A Data Warehouse System for Estimating Fuel Consumption from Multiple Traffic Data Sources

Rune Bak Jacobsen
rbja16@student.aau.dk
Aalborg Universitet

Søren Andreas Abildskov Hansen
saah16@student.aau.dk
Aalborg Universitet

***Abstract*—Given the need to reduce greenhouse gas emissions throughout society, methods are needed to estimate current emissions reliably so that the effects of initiatives towards reducing emissions can be assessed. This paper focuses on traffic and how, given traffic data, we can provide insight into the total emissions of the municipality. The data sources are Induction Loop Derived Data and Floating Car Derived Data. A data warehouse design is proposed, and the paper elaborates on the benefits of the design. A fuel estimation model uses the processed traffic data to estimate fuel consumption and CO_2 emissions.**

Furthermore, the model uses custom vehicle sets to reflect the real-life traffic of a municipality. The paper offers insight into the strengths and weaknesses of the approach and explores alternative methods for modifying and improving the approach. The system provides estimations for fuel consumption and CO_2 from spatial and temporal queries through an interface.

1. Introduction

A. Motivation

On December 12, 2015, 196 parties to the UN adopted The Paris Agreement [1]. The agreement's goal is to limit greenhouse gas emissions to slow global warming. Subsequently, Danish municipalities created the "DK2020" [2] plan, which represents the commitment to reaching the goals defined in The Paris Agreement. Currently, 66 out of 98 Danish municipalities have joined the plan. The plan centers around a collaborative effort where the municipalities can share experiences and solutions to create a framework to prevent climate change. With the growing commitment to reduce greenhouse gas emissions, it is necessary to create new solutions that support this development.

In Europe, out of the total CO_2 emissions, nearly 30% comes from traffic, of which 72% is from road traffic. Thus, a large part of greenhouse emissions comes from traffic. This paper studies how to provide fuel consumption and CO_2 emission estimates given two different sources of traffic data. Thus, the solution gives insight into the current situation and plan for future scenarios regarding fuel consumption and CO_2 emission estimates.

B. Contributions

This paper introduces a system to compute fuel estimations of vehicles traveling in a road network from two traffic data sources. As fuel and CO_2 emissions are proportional to each other, the paper focuses on fuel estimations. The traffic data sources are Induction Loop Derived Data (ILDD) and Floating Car Derived Data (FCDD). The data in the ILDD is obtained from stationary measurement stations, consisting of data from **points**, which are a set of latitude and longitude coordinates. The data in the FCDD is obtained from GPS devices in vehicles given from **road segments** as a stretch of road.

Additionally, we use OpenStreetMaps (OSM) as a common representation. Separated, these data sources cannot represent the density of the traffic in a traffic network. However, the strengths of the data sources cancel out the weaknesses of the opposite source. Thus, we integrate the data sources into a common representation. However, as the data sources have different spatial and temporal representations, the integration is not straightforward. It includes, among others, map matching of ILDD and FCDD to OSM road segments. Section 6 describes the entire data integration process. The FCDD and ILDD register the Annual Average Daily Traffic [3] (AADT), which is the yearly number of vehicles on a given road. Throughout this paper, the AADT will be referred to as *hits* for a given road segment. The FCDD and ILDD also register the average speed.

However, the FCDD does not register the total density of traffic on its road segments as it only registers a subset of all vehicles. This is due to the data being vehicle dependent and not location dependent. Likewise, ILDD does not capture the entire traffic flow since it registers data from stationary stations located at specific points. Unfortunately, not every road segment has a measurement station, as the deployment maintenance of such stations is costly. Thus, stations register complete data, but their coverage of a road network is sparse. In order to extrapolate the hits from the FCDD to the complete hits number given from the ILDD, we use linear regression. The approach creates a model which can predict the ILDD hits value given an FCDD hits value. Additionally, a model to predict the ILDD average speed given the FCDD average speed us also used. When the data integration of the sources is complete, we utilize a fuel estimation model to estimate fuel consumption on road segments. The fuel estimation model takes different vehicle parameters as input, capturing the variations among vehicles and affecting fuel consumption. As an example, a small car usually uses less fuel than a large car, and a truck uses more fuel than a large car. For that reason, the system supports creating custom vehicle types. This works by specifying the parameters of the given vehicle to be used in the fuel estimation model. The vehicle parameters include, e.g., the weight of the vehicle and the idle fuel consumption.

Furthermore, one can specify the vehicle distribution of

trucks, buses, and cars. This allows analyzing different scenarios such as traffic with or without electric buses. Thus, it is possible to see how fuel usage will decline as electric vehicles increase. The results of the system are accessible in a web application. The application takes a temporal input in the form of a number of days and a time period. Thus, it is possible to specify the time period, e.g., between 15.00 and 16.00, down to a 15-minute range. The web application also contains a map from which the users can draw polygons to query on the road network. This usage enables, e.g., investigation of fuel usage in a city versus on motorways. Appendix A shows an example of the result page from the web application.

C. Overview

The paper first covers the related works in Section 2. Next, Section 3 describes the use cases of the system. Following this, Section 4 explains the system flow. Here we give an overview of the functionality of the entire system. Section 5 describes the used fuel estimation model SIDRA-RUNNING. We go through which parameters are necessary for the model to function and support vehicle sets. Section 6 explains the data sources and integration and covers the steps from raw data to a common representation. Section 7 explains the data warehouse architecture. Section 8 looks at linear regression models to predict accurate hit numbers and accurate average speed numbers. Section 9 reiterates our use cases and showcases the results. Following this, Section 10 will discuss the decisions throughout the paper and the processes leading to the final results. Finally, Section 11 will summarize and conclude the paper.

2. Related Work

EcoMark 2.0 [4] studies multiple fuel estimation models to compare their results to ground truth data measured from vehicles. The paper tests both instantaneous and aggregated fuel models, where instantaneous models use point data and aggregated models use road segment data. As for relevance to this paper, they concluded that the SIDRA-RUNNING model was suitable for assigning eco-weights to road segments, thus making the model viable for fuel consumption estimations. They also used the SIDRA-Avg model, but the model does not apply to all road categories, as it becomes unreliable when the average speed excels 50 km/h. For this paper, we use the fuel model SIDRA-RUNNING based on the results from *EcoMark 2.0*.

"Estimating traffic volume on Wyoming low volume roads using linear and logistic regression methods" [5] describes a method to try and estimate the AADT traffic density measurement on low-volume roads in Wyoming using linear and logistic regression. The intended usage of the method was for low-volume rural roads. The study used linear and logistic regression to try and give an estimate based on several predictor variables, which include, e.g., the surface of the roads and the population near the roads. They found that land usage near the roads was the most significant predictor variable. Their model scored an R^2 value varying from 0.44 up to the best score of 0.64. For this paper, we use a similar approach as we also will use a linear regression model. We take a different approach for this paper as we use a second AADT measurements source for the model instead of variables surrounding the roads.

In *"An Advanced Data Warehouse for Integrating Large Sets of GPS Data"* [6], they implemented data warehouses for GPS data using PostgreSQL [7] with the PostGIS [8] extension. For the ETL flow, they used a programmatic approach using pygrametl, which *"pygrametl - ETL programming in Python"* [9] and *"Programmatic ETL"* [10] documents. We will use PostgreSQL with the PostGIS extension for the data warehouse and pygrametl for the ETL process for this paper.

3. Use Cases

As described in Section 1-A, the paper aims to aid Danish municipalities in estimating traffic fuel usage. In order to get an understanding of the use cases, we cooperated with a Danish consultant company which specializes in traffic data solution for the Danish municipalities. The usage of the system is intended to be made available by said company to the Danish municipalities.

A. Yearly Estimate

One of the primary requests is to get a yearly estimate of the fuel consumed and CO_2 emitted in a user-specified geographical region. This is the primary feature, and thus the system needs to present this properly. The feature considers multiple traffic states with free-flowing traffic and spike hours, based on our data sources. Another aspect is to distinguish weekdays and weekends as the total traffic will be different. Thus, we must calculate the yearly total estimate accordingly.

