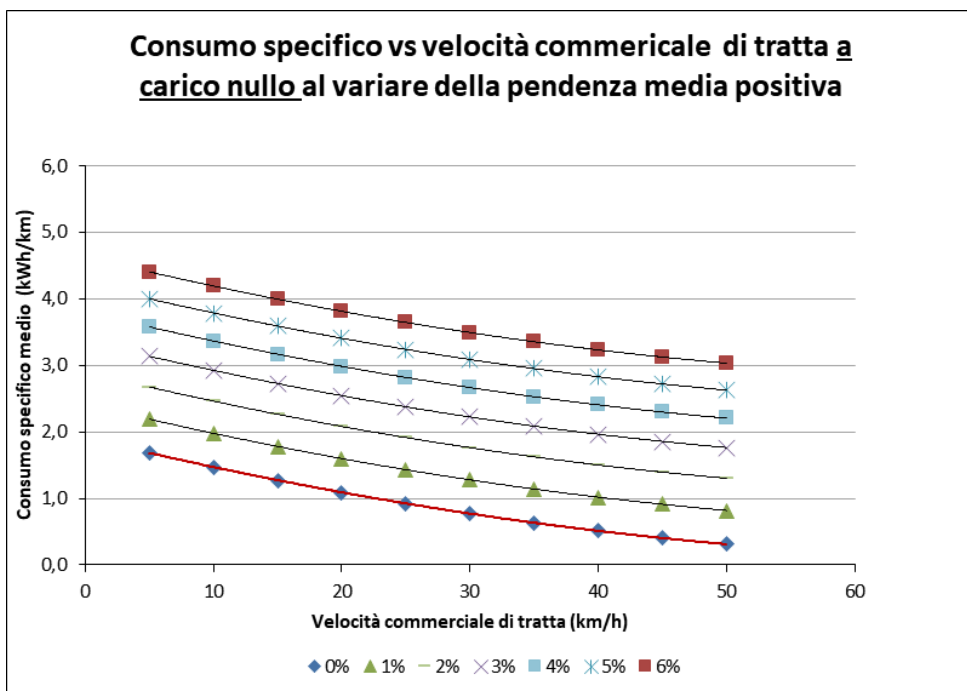


Figura 17: Curve di consumo dei Bus elettrici a batteria a carico nullo estrapolate per pendenze positive

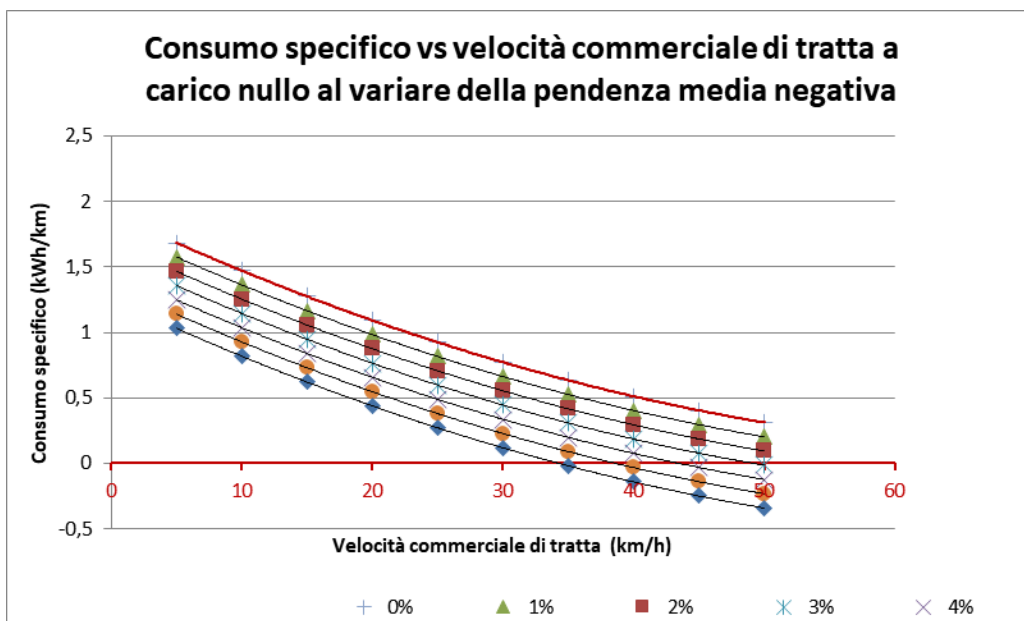


Figura 18: Curve di consumo a carico nullo estrapolate per pendenze negative

Si osservino, nel caso di gradiente negativo fortemente accentuato e per velocità medie di tratta superiori a 35 km/h, valori di consumo specifico medio negativi. Si tratta di un'eventualità, piuttosto rara a verificarsi nella realtà, connessa alla possibilità di recupero energetico in frenata e, appunto, in discesa, allorché il motore elettrico preposto alla trazione si converte in generatore in grado di trasformare l'energia cinetica del veicolo in energia elettrica che va ad essere immagazzinata all'interno della batteria di bordo o, in alcuni casi, ad essere utilizzata direttamente dai dispositivi ausiliari.

Si osservi altresì che i valori di consumo specifico rappresentati nelle curve sono da intendersi esclusivamente per le necessità di trazione e degli ausiliari minimi e non anche per le necessità di condizionamento di bordo.

5.5 *Le componenti WTT*

Con la diffusione di autoveicoli elettrici, per avere un quadro completo dal punto di vista energetico e di impatto globale non è più sufficiente valutare i consumi e le emissioni di gas serra (GHG) in fase d'uso, dal momento che quest'energia elettrica ha processi di produzione con consumi ed emissioni molto diverse rispetto ai carburanti tradizionali.

E' importante, pertanto, valutare consumi ed emissioni nelle fasi di estrazione delle materie prime, trasformazione, trasporto fino al serbatoio (dal pozzo al serbatoio – Well To Tank) di tutti i vettori energetici.

Per poterlo fare sono stati elaborati dei fattori di emissione in funzione dei consumi su strada calcolati dal software, basandosi su dati di studi internazionali e nazionali.

Per i carburanti tradizionali (Benzina, Gasolio, Gas Naturale Compresso e GPL) le fasi WTT riguardano le operazioni di estrazione di petrolio e gas naturale (perforazioni, separazione tra petrolio, gas e acqua, riduzione e controllo della pressione, stabilizzazione del greggio e trattamento del gas), lo stoccaggio e il trasporto verso le raffinerie o il pompaggio nei gasdotti verso i depositi, e poi il trasporto fino ai distributori per la vendita.

I fattori dell'estrazione e del trasporto del petrolio sono stati stimati a partire dai dati contenuti nel Report JRC WTT versione 4 e sua Appendice 2⁶ [13] [14], ricalibrati in funzione delle importazioni italiane di greggio pubblicate dall'Unione Petrolifera [15].

Il fattori della fase di raffinazione sono stati, invece, calcolati a partire dall'Inventario delle Emissioni di GHG di ISPRA [16], sulla base dei dati di produzione contenuti nel Bilancio Energetico Nazionale, pubblicato annualmente dal Ministero dello Sviluppo Economico [17].

Per la fase di distribuzione vengono assunti i coefficienti del JEC che valuta il mix di modi di trasporto, comprendendo anche l'energia elettrica delle stazioni di servizio.

Il Gas Naturale Compresso, invece, non deve subire ulteriori trasformazioni dopo la fase di estrazione e separazione, per cui si hanno ulteriori consumi e emissioni solo per le operazioni di pompaggio nella rete,

⁶ Il nuovo rapporto JR [14]C (Report WTW version 5 – 2020) ha dei valori differenti per alcune fasi di produzione del petrolio che però in questo studio non sono stati usati perché si sono preferiti dei fattori più specifici del IOGP (associazione internazionale dei produttori oil&gas) in esso contenuti, anche se più datati. Si ritiene necessario un maggiore approfondimento della questione, i cui risultati verranno acquisiti nell'aggiornamento futuro del software

stoccaggio nei depositi, distribuzione e compressione nelle stazioni di servizio. I dati per queste fasi sono mutuati dai rapporti ISPRA [16].

I fattori per il GPL sono ricavate percentualmente dai fattori delle fasi sopradescritte per petrolio e gas rispetto alla composizione del carburante (miscela al 60% di gas liquidi e al 40% di prodotti di raffineria).

I consumi e emissioni specifici di GHG per la produzione e la distribuzione dell'energia elettrica consumata in Italia sono calcolati a partire dai dati TERNA [18] e del Bilancio Energetico Nazionale [17]. L'energia elettrica consumata in Italia è prodotta per più della metà da centrali termoelettriche, alimentate per quasi il 70% da gas naturale, e poi dal carbone e in minima parte da fonti di rifiuti non riciclabili. La produzione da fonti rinnovabili copre più del 30% dei consumi.

Per i consumi ed emissioni della fase di produzione delle materie prime (estrazione gas e carbone etc.) per la produzione termoelettrica, sono stati usati i fattori di emissione del JEC pesati sulle percentuali di elettricità immessa in rete, secondo i dati TERNA.

Per i consumi e i GHG della fase di produzione, valgono i dati di efficienza della produzione termoelettrica e dei fattori di emissioni di GHG del rapporto ISPRA su produzione e consumo di energia elettrica in Italia [19], e la quota dell'energia prodotta da fonti rinnovabili (FER) – principalmente da centrali idroelettriche e impianti fotovoltaici – e immessa in rete, secondo le statistiche TERNA.

E' stata poi considerata una quota percentuale di consumo ed emissione dovuta a perdite di rete – dato TERNA – e di perdite in fase di ricarica delle batterie, dato ricavato da sperimentazioni ENEA.

I fattori specifici così calcolati sono riportati nella seguente Tabella 7:

Tabella 7: fattori di consumo e emissione di GHG dal pozzo alla ruota in funzione dei consumi finali per alimentazione

WTT	CO2 eq	EC fossil
Alimentazione	g/MJ	MJ/MJ _{fin}
Benzina	11,20	0,15
Gasolio	11,18	0,15
Gpl	11,21	0,13
Gas naturale compresso	13,00	0,17
Energia elettrica	106,16	0,93

I consumi totali WTW dei vari vettori energetici saranno quindi la somma dei consumi in fase d'uso (TTW) e dei consumi WTT calcolati con i fattori in Tabella 7.

5.6 La struttura della flotta

Sia per quanto riguarda il caso degli autobus che quello delle vetture private è stato necessario associare alle tratte percorse una tipologia di veicolo. Questo per quanto riguarda gli autobus ha comportato

principalmente una scelta tecnologica di alimentazione, invece per quanto riguarda i veicoli privati è stato necessario differenziare oltre che per tecnologia anche per dimensione del veicolo (SUV, veicoli medi, veicoli mini).

All'interno del codice emucons è stato scelto di associare in modo casuale ad ogni tratta percorsa una tipologia di veicolo a partire dalla distribuzione di veicoli disponibile per l'ambito territoriale considerato, nel nostro caso la città di Roma.

Le distribuzioni di veicoli sono state ricavate a partire da dati della motorizzazione civile per gli autobus e dell'Autoritratto ACI per le autovetture [20]. Per quest'ultime il parco circolante è dato suddiviso per alimentazione, cilindrata e classe Euro a livello delle provincie, mentre nel caso dei comuni viene dato solo per classe Euro.

Le autovetture del comune di Roma sono state, in mancanza dei dati necessari, disaggregate secondo la distribuzione delle autovetture della Provincia di Roma, di cui costituiscono più del 65%. Le distribuzioni di veicoli sono rappresentate nelle figure seguenti rispettivamente per quanto riguarda gli autobus e le autovetture private (Figura 19, Figura 20, Figura 21, Figura 22).

Si osserva per gli autobus una predominanza di veicoli diesel di dimensione standard (15-18 tonnellate), con standard emissivi in prevalenza appartenenti alla categoria Euro 3 e quindi con un'età dell'ordine dei 15 anni. Sono presenti anche veicoli Euro 5 ed euro 6, e una residua ma non completamente trascurabile quota di veicoli Euro 2.

Il parco veicoli include anche una importante quota di autobus a metano, mentre gli autobus elettrici, seppur presenti, sono pochi e di piccole dimensioni.

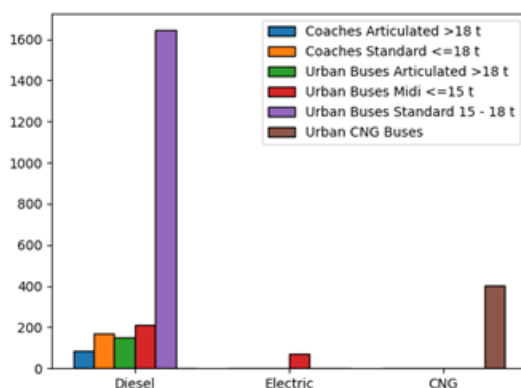


Figura 19: Distribuzione dei mezzi utilizzata in emucons e da cui sono stati estratti in modo casuale i veicoli da associare alle tratte considerate - numero di autobus suddiviso per tecnologia di trazione e tipologia di mezzo

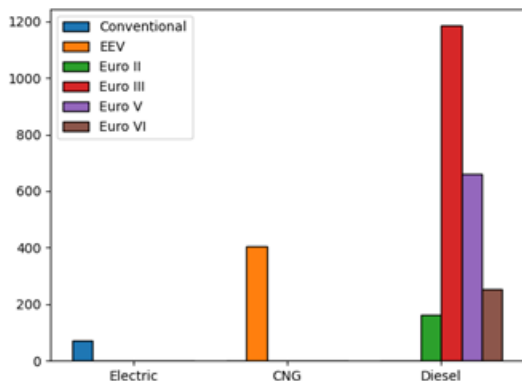


Figura 20: Distribuzione dei mezzi utilizzata in emucons - numero di autobus suddiviso per tecnologia di trazione e Standard Emissivo

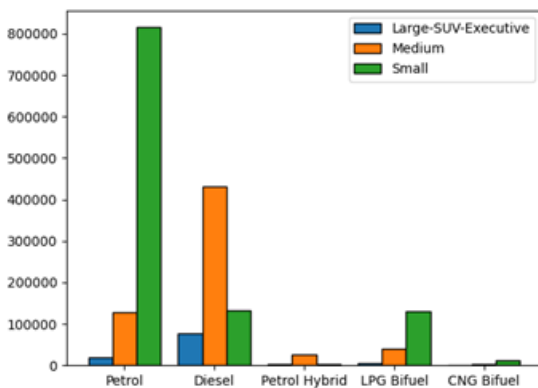


Figura 21: Distribuzione dei mezzi utilizzata in emucons - Numero di veicoli privati suddiviso per tecnologia di trazione e tipologia di veicolo

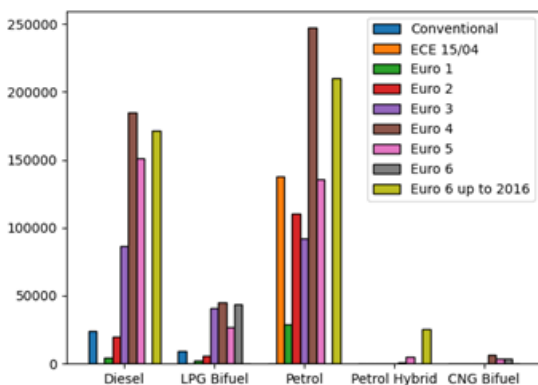


Figura 22: Distribuzione dei mezzi utilizzata in emucons - Numero di veicoli privati suddiviso per tecnologia di trazione e Standard Emissivo

Il parco privato vede una predominanza di veicoli a benzina, seguiti da veicoli diesel, con una diversa distribuzione delle dimensioni. Se infatti i veicoli a benzina sono in gran prevalenza di piccole dimensioni, l'alimentazione diesel è prevalente per le vetture medie e i SUV. La distribuzione delle categorie emissive è

per tutte le alimentazioni più piatta rispetto a quella degli autobus, con grandi quantità di veicoli di recente costruzione (euro 5 o euro 6) ma veicoli ancora non in sostituzione per quanto riguarda gli Euro 4 e solo in parziale sostituzione per gli euro 2 ed euro 3. Una distribuzione simile, anche se con totali molto inferiori caratterizza i veicoli GPL, mentre i veicoli a metano e soprattutto ibridi sono rappresentati da flotte più moderne.

