Received 7 June 2022, accepted 28 July 2022, date of publication 1 August 2022, date of current version 5 August 2022.
Digital Object Identifier 10.1109/ACCESS.2022.3195569

APPLIED RESEARCH

A Long-Distance Smart Driving Service Based on Floating Car Data and Open Data

DAVID ESCUÍN[1], LORENA POLO[1], DAVID CIPRÉS[1], CARLOS MILLÁN[1], AND JORGE CARCAS[2]
[1]Instituto Tecnológico de Aragón, 50018 Zaragoza, Spain
[2]Grupo Sese, 50014 Zaragoza, Spain
Corresponding author: David Escuín (descuin@itainnova.es)

ABSTRACT The purpose of this manuscript is to present the Smart Driving Service (SDS), a customized mobile application, and a complex microservices framework intended not only for professional drivers but also for novel people who need help during the driving time in their long-distance journeys. The European regulation on driving times, breaks and rest periods for drivers engaged in the carriage of freight is implemented in the system. Additionally, it is necessary to have a feedback report to detect the behavior of drivers and what to do differently to improve driving. This issue is addressed by implementing a Route Performance Index (RPI) to measure the driver compliance. The proposed service has been running in a production stage for 6 months with a reduction in consumption of 2 liters/100 km. Considering that the company runs more than 100M km per year, the savings in fuel are relevant apart from the environmental impact reduction.

INDEX TERMS Driver assistance, route navigation, floating car data, telemetry, truck sensors, open data.

I. INTRODUCTION

The purpose of this manuscript is to present the Smart Driving Service (SDS), a customized mobile application and a complex microservices framework that is intended for not only professional drivers but also for novel people who need help during the driving time in their long-distance journeys.

The project is supported by one of the Europe's largest Logistics Solutions Providers (hereafter, the LSP) in the areas of road transport, logistics, industrial services and supply chain management. In particular, the LSP proposed to analyze and leverage thousands of miles of records from truck telemetry (best known as floating car data, FCD) that, combined with Open Data Services (maps and weather), could improve vehicle efficiency and a fuel consumption reduction.

The service, already running in a production environment, consists of a real-time intelligent driving assistant that provides drivers with the following features:

• Offers navigation over a predefined network built with homologated paths that the fleet company considers as valid for driving (such as avoiding tolls, big slopes, congestion areas, and motorways imposed by specific customers).

• Suggests a recommendation of speed for all the arcs on the route extracted from the processed historical data from telemetry and integrated with weather forecasts and other network features such as maximum speed and slopes.

• Informs about advisable stops with security cameras and surveillance services.

• Remembers to drivers when and where to refuel (mainly in gas stations with agreements)

• Calculates the Estimated Time of Arrival (ETA) and decides whether to speed up or slow down dynamically according to the delivery time windows.

The European regulation on driving times, breaks and rest periods for drivers engaged in the carriage of freight is implemented in the system. The Smart Driving Service includes a stand-alone driving simulation tool that estimates the ETA for a particular driver and route. The current state of the driver's tachograph is known periodically and the SDS adapts the speed recommendations to the route profile. The drivers have the SDS application on their mobile phones and work as a voice guidance navigation system. They follow

The associate editor coordinating the review of this manuscript and approving it for publication was Aysegul Ucar.

the continuous spoken turn-by-turn instructions (hands-free), especially in urban and resting areas. To avoid interruptions in low coverage areas, the information of the recommendations is cached. The drivers do not have to use the mobile phone when circulating. The usage is strictly limited in the loading and unloading operations or in the resting areas, in line with the European regulations on mobile devices in vehicles.

Fleet and customer managers can track the vehicles and the speed recommendations using customized dashboards. The ETA is continuously updated when vehicles are fully stopped for some period of time in a traffic jam or the driver has ended their break. Smart Driving harmonizes the work of drivers by providing unified routes, speed recommendations, and shared resting locations. A testing period of more than 6 months achieved an average fuel saving of 2 liters per 100 km. The LSP owns more than 1000 vehicles and 100M kilometers per year.