B. Estimate Scaling

In addition to the yearly estimate, the yearly development of the total emissions is also relevant to investigate. As the number of vehicles increases in municipalities, the total fuel and CO_2 will also increase. Thus, a parameter to alter the total number of vehicles can provide additional information for future scenarios.

C. Vehicle Sets

Finally, an aspect that affects the total fuel used is the type of vehicles driving in the municipalities. If the vehicles are primarily gas and diesel vehicles, the total fuel used will be greater than a scenario where a subset of the vehicles are electric vehicles. The ability to model an increase in electric vehicles means seeing how electric vehicles impact the total fuel and CO_2 usage.

4. System Design & Flow

We created the system as a web application, from which all the functionality is available. To be specific, the functionality manages the traffic data upload, applies the fuel estimation model with different parameter sets, and presents the results. Figure 1 shows the flow of the system.

The center column consists of the three main pages, *Upload*, *Parameters*, and *Cases*, and the ETL flow. Each of

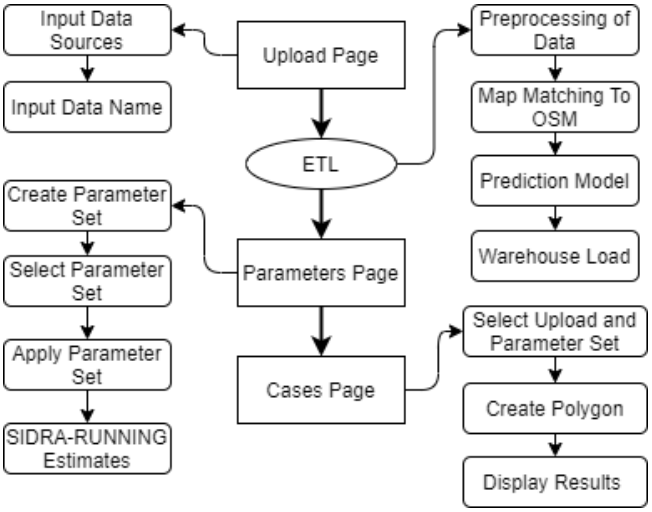

Fig. 1: System flow

them has their process flow mapped through the rounded rectangles following their small arrow. The system starts from the *Upload* page, where the users upload the FCDD and ILDD. The ETL flow then processes the data and integrates it into a common source. The ETL process is described further in Section 6 and Section 8.

When the ETL flow is complete, the user can, through the *Parameters* page, create custom parameter sets and apply them to the uploaded data. A parameter set contains a vehicle growth parameter which is a percentage indicator to scale the total number of vehicles from the uploaded traffic data. A parameter set also contains a vehicle set that represents the vehicles present in the road network. Furthermore, the user can specify vehicle sets, giving complete control of which vehicles to apply to the traffic data and the percentage each vehicle should represent. It is possible to apply many parameter sets to the same uploaded data, enabling comparison between multiple models of the data.

After applying the chosen parameter sets, the user can go to the *Cases* page to view the results. The *Cases* page allows the user to input an upload name, an applied parameter set, a time period, a number of days, and a polygon. Then, the system returns the fuel consumption for the area in the given time period. The user creates polygons using Leaflet[1], a library that allows the user to draw a polygon point by point. The user can then compare different model results from the uploaded traffic data. Appendix D shows the different functionality of the web application.

5. SIDRA

To provide fuel estimates based on the data sources, we use a model that utilizes the data features. As mentioned in Section 2, Eco-Mark-2.0 tested several fuel estimation models to determine how closely they compared to ground truth fuel data. One of the methods used was the SIDRA-RUNNING model, which showed promising results based on road segment data. Another option was the SIDRA-AVERAGE model, which also uses segment data. However, the original paper [11] declared the model unfit to estimate fuel when the average speed exceeds 50 km/h. Furthermore, since the traffic data concerns both rural and urban driving, the model is not viable. Based on SIDRA-RUNNING's features and given the inapplicability of SIDRA-AVERAGE, we use SIDRA-RUNNING as the fuel estimation model.

A. Model Description

The SIDRA-RUNNING model uses variables regarding a road segment and a vehicle to estimate the fuel required to traverse that given road segment with the given vehicle. For a road segment, the model uses the variables listed below from the input data. However, the model uses additional variables not available from the data. Thus, they are estimated based on the calculations from the original paper [11]. To summarize, the variables we use from our data sets are the following two.

- Average speed
- Total segment distance

In addition to the driving behavior variables, the SIDRA-RUNNING model also uses vehicle-specific parameters, as there is a difference between the fuel usage among vehicles, e.g., a heavy truck uses more fuel than a small car. In order to accurately provide fuel estimations that also account for vehicle diversity, we are taking two approaches. First, we use the vehicle set defined by Akçelik et al. in [12]. This set contains three car types, one bus type and one truck type, which were created and calibrated by the authors of the SIDRA models. Additionally, we include an electric vehicle type with a fuel usage of 0 to take into account the growing percentage of electric vehicles. Second, we support custom vehicle creation by the users of our system to accommodate for the evolution of vehicles, e.g., vehicles driving farther using less fuel. The result of the model is an estimate of the fuel consumed on a road segment given in mL. To get the CO_2 emitted, the fuel estimate is multiplied by 2.29 for gas or 2.66 for diesel [13].

The SIDRA-RUNNING model is given as:

$$
\begin{aligned}
F_s &= \begin{cases} \alpha + t_i + f_r + x_s & for\ 0 < f_r + x_s \\ \alpha + t_i & otherwise \end{cases} \\
f_r &= f_i/v_r + A + Bv_r^2 + Q_1 + Q_2 + 0.0981 K_G \beta_1 MG \\
Q_1 &= K_{E1}\beta_1 M E_{k+} \\
Q_2 &= K_{E2}\beta_2 M E_{k+}^2 \\
K_{E1} &= \begin{cases} 0.675 - 1.22/v_r & for\ 0.5 < 0.675 - 1.22/v_r \\ 0.5 & otherwise \end{cases} \\
K_{E2} &= 2.78 + 0.0178 v_r \\
K_G &= \begin{cases} 1 - 1.33 E_{K+} & for\ G < 0 \\ 0.9 & otherwise \end{cases} \\
E_{K+} &= \begin{cases} 0.35 - 0.0025 v_r & for\ 0.15 < 0.35 - 0.0025 v_r \\ 0.15 & otherwise \end{cases}
\end{aligned}
\tag{1}
$$

[1]https://leafletjs.com

In the model, F_s is the fuel consumed in mL, f_r is the fuel consumption in mL/km, x_s is the segment length in km, v_r is the travel speed in km/h, t_s is the travel time in seconds, t_i is the idle time in seconds, E_{K+} is the sum of positive kinetic energy changes per unit mass per unit distance, and G is the road grade. The parameters f_i, α, A, B, β_1, β_2, and M are vehicle parameters, and lastly K_{E1}, K_{E2}, and K_G are calibration parameters. The complete model can be found in the literature [11].

B. Vehicle Sets

When using SIDRA-RUNNING, many of the variables are vehicle specific, and thus the result becomes vehicle specific. As there is a wide variety in the type of vehicles in everyday traffic, the system must account for this. To accomplish this, it is possible to create custom vehicles with the parameters specified. This allows for custom vehicle sets creation consisting of a set of vehicles and the percentage each vehicle makes of the full set. Currently, the vehicles separate into three categories; cars, buses, and trucks. The reason is to be able to distinguish different vehicle types from each other. It is also possible to create electric vehicle types to reflect the number of electric vehicles currently driving, which do not consume fuel such as petrol or diesel. For this paper, we do not consider the electric energy for electric cars in terms of CO_2 output.

6. Data

A. Overview

As mentioned in Section 1-B, the paper uses the traffic data sources FCDD and ILDD and uses a map representation from OSM. Since FCDD uses a proprietary map and ILDD only provides coordinates of its stations, we use OSM's road network to connect them.