5.7 Descrizione del software Emucons

Come già accennato Emucons contiene due programmi principali, `carmain.py` e `busmain.py` che permettono il calcolo dei consumi e delle emissioni di CO₂ sia TTW che WTT rispettivamente per veicoli privati e autobus. I due programmi si appoggiano ad una classe (`educons`) che contiene tutti i metodi necessari per i conti.

I file di ingresso sono file csv o excel contenenti righe che rappresentano una frazione di un viaggio (per i veicoli privati) oppure una tratta da fermata a fermata per gli autobus. Il formato di questi file è il formato `gtfs enea` per gli autobus, un formato in uscita da un database sql per quanto riguarda i veicoli privati.

I file di uscita hanno la stessa struttura dei file di ingresso con l'aggiunta di diverse colonne che rappresentano il tipo di mezzo associato alla linea o al viaggio, i consumi in MJ, le emissioni di CO₂, i consumi nelle unità di misura specifiche del combustibile, le unità di misura e energia e emissioni co2 WTT.

I file di distribuzione del parco veicoli è rappresentato da file excel contenenti colonne rappresentative dei veicoli e numero di veicoli associato.

Questa è la struttura dei due main:

5.7.1 `carmain.py`:

`carmain` è il main file per generare i consumi del parco auto a partire dalle traiettorie fornite nell'ambito del progetto emu. Il programma legge le traiettorie (`myfile`), associa ad ogni traiettoria una tipologia di veicolo a partire dalla distribuzione di veicoli disponibile per la città considerata (`mydist`) e calcola consumi e emissioni di CO₂ per ogni traiettoria utilizzando le equazioni disponibili in Copert.

Le funzioni sono disponibili nella classe `emucons`, condivisa con l'altro main del progetto, (`consbus`), finalizzato al calcolo dei consumi per il parco autobus. Nel caso di motori bifuel, i consumi in MJ e la CO₂ sono riferiti al totale, i consumi nelle unità del combustibile sono divisi e riportati in colonne separate.

Essendo ogni riga associata a un veicolo diverso l'implementazione è basata su cicli `lambda x` per ogni riga. Dato il numero di righe dell'ordine del milione in particolare il ciclo relativo alle curve copert è relativamente lento. Per questo il ciclo copert è stato parallelizzato utilizzando `mapply`. In tal modo l'esecuzione su un pc con 8 processori i7 di recente generazione richiede circa 15 minuti di tempo macchina. Ulteriori modifiche potranno riguardare la sostituzione di un ciclo `lambda x` con funzioni basate su `groupby`.

La struttura di carmain.py è come segue:

- 1) Lettura file di ingresso, file di distribuzione veicoli, file funzioni Copert
- 2) Filtraggio righe con percorrenza e con tempo di percorrenza nulli
- 3) Associazione random di un veicolo ad ogni riga rispettando la distribuzione data
- 4) Calcolo consumi in MJ usando le curve copert
- 5) Conversione consumi nelle unità del combustibile
- 6) Calcolo CO₂ emessa
- 7) Calcolo energia e CO₂ WTT
- 8) Pulizia
- 9) Scrittura file di output (excel o csv)

In uscita (pathout) viene prodotto lo stesso file di ingresso con l'aggiunta di alcune colonne che rappresentano il veicolo associato alla riga, consumi in MJ, consumi nelle unità del combustibile considerato e emissioni di CO₂ in grammi.

5.7.2 Busmain.py

Busmain è il main file per generare i consumi del parco autobus a partire dalle traiettorie fornite nell'ambito del progetto emu. Il programma legge le traiettorie (pathin), associa ad ogni traiettoria una tipologia di veicolo a partire dalla distribuzione di veicoli disponibile per la città considerata (b2lpath) e calcola consumi e emissioni di co2 per ogni traiettoria utilizzando le equazioni disponibili in Copert.

Le funzioni sono disponibili nella classe emucons, condivisa con l'altro main del progetto, (conscar), finalizzato al calcolo dei consumi per il parco autobus. Essendo ogni riga associata a un veicolo diverso l'implementazione è basata su cicli lambda x per ogni riga. Dato il numero di righe dell'ordine del milione in particolare il ciclo relativo alle curve copert è relativamente lento. Per questo il ciclo copert è stato parallelizzato utilizzando mapply. In tal modo l'esecuzione su un pc con 8 processori i7 di recente generazione richiede circa 15 minuti di tempo macchina. Ulteriori modifiche potranno riguardare la sostituzione di un ciclo lambda x con funzioni basate su groupby.

La struttura di busmain.py è la seguente:

- 1) Lettura file di ingresso, file di distribuzione veicoli, file funzioni Copert
- 2) Associazione random di un veicolo ad ogni riga rispettando la distribuzione data
- 3) Calcolo consumi in MJ usando le curve copert
- 4) Conversione consumi nelle unità del combustibile
- 5) Calcolo co2 emessa
- 6) Calcolo energia e co2 WTT
- 7) Scrittura file di output (csv)

In uscita (pathout) viene prodotto lo stesso file di ingresso con l'aggiunta di alcune colonne che rappresentano il veicolo associato alla riga, consumi in MJ, consumi nelle unità del combustibile considerato e emissioni di CO₂ in grammi.

5.7.2.1 Classe emucons

La classe **emucons** contiene i seguenti metodi, alcuni dei quali sono comuni ai due main, altri, pur svolgendo funzioni simili, sono stati duplicati a causa delle differenze di struttura dei file di ingresso.

5.7.2.2 Readcopert

Metodo per la lettura del file excel contenente le curve copert. Utilizza le variabili generali della classe:

- a) :param sn: nome del foglio excel da leggere, attualmente uguale a 1
- b) :param sr: numero di righe da saltare all'inizio del file al momento della lettura, attualmente 0
- c) :return: self.xl: dataframe contenente i parametri delle curve copert

5.7.2.3 Selectenecons

Seleziona tutte le righe relative ai bus per la tipologia di inquinante o consumo dal dataframe di copert:

- a) :param sh: dataframe di copert
- b) :param poll: consumo o inquinante da selezionare
- c) :return: sottoinsieme di sh filtrato

5.7.2.4 Selectunique

Seleziona una lista di tutte le possibili categorie di veicoli disponibili. Se cut è True fornisce solo un sottoinsieme di colonne:

- a) :param sh: copert dataframe
- b) :param cut: Riduce il numero di colonne in uscita
- c) :return: sottoinsieme filtrato di sh

5.7.2.5 Selectlines

Identifica tutte le linee presenti nel file gtfs e se cut è True fornisce solo l'identificativo

- a) :param sh: dataframe del gtfs
- b) :param cut: Seleziona solo la colonna route_id
- c) :return: dataframe filtrato

5.7.2.6 Selectroutes

Identifica tutte le possibili traiettorie. Se cut è True restituisce solo la colonna delle trajectory

- a) :param sh: dataframe
- b) :param cut: Seleziona solo la colonna idtrajectory
- c) :return: dataframe filtrato

5.7.2.7 Writsheet

Scrive un file excel con il dataframe passato.

Attenzione xlsxwriter è obsoleto, da sostituire con openpyxl

- a) :param sh: dataframe da scrivere
- b) :param pathfile: path completo del file da scrivere

5.7.2.8 readENEAgdfs

Legge il file csv contenente l'elaborazione del gtfs in formato Enea

- a) :param pathfile: nome con path del file da leggere
- b) :return: dataframe con il contenuto del file

5.7.2.9 Bus2line

Costruisce un dataframe a partire dalla distribuzione delle linee (pathd) con la distribuzione degli autobus per la città considerata e associa ad ogni trip un tipo di autobus in modo random

- a) :param city: città considerata
- b) :param lines: dataframe contenente le linee evidenziate nel gtfs
- c) :param pathd: path del file contenente la distribuzione del parco mezzi
- d) :return: dataframe lines con una tipologia di bus associata ad ogni linea

5.7.2.10 Car2route

Associa ad ogni percorso un veicolo a partire dalla distribuzione di veicoli fornita:

- a) :param lines: dataframe con i tragitti, che corrisponde in questo caso (veicoli privati) al dataframe di ingresso
- b) :param distr: distribuzione dei veicoli
- c) :return: lines con associata una tipologia di veicolo per ogni percorso

5.7.2.11 computecons

Calcola i consumi per gli autobus a partire dal tipo di mezzo, pendenza, tempo di percorrenza, distanza.

Utilizza le curve copert:

- a) :param cat: Identificativi mezzo
- b) :param fuel:
- c) :param seg:
- d) :param euro:
- e) :param tech:
- f) :param fromel: elevazione di partenza
- g) :param toel: elevazione di arrivo
- h) :param tim: tempo tratta

- i) :param dist1: distanza fornita da gtfs
- j) :param dist2: distanza calcolata dallo shape
- k) :param dist3: distanza calcolata in modo euclideo da fermata a fermata
- l) :param onlyb: selezione del file copert
- m) :param city: città considerata
- n) :param which: se True calcola i consumi effettivi e non quelli specifici
- o) :return: consumo calcolato

5.7.2.12 copFun

Funzione base per i calcoli Copert basati sulle curve definite in funzioni della velocità. Chiamata da computecons:

- a) :param myline: linea del dataframe copert contenente i parametri della funzione
- b) :param v: velocità
- c) :return: consumi specifici

5.7.2.13 Convertcons

Converte i consumi in MJ/km nelle unità associate al tipo di combustibile. Restituisce più di una colonna, separate al momento della chiamata nei main

- a) :param cons: consumi
- b) :param fuel: combustibile
- c) :param bus: if True fa i conti per i bus senza bisogno di considerare i bifuel e lo sdoppiamento delle colonne
- d) :param cons1: nel caso di un bifuel consumi del secondo carburante
- e) :return: consumi e unità utilizzate

5.7.2.14 CO₂

Calcola le emissioni di co2 da Tank To Weel (TTW). Utilizza self.pp che rappresenta la percentuale di utilizzo della benzina nel caso di bifuel

- a) :param fuel: tipo di combustibile
- b) :param cons: consumi del primo combustibile
- c) :param cons1: consumi del secondo combustibile
- d) :return: emissioni di co2 in grammi

5.7.2.15 carcons

Calcola i consumi nel caso dei veicoli privati. Utilizza self.pp che rappresenta la percentuale di utilizzo della benzina nel caso di bifuel

- a) :param cat: identificazione del veicolo

- b) :param fuel:
- c) :param seg:
- d) :param euro:
- e) :param techp:
- f) :param techd:
- g) :param tim: tempo impiegato per percorrere il tragitto
- h) :param dist1: distanza del tragitto
- i) :param onlyb: sottoinsieme del file copert riferito ai consumi
- j) :param which: se True calcola i consumi reali e non quelli specifici
- k) :return: dataframe con consumi

5.7.2.16 Wttene

Calcola l'energia necessaria per il tragitto da Well To Tank (WTT). Utilizza self.pp che rappresenta la percentuale di utilizzo della benzina nel caso di bifuel

- a) :param fuel: tipo di combustibile
- b) :param cons: consumi del primo combustibile
- c) :param cons1: consumi del secondo combustibile
- d) :return: energia in MJ aggiuntiva per la gestione WTT

5.7.2.17 Wttco2

Calcola le emissioni di CO₂ da Well To Tank (WTT). Utilizza self.pp che rappresenta la percentuale di utilizzo della benzina nel caso di bifuel

- a) :param fuel: tipo di combustibile
- b) :param cons: consumi del primo combustibile
- c) :param cons1: consumi del secondo combustibile
- d) :return:

5.8 *Analisi dei risultati*

In Figura 23 sono rappresentati i consumi totali per i veicoli privati calcolati dal software emucons per un totale di 21680997 tratte considerate. Le curve sono differenziate per tecnologia (diesel, benzina...) e classe di veicoli (SUV, veicoli medi, piccoli). L'andamento dei consumi calcolati ricalca come atteso quello delle curve Copert discusso in precedenza.

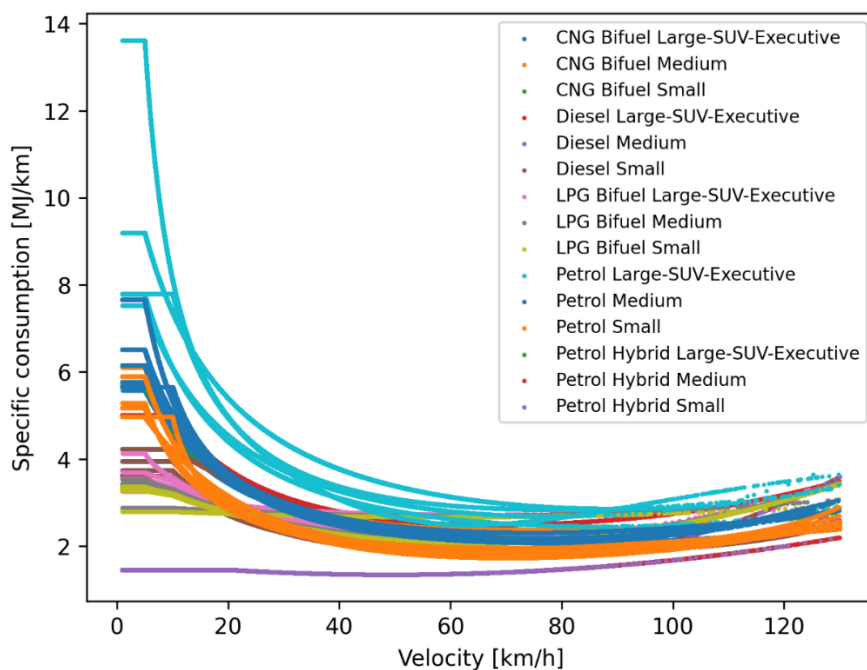


Figura 23: Consumi simulati dal software emucons per le tratte disponibili. Il numero di punti è pari a 21680997. I colori rappresentano diverse tecnologie e categorie di veicoli. La presenza di più curve per la stessa classe è dovuta a diversi Standard di emissione.