This paper describes and highlights how the integration of FCD with open datasets has resulted in a production-already smartphone application that improves the driver's experience and reduces the cost of the LSP. The remaining part of this paper is organized as follows. Section 2 gives the literature review of relevant research that focuses on the use of FCD for traffic and speed recommendations. Section 3 describes the multiple datasets integrated into the Smart Driving solution: telemetry, stops, radars and an innovative homologated route network. Section 4 presents the Smart Driving recommendation system and its primary software components. The SDS architecture is explained in Section 5, and Section 6 describes the smartphone applications installed on trucks. Section 7 describes the SDS control panel, which allows fleet managers to interact and know the status of vehicles in real time. Sections 8 and 9 briefly present a discussion and conclusions, respectively.

II. LITERATURE REVIEW

Analyzing the state of the art of this research implies a review of the different building blocks that integrate the technological solution presented in this paper. In the literature, there are various research papers that address the different technologies covered in this paper: i) floating car data, ii) the Advanced Driving Assistance System, iii) driver behavior, and iv) sensors.

The usage of FCD is intended not only for ADAS but also for the creation and maintenance of navigation map databases. As explained in [1], these maps are enhanced to offer dynamic route guidance, delay description and road capacity balancing and routing algorithms. As technological advances have permitted an increasing amount of data in real time, map databases are constantly extended and updated with images of the road and estimation of the local traffic level, apart from other vehicle sensors [2]. [3] goes a step forward and addresses the map-matching problem to build the road network from scratch, owing to all representation points that are connected by a Delaunay triangulation network and applying a shortest path searching approach between the connected representation point pairs. The authors in [34] evaluate the accuracy of map-matching algorithms by using FCD to identify trajectories and extract traffic patterns. A map-matching system is executed in our solution, but this is not the core of our research.

Papers related to driving assistance are the most found in the literature because of the tremendous impact on the fuel savings and safety warnings. This is also motivated with the use of newly emerging sensor technologies to provide real-time traffic information and driving patterns. Some works put the focus on the frameworks as done in [4] which shows road application models for a smart roadside system and sensors with a speed advisory system for highways. Most of the papers address the topic of eco-driving and how to reduce the fuel consumption and optimize the use of energy. The work presented in [5] exposes a driving assistance prototype called Driving coach that collects weather, traffic and vehicle information to provide drivers fuel-efficient driving hints and monitoring metrics of their performance. Neural networks are also used to evaluate traffic data. The paper in [42] studies Deep Convolutional Neural Networks (DCNNs) for the accurate estimation of space–time traffic speeds given sparse data on freeways. The authors propose a methodology that allows to effectively train data on several domains and reconstruct congestion types. And Long Short-Term Memory Neural networks (LSTM) are also investigated to derive traffic speed predictions from FCD as explained in [35]. In this paper, the authors improve the prediction accuracy with characteristics of historical average speed. A large experiment with more than 100 drivers and 8000 km showed in [6] helped to develop a very simple microscopic model to estimate vehicle fuel consumption with infield instantaneous measurements. Positive remarks about eco-driving and its relation to the road type are also found in [7]. This paper presents a methodology for different road sections processed using the R code to evaluate the specific impacts on fuel savings. Adopting a new driving style for the first time could affect the driver's acceptance and undermine the efficacy of new technologies. The purpose of the research in [8] is to measure and evaluate the user's responses to the first-time use of eco-driving assistance technology. And the paper in [36] presents a general analysis methodology aimed at processing FCD to reconstruct the routes followed by the drivers and evaluate the possibility of modeling drivers' route choice.

Several studies have been conducted on the impact of driving behavior on fuel efficiency. Wijayasekara [9] presents a low-cost framework and a hardware setup for prompting drivers on fuel efficient behavior with the help of rich information and intuitive un-obstructive visualization. Muslim [10] focuses on the road transportation to generate a new dimension called "green driver" to establish the green driver's behaviors related to fuel saving and emission reduction. In this research, twenty-one variables classified into four clusters were identified to conclude that driver personalities should be integrated for green driving accreditation.

Estimations on energy consumption can be found in [31]. In this paper, authors present an approach to estimate traffic energy consumption via traffic data aggregation in probability distributions. By using a microscopic traffic simulator, they compared the estimated energy consumption to the measured energy consumption.

Undoubtedly, controlling vehicle speed is a promising method for lowering fuel consumption. To do so, it is mandatory to involve drivers as they are the last link of the chain and the final users of the ADAS. Driving behavior has a significant impact on vehicle fuel consumption. This aspect is a key factor in our research and is covered by most solutions. The paper in [37] pays attention to the reliability of the data gathering by comparing the speed data obtained from FCD with those recorded by inductive-loop detectors. The conclusions show that the FCD can be proposed for the monitoring and operation of mobility along road networks.