The traffic information from FCDD and ILDD are contrasting. The FCDD consists of traffic data for many road segments, which, when combined, creates a road network. In this regard, the data is complete. However, the registered traffic data in FCDD is sparse, as it does not reflect every vehicle that has been in the road network. This is because the registrations are from, e.g., navigation systems. ILDD only provides the specific point for each measurement station. These are stationary and registers the total traffic at a given time in a specific place. The traffic data from ILDD is then complete as it registers all vehicles at a specific point. In regards to a road network, it is a sparse data source since there are only a limited number of stations. Appendix B gives an overview of the FCDD road segments and the ILDD points.

Both FCDD and ILDD support varying time granularity, with the finest being 15-minute intervals. The system proposed in this paper supports these temporal representations. This section describes how we integrate the finest time granularity as it requires additional processing. Furthermore, it enables the system to make more fine-grained temporal queries.

The data used for this paper is from February 2019 to February 2020 for Odense municipality. The time granularity is 15-minute intervals. The ILDD is composed of 85 measurement stations, and the FCDD road network is composed of 58.484 road segments.

B. OpenStreetMaps

To combine FCDD and ILDD to a common representation, we use OpenStreetMaps' road network (OSM). The road network consists of a number of road segments. The road segments can vary in length and structure. Figure 2 shows an example with five segments where a single blue line with a black border represents a road segment.

Fig. 2: OpenStreetMaps road segments

The road segments in OSM also have associated metadata. Table 1 shows an example of the information used from OSM.

Id	Name	Highway	Max Speed
121390376	Skovalléen	tertiary	50

Table 1: Example of OSM road segment

C. Floating Car Derived Data

As mentioned, the FCDD provides traffic data assigned to road segments. Since there are different ways to model road networks, the road segments can vary in length. This is the case with the FCDD road network compared to the OSM road network. To illustrate, Figure 3 shows the same street with road segments from the FCDD. The road segments have a black border to indicate where they begin and end. As mentioned, there are a total of 58.484 road segments for the FCDD road network.

Table 2 shows an example of the traffic data associated with a road segment. Hits is the number of vehicles registered for the road segment in the time period, AvgSp is the average speed in km/h of the vehicles, and AvgTt is the average travel time in seconds.

id	length	frc	speedlimit	streetname	AvgSp	AvgTt	Hits
13203	24,07	6	35	Skovalléen	33,55	3,82	201

Table 2: Example of FCDD row, weekday during 11.45–12.00

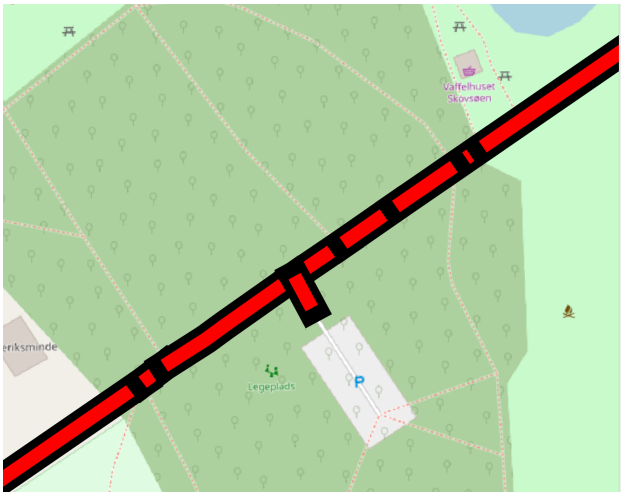

Fig. 3: FCDD Segments Example

This paper uses the data in 15-minute intervals, e.g., from 11.45–12.00, and in two periods. The first period is Tuesdays, Wednesdays, and Thursdays to represent a weekday, and the second period is Saturdays to represent a weekend day. Notice that the FCDD does not have a date specified with it. This is due to the data is aggregated. This means that the vehicles registered in a time period are for the entire year. Looking at Table 2 again, the 201 hits registered are from every Tuesday, Wednesday, and Thursday between 11.45 and 12.00 for the entire year.

D. Induction Loop Derived Data

The ILDD is point based and registers every vehicle that passes a point with a measurement station on the road. To illustrate this, Figure 4 shows a red dot which the location of a single ILDD measurement station.

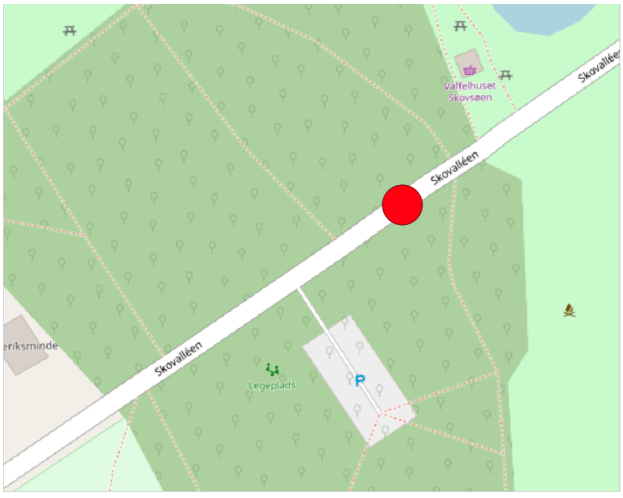

Fig. 4: ILDD Example

In the data used for this paper, there is a total of 85 measurement stations. These measurement stations register traffic information from their specific location, such as the total number of vehicles and their average speed. Table 3 shows an example of this.

Road Id	Place Id	Road Name	Date	Direction	Count	Speed
...	...	Skovalléen	01.02.2019	-	57	34,1

Table 3: Example of ILDD row, 01.02.2019 during 11.45–12.00

Contrary to FCDD, ILDD contains dates, which means there are rows for every day of the year, with 15-minute intervals associated with it.

E. Data Integration

Both FCDD and ILDD have their strengths and weaknesses. The goal is to integrate the two sources into a single solid data source. This integration includes the date representation in ILDD and the lack of it in FCDD. Also, the integration corrects the spatial representations to a common representation.

The data integration integrates FCDD, ILDD, and OSM into a unified format. Furthermore, the resulting rows capture the combined data of the road segments. The process is three steps. One step for the ILDD, one step for the FCDD, and the last step is merging the two.

1) ILDD Transformation: Each measurement station registers independent data from the other. For that reason, we perform the transformation for every station. Since we want to integrate the ILDD and the FCDD, we group the individual dates to reach the same aggregation. This works by setting the date as a weekday or a weekend day. Following that, it is possible to group the measurement station based on if it is a weekday or not.

Example: Table 4 shows four rows from the measurement station on *Skovalléen* during 11.45–12.00. Depending on the date of the measurement, a weekday flag determines whether it is a weekday or weekend day. We omit the coordinates of the measurement stations for brevity.

Road Name	Date	Direction	Count	Speed	Weekday
Skovalléen	01.02.2019	-	57	34,1	true
Skovalléen	02.02.2019	-	84	35.4	false
Skovalléen	03.02.2019	-	60	34,1	false
Skovalléen	04.02.2019	-	37	37.4	true

Table 4: ILDD station on *Skovalléen* during 11.45–12.00

As the data for each station spans multiple days, we group it to fit a generic weekday and weekend day. We do this by taking the average of the hits and the average of speed from the days.

Example: Table 5 shows two rows from the same measurement station as before. These two rows are found by taking rows 1 and 4 from Table 4 and averaging their speed and count. Row 2 is found by taking rows 2 and 3, the weekend days, and averaging their speed and count.

This results in a single row, for a specific interval, for each station. Furthermore, the aggregation for speed and count is now comparable with FCDD, as they both have the same time format.

Road Name	Direction	Count	Speed	Weekday
Skovalléen	-	72	35,75	true
Skovalléen	-	47	34,75	false

Table 5: ILDD station *Skovalléen* during 11.45–12.00 with grouped values

Following this, the stations need to be map matched on to OSM. The map matching process is naive by matching the stations to their nearest road segment. As there were no falsely mapped points in our data source, the method is valid. This process results in the road segment now containing a corresponding OSM id.