Per molte classi sono evidenti curve di consumo differenziate che dipendono dalla presenza di un insieme di standard emissivi (si veda Figura 24). Le curve Copert hanno validità in un range di velocità compreso tra un minimo, che varia a seconda della categoria del veicolo tra 5 km/h e 20 km/h, e un massimo, generalmente di 130 km/h per i veicoli privati. Nell'ambito del software emucons è stato scelto di considerare solamente velocità inferiori a 130 km/h, eliminando tutte le tratte percorse con velocità maggiori di 130 km/h (16402 tratte, che rappresentano circa lo 0.075% del totale). Al contrario, per quanto riguarda velocità di tratta inferiori alle velocità minime delle curve Copert, è stato scelto di utilizzare i valori di consumo associati alla velocità minima ammessa in Copert. Questo spiega i tratti piatti per velocità basse rappresentati in Figura 23. Informazioni analoghe sono presentate in Figura 24, dove i diversi pannelli rappresentano in modo separato le tecnologie disponibili. In aggiunta ogni pannello riporta anche la divisione dei consumi nelle diverse categorie emissive che caratterizzano la tecnologia considerata. Risulta evidente come la categoria veicolare con i consumi maggiori sia rappresentata dai Large SUV- Executive, in particolare le prime generazioni, con un marcato miglioramento dei consumi associato alla variazione delle tecnologie emissive (Euro 2, Euro 3, ... Euro 6).

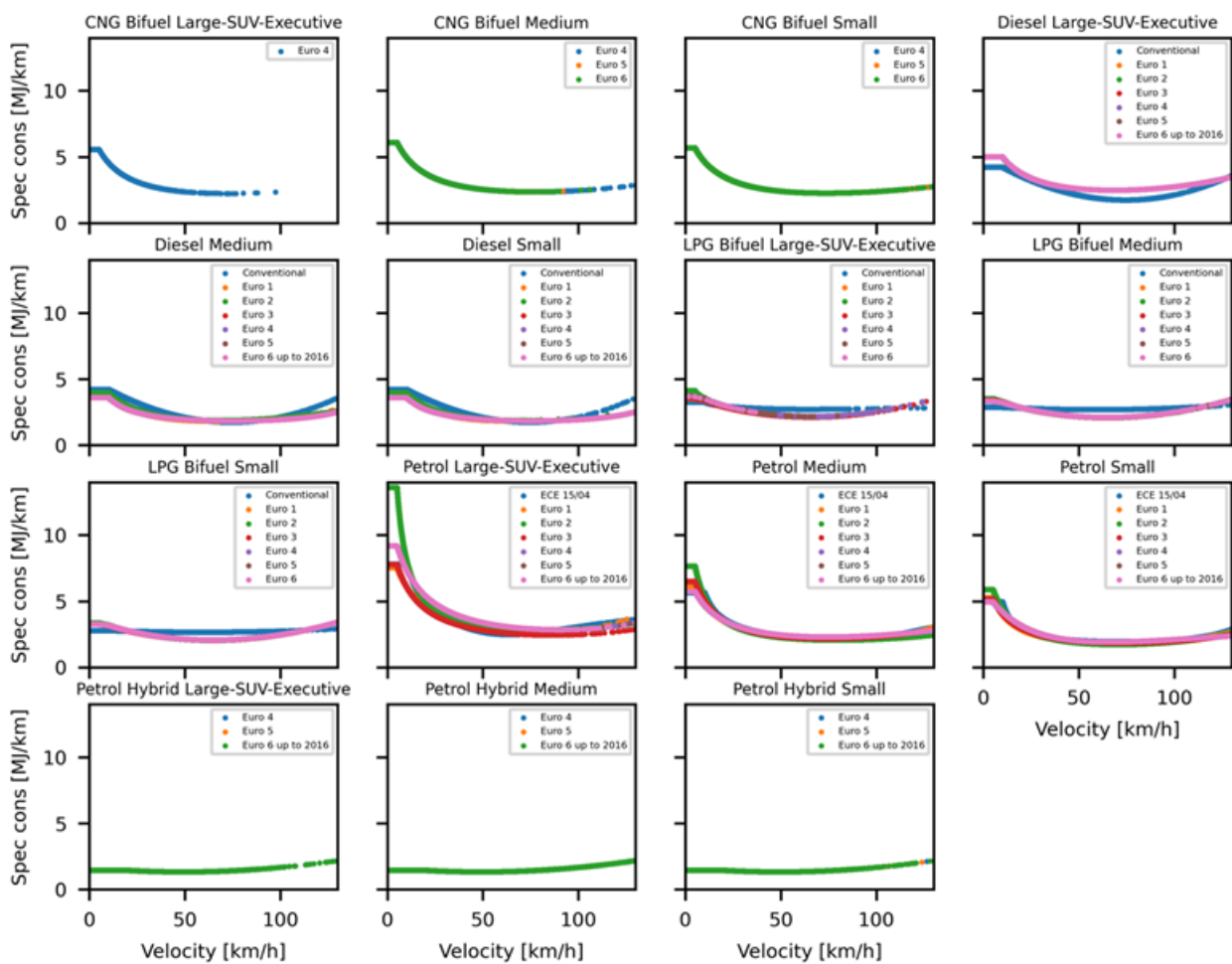


Figura 24: consumi simulati dal software emucons per le tratte disponibili. La presenza di più curve per la stessa classe è dovuta a diversi Standard di emissione, che sono rappresentati in ogni pannello per ogni tecnologia considerata

Risulta anche evidente l’impatto dei veicoli bi-fuel e soprattutto dei veicoli ibridi nel ridurre i consumi a basse velocità.

Nel caso dei veicoli CNG bifuel, Large SUV-Executive e in parte anche veicoli di dimensione media, le curve non risultano complete a causa del limitato numero di veicoli nel parco auto, che ne determina un uso limitato rispetto alle altre tecnologie.

Dall’analisi dei pannelli di Figura 24 si nota come l’impatto sui consumi dei diversi standard emissivi (classificazione Euro) sia soprattutto legato alle basse velocità, come evidente ad esempio per i veicoli SUV e le auto di dimensioni medie a benzina. E’ inoltre utile notare come l’incremento dei consumi per basse velocità sia in generale più marcato per i veicoli a benzina rispetto alle altre tecnologie.

Analogamente al caso dei veicoli privati in Figura 25 sono rappresentati i consumi simulati da emucons per gli autobus per i giorni feriali (2 Ottobre 2020), in Figura 26 per i giorni prefestivi (3 Ottobre 2020) e in Figura 27 per i giorni festivi (4 Ottobre 2020).

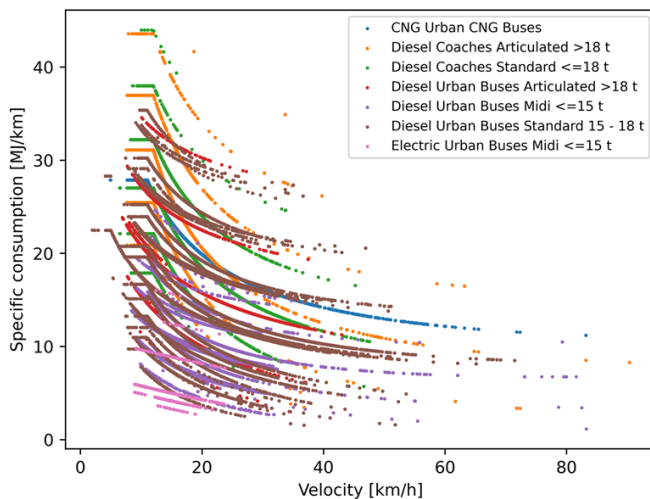


Figura 25: Consumi simulati dal software emucons per le tratte percorse dagli autobus per giornate feriali

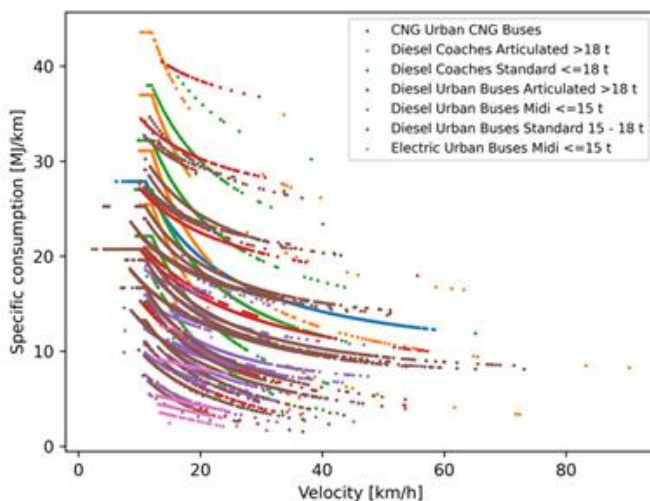


Figura 26: Consumi simulati dal software emucons per le tratte percorse dagli autobus per giornate prefestive

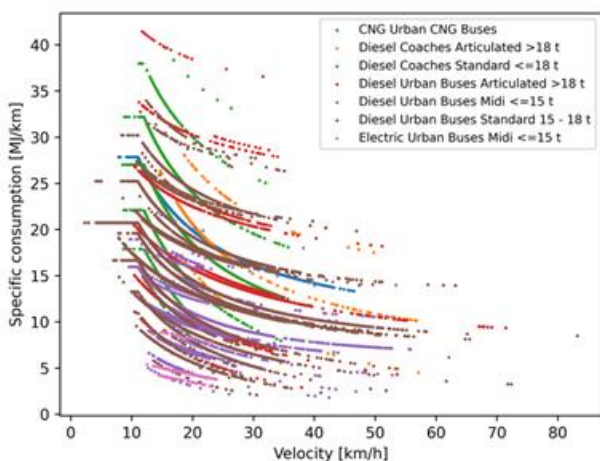


Figura 27: Consumi simulati dal software emucons per le tratte percorse dagli autobus per giornate festive

Per il caso feriale risultati analoghi sono anche presentati in Figura 28 dove è disponibile anche una classificazione per standard emissivo. In questo caso, le diverse curve che vengono evidenziate dalla

sequenza di punti dipendono non solo dalla presenza di differenti standard emissivi ma anche dal fatto che i consumi sono relativi a tratti di strada caratterizzati da pendenze variabili.

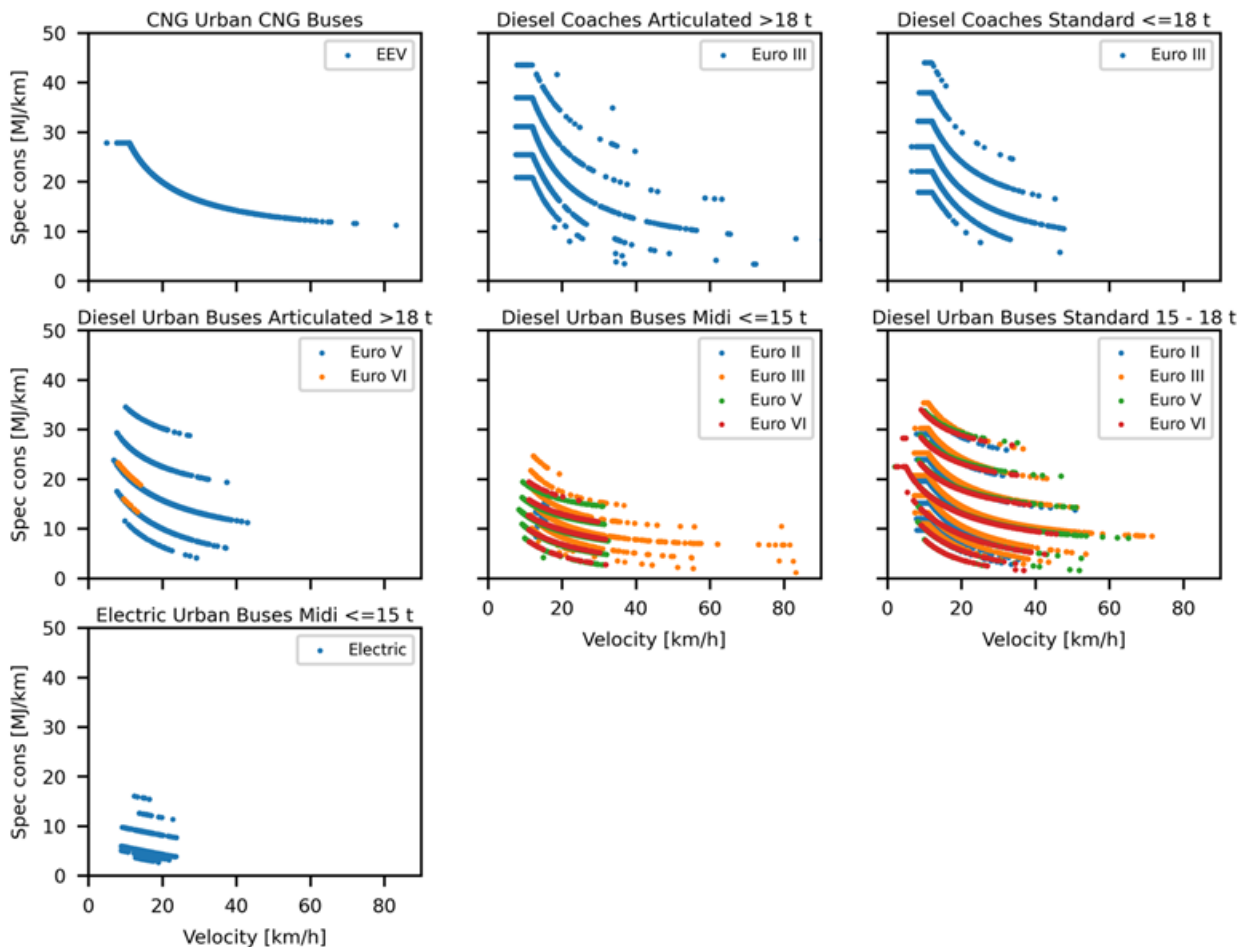


Figura 28: Consumi simulati dal software emucons per le tratte percorse dagli autobus

Come atteso, nei giorni prefestivi e festivi il parco mezzi utilizzato è ridotto e questo spiega l'assenza di alcune curve presenti nel grafico per il giorno feriale. Inoltre, analogamente al caso dei veicoli privati, alcune tecnologie sono presenti nel parco veicoli in modo limitato, per cui non tutto il range di velocità è coperto dai dati disponibili. Notiamo in generale che il range di velocità degli autobus articolati è ridotto rispetto agli altri modelli, così come quello dei veicoli elettrici.

Essendo la maggioranza dei mezzi disponibili con tecnologia diesel, si nota un impatto relativamente piccolo (anche se non trascurabile) delle variazioni di standard emissivo, coerentemente con le differenze evidenziate nel caso dei veicoli privati fra trazione diesel e trazione a benzina.

Il software Emucons permette anche di simulare altre variabili, in generale derivate dai consumi TTW descritti in precedenza. In Figura 29 e Figura 30 sono riportate le emissioni di CO₂ rispettivamente per veicoli privati e autobus. Le strutture dei grafici sono leggermente diverse rispetto alle corrispettive strutture dei consumi, a causa delle differenze nei coefficienti di emissione per le tecnologie considerate.