In the prototype system presented in [11], the authors introduced an optimal control model of acceleration that mimics drivers' behavior to trade-off driving preferences as the space to the preceding vehicle or the way corners are taking. A breakthrough innovation to derive speed recommendation using oriented agents can be found in [12]. That research combines, integrates, and evaluates multiple information sources (weather and routes) that cohesively align vehicle information to estimate the status of the vehicle and provide recommendations for speed. Dynamic programming is the methodology used in [13] to minimize the energy in passenger vehicles over a Raspberry Pi and providing speed recommendations to drivers. The authors report an 8% reduction in fuel consumption compared to the test data. Another dedicated study on the relationship between driving behavior and fuel consumption is reported in [14] with the execution of clustering algorithms and the generation of their corresponding models. The deep learning-base models were able to predict the fuel consumption associated with different driving behaviors. Other work that makes use of machine-learning functionality to forecast energy consumption for eco-driving assistance system can be found in [15] with demonstrated experimental results. It is equally important to seek the strategy on how to persuade drivers to change their driving behavior. In the paper presented in [16], the authors identify functional, design, safety, and persuasive features for systems supporting fuel-efficiency. As a conclusion, they highlight the needs for overall situation assessment when it comes to eco-driving. The paper in [17] addresses similar issues. In this occasion, they explore the use of wearable devices in a dynamic driving environment to show statistically significant differences between various levels of driving demand. And the paper presented in [38] gives details on how to use real-time data from GPS and automotive radar to perform a predictive optimization of a vehicle's speed profile and coaches a driver into fuel-saving and CO2-reducing behavior. The authors demonstrate the feasibility of using data from vehicle sensors to achieve a reduction in fuel consumption. Good eco-driving practices can be found in [39], which presents how fuel consumption can be optimized in a waste collection fleet by installing in-board driving assistance devices and providing real-time feedback.

For a deep analysis on how sensor technology can be integrated with the transportation infrastructure to achieve a sustainable Intelligent Transportation System (ITS) and how safety, traffic control, and infotainment applications can benefit from multiple sensors deployed in different elements of an ITS, it is recommended to take the paper [18] into consideration. A clear example of this is described in [19], in which the authors propose an approach for designing a vehicle collision warning system based on the fusion of multiple sensors and wireless vehicular communications. Another example is found in [41]. This paper examines the implementation of a cooperative intelligent transport system by exploiting road traffic and weather data. As a result, the pilot platform makes the system suitable for use in heavy vehicles in real-time.

Electric vehicles deserve special attention. In addition to optimizing power and battery life, researchers have conducted studies on how to facilitate drivers with assistance systems to reduce accidents and fatalities. One method to prove this is shown in [20]. This paper describes a system to assist drivers in vital tasks, such as braking, owing to the implementation of a simulator and an automated control algorithm. Similar methodologies are proposed by the authors in [21] where the effects of driving style are analyzed with hybrid electric vehicles (HEVs). This manuscript describes a statistical pattern recognition approach to classify drivers into six groups, from moderate to aggressive, using kernel density estimation and entropy theory.

With respect to route navigation and traffic congestion control, FCD systems have provided a vast amount of literature. Interesting works are found in [22], which presents an architecture for real-time traffic-dependent navigation, and in [23]. In this paper, the authors describe a Dijkstra algorithm-based system for calculating the shortest route in a parking lot environment. The paper in [33] investigates the opportunities offered by FCD to infer the number of delivering activities per tour with light-good vehicles. The technological background adopts vehicle-to-everything (V2X) and driverless technologies. A measurement indicator called the traffic state level (TSL) formulated by [24] evaluates the speed provided per consecutive road segments through the FCD as a data source. The detection of traffic partners from FCD is other challenge that stimulate concern among researchers. The work explained in [25] shows a case study carried out in Ankara, where a 1-min interval FCD allows the authors to transform the average speed values into a qualitative 4-scale state parameter based on the level of service and identify bottlenecks. The connected vehicles gain special importance in [26] to regular in real time the traffic signals and in [30] demonstrates how the combined use of data from idle and active devices improves congestion detection performance in terms of coverage, accuracy, and timeliness. The authors apply a new method to real mobile signaling data obtained from an operational network and present an extensive validation study based on the ground truth obtained from a rich set of reference datasets.