Example: Table 6 shows the rows after the matching process.

Road Name	Direction	Count	Speed	Weekday	OSM id
Skovalléen	-	72	35,75	true	121390376
Skovalléen	-	47	34,75	false	121390376

Table 6: ILDD station *Skovalléen* during 11.45–12.00 with grouped values and OSM id.

2) FCDD Transformation: The transformation of FCDD focuses on moving the FCDD road segments to OSM road segments. We do the following process for every FCDD road segment provided from the data.

As the FCDD road segments differ from OSM road segments, a naive map matching process is inaccurate as it resulted in too many falsely matched cases. Instead, we use Project OSRM2[2]. This provides a routing API for OSM with a matching service that matches GPS points to OSM road segments. After the map matching, each FCDD road segment has a corresponding OSM id. As multiple FCDD road segments can have the same OSM id, we map the data together.

Example: Table 7 shows many FCDD road segments which map to a single OSM road segment. The column OSMid represents the OSM road segment. We omit the Streetname, speedlimit, and length for brevity.

id	AvgTt	AvgSp	Hits	OSM id
3297	3.87	34.63	425	121390376
8554	10.98	33.41	431	121390376
8555	1.24	34.93	440	121390376
8556	7.08	35.18	440	121390376
8557	2.37	34.89	433	121390376
8559	2.61	33.73	435	121390376
8560	2.97	32.88	437	121390376
19942	0.97	34.69	443	121390376
19943	3.11	34.72	442	121390376
25463	6.61	37.10	431	121390376
25464	0.96	37.76	435	121390376
25465	6.10	37.07	436	121390376
25466	2.35	35.92	429	121390376
25467	1.38	32.75	1	121390376
25468	2.45	35.12	428	121390376

Table 7: All FCDD rows for weekday during 11.45–12.00

To integrate this, we aggregate all segments with identical OSM ids to a single row. This is done by taking the average of the average speed values and using the max value of the hits values. We disregard the travel time as it can be calculated using the length of the OSM road segment and the average speed. Table 8 shows the result.

As mentioned in Section 6-C, the hits are a sum from the entire year. This value needs to be an average for a weekday to integrate with ILDD. Thus we divide it by the number of days from the data. It is, however, a more complicated process as, e.g., the FCDD omits vacation days. In our case, the number of Tuesdays, Wednesdays, and Thursdays sum to 120. Thus we need to divide the hits by 120. The average hits are shown in the column named average hits.

AvgSp	Hits	Average Hits	OSM id
34.985	443	3,6	121390376

Table 8: Single FCDD row created by OSM id for weekday during 11.45–12.00

3) Result: With ILDD and FCDD matched to OSM and ILDD aggregated to the identical date format as FCDD, merging the two data sources is now possible. The merging process is done based on the OSM id, whether it is a weekday and the corresponding interval. This process results in rows with both FCDD and ILDD but also yields rows where there only is FCDD. The total output is a collection of 9.164 road segments with 14 different categories describing the road.

Example: Table 9 shows the data resulting from merging the FCDD and the ILDD.

ildd_hits	ildd_speed	fcdd_speed	fcdd_hits	OSM id
44	36.091	34.985	3	121390376

Table 9: ILDD and FCDD data merged for weekday 11.45–12.00

7. Data Warehouse Design

In order to store and analyze the data, we use a data warehouse. For the relational database representation, we use a star schema. This representation has a fact table that holds a row for each fact. The star schema has at most one dimension table for each dimension. The fact table has a column for each measure. It also has one column for each dimension table that contains a foreign key value that references the primary key values in a particular dimension table. We create the design based on the principles from [14], and the complete design is in Appendix C. We use Kimball's [15, 16] multidimensional modeling process. This process has four subprocesses:

1) Choose the process(es) to model.
2) Choose the granularity of the process.
3) Design the dimensions.
4) Choose the measures.

A. Process Modeling

For step 1, there are two processes to model, the process of driving and the process of fuel usage during driving. For step 2, we need a data granularity that best matches the analysis needs. The granularity of the driving process becomes every road segment, for every time period, with the number of

[2]http://project-osrm.org/

id	dim_time	dim_road	dim_junk	fcdd_hits	fcdd_speed	ildd_hits	ildd_speed	predicted_hits	predicted_speed
1275247	355	23560	3	76	112.6887	473	110.8142	TBD	TBD
1235418	348	22372	3	6	38.7025	79	44.4272	TBD	TBD

Table 10: Example of two rows from the fact tabel `fact_osm`

id	length	speed_limit	name	osm_id	category	direction	ildd_id	ildd_point
22328	191.3711	50	Højsagervej	372439609	tertiary	+	461 420506-0 4/ 340	...
22452	3012.0817	110	Fynske Motorvej	414187457	motorway	-	0 40-0 159/ 400	...

Table 11: Example of two rows from the road dimension `dim_road`

vehicles recorded and their average speed. The granularity of the fuel usage of driving becomes every road segment, for every time period, with the number of vehicles, their average speed, and the estimated fuel usage. Section 7-B and Section 7-C covers the design of dimensions and the selection of measures.

B. Road Segment Fact

The process of driving is defined by the input traffic data, consisting of ILDD and FCDD. The characteristics of this process are that a given road segment in a given time period has an average speed and a number of hits.

In the schema, the fact table `fact_osm` stores this information. Table 10 shows an example of the fact table rows. It shows the fact table `fact_osm` with the columns `dim_time`, `dim_road`, and `dim_junk` that hold foreign keys to their respective dimensions. Furthermore, the fact table has columns for the measures: `fcdd_hits`, `fcdd_speed`, `ildd_hits`, `ildd_speed`, `predicted_hits` and `predicted_speed`.

The time dimension, `dim_time`, holds the time information about a specific fact row. Table 12 shows an example of rows stored in the time dimension. As mentioned in Section 6, the data used in this paper is separated into 15 minutes intervals and aggregated to a weekday or a weekend day. The time dimension stores information. To do so, the table has three columns, a `weekday` flag where true indicates it is a weekday and false indicating it is a weekend day. The attribute `timerange` stores the range of time, e.g., 15.00–15.15, and a primary key integer stored in `id`. This enables temporal queries through the time dimension. An example of this could be a query for all the data between 13:00–14:30. As the time dimension covers each 15-minute interval during the day and separates weekday and weekend day, the total row count for the dimension is 192.

id	weekday	timerange
62	false	[1515,1530)
158	true	[1515,1530)

Table 12: Example of two rows from the time dimension `dim_time`

To support spatial queries, the data about road segments put into the road dimension, `dim_road`. Table 11 shows an example of the rows stored in this dimension table. The dimension table has nine columns. The attributes `name`, `speed_limit`, `length` are self-explanatory. The attribute `osm_id` record the road segment's id in OSM from the matching process described in Section 6. The attribute `ildd_point` records the geometry of the ILDD measurement station, and `ildd_id` is its corresponding id. The attribute category is the road segments category from OSM, e.g., 'motorway'. The attribute `direction` is the direction the ILDD measurement was, and the `id` column stores the primary key for the table.