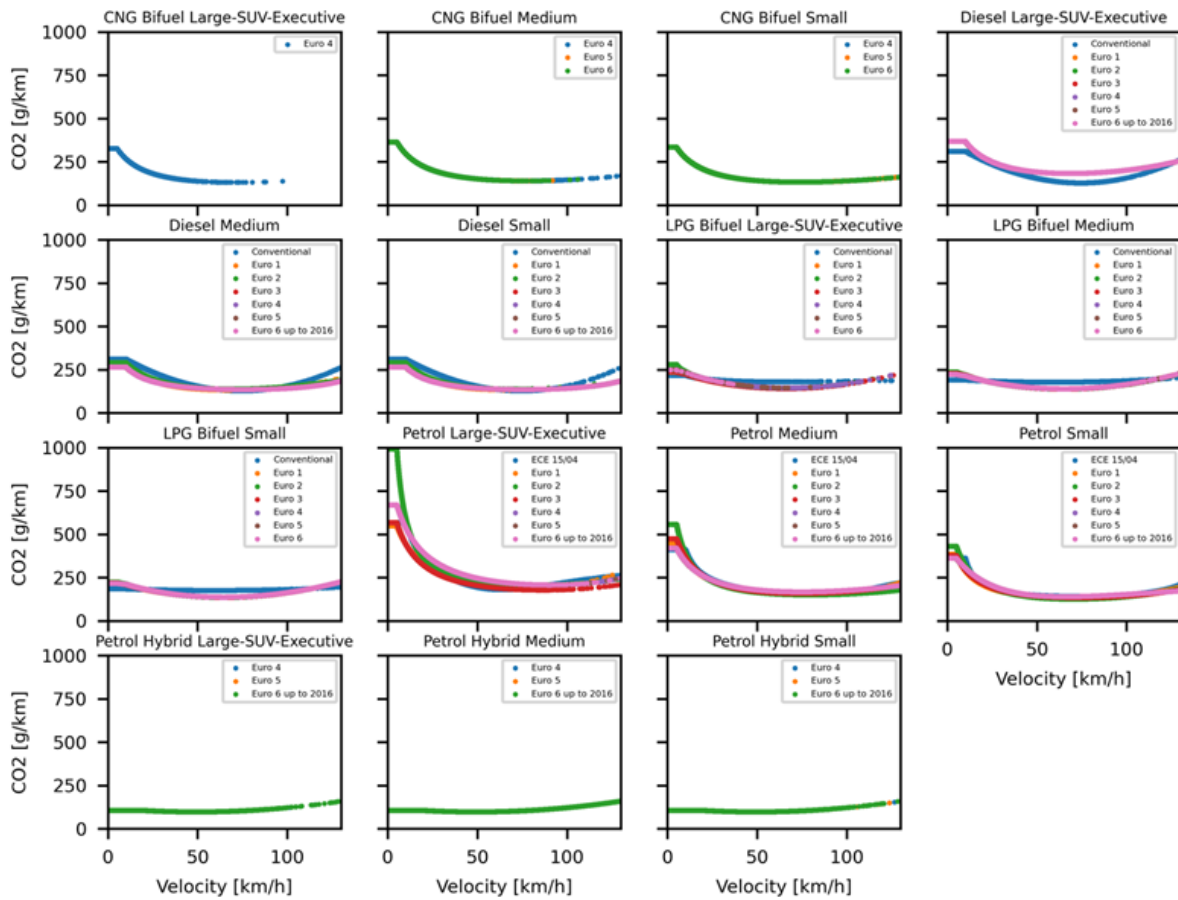


Figura 29: CO2 per km percorso calcolata a partire dai consumi TTW per i veicoli privati

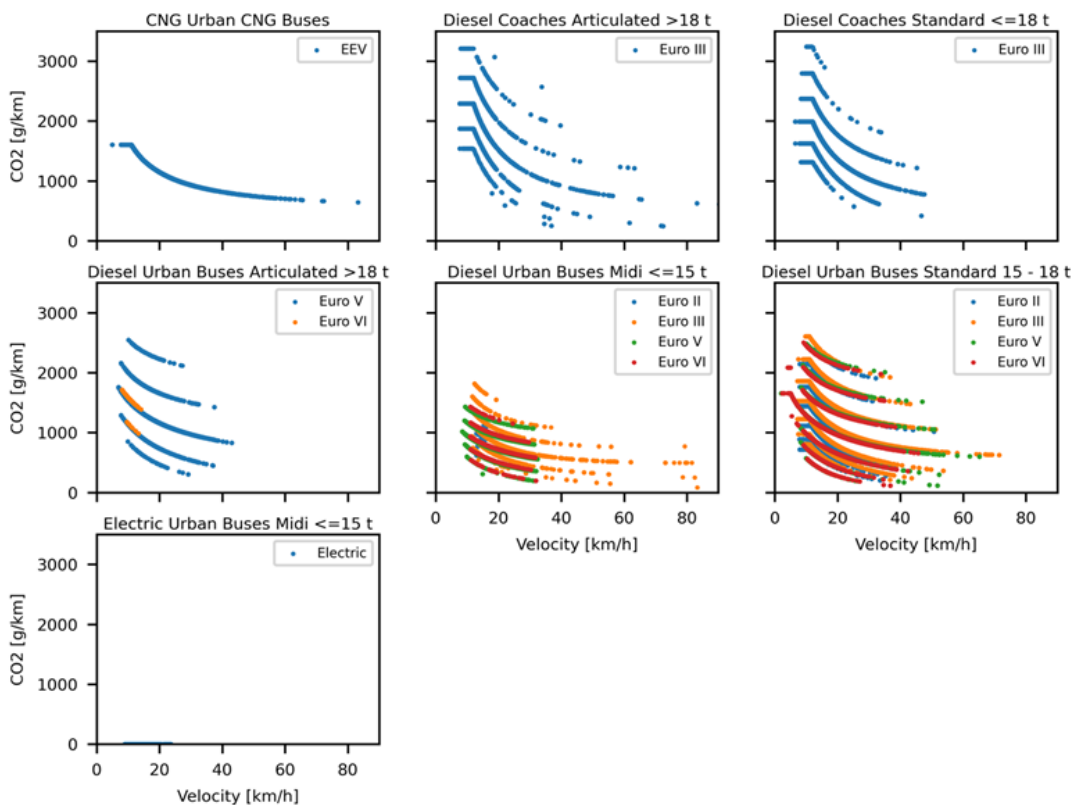


Figura 30: CO2 per km percorso calcolata a partire dai consumi TTW per gli autobus nel corso di una giornata feriale

In Figura 31 e Figura 32 è riportata l'energia spesa in ambito WTT rispettivamente per veicoli privati ed autobus, a partire dai consumi specifici calcolati per il TTW.

In generale l'energia addizionale WTT è, come atteso, una frazione di quella consumata per trazione, mostrando comunque la stessa struttura, con piccole differenze associate ai coefficienti specifici per le diverse tecnologie.

Una valutazione analoga può essere fatta per le emissioni di anidride carbonica nella fase WTT, presentate rispettivamente in Figura 31 e Figura 32. E' utile notare come le emissioni più alte siano nel caso degli autobus quelle associate alla trazione elettrica, anche se nella somma totale delle emissioni di CO2 è predominante la parte TTW, che sono nulle per gli autobus elettrici.

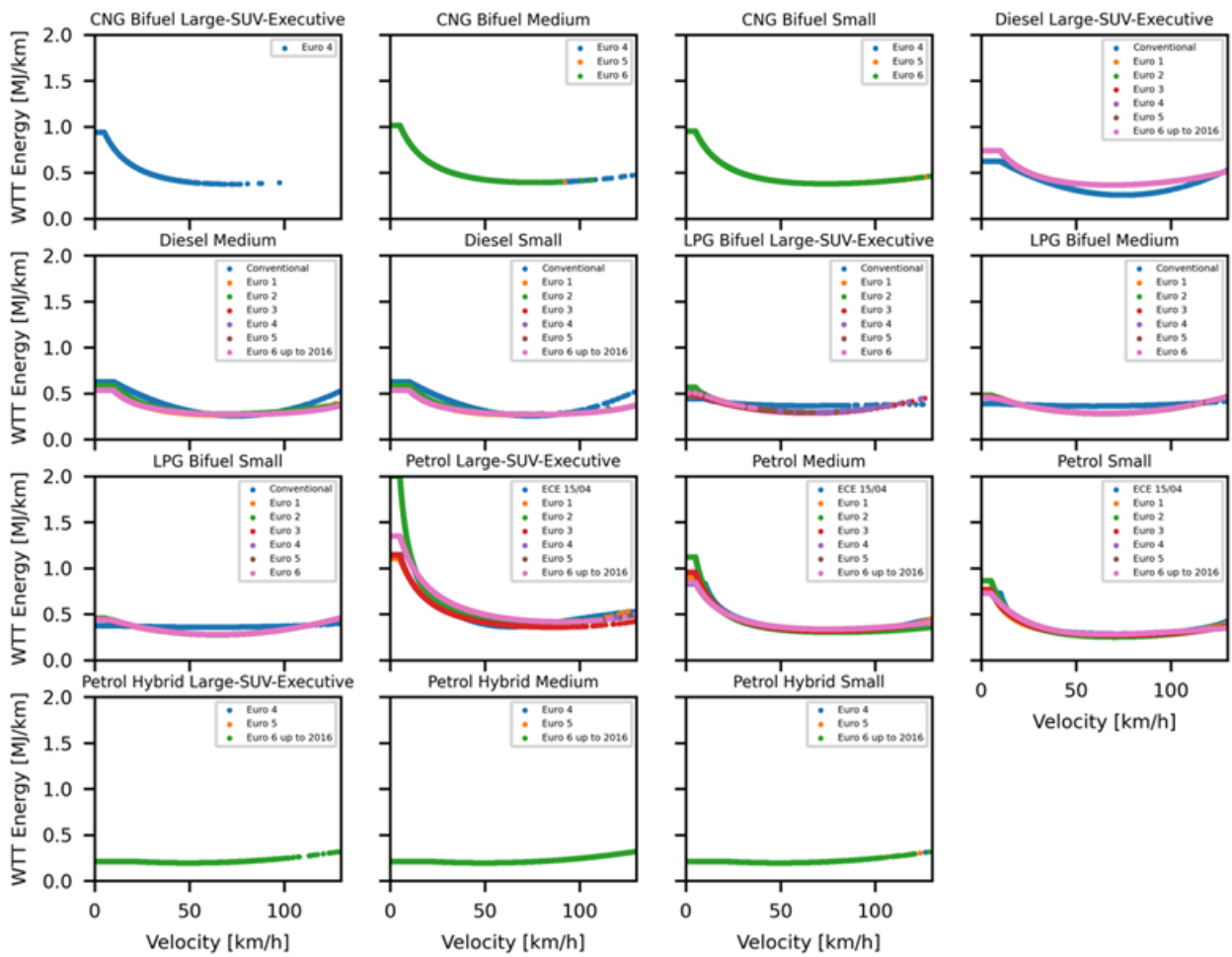


Figura 31: Energia Well To Tank (WTT) per km percorso calcolata a partire dai consumi TTW per i veicoli privati

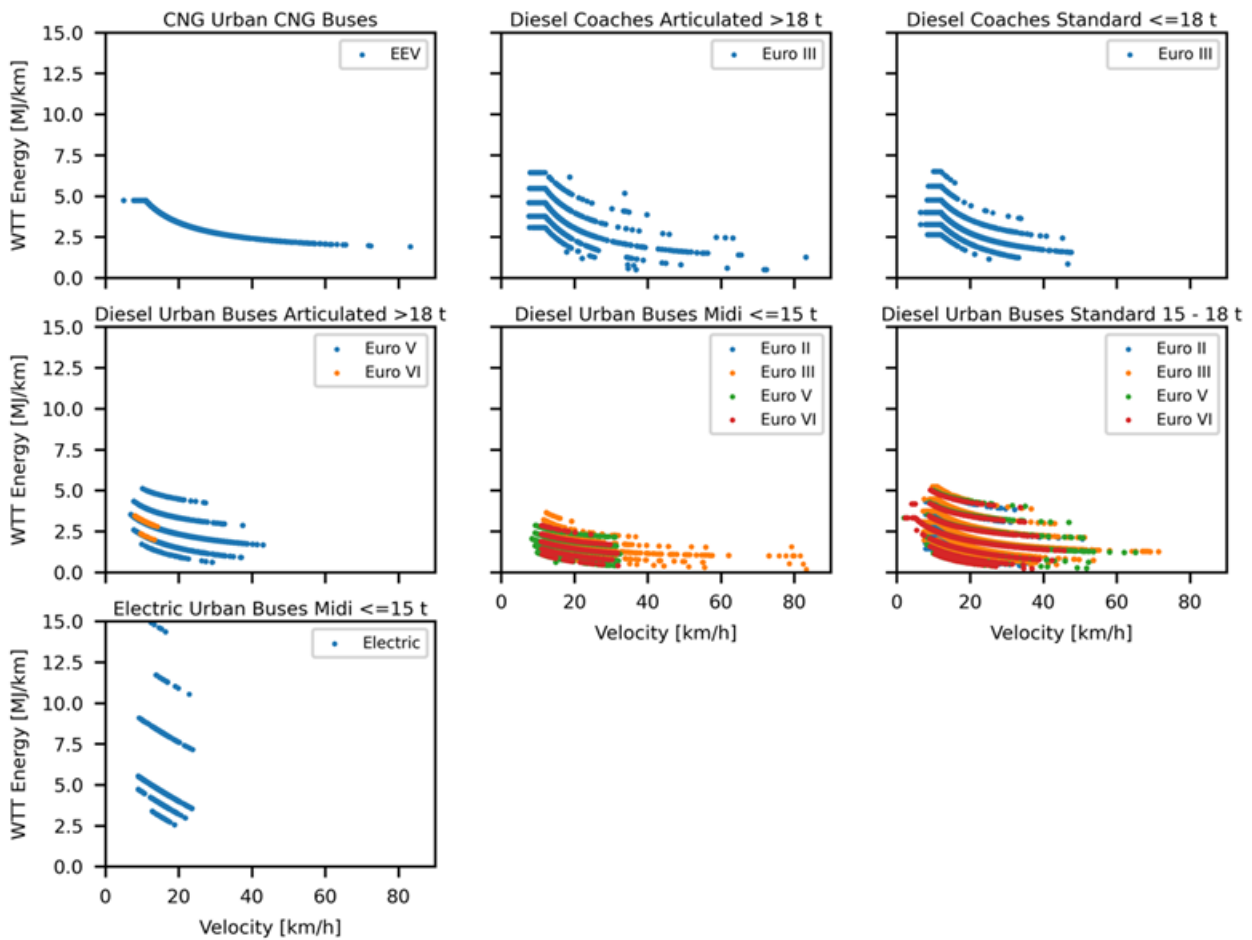


Figura 32: Energia Well To Tank (WTT) per km percorso calcolata a partire dai consumi TTW per gli autobus nel corso di una giornata ferial

6 Il calcolo dei consumi per i modi di trasporto su ferro

Il consumo di energia totale dei modi di trasporto su ferro è dato dalla somma dell'energia per la trazione E_{in} e dell'energia per i servizi ausiliari E_{aux} meno l'energia recuperata E_{maxESS} , nel caso di veicoli equipaggiati da ESS durante la frenata:

$$E = E_{in} + E_{aux} - E_{maxESS} \tag{10}$$

Per il calcolo dei consumi di trazione E_{in} è stato utilizzato il modello di calcolo Backward che considera il flusso di potenza inverso, calcolando la forza di trazione richiesta alle ruote e l'efficienza dei componenti del sistema a ritroso verso il motore. Il modello offre una valutazione affidabile del consumo di energia del veicolo sulla base del duty cycle e di dati caratteristici più aggregati dei veicoli. In questo metodo si parte dai diagrammi di marcia dei convogli (accelerazioni, velocità e distanze percorsa nel tempo) per calcolare la potenza alle ruote e, successivamente, la potenza assorbita alla catenaria e il consumo totale di energia. I diagrammi di marcia rappresentano le caratteristiche dei servizi offerti tra le fermate nel rispetto degli orari, con cicli di guida standard comprendenti fasi di accelerazione, cruising e decelerazione. Pertanto, i

diagrammi di marcia rappresentativi degli orari sulle singole tratte sono definibili sulla base dei dati GTFS disponibili.