Driving simulators are less often found in the literature but provide valuable knowledge. One example is found in [27], which presents a study designed with 12 potential eco-driving interfaces to advise the driver with the most fuel-efficient accelerator pedal angle in real time, accompanied with audio alerts. Another interesting simulator is described in [28]. Here the authors present an ADAS-based application to provide the optimal value of the target vehicle speed by considering the road gradients and speed limits of the upcoming road segments. One step forward is done in [40]. In this paper, the authors evaluate drivers' behavioral responses to scenarios when driving vehicles with an ADAS compared to vehicles without an ADAS. The results show that the ADAS influences driving behavior by making drivers less aggressive and harmonizing the driving environment. As in our research, the driver experience guided the solution and was implemented on a prototype.

To conclude the existing research, game theory can play an important role in providing speed recommendations. The authors in [32] solved the challenge by modeling the problem as a game in which drivers are the players and the speed of the vehicle is their strategy. Through evolutionary dynamics, the drivers receive the recommended speed at specified time intervals.

The conducted literature review shows the benefits of using FCD for traffic forecasting and speed recommendations systems. However, to achieve a reliable speed value, it is necessary to integrate other datasets apart from telemetry as updated maps and to know the accuracy value of the driver's tachograph in real time. The Smart Driving Service seeks to provide a software product that is harmonized and validated with all the actors on the road transport, from the drivers to fleet managers and customer account supervisors.

III. DATASETS

Prior to describing the SDS architecture and the algorithms in detail, this section presents the list of the heterogeneous datasets used in the SDS.

A. FLOATING CAR DATA

The floating car data coming from a fleet of more than 100 telemetry device-equipped vehicles are retrieved every 1-min interval. Trucks are equipped with devices certified by UNE-EN 12830: 2019. The equipment has 2 Can Bus, a Bluetooth, USB, and WIFI input and output modules, and all connectivity is certified by automotive device manufacturing providers.

The devices collect information from the Electronic Braking System (EBS) to retrieve trailer covered distance, trailer speed, supported weight by the three axes, safety systems, breakdown lights, Anti-Block System, Yaw Control, Roll-Over Prevention and Traction Control system, pressures, maintenance, and general status of the vehicle. The vehicle position, consumption (L/100 km), and speed are sent with a timestamp. The average values calculated internally are also stored in the cloud.

These trucks travel approximately 100 million kilometers per year over Europe. Through a vehicle sensor placed in each vehicle, vehicle sensor data are processed and transmitted to a cloud storage server (Azure Cosmos DB) for further processing and obtaining insights into the road transport driving environment. The devices provide location coordinates, current speed, instant consumption, fuel level, accumulated fuel, kilometers accumulated, altitude, and other values.

Apart from this, drivers must obviously meet the European regulation [29] on driving times, breaks and rest periods in concordance with the digital tachograph. As a general rule, the regulation lays down that drivers must freely dispose of their time for at least 11 hours during their daily rest period and at least 45 hours for the regular weekly rest period. With regard to their daily driving time, drivers cannot exceed 9 hours although this value can be extended to at most 10 hours not more than twice during the week. The maximum weekly driving time is 56 hours being 90 hours the total accumulated driving time during any two consecutive weeks. Alternatively, there are other options not explained here for being less relevant.

These driving restrictions and normative are coded in the Driving Simulator Service (DSS) that is consumed by SDS to valid route proposals. The DSS receives as input the distance in kilometers and the tachograph status of the driver. As an output, returns the estimated status of the tachograph at the destination, the ETA, and the locations where to stop and resting time. An external web service (TIS, Tachograph Information Service) provided by the telemetry sensor manufacturer processes driver cards and provides tachograph information in real time with a delay of 10 minutes. This information includes the driving time from the last stop, the accumulated daily and weekly driving time, and the daily and weekly accumulated resting hours. The DSS also computes the tachograph simulation for the two-driver driving.

B. MAP MATCHING SYSTEM

The vast amount of GPS data retrieved from vehicles is subjected previously to a preprocessing system to match the records with a logical model of a geographic information system, best known as map matching algorithm. It would be impossible to use raw data because of the GPS positioning accuracy of each device. Even when a vehicle is stopped, there is always a certain deviation between the points, resulting in the GPS positioning points being scattered on both sides of the road. Another possible scenario is a vehicle driving on the left side, which can result in two road segments. This is perfectly avoided with the bearing reported by the device, which guides the system to identify the real segments on the logical maps.