The last dimension which `fact_osm` uses is the junk dimension, `dim_junk`. The information contained in this dimension is the information needed in the schema but did not have enough precedence to have an entire dimension for itself or did not fit into any of the existing dimensions. This schema contains a `load_number`, a `load_name` for the data inserted, and a primary key `id`. Table 13 shows an example of the rows contained in this dimension.

id	load_number	load_name
1	1	First import
2	2	Second import

Table 13: Example of two rows from the junk dimension `dim_junk`

A star scheme can result in redundancy. This is this case for the time dimension in our schema. Here the weekday value will be the same for each interval of the day. This means we have 96 duplications per weekday entry since there are 96 intervals during a day. It can be a good idea to have this kind of redundancy in the dimensions tables as it can support more straightforward queries

C. Fuel Estimate Fact

The second process concerns the fuel consumption associated with driving. The fact table, `fact_sidra`, stores this information. This utilizes a mix of a star schema and a snowflake schema as one of its dimensions is a snowflake dimension. The dimension in question is the parameter dimension `dim_parameter`. A snowflake dimension contains several tables for the dimension. The dimension tables contain a key, columns holding textual descriptions of the values, and possibly columns for level properties. Tables for lower levels also contain a foreign key to the containing level. The dimension `dim_parameter` contains seven columns, the primary key `id`, the parameter set name `name`, a value to increase the hits as an indication for future traffic called `hit_increase`, the percentage of which each vehicle type amounts, and lastly, a foreign key to `vehicle_set`. The `vehicle_set` table has three columns, `id`, the percentage of which a given vehicle makes of the full set called `vec_percentage`, and lastly, a foreign key to `vehicle`. The `vehicle` table has eight

id	dim_time	dim_road	dim_parameter	dim_junk	car_fuel	truck_fuel	bus_fuel	hits	avg_speed
572495	492	28804	20	3	62.4881	272.5049	219.3438	37	36.4212
1086607	492	28606	20	3	30.7307	125.3208	102.0861	18	28.0105

Table 14: Example of two rows from the fact tabel `fact_sidra`

id	hit_increase_percentage	vehicle_set_id	name	car_percentage	truck_percentage	bus_percentage
15	0	15	Standard	75	15	10
16	0	16	Electric 15	75	15	10

Table 15: Example of two rows from the parameter dimension `dim_parameter`

columns, its primary key `id`, and the rest are parameter values for the SIDRA-RUNNING model Section 5.

The reason for the parameter dimension being a snowflake dimension table is that the system supports applying different vehicle types to the SIDRA-RUNNING model. The vehicle table stores the vehicles, which allows the users of the system to add custom vehicles. By storing the custom vehicles, it is possible to reuse them in different vehicle sets without inputting them every time. Table 15 shows an example of the data contained in the parameter dimension. Table 16 shows an example from the `vehicle_set` dimension, and Table 17 shows an example of the data in the vehicle dimension.

id	vehicle_id	percentage
15	3	35
15	4	25

Table 16: Example of two rows from the vehicle set table `vehicle_set`

id	category	name	fi	a	b	beta1	m
3	100	Small Car	891	9.6	0.00373	0.1012	1250
4	100	Medium Car	1504	19.39	0.00406	0.109	1810

Table 17: Example of two rows from the vehicle table `vehicle`

The fact table `fact_sidra` also uses the road dimension `dim_road`, the junk dimension `dim_junk`, and the time dimension `dim_time`. It has a five measures `car_fuel`, `truck_fuel`, `bus_fuel`, which stores the calculated fuel estimate for the specific road segment for cars, trucks, and buses, respectively, and `hits` and `avg_speed`, which enables faster querying. Table 14 contains an example of the rows in the fact table.

D. Extract-Transform-Load

As the data integration handles all extraction and transformation, the data is ready to be loaded into the warehouse. We use pygrametl [9, 10] to load it into the data warehouse, as the data integration is written in Python.

After the data loads into the database, the table size for `dim_road` is 1.048 kB consisting of 9.241 rows, for `dim_time`, the size is 72 kB consisting of 192 rows, for `dim_junk`, the size is 32kB consisting of 1 row, and for the fact table `fact_osm` the size is 125 MB consisting of 994.351 rows. The dimension `dim_parameter` table size is 32 kB for 3 rows, with `vehicle_set` table size being 24kB with 21 rows and `vehicle` table 32 kB with 6 rows. The `fact_sidra` size is after one parameter run 125 MB with 988.299 rows. For two parameter runs, the size scales accordingly to 250 MB with 1.976.598 rows

8. Prediction

When estimating fuel consumption, a necessary variable is the total number of vehicles on a road segment. This number fluctuates throughout the day, with mornings and afternoons having more traffic than the rest of the day. As described in Section 6, we have two sources with their number of vehicles registered on their road segments. As mentioned in Section 6, the hits in FCDD are incomplete as they are only counting a subset of the vehicles. Table 18 shows an example of the absolute difference in hits between the data sources. The numbers stem from all road segments, which have both ILDD and FCDD, which is 165. When also divided by time, we get 31.428 records of data for the models. Furthermore, we do not have access to the raw data from either source, so we can not clean the data ourselves. Thus, we assume that domain experts have processed the data, and we set the ILDD as our ground truth data.

FCDD HITS	ILDD HITS
138.534	1.026.361

Table 18: Total hits from data sources on common road segments

Another area of focus is the average speed of the vehicles driving on a given road segment. As both data sources estimate the average speed, we assume that the data registered in ILDD is correct and deem it ground truth.

This section describes the usage of linear regression models to estimate the attributes on a given road segment from the incomplete FCDD using the ILDD. We use the R^2 metric to determine the validity of the models. The models are created using the scikit-learn [17] library in Python.

A. Hits Model

For the hits model, we aim to get a reliable model that can predict FCDD hits to ILDD hits. The initial model is a flat linear model with only two variables: the FCDD hits and the ILDD hits. The model gave an R^2 score of **0.933**, which is excellent. We tried to improve the model by using a multi-linear regression model with additional variables such as speed and time. This model showed improvement but not to a significant degree as the R^2 value only increased to **0.94**. To simplify the model, we use the model created using only the FCDD hits to predict. However, this model showed an apparent flaw when predicting hits on road segments with very low

hits from the FCDD. When applied, the model would scale 0 hits from the FCDD to ~40 hits. This inflates the number of hits during the night as many road segments have zero or close to zero hits. To solve this, we decided to use two models to predict hits. One model using the hits from FCDD below 3, and another model using the hits from the FCDD being 3 or greater. The "low" model scored an R^2 value of **0.56**. For the numbers [0,1,2], the model returned [4.39, 14.36, 24.32], which are very close to the averages from the ILDD. To compare, we took the average hits count from the ILDD when the hits from the FCDD were [0,1,2]. The results showed the averages to be [4.31, 14.83, 24.09], which corresponds to the model's output. The "large" model decreased a bit in the R^2 score to **0.89**. However, we still consider it a valid model to predict roads and predict the hits when the FCDD hits are greater than 3.

B. Speed Model

The speed model also yielded excellent results. The R^2 value for this model was **0.931** when using only the FCDD speed to predict the ILDD speed. We attempted to further improve the model by using multi-linear regression as well. We added road segment length, speed limit, and category, as well as the time. There was an improvement to the model as R^2 value increased to **0.933**. The increase was too insignificant to apply the added dependency to the model.

C. Road Segments Without Data

The last aspect of getting accurate hits to the data is to estimate hits to road segments without ILDD or FCDD. From OSM, it is possible to get attributes about the road segments, e.g., their length, category, and speed limit. We attempted to apply a multi-linear regression model with these parameters to predict the ILDD hits on a road segment. This model only resulted in an R^2 score of **0.209**, which is too low.

The second attempt to enrich the road segments without data was to predict the average speed on the road segments. We used the attributes for each road segment from OSM and obtained a model that achieved an R^2 value of **0.551**. This value, although better, is also insufficient to predict values without too many inaccuracies.

To handle road segments without data, we thus take a different approach than a linear regression model. Instead, we use an average k-nearest neighbor (kNN) approach for the unknown road segments. We collect the k nearest neighbors' speeds and hits and take the average for a given road segment without data. We use this method based on the assumption of homophily in a road network, where road segments close in proximity will share similar attributes. By using this method, we also account for road category such that, e.g., city roads are not affected by nearby highways.

9. Results

In order to validate the system, we look at the use cases defined in Section 3. The cases center around three aspects; yearly estimates, estimate scaling, and vehicle sets. Three cases will demonstrate the yearly estimates. One with a full year and the whole day, one with a full year during 06:00–22:00, and one with a full year during 06:00–16:00.The estimates scaling will be shown by the same cases as the yearly estimates, but with a 5% vehicle growth. To show the implementation of vehicle sets, we have created a *Standard* set, *Standard Growth 5%* set, and an *Electric* set. Figure 5 shows the sets.