E_{in} si ottiene integrando la potenza input P_{in} nella catenaria nel tempo:

$$E_{in} = \int P_{in} dt \quad (11)$$

Per stimare P_{in} si avrebbe bisogno delle seguenti fasi di calcolo e di tutti i dati di input riportati in Tabella 8.

$$P_{in} = \frac{P_{out}}{\eta} \quad (12)$$

$$P_T \text{ or } P_{out} = 2.778 \cdot 10^{-7} v F_T \quad (13)$$

$$F_T = ma + F_R \quad (14)$$

$$F_R = 2.67W + 88.96n + 9.95fWv + 22.26Kv^2 + 88.96WG \quad (15)$$

Dove:

P_{in} è la potenza input o potenza nella catenaria [W]

P_T è la potenza output o potenza alle ruote [W]

F_T è la forza di trazione [N]

F_R è la resistenza alla guida totale [N]

v è la velocità [m/s]

A seconda delle fasi del ciclo di marcia, varia la velocità e quindi la P_{in} .

La fase di accelerazione si considera di moto uniformemente accelerato secondo la seguente formula:

$$v = at \quad (16)$$

Sostituendo la funzione(15) nella (14) e la funzione F_R così calcolata nella equazione (13) per il calcolo di F_T da immettere nella formula (11) si ottiene la potenza input P_{in} in funzione del tempo nelle due fasi suddette.

Sostituendo la funzione P_{in} nella formula (10) e integrando nel tempo t_1 della fase di accelerazione, si ottiene la seguente formula per l'energia E_{in}^{acc} espressa in kWh:

$$E_{in}^{acc} = 2.778 \cdot 10^{-7} \frac{a}{\eta} \left(ma \frac{t_1^2}{2} + 2.67W \frac{t_1^2}{2} + 88.96n \frac{t_1^2}{2} + 9.95fWa \frac{t_1^3}{3} + 22.26Ka^2 \frac{t_1^4}{4} + 88.96WG \frac{t_1^2}{3} \right) \quad (17)$$

Il consumo di trazione in kWh della fase di *crusing* a velocità costante v_{max} e per la durata del tempo t_2 sarà dato dalla seguente formula:

(18)

$$E_{in}^{v_{max}} = 2.778 \cdot 10^{-7} \frac{v_{max} t_2}{\eta} (2.67W + 88.96n + 9.95fWv_{max} + 22.26Kv_{max}^2 + 88.96WG)$$

Il consumo degli ausiliari, anch'esso dipendente dal tempo, invece, si calcola a partire dalla seguente formula:

(19)

$$E_{aux} = \int P_{aux} dt$$

Considerando la potenza P_{aux} costante nel tempo, il consumo E_{aux} si ottiene moltiplicando la potenza per il tempo.

Nella Tabella 8 sono state riassunte tutte le variabili necessarie e le relative formule per il modello di calcolo elettrico dei sistemi ferroviari.

Tabella 8 Definizione dei dati input del modello cinematico per il calcolo del consumo di energia

Variabile	Descrizione	UdM
m	Massa del veicolo (carico) = $(1 + \lambda) \cdot m_{tara} + l \cdot m_{pas}$	[kg]
m_{tara}	Massa di tara	[kg]
m_{pass}	Carico di merci o passeggeri	[kg]
λ	Fattore massa rotante	
l	Fattore di carico	
a	Accelerazione istantanea ≤ min {a massima forza di trazione e frenata del treno, a massima per condizioni comfort dei passeggeri}	[m/s ²]
v	Velocità istantanea ≤ min {v massima del treno, v limitata dalla tratta}	[m/s]
η	Efficienza dei componenti di trazione = $\eta_{axle\ gear} \times \eta_{motor} \times \eta_{motor\ inverter} \times \dots \times \eta_{pantograph}$	
P_{aux}	Potenza degli ausiliari = $P_{const} + P_{cool}$	[W]
P_{const}	potenza richiesta da parte di ausiliari come il sistema HVAC	[W]
P_{cool}	potenza totale per il raffreddamento del motore	[W]
E_{maxESS}	Capacità massima del sistema di accumulo dell'energia = $\eta_{charge} \cdot v \cdot F_T^- \cdot \Delta t_{charge}$	[Wh]
η_{charge}	Efficienza della ricarica dell'ESS	
Δt_{charge}	Tempo richiesto per la ricarica completa dell'ESS	[s]
W	Peso totale del veicolo (carico)	[t]
n	Numero di assi	
f	Coefficiente di attrito tra rotaia e ruota che varia tra 0,001 e 0,01	[s/m]
K	Coefficiente di massa per resistenza aerodinamica	[N/(m/s) ²]
G	Pendenza in percentuale (positivo in salita e negativo in discesa)	[%]
t₁, t₂	Tempo di funzionamento in accelerazione e cruising	[s]

Nella Figura 33 sono riportati i profili giornalieri di consumo di elettricità in kWh nell'area metropolitana di Roma stimati con il modello Backward a partire dai dati GTFS relativi alle modalità di trasporto su ferro: tram, metro e treni locali.

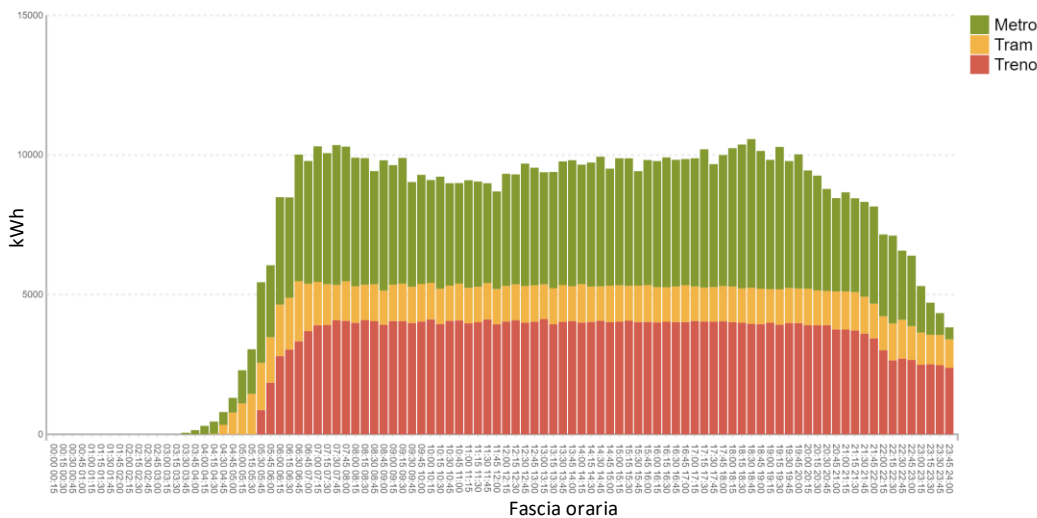


Figura 33: Profilo giornaliero dei consumi elettrici di metro, tram e treni locali.

7 Struttura ed interfaccia della piattaforma software

In questa sezione sono descritte la struttura della nuova piattaforma web finalizzata all’archiviazione e rappresentazione dei risultati provenienti da diverse metodologie di elaborazione dati. L’ambiente software proposto espone la tipica architettura *client-server* che garantisce l’accessibilità web da qualsiasi postazione collegata in rete senza la necessità di utilizzo di software dedicati.

La piattaforma si compone di un *front-end* e di un *back-end*; il primo consente all’utente di interagire direttamente con l’ambiente software attraverso una rappresentazione visiva delle sue funzionalità e, in tal modo, di visualizzare e, eventualmente, confrontare le elaborazioni prodotte dal *back-end*. In altre parole, la configurazione del *front-end* assicura, mediante un apposito editing grafico semplice ed intuitivo, l’impostazione dei parametri del sistema di simulazione della mobilità urbana e dei risultanti consumi energetici.

La componente di *back-end* non è accessibile né visibile all’utente finale e si riferisce al software sviluppato per generare tutte quelle informazioni richiamate dal *front-end* in fase di visualizzazione. I formati scelti per l’interscambio dati sono il *JSON* [21] e il *CSV*⁷. Una volta aggregati omogeneamente, tali pacchetti di informazione sono spediti al *front-end* a richiesta, utilizzando il minor spazio disco possibile in modo da garantire un’efficienza adeguata nella fase di visualizzazione da parte del *front-end*. La generazione dei dati avviene attraverso il linguaggio *Python* [22], linguaggio ad alto livello orientato agli oggetti, in grado di garantire uniformità e scalabilità e al contempo di realizzare script multithreading per migliorare la velocità di produzione dei dati.

7.1 Descrizione della piattaforma e rappresentazione dei dati

⁷ CSV (Comma-Separated Values) è un formato basato su file di testo tipicamente utilizzato per la rappresentazione di una tabella di dati

7.1.1 Front-end

L'applicazione front-end deve consentire la rappresentazione delle relazioni di base e dei risultati generati dalle simulazioni sia in forma disaggregata attraverso una geo-referenziazione su mappa digitale, che in modalità aggregata, mediante la generazione di grafici e tabelle riassuntive che garantiscano una comprensione immediata delle informazioni esposte.

La componente *front-end* include inoltre tutte le finestre di dialogo necessarie per impostare i parametri delle funzioni dedicate alla simulazione di scenari di mobilità urbana.

I linguaggi di programmazione scelti per la realizzazione della componente *front-end* della piattaforma sono: *PHP* [23], *JAVAScript*⁸, *HTML* e *CSS*. Inoltre, alcune librerie open source quali *D3.js* [24], *Jquery* [25] (librerie *JAVAScript*) sono state selezionate per gestire le diverse informazioni da visualizzare.

Specifici moduli o schede (*form*) consentono all'utente di inviare input e dati al *back-end*; quest'ultimo è in grado di elaborare tali informazioni e restituire i risultati del particolare processo di esecuzione svolto. Sono stati scelti formati standard per la restituzione dei dati da parte del *back-end*, quali il formato *JSON* [21] e *GEOJSON* [26].

Il collegamento tra *front-end* e *back-end* è realizzato mediante un'interfaccia di programmazione delle applicazioni conforme ai vincoli dell'architettura *REST (REpresentational State Transfer)* che assicura l'invio e la ricezione dati tramite protocollo *http*.

I *layers* sviluppati per la visualizzazione delle informazioni afferiscono a due macro categorie: trasporto pubblico locale (TPL) e *Floating Car Data (FCD)*. Al fine di fornire informazioni a diverse scale di rappresentazione, è stato necessario aggregare i dati in funzione della specifica visualizzazione da esporre.

Ad esempio, in Figura 34 sono state riportate su mappa geo-referenzata tutte le fermate effettuate dal TPL nella provincia di Roma, aggregate per ora del giorno. In Figura 35 sono invece riprodotti il numero di transiti effettuati dalle diverse linee del TPL riferiti al singolo arco stradale. Infine, in Figura 36, una zonizzazione mediante celle esagonali consente di aggregare i dati di transito delle linee del TPL in modo da ottenere delle informazioni relative a specifiche aree.