The SDS backend leverages the OpenStreetMap (OSM) to match the GPS data with the Graphhopper Map Matching Engine. Map matching algorithms are executed offline, meaning that the data are first recorded in real time, but later matched to the road network. This happens on weekends when most drivers are taking their weekly resting time, and the amount of telemetry data is much lower. Executing offline map matching results in a good compromise between performance and accuracy.

FIGURE 1. **Observations (speed values) identified by the map matching system over the matched segment road.**

As a result of executing a map matching algorithm, a Mongo DB database is updated with the list of speed values for each segment (also known as ''arc'') in the OSM logical model. Each segment length varies from meters to kilometers, depending on the granularity of the model. Prior to persisting the segment speeds in the database, outliers are excluded, as is found in situations with traffic jams or bad weather that produce excessively low speeds. The experience of drivers has demonstrated that speeds lower than 50 km/h are unrepresentative of road transport reality and, furthermore, could disrupt the smart recommendation system based on the handling of the historical speed data, as explained later.

Traffic congestion, road incidents, and adverse climate conditions, together with the lack of knowledge on the decisions adopted by the driver, make it extremely difficult to draw accurate conclusions about what really happened. One aspect that helps is to match speed with the loads. For example, suppose a road segment that is reported with these historical speed values (in km/h): 79 (2 times), 85, 84,82,81,80,76,75,64 and 63 km/h. The legal speed limit for trucks is 90 km/h. According to the data, one may think that the location has a notable slope because the highest reported speed value is 79. In addition, there is no information related to whether the driver turned on heating or air conditioning to reduce fuel consumption by slowing the speed. Thus, it is highly recommended that the trailer load be registered with the matched speed to derive a more reliable speed.

At the end of this algorithm, the database stores a list of historical speed values for each arc done by the vehicles.

1) RESTING AREAS

The DSS needs to know the exact position of the resting areas. One approach to enrich the SDS with this information would be to use external services dedicated to truck routing for fleets. However, the SDS seeks to identify resting areas with artificial intelligence by analyzing telemetry data and matching them with tachograph information. The main reason for this is that the SDS should be primarily built and led by the driver's behavior and their routines. In this sense, the locations supported by most drivers should be considered as valid resting areas for the SDS.

Identifying a resting area from FCD is challenging. The first approach is to consider that a large list of zero values close in time on vehicle speed could indicate that the vehicle is stopped. In such cases, vehicle positioning may become unstable because of changes in the operation or stopped in a traffic jam. This issue can be fixed by matching the position with map matching and estimating the vehicle heading. Some testing data have revealed that 15 min of zero speed is a good criterion for assuming that the vehicle is stopped. Thus, if many vehicles are at the same point, this location is a provisional point that is confirmed with the Overpass API, an inverse geocoding API, based on OSM, which returns the data associated for a given GPS point, if any. The Overpass API is a powerful application for extracting and filtering data, even for an area or a list of GPS coordinates.

2) RADARS

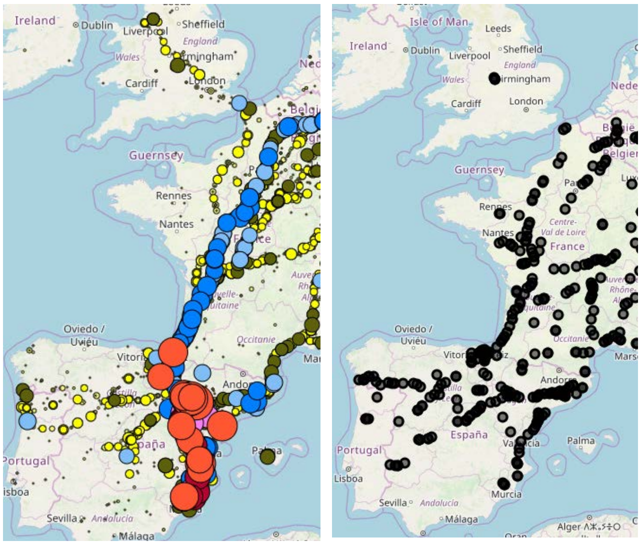
Another key point for the success of the SDS is the fact that drivers are interested in knowing when they are approaching a radar. This additional feature, not even contemplated at the beginning of the project because it does not affect the route planning, has led to a significant positive effect on the acceptance of the SDS by drivers. In this sense, as done before with the resting areas, the SDS periodically updates the list of radars by requesting the Overpass API with the appropriate tag filters.