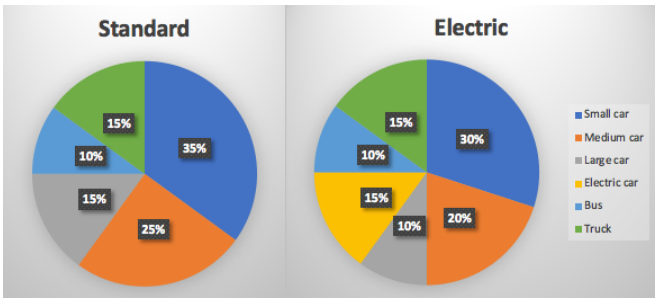

Fig. 5: Vehicle sets

Additionally, we wanted to enable users of the system to create spatial queries, to get results based on a specified geographical region. To demonstrate this, the results of the cases are recorded from two different geographical regions, as demonstrated in Figure 6 and Figure 7. The figures have the user input on the right and the returning road segments on the left. To summarize, the results will be an example of how to use the first version of the system and explores the core functionality.

Table 19 shows the results from the large polygon. The results of the queries are displayed in Table 20 and Table 19 and show the small and large polygon. The large polygon has a total of 5259 road segments, and the small polygon has a

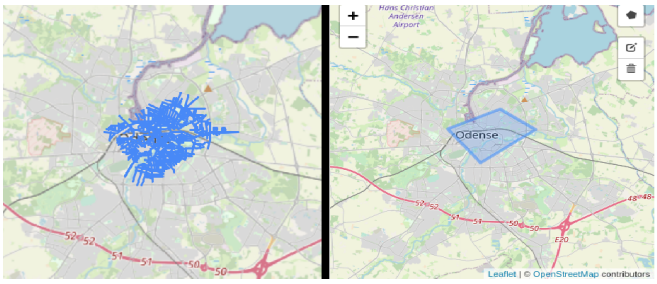

Fig. 6: Small polygons with results

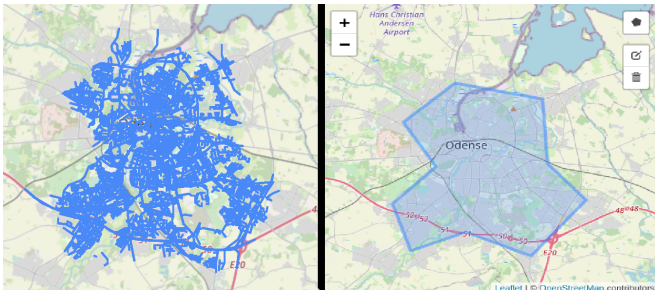

Fig. 7: Large polygon with results

Parameter Set	Total Fuel	Car Fuel	Bus Fuel	Truck Fuel	Road Segments	Time Frame
Standard	161.916K L	68.460K L	30.656K L	62.799K L	5259	00:00-23:45
Standard	150.010K L	63.424K L	28.404K L	58.182K L	5259	06:00-22:00
Standard	99.643K L	42.096K L	18.914K L	38.633K L	5259	06:00-16:00
Electric	147.471K L	54.015K L	30.656K L	62.799K L	5259	00:00-23:45
Electric	136.628K L	50.041K L	28.404K L	58.182K L	5259	06:00-22:00
Electric	90.758K L	33.211K L	38.633K L	18.914K L	5259	06:00-16:00
Standard Growth: 5%	169.426K L	71.638K L	32.074K L	65.714K L	5259	00:00-23:45
Standard Growth: 5%	157.000K L	66.381K L	29.723K L	60.895K L	5259	06:00-22:00
Standard Growth: 5%	104.302K L	44.065K L	19.796K L	40.441K L	5259	06:00-16:00

Table 19: Large polygon results

Parameter Set	Total Fuel	Car Fuel	Bus Fuel	Truck Fuel	Road Segments	Time Frame
Standard	16.731K L	7.024K L	3.371K L	6.336K L	940	00:00-23:45
Standard	14.809K L	6.215K L	2.985K L	5.609K L	940	06:00-22:00
Standard	9.606K L	4.036K L	1.936K L	3.634K L	940	06:00-16:00
Electric	15.234K L	5.527K L	3.371K L	6.336K L	940	06:00-16:00
Electric	13.485K L	4.891K L	2.985K L	5.609K L	940	06:00-22:00
Electric	8.746K L	3.176K L	1.936K L	3.634K L	940	00:00-23:45
Standard Growth: 5%	17.457K L	7.329K L	3.518K L	6.611K L	940	00:00-23:45
Standard Growth: 5%	15.453K L	6.486K L	3.115K L	5.852K L	940	06:00-22:00
Standard Growth: 5%	10.027K L	4.212K L	2.020K L	3.793K L	940	06:00-16:00

Table 20: Small polygon results

total of 940 road segments.

As shown in Table 19, the large polygon results show that the total fuel for full days with the standard set is 161.916K liters of fuel. It is worth noting that despite cars being 75% of the vehicle set and trucks only being 15%, the trucks are contributing to approximately 38% of the total fuel emitted. This could relate to the model, which applies every vehicle type defined in the parameter set to every road segment. In other words, the road segments located in residential areas have the truck percentage applied to them, which inflates the fuel from trucks.

Another point to note from the results is the improvement when removing nightly hours. When comparing the *Standard* parameter set with time periods 00:00–23:45 and 06:00–22:00, we see an 8% difference. This aligns well with the assumption that the majority of traffic is during the day.

The *Electric* parameter set also shows promising results, as the total for the cars for a full day is 54.015K liters of fuel. This total is approximately 22% less than the *Standard* set and can reflect how increasing the total amount of electric vehicles can affect the total fuel usage.

When comparing the *Standard* parameter set to the *Standard Growth 5%* parameter set, we see that the 5% increase result in a total fuel difference for the full day of approximately 5%. This enables scaling the different uploads to fit future growth. Table 20 shows the results for the small polygon and shows similar results for the center of the municipality.

10. Discussion

A. Data Integration

The data integration proved to be a difficult task. Both data sources had strengths, and our goal was to integrate these into a common source that incorporated those strengths.

One of the weaknesses of the FCDD was that it did not have a date attribute. If a date had been accessible, merging the data sets would have allowed us to enrich the data with a specific date. Furthermore, it would enable date-specific temporal queries, e.g., before a holiday.

Both resources provided traffic data, yet neither used the same representation. To fix this, we map matched both to OSM. For ILDD, this was simple: single points needed to be matched to road segments. FCDD was more challenging as we tried with the same approach as ILDD, but too many false positives were triggered. To get an accurate representation, we needed a map matching service. It would have been optimal if both data sources could use the naive map matching approach as the runtime of the map matching service is high. However, after the initial map match of the FCDD, we store the relations between them such that it is not needed to run again unless the users provide a new area. Thus, the initial load of a data set has a longer load time.

The result of the data integration proved very well. It resulted in the data sets being correlated. The prediction models confirm this as they were almost linear.

B. Traffic Direction

As FCDD provides road segments, which, when combined, form a road network with the driving directions from the road segments. The ILDD had problems here. Each point from the ILDD registers at a specific point and does so for both directions. This is, however, only documented with a + for one direction and - for the other. This data is not translatable to the common source, which means it is lost. Thus we use the total from ILDD and assign it the road segment. This means we need to take the total for FCDD to make it representative. This is not ideal, as we lose some information regarding traffic flow direction. However, since the system aims to estimate fuel for larger areas and not single roads, the damage is minimal. This is because the polygons created from the web application will rarely include both directions for a given road segment.

So as we do not lose any information about the number of vehicles on a given road segment, the damage is minimal.

C. Predictions

We decided to use a linear regression model to predict hits from FCDD to ILDD. The original model showed excellent results in terms of R^2 value, but the model's actual usage showed problems in predicting the nightly values of roads. We decided to use a *low* model and a *high* model to get more accurate predictions on the road segments with low values.

As we deemed this method acceptable, it could also be interesting to look at other ways to predict the data. As our data is mapped to a road network, prediction methods utilizing graphs could have been explored.