⁸ *JAVAScript: linguaggio di programmazione*

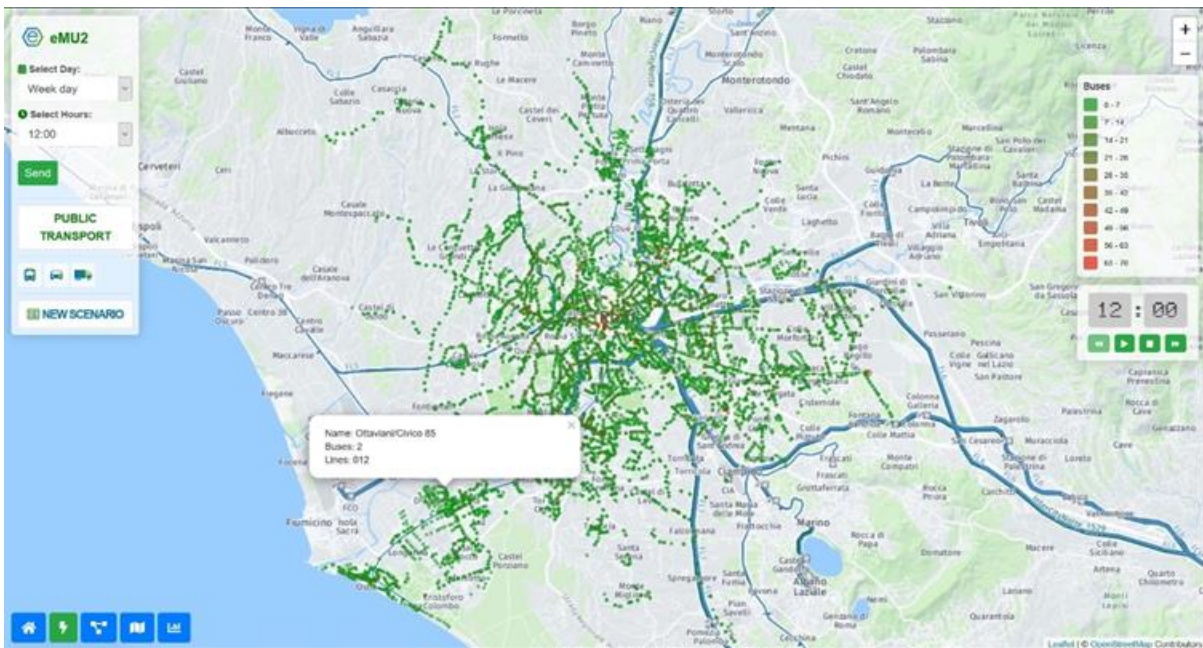


Figura 34: Layer TPL delle fermate

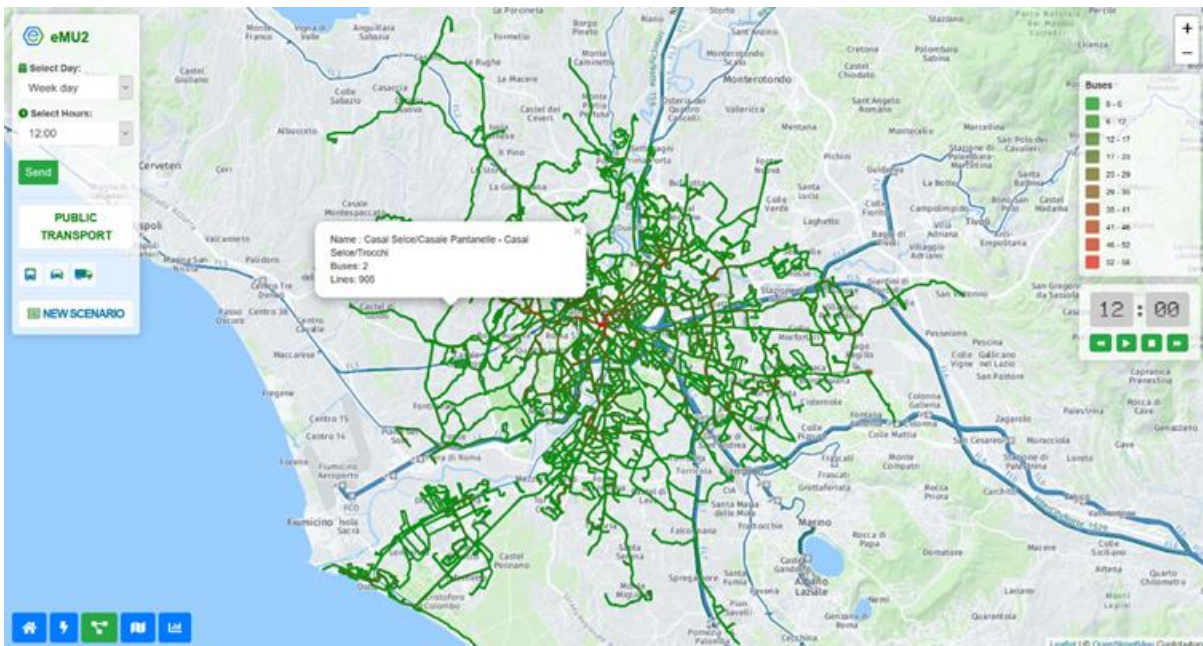


Figura 35: Layer TPL dei tracciati lineari

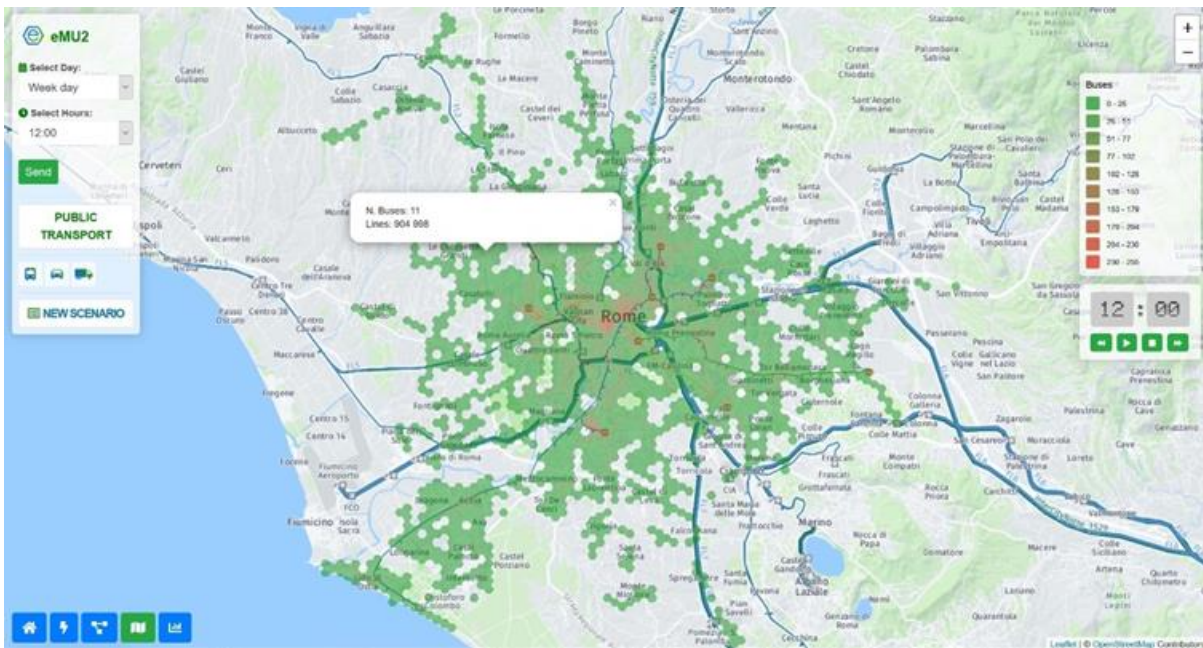


Figura 36: Layer TPL della zonizzazione esagonale

Nelle tre diverse rappresentazioni realizzate è possibile selezionare la fascia oraria e la tipologia di giornata (i.e. lavorativo, prefestivo o festivo). Inoltre, mediante uno specifico player, è possibile anche realizzare una visualizzazione dinamica in grado di evidenziare le variazioni dell’impatto dei transiti del TPL su mappa per un determinato giorno, sulla base dell’orario di riferimento. Il player consente anche una progressione step-by-step bidirezionale (sia avanti che indietro nel tempo) per intervalli di 1 ora.

Selezionando mediante mouse una fermata, un arco o una cella esagonale, è possibile evidenziare ulteriori informazioni riguardanti ad esempio i nomi delle fermate, il numero di passaggi orari, i nomi delle linee associate, ecc.

La rappresentazione del trasporto privato è stata sviluppata sulla base di dati geo-localizzati di posizione, velocità e direzione di viaggio provenienti dal campione di veicoli equipaggiati con unità di bordo. Anche in questo caso i dati sono stati aggregati ed elaborati a diverse scale in funzione della specifica visualizzazione da generare. Ad esempio, sempre sulla base di una zonizzazione esagonale, in Figura 37 sono rappresentati il numero di spostamenti FCD attratti, mentre in Figura 38 si riportano gli spostamenti generati da ogni singola cella. Inoltre in Figura 39 sono riprodotti il numero delle auto equipaggiate in sosta per singola cella e in Figura 40 quello delle auto residenti nella specifica cella.

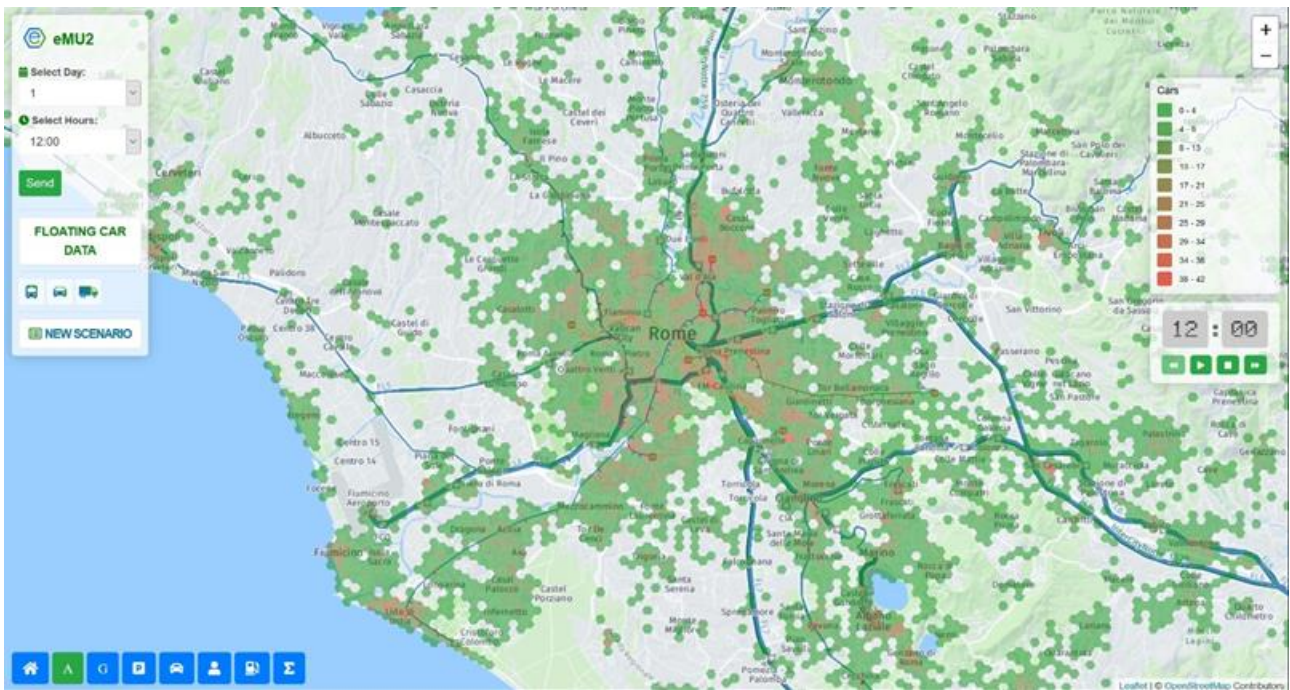


Figura 37: Layer FCD degli spostamenti attratti

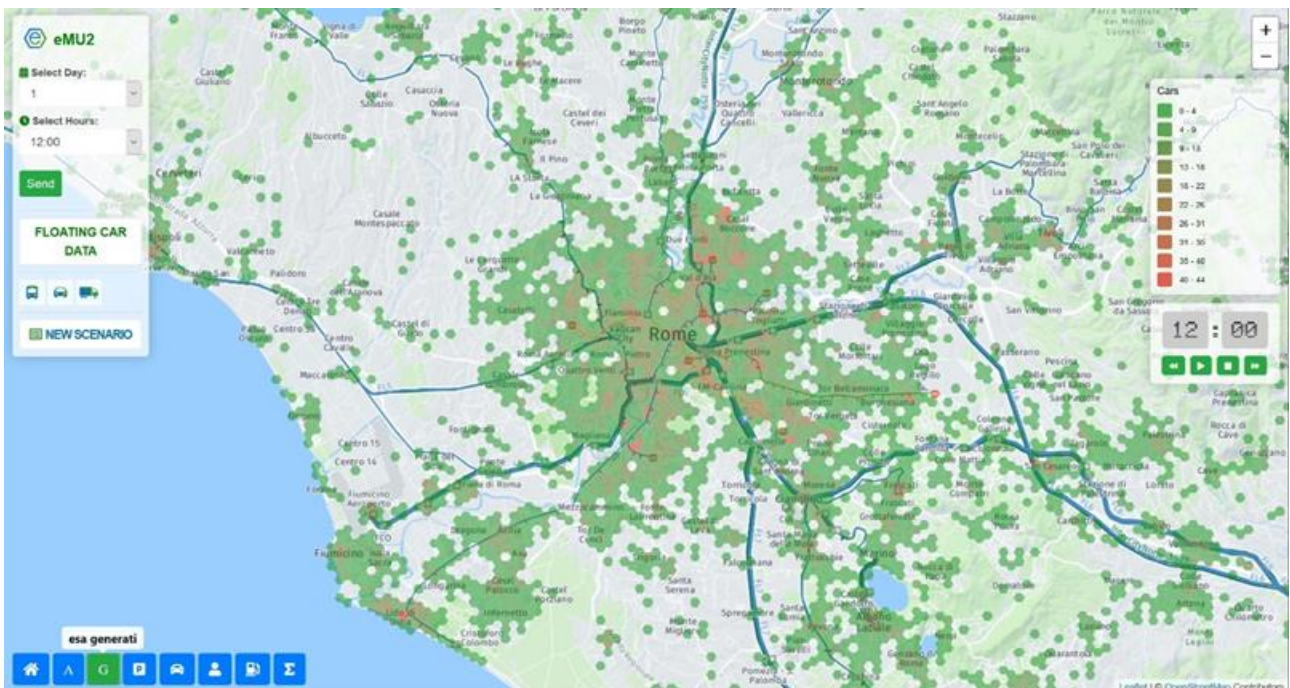


Figura 38: Layer FCD degli spostamenti generati

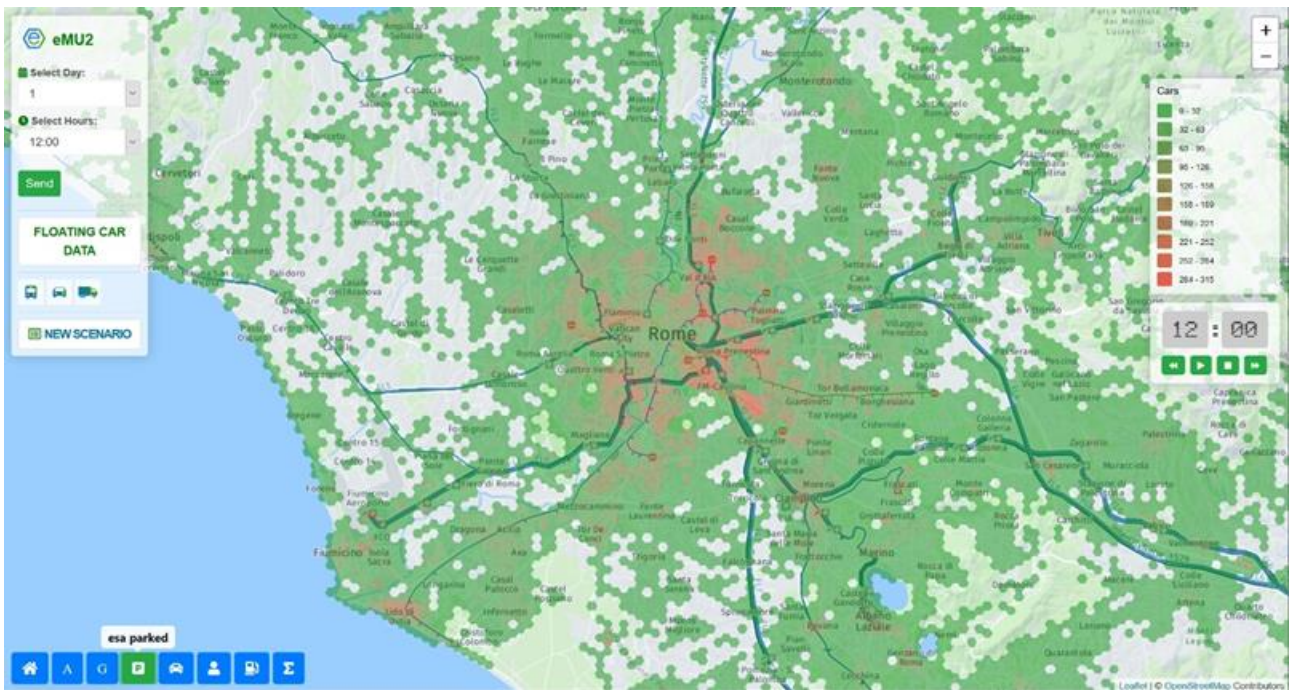


Figura 39: Layer FCD delle auto in sosta

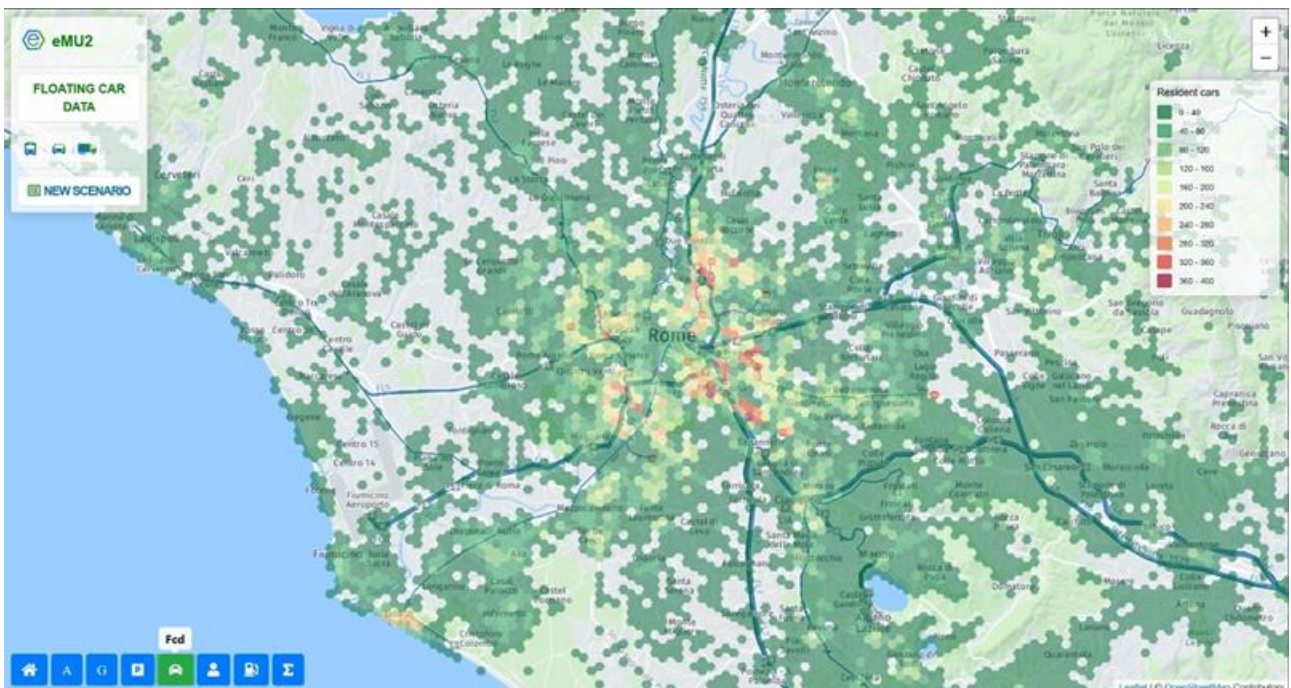


Figura 40: Layer FCD delle auto residenti

Per le suddette rappresentazioni è possibile selezionare la fascia oraria e il particolare giorno del mese.