C. HOMOLOGATED ROUTES NETWORK

The Homologated Routes Network (HRN) is one of the key pillars of successful SDS. Owing to this module, drivers drive along the routes defined by the LSP and not those retrieved from map providers, such as Google Maps or Bing. The reason is obvious: the experience of the drivers and fleet managers provides better routes for trucks than those provided by third-party services. This approach also helps to unify the driving criteria. For example, to cross Paris (one of the hottest areas in Europe in terms of traffic), there are several alternatives, and none of them obtained by external map providers satisfies the company.

With the HRN, the trucks drive where the company wants them to drive, and this is achieved by building their own routing and mapping system. The HRN ensures that all drivers drive across the same roads and highways. This is very useful for novel or new drivers that find it easy to join the company.

Of course, creating a specific GIS (Geographical Information System) for SDS would be almost unmanageable. Fortunately, this is not necessary because the number of roads, motorways, and highways in the LSP network is not very large. The HRN only stores the critical routes, those that are not returned by third-party APIs, and those that overlap with the general routes. For example, to avoid a



(a) (b)

FIGURE 2. Overpass API results. (a) The figure shows the locations where vehicles have been stopped for more than 15 min. The higher the circle, the more vehicles were found in the dataset. (b) Radar locations found in the OSM.

toll road or crossing through Switzerland, the HRN should have an entry (origin and destination coordinates only) for that homologated path. Other reasons for building a customized path are imposed by the final customers (the freight destinations).

Figures 3 and 4 show representations of the HRN that explain the HRN data model. Each colored line represents the beginning and end of the desired homologated path. Only the coordinates of the start and end points are stored in the HRN and not the full list of arcs of the route. In addition, because the sections are on roads or freeways on the map, the number of paths is double the number shown in the drawing, as both directions must be considered. For example, as the usual route from Bordeaux to Poitiers would be via A10 and this alternative does not convince the company, two sections are created in the Angoulome area. Then, how does SDS build the HRN?

SDS periodically builds and refreshes the HRN in memory. Prior to calculating the SDS routes, all homologated paths must be connected between them. If the destination of any homologated path is less than 100 km from the start of another homologated path, both paths are automatically merged, and distances are recalculated. At the end of this process, the homologated network is built on memory and is ready to be used as an input for a new route request.

Each new request for routing comes with an origin (usually the vehicle location) and a destination. Whether the origin and the destination are both less than 25 km to some homologated path, SDS makes use of the HRN by executing the Dijkstra algorithm. Otherwise, a conventional call to the Graphhopper Directions API is invoked. For example, for travelling from Lisbon to Duisburg, the HRN would retrieve a solution as can be derived from the Figure 4. The solution is significantly different from that retrieved by the Graphhopper Routing Service.[1]

However, from Seville to any other location in Europe, SDS would make a request to the third-party API as no homologated paths in the Seville area are found in the HRN.

Fleet managers and experienced drivers manage and admit updates in the HRN. The volume of customers, their needs, tolls, and other parameters decisively influence the path choice. The HRN does not suffer from scalability issues, and the response to SDS is much faster than when invoking a third-party API call.

FIGURE 3. A detailed view of the homologated routes in the Angouleme area forces drivers to travel from Bordeaux to Poitiers via colored routes.

IV. SMART DRIVING SERVICE

The previous section has described all the types of datasets required for the speed-recommendation system and how they are obtained. The SDS calculates the route over the HRN as explained before. This section describes the intelligent system that recommends the speed to the driver at each moment according to the point (or arc) where the driver is situated.

Each route request, in addition to the origin and destination coordinates, receives the delivery time window as an input. The SDS must recommend speed values that ensure that the arrival of the vehicle occurs within the time window with minimum fuel consumption. Note that one might think that driving at the legal minimum speed would result in the lowest possible fuel consumption, provided the arrival to the destination is on time. However, the trucks would not be profitable in

[1] https://graphhopper.com/maps/?point=Lisbon%2C%20Portugal&point =51.433464%2C6.778564&locale=es-ES&elevation=true&profile =car&use_miles=false&selected_detail=Elevation&layer=Omniscale