D. Vehicle Specific Road

As mentioned in Section 9, the trucks amounted to approximately 38% of the total fuel. One of the reasons for the high percentage is that trucks are present on every road segment for our system. For residential areas, trucks are not as common as they are for industrial districts or motorways. Vice versa, cars are probably more likely to drive on residential area roads than in industrial districts compared to trucks. Thus, a possible improvement would be assigning specific vehicle percentages to road types instead of the total number of road segments.

11. Conclusion

Throughout this paper, we have presented a fuel estimation system based on real data sources. These data sources were integrated to combine their strengths, and the result was a nearly linear correlation between them. The system supports temporal and spatial queries, full customization in vehicle parameters for the fuel estimation, and multiple vehicle types. Additionally, it is possible to specify a traffic growth percentage and create vehicle sets with electric vehicles. All of which stem from the use cases presented in Section 3.

To support the cases, we have used a data warehouse design structure. This proved beneficial as it made fuel estimation computation fast and straightforward. Additionally, it made it possible to store and compare different results. Furthermore, it made querying simple both in temporal and spatial regard.

Since neither data source captured every vehicle in the road network, we created a prediction model to improve the estimate of the number of vehicles in the road network. This linear regression model scored an R^2 of 0.89, which was excellent. The same method was applied for predicting the speed, which also gave an excellent R^2 score of 0.931.

Another strong point of the system is extendable of the components. The system relies on three central processes: the ETL, the predictions model, and the fuel estimation model. The prediction and fuel estimation models chosen for this system are possible to change if a better option becomes available. This ensures the lifespan of the system as outdated components are replaceable or updateable.

To summarize, we have developed a system to aid the municipalities of Denmark in estimating their fuel consumption. We have developed support for both larger areas and a more local focus by providing spatial query support. We have created a system that allows the municipalities to create many different cases based on the same data or upload different data sets and compare them on the same parameter sets.

Acknowledgement

We would like to thank Rambøll and their representatives Stig G. Søbjærg and Brian R. Jeppesen for their help and input to the project as well as the provided data. We would also like to thank Christian S. Jensen and Kristian Torp for their supervision of the project.

We take full responsibility for any shortcomings of the system.

References

[1] UN. *The Paris Agreement*. URL: https://unfccc.int/process-and-meetings/the-paris-agreement/the-paris-agreement. (accessed: 29.03.2021).

[2] Realdania. *DK2020*. URL: https://realdania.dk/projekter/dk2020. (accessed: 29.03.2021).

[3] Wikipedia contributors. *Annual average daily traffic — Wikipedia, The Free Encyclopedia*. [Online; accessed 2-June-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Annual_average_daily_traffic&oldid=1024109560.

[4] Chenjuan Guo, Bin Yang, Ove Andersen, Christian S. Jensen, and Kristian Torp. "EcoMark 2.0: empowering eco-routing with vehicular environmental models and actual vehicle fuel consumption data". English. In: *Geoinformatica* 19.3 (July 2015), pp. 567–599. ISSN: 1384-6175. DOI: 10.1007/s10707-014-0221-7.

[5] Dick Apronti, Khaled Ksaibati, Kenneth Gerow, and Jaime Jo Hepner. "Estimating traffic volume on Wyoming low volume roads using linear and logistic regression methods". In: *Journal of Traffic and Transportation Engineering (English Edition)* 3.6 (2016), pp. 493–506. ISSN: 2095-7564. DOI: https://doi.org/10.1016/j.jtte.2016.02.004. URL: https://www.sciencedirect.com/science/article/pii/S209575641530670X.

[6] Ove Andersen, Benjamin Bjerre Krogh, Christian Thomsen, and Kristian Torp. "An Advanced Data Warehouse for Integrating Large Sets of GPS Data". In: *Proceedings of the 17th International Workshop on Data Warehousing and OLAP*. DOLAP '14. Shanghai, China: Association for Computing Machinery, 2014, pp. 13–22. ISBN: 9781450309998. DOI: 10.1145/2666158.2666172. URL: https://doi.org/10.1145/2666158.2666172.

[7] PostgreSQL Global Development Group. *PostgreSQL*. en. Jan. 2021. URL: https://www.postgresql.org/ (visited on 01/04/2021).

[8] *PostGIS — Spatial and Geographic Objects for PostgreSQL*. URL: https://postgis.net/ (visited on 01/04/2021).

[9] *pygrametl - ETL programming in Python*. URL: https://chrthomsen.github.io/pygrametl/ (visited on 01/06/2021).

[10] Christian Thomsen, Ove Andersen, Søren Kejser Jensen, and Torben Bach Pedersen. "Programmatic ETL". In: *Business Intelligence and Big Data*. Ed. by Esteban Zimányi. Cham: Springer International Publishing, 2018, pp. 21–50. ISBN: 978-3-319-96655-7.

[11] D.P. Bowyer, Rahmi Akçelik, and D.C. Biggs. "Guide to fuel consumption analyses for urban traffic management". In: *ITE J.; (United States)* 56 (Nov. 1986), p. 31.

[12] Rahmi Akçelik, Robin Smit, and Mark Besley. "Recalibration of a vehicle power model for fuel and emission estimation and its effect on assessment of alternative intersection treatments". In: Apr. 2014.

[13] *Learn the facts: Fuel consumption and CO2*. URL: https://www.nrcan.gc.ca/sites/www.nrcan.gc.ca/files/oee/pdf/transportation/fuel-efficient-technologies/autosmart_factsheet_6_e.pdf. (accessed: 29.03.2021).

[14] Christian S. Jensen, Torben Bach Pedersen, and Christian Thomsen. *Multidimensional Databases and Data Warehousing*. English. Synthesis Lectures on Data Management 9. Morgan & Claypool Publishers, 2010. ISBN: 9781608455379. DOI: 10.2200/S00299ED1V01Y201009DTM009.

[15] Ralph Kimball and Margy Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. 2nd. USA: John Wiley & Sons, Inc., 2002. ISBN: 0471200247.

[16] Ralph Kimball, Margy Ross, Warren Thornthwaite, Joy Mundy, and Bob Becker. *The Data Warehouse Lifecycle Toolkit: Practical Techniques for Building Data Warehouse and Business Intelligence Systems*. 2nd. USA: John Wiley & Sons, Inc., 2008. ISBN: 9780470149775.