Inoltre, selezionando mediante mouse una cella esagonale, è possibile evidenziare ulteriori informazioni riguardanti ad esempio il numero di auto generate, emesse, in sosta e residenti in relazione alla cella considerata.

Infine, mediante uno specifico player, è possibile anche realizzare una visualizzazione dinamica in grado di evidenziare le variazioni su mappa del numero di veicoli generati, emessi ed in sosta per un determinato

giorno, sulla base dell'orario di riferimento. Il player consente anche una progressione step-by-step bidirezionale per intervalli di 1 ora.

Ulteriori elaborazioni sono state utilizzati all'interno della macro-categoria FCD sulla base della succitata zonizzazione esagonale. In particolare è stato possibile sviluppare un layer per la visualizzazione della popolazione residente (Figura 41) e un layer raffigurante la rete stradale i cui archi sono visualizzati con spessore variabile sulla base del numero di transiti giornalieri ricostruiti con la procedura di map-matching delle traiettorie FCD sul grafo stradale (Figura 42).

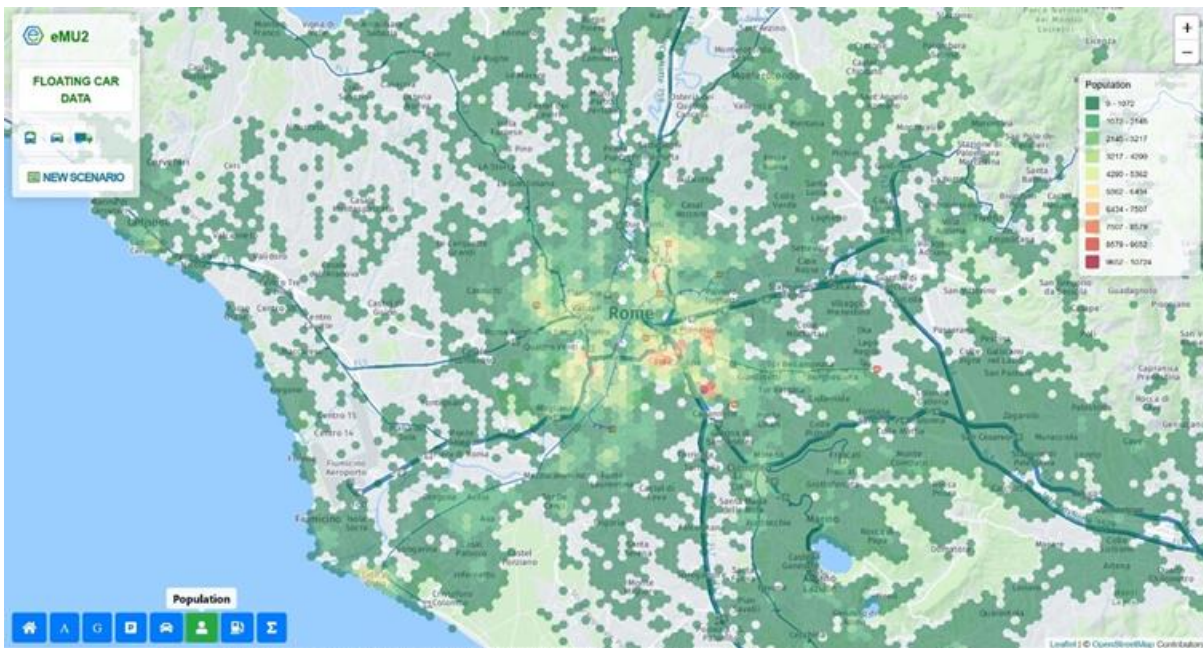


Figura 41: . Layer FCD: popolazione residente

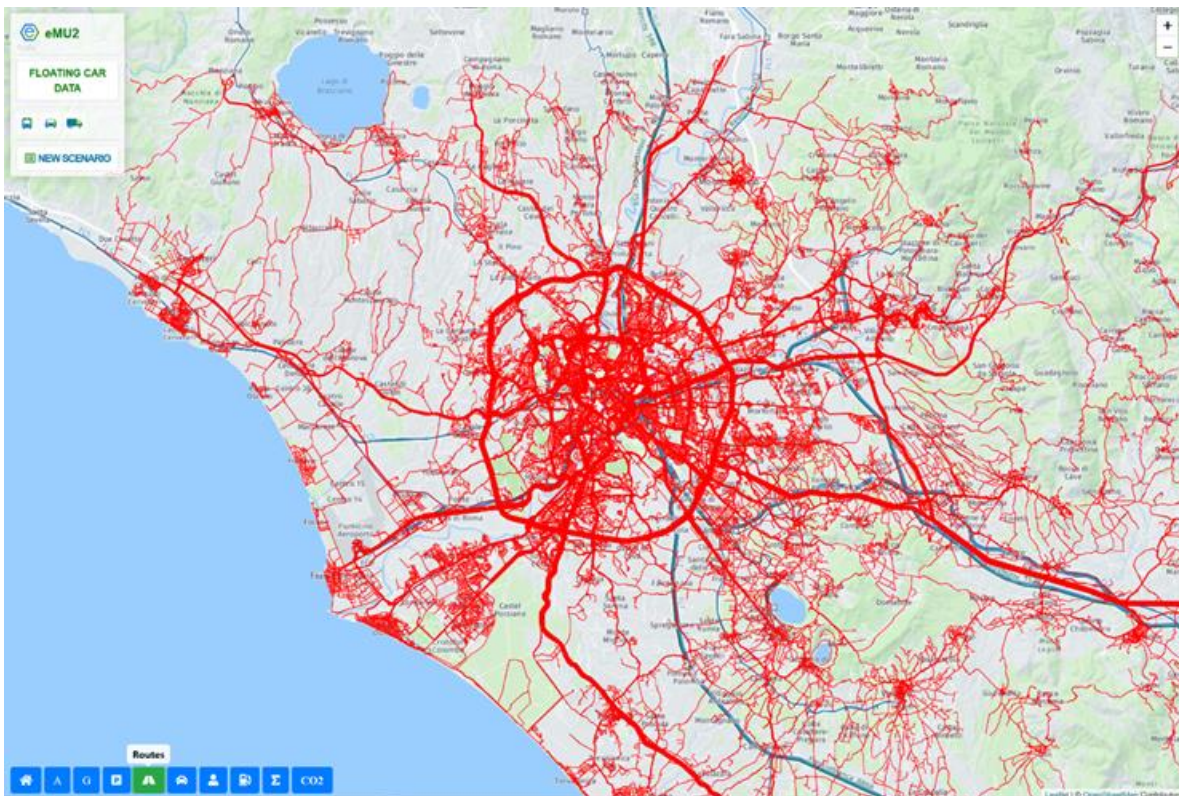


Figura 42: Rappresentazione del grafo sulla base dei flussi giornalieri FCD

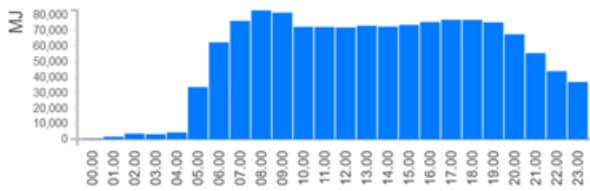
All'interno di entrambe le sezioni riguardanti il TPL e gli FCD, sono state inoltre predisposte delle banche dati per la realizzazione di rappresentazioni grafiche su base giornaliera e mensile.

Un esempio relativo al consumo giornaliero della flotta di autobus del TPL di Roma è riportato in Figura 43. In tale rappresentazione mediante istogrammi, il consumo espresso in Mega Joule (MJ) è calcolato per ciascuna fascia oraria in riferimento alla tipologia di carburante. Sempre sulla base della particolare ora del giorno, sono inoltre riportate anche le rappresentazioni grafiche dei valori di consumo totale così come il numero complessivo di mezzi considerati.

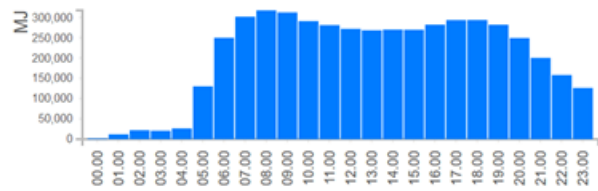
Day Summary



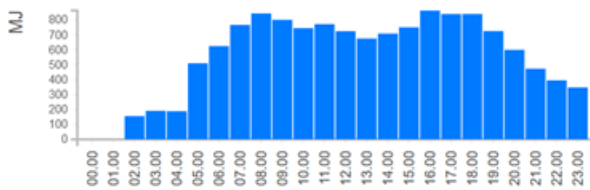
BUS Diesel



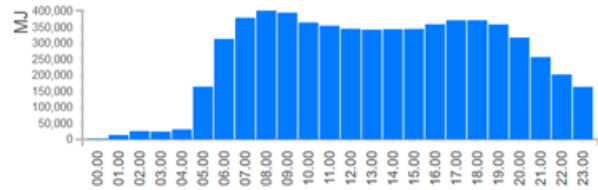
BUS CNG



BUS Electric



Total Consumption



BUS Total

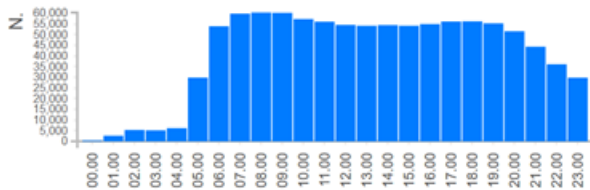


Figura 43: consumi e statistiche TPL

Facendo invece riferimento alla sezione FCD, i dati forniti consentono, ad esempio, la rappresentazione dei consumi dei veicoli privati su base oraria per le diverse alimentazioni considerate (Figura 44) una volta selezionato uno specifico giorno del mese. Le tipologie di carburante considerate sono: *Petrol Hybrid*, *Petrol*, *LPG Bifuel*, *Diesel*, *CNG Bifuel*.

Monthly consumption per day



01 Change day

Car Consumption

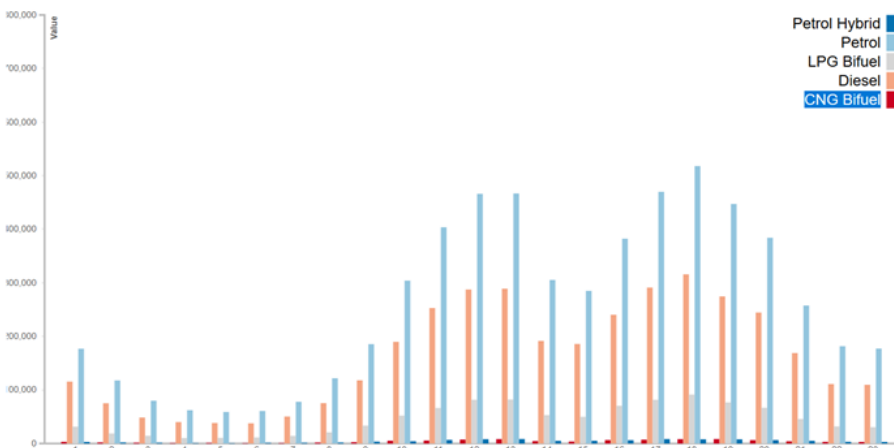


Figura 44: Consumi orari del campione FCD per tipologia di carburante

A titolo di esempio in Figura 45, sono riportati i profili orari dei veicoli FCD in viaggio, nonché i profili giornalieri di traffico per i veicoli residenti e non.

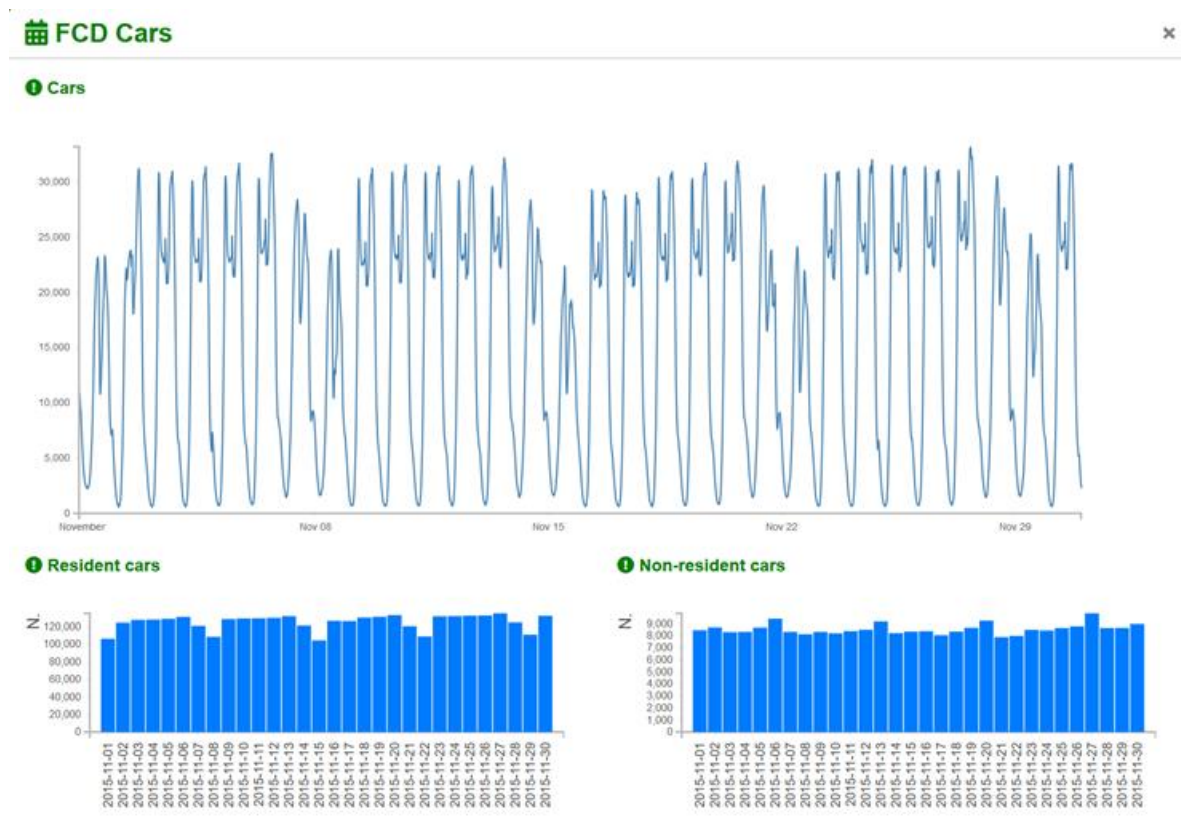
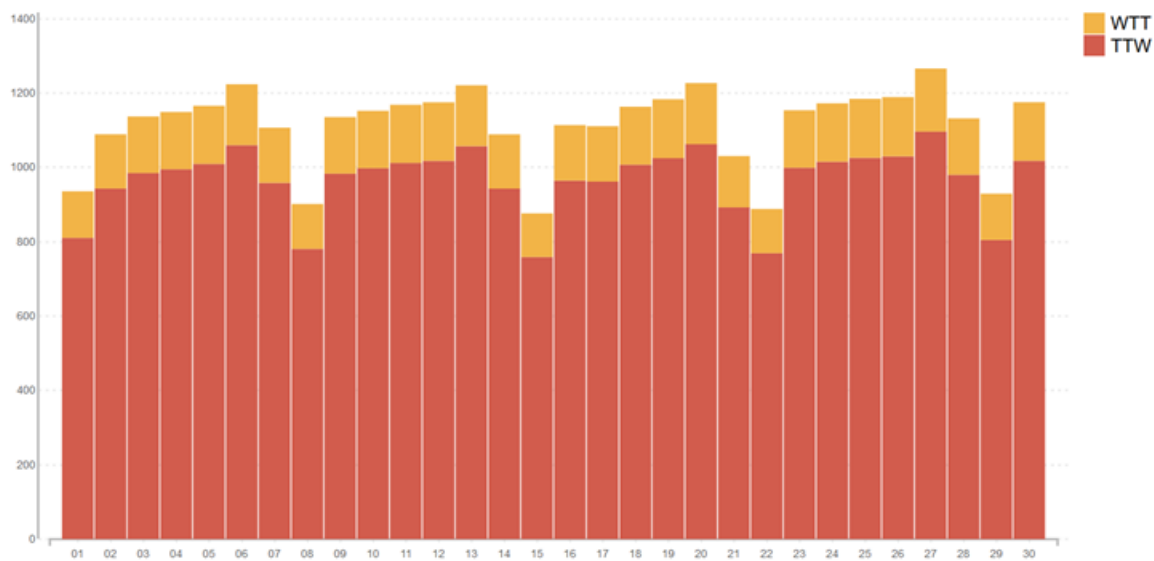