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

APPENDIX A
WEB APPLICATION

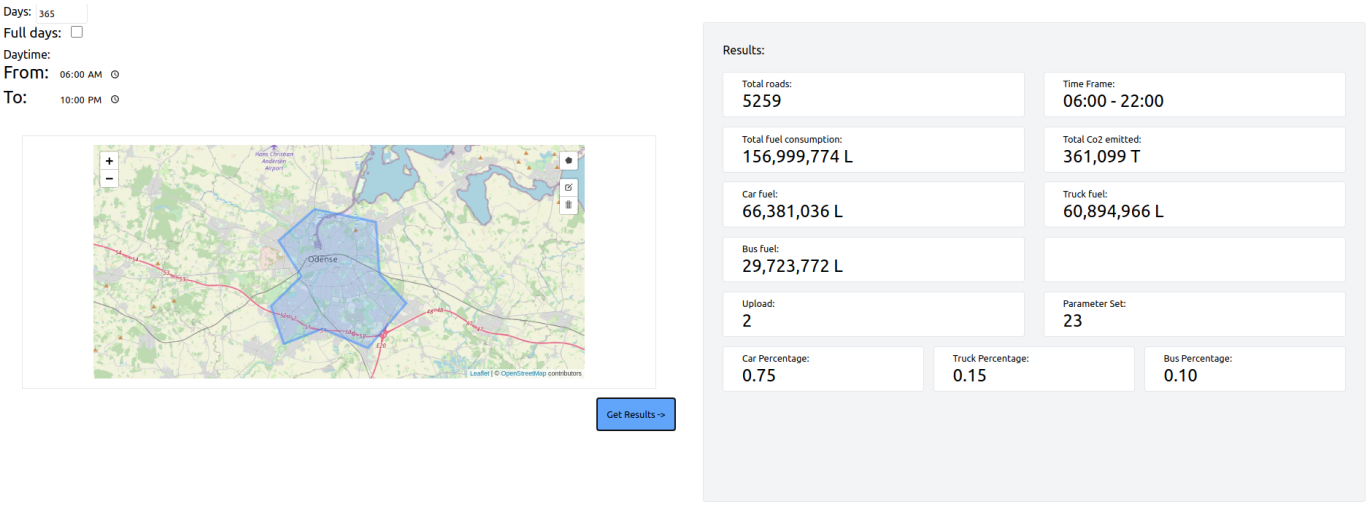

Fig. 8: Example of the result page of the system

APPENDIX B
ROAD NETWORK

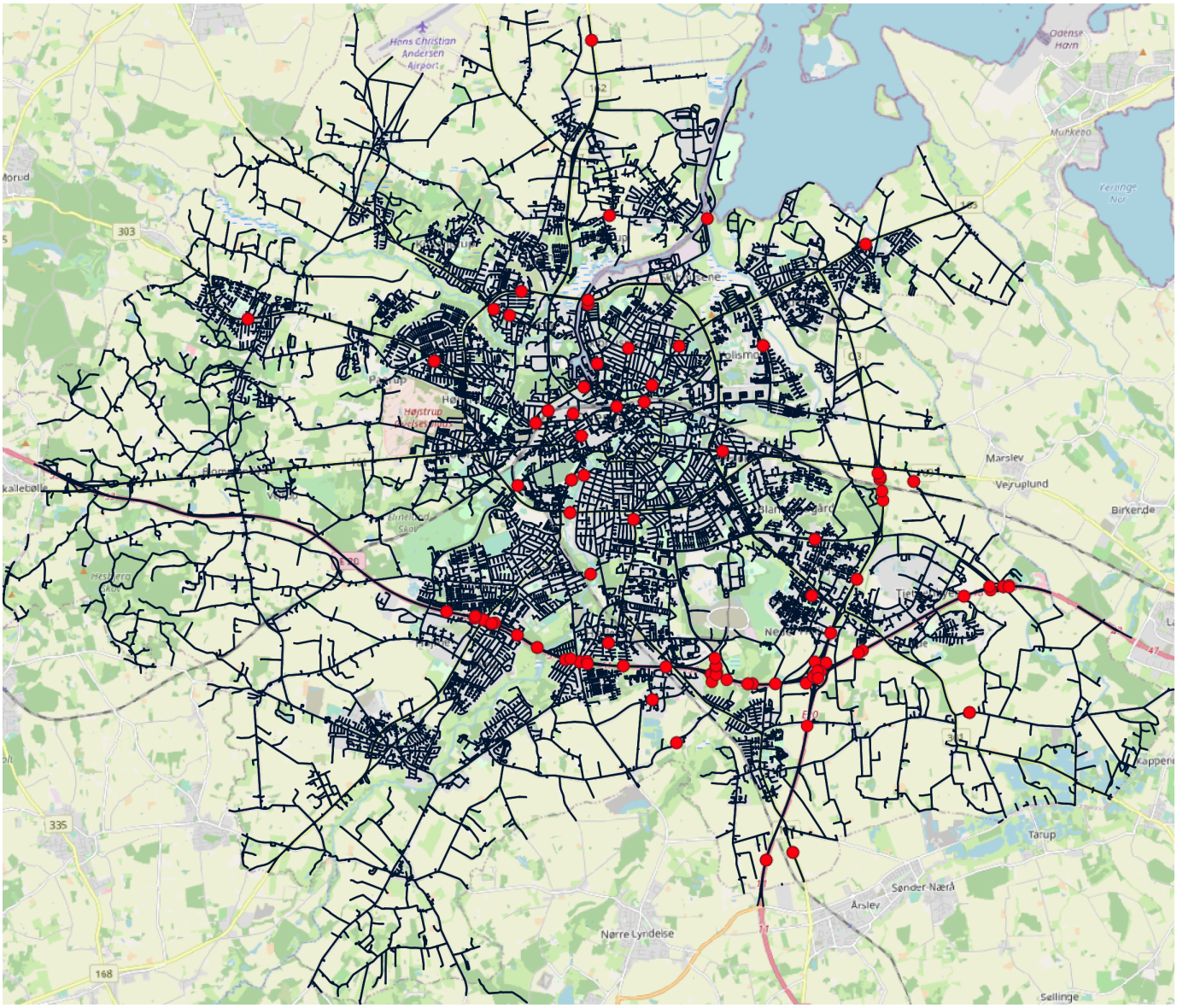

Fig. 9: FCDD road segments (black lines) and ILDD points (red dots)

APPENDIX C
DATA WAREHOUSE APPENDIX

Fig. 10: Data warehouse design

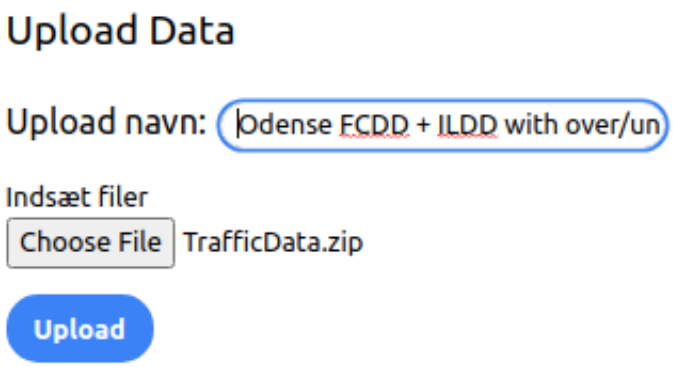

Fig. 11: Upload a new data set

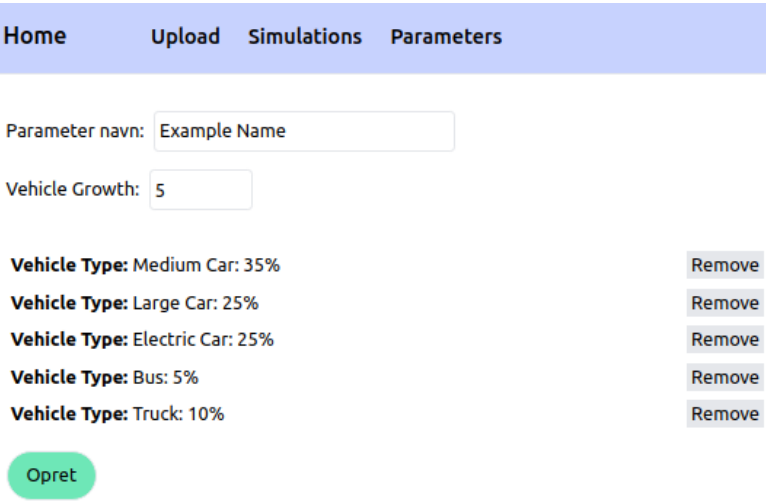

Fig. 12: Create a parameter set

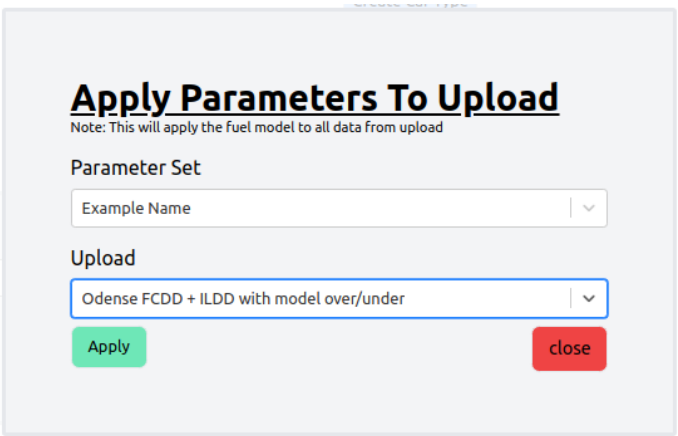

Fig. 13: Apply the parameters to the uploaded data