Figura 45: statistiche di traffico dei FCD

Infine in Figura 46 sono riportati i risultati del calcolo delle emissioni di CO2 elaborati per giorno del mese evidenziando sia la quota *Well To Tank* che quella *Tank To Wheel*.

Monthly consumption CO2



Close

Figura 46: Emissioni orarie di CO₂ del campione FCD

7.1.2 Back-end

La componente di *back-end* è stata sviluppata in linguaggio *Python 3.7* [22], linguaggio ad alto livello orientato agli oggetti in grado di garantire una elevata compatibilità con i diversi modelli da integrare nella piattaforma. In tal modo l'obiettivo principale è quello di realizzare un insieme di funzioni che metta in comunicazione i dati e modelli sviluppati con il *front-end* descritto in precedenza nonché di interfacciarsi con un database *MySQL* [27] per il recupero e la visualizzazione dei dati precedentemente processati.

Il *back-end* è la parte software che consente l'effettivo funzionamento del simulatore garantendo la ricezione in input delle varie impostazioni relative agli scenari, l'esecuzione delle simulazioni, il salvataggio dei risultati e l'esposizione all'utente di una serie di feedback mediante *front-end*.

I dati presenti nell'archivio sono organizzati e analizzati utilizzando diversi livelli di zonizzazione del territorio riguardanti l'intera area geografica di applicazione delle analisi, nello specifico l'area corrispondente alla provincia di Roma.

L'infrastruttura informatica principale individuata per la sperimentazione della piattaforma si avvale dell'integrazione di diversi sistemi e strumenti *open source*.

Sono stati realizzati degli script che permettono la generazione dei dati a partire da diverse sorgenti di input. Si possono generare ad esempio dati da un insieme di tabelle registrate su un database *MySQL* [27] oppure selezionando dei file *CSV* o *JSON* [21] per la generazione di dati da visualizzare tramite *front-end*.

Sono stati utilizzati per la generazione delle visualizzazioni sul TPL anche i dati *GTFS* della provincia di Roma.

I file generati dagli script *Python* [22] sono stati realizzati utilizzando il minor spazio possibile su disco per garantire una velocità e fluidità dell'interfaccia di *front-end*; il formato scelto è *JSON* [21] in quanto garantisce una elevata compatibilità con la piattaforma di *front-end* e.

7.1.3 Struttura file sorgenti

La generazione dei file visualizzati dal front-end è stata realizzata a partire da una serie di file e tabelle registrate su di un database *MySQL* [27]77. Di seguito si riporta una descrizione delle strutture dati utilizzate e della modalità di utilizzo in fase di visualizzazione:

7.1.3.1 Consumi_Bus (DB MySQL – lavorativo, prefestivo e festivo):

Questi dati messi in relazione con il file degli stop, considerando le fermate, generano i dati visualizzati nel layer “TPL: fermate” (Figura 34).

Gli stessi dati messi in relazione con il file degli stop, considerando le tratte da una fermata alla successiva, generano i dati visualizzati nel layer “TPL: tracciati lineari” (Figura 35).

Gli stessi dati messi in relazione con il file degli Hexagon_cover, generano i dati visualizzati nel layer “TPL: esagoni” (Figura 36).

Gli stessi dati sono utilizzati anche per il grafico “TPL: consumi e statistiche” (Figura 43).

La Struttura della tabella è la seguente:

route_type, route_id, trip_id, from_stop_id, from_stop_name, from_lat, from_lon, from_ctime, to_stop_id, to_stop_name, to_lat, to_lon, to_ctime, progressive_edge, timeinsec, dist_metri, dist_metri_shape, geom, fromstop_id_elevation, tostop_id_elevation, Category, Fuel, Segment, Euro Standard, Technology, Consumption [MJ], Consumption [Units], Units, CO2 [g], WTT Energy [MJ], WTT CO2 [g]

7.1.3.2 Stops.txt (GTFS data):

Questo file di testo contiene le informazioni in formato GTFS delle fermate effettuate dal TPL nella provincia di Roma. Il file comprende anche le stazioni ferroviarie e metropolitane.

La Struttura del file è:

stop_id, stop_code, stop_name, stop_desc, stop_lat, stop_lon, zone_id, stop_url, location_type, parent_station, stop_timezone, wheelchair_boarding

7.1.3.3 Hexagon_cover (file GEOJSON [26]):

Il file descrive la zonizzazione dell'area della provincia di Roma sulla base di esagoni regolari.

La struttura del file è di seguito riportata:

```
{ "type": "FeatureCollection",
```



```

"features": [
  {
    "geometry": {
      "type": "Polygon",
      "coordinates": [
        [
          [
            Lon,
            Lat
          ],
          .....
        ]
      ]
    }
  },
  .....

```

7.1.3.4 Esa_attratti (file JSON [21]):

Questi dati, messi in relazione al Hexagon_cover, generano i dati visualizzati nel layer “FCD: spostamenti attratti” (Figura 37).

La Struttura file è la seguente:

```
{[idesagono]:{[datetime]:[numero veicoli]},,.....}
```

7.1.3.5 Esa_generati (file JSON):

Questi dati, messi in relazione al Hexagon_cover, generano i dati visualizzati nel layer “FCD: spostamenti generati” (Figura 38).

La Struttura file è:

```
{[idesagono]:{[datetime]:[numero veicoli]},,.....}
```

7.1.3.6 Esa_parked (file JSON):

Questi dati, messi in relazione al Hexagon_cover, generano i dati visualizzati nel layer “FCD: auto in sosta” (Figura 39).

La Struttura file è:

```
{[idesagono]:{[datetime]:[numero veicoli]},,,....}
```

7.1.3.7 Hexagon FCD (file CSV):

Questi dati, messi in relazione al Hexagon_cover, generano i dati visualizzati nei layer “FCD: auto residenti” (Figura 40) e “FCD: popolazione residenti” (Figura 41).

La Struttura del file è:

idcella, popolazione, n_fcd

7.1.3.8 Matched_routes (file GEOJSON):

Questi dati costituiscono la sorgente del layer “FCD rappresentazione del grafo sulla base dei flussi giornalieri” (Figura 42).

La Struttura del file è:

```
{ "type": "Feature", "properties": { "flow": [numero di passaggi ],  
"geometry": { "type": "LineString", "coordinates": [ [ Lon, Lat ], [ Lon ,Lat],..... ] } },.....
```

7.1.3.9 Consumi_ore_giorno_mese (file CSV):

Questi dati rappresentano la sorgente del grafico “FCD: Statistiche di traffico” (Figura 45).

La Struttura del file è:

giorno, ora, tripdistance_km, triptime_h, fuel, mj, consumption_units, units, second_cons_units2, units2, co2_g, wtt_mj, wtt_co2_g

7.1.3.10 CO2 (file CSV):

Questi dati costituiscono la sorgente del grafico “FCD: emissioni di CO2” (Figura 46).

La Struttura del file è:

giorno, ttw_co2_tons, wtt_co2_tons

8 Conclusioni

Durante la seconda annualità è stato sviluppato il nuovo ambiente software mantenendo la tipica architettura Client-Server per essere accessibile via web da qualsiasi postazione collegata ad Internet senza la necessità di dover installare alcun software dedicato.

Sono state sviluppate le procedure software per il processamento e l'analisi di grandi quantità di tracce digitali di viaggi automobilisti rese disponibili dalle moderne tecniche FCD che consentono di ottenere informazioni più affidabili sui viaggi in termini di lunghezza dei percorsi e tempi di percorrenza, e di valutare i flussi origine-destinazione.

Sono state implementate le procedure di elaborazione ed integrazione degli open data in formato GTFS che descrivono percorsi, fermate e orari dei servizi di trasporto pubblico locale.

Sono state infine implementate le procedure di calcolo dei consumi energetici disaggregati per vettore energetico e modo di trasporto e di stimare le emissioni di gas serra considerando l'intero ciclo di vita dei vettori energetici utilizzati.

9 Riferimenti bibliografici

1. C. Liberto e G. Valenti, «Simulatore di scenari urbani di mobilità veicolare elettrica: specifiche interfaccia utente, verifica del sistema integrato di modelli e valutazione di scenari di elettrificazione della mobilità veicolare», Rapporto RdS/PAR2017/241, 2018
2. <https://www.octotelematics.com/it/home-it/>
3. <http://www.vemsolutions.it>
4. F. Karagulian, G. Messina, G. Valenti, C. Liberto e F. Carapellucci, «A Simplified map-matching algorithm for Floating Car Data», Advanced Information Networking and Applications, vol. 3, p. 227, 2021
5. E. Dijkstra, «A note on two problems in connexion with graphs» Numerische Mathematik, 1959
6. <https://developers.google.com/transit/gtfs>
7. <https://transitfeeds.com/p/roma-servizi-per-la-mobilita/542>
8. <https://www.sinanet.isprambiente.it/it>
9. http://resources.arcgis.com/en/help/main/10.1/index.html#/Esri_ASCII_raster_format/009t000000z000000/
10. <https://www.emisia.com/utilities/copert/documentation/>
11. P. Boulter, P. McCrae, R. Jourmard, M. André, M. Keller, P. Sturm e S. Hausberger, «ARTEMIS: Assessment and reliability of transport emission models and inventory systems - Final Report, Deliverable 15» TRL, Crowthorne, UK, 2007
12. INFRAS, «The Handbook Emission Factors for Road Transport (HBEFA)», 2004
13. JEC, «Well-To-Tank Report version 4.a,» 2014. [Online]
14. JEC, «Well-To-Tank Report version 4.a, appendix 2» 2014
(https://ec.europa.eu/jrc/sites/jrcsh/files/wtt_appendix_2_v4a.pdf)
15. Unione Petrolifera, «Relazione Annuale 2019» 2019
(<https://www.unione petrolifera.it/download/relazione-annuale-2019/?wpdmdl=11482&refresh=5e31892ceb5261580304684>)
16. Istituto Superiore per la Protezione dell'Ambiente, «National Inventory Report - Year 2018» 2018
(<http://www.sinanet.isprambiente.it/it/sia-ispra/serie-storiche-emissioni/national-greenhouse-gas-inventory-system-in-italy/view>)
17. Ministero dello Sviluppo Economico, «Bilancio Energetico Nazionale, dati 2018» 2018
(<https://dgsaie.mise.gov.it/ben.php>)
18. TERNA, «Dati Generali 2017-2018» 2018 (https://download.terna.it/terna/1-Sez_DATI%20GENERALI_8d7304e358d68bd.pdf)

19. ISPRA, «Serie storiche emissioni: fattori di emissione per la produzione ed il consumo di energia elettrica in Italia» 2020 (<http://www.sinanet.isprambiente.it/it/sia-ispra/serie-storiche-emissioni/fattori-di-emissione-per-la-produzione-ed-il-consumo-di-energia-elettrica-in-italia/view>)
20. ACI, «Autoritratto 2019» 2019 (<http://www.aci.it/laci/studi-e-ricerche/dati-e-statistiche/autoritratto/autoritratto-2019.html>)
21. «JSON (JavaScript Object Notation): semplice formato per lo scambio di dati» (<https://www.json.org/>)
22. «Python: linguaggio di programmazione» (<https://www.python.org/>)
23. «PHP: linguaggio di programmazione» (<https://www.php.net>)
24. «D3.js: libreria JAVAScript» (<https://d3js.org/>)
25. «Jquery: libreria JAVAScript» (<https://jquery.com/>)
26. «GEOJSON: semplice formato per dati geografici» (<https://GEOJSON.org/>)
27. IOGP, «Report 373-07-02, Geomatics Guidance Note7, part2. Coordinate Conversions & Trasformations including Formulas», Marzo 2019 (<http://www.epsg.org/Portals/0/373-07-2.pdf>)
28. G. Boaga, «Trattato di geodesia e topografia con elementi di fotogrammetria», Padova, CEDAM, 1948

10 Abbreviazioni ed acronimi

Tabella 9: nomi estesi delle abbreviazioni

Sigla	Significato esteso
ASSTRA	Associazione Trasporti
CNG	Compressed Natural Gas
CORINAIR	CORe Inventory AIR emissions
COST	COoperation in the field of Scientific and Technical Research
CSV	Comma-Separated Values
DEM	Digital Elevation Model
DTM	Digital Terrain Model
EEA	European Environment Agency
EGR	exhaust gas recirculation
EMEP	European Monitoring and Evaluation Programme
FCD	Floating Car Data
GHG	Green House Gas
GPL	Gas di Petrolio Liquefatto
GPRS	General Packet Radio Service
GPS	Global Position System
GSM	Sistema globale per comunicazioni mobili (Groupe Spécial Mobile)
GTFS	General Transit Feed Specification
HBEFA	HandBook Emission Factors for road transport
IPCC	Intergovernmental Panel on Climate Change
LNG	Liquefied Natural Gas
MiSE	Ministero dello Sviluppo Economico
OBU	On Board Unit
OSM	Open Street Map
SCR	Selective Catalytic Reduction
SUV	Sport Utility Vehicle
TPL	Trasporto Pubblico Locale
TTW	Tank To Wheel
UTM	Universale Trasversa di Mercatone
WGS84	World Geodetic System 1984
WTT	Well To Tank