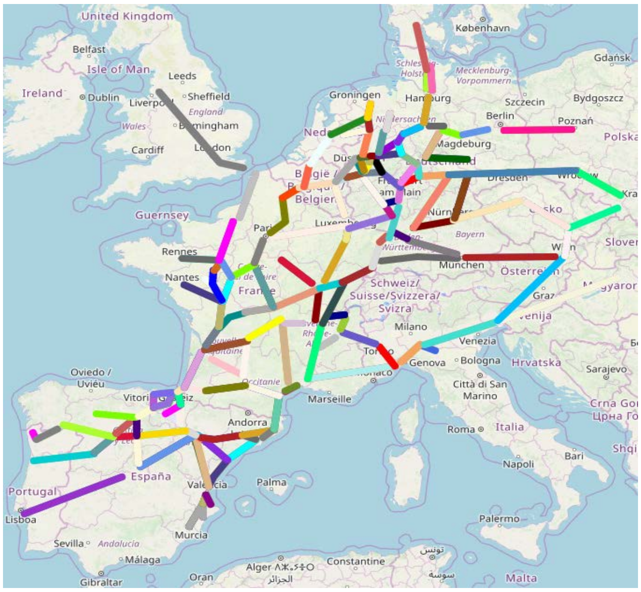


FIGURE 4. This picture shows the homologated route network in colored lines. The routes were built near LSP customers to make SDS use them at the time of constructing a driver journey.

monthly kilometers, and the driving experience for the drivers would be discouraging. In this respect, the minimum speed for trucks under normal conditions is set at 70 km/h, a value agreed upon by drivers.

As stated before, the eligible speed values are obtained from the processing of the floating car data to ensure that the estimated values are real and can be reproduced. A major technical challenge is dealing with slopes in loaded trucks when there are no speed records in the historical data. To address this problem, the OSM elevation API is used to obtain the gradient of each arc of the route and its legal maximum speed, if any. With this information and considering that the maximum legal speed for trucks is 90km/h, the initial minimum and maximum speed values are calculated. Next, to apply gradient effects, a factor is applied to the maximum speed. Only if the maximum speed falls below the minimum speed, the last one is updated. The speed reduction factors, reflected in Table 1, are applied depending on the load weight. As agreed by the drivers, a gradient of less than 1.5% is considered irrelevant.

TABLE 1. Speed reduction factors for the slopes.

Gradient (%)	< 6000 kg	< 15000 kg	other
≤ 2.0	99%	95%	90%
≤ 2.5	95%	90%	85%
≤ 3.0	90%	85%	80%
> 3.0	85%	80%	75%

Factors applied to max speed when a gradient is greater than 1.5%.

We performed a test period of one month to adjust this value with simulations of the trailer load with the matched speed to derive a more reliable speed. A value higher than 1.5 leads to unrealistic speeds being calculated, especially for high loads. In any case, the influence of Table 1 is very low compared to that of the complete system because road transport avoids driving on slopes and is only used when strictly necessary. In fact, the homologated route network s built with hardly any slopes on the roads.

A. DRIVING MODES

Having said all this, the SDS implements two driving modes: normal driving mode and maximum speed driving mode. Note that for each arc, a list of speed histories obtained with map matching is available. Sometimes, the list is empty; therefore, the system can only make decisions based on the maximum permitted speed.

To arrive at a value that is accurate and precise, we conducted an experiment measurement in concordance with the logistics perspective and driver satisfaction. The scientific pillar of the application is important, as is the operational and software design point of view.

The logistics point of view refers to the assumption that transport operators place customer satisfaction slightly above cost and, in order to mitigate incidents, anticipate problems as soon as the shipment departs. This is a conservative and incident-prevention mode. Therefore, higher values are preferred over low speed values.

However, it is also important to consider driver satisfaction. "A tendency to lower speed values leads to more boring driving and less attention on the road" is a common saying among drivers.

1) NORMAL DRIVING MODE

The Normal Driving mode is the default criterion when the SDS receives a new request for routing. This mode supposes that the driver disposes enough time to perform the service, and thus, the fuel saving could be the highest. These steps are described in Algorithm 1. The following operational decisions are worth noting.

1) The 60th percentile (P_{60}) is considered a good trade-off between the travel time and consumption.
2) The speed up strategy is intended to increase the speed at the beginning of the route, and not in the complete route. This is achieved by obtaining a higher percentile from the speed history or by using 100% of the maximum permitted speed.
3) In contrast, in the case of arrival before the time windows, the recommendation system gradually reduces the speed in the last arcs of the route.

2) MAXIMUM SPEED DRIVING MODE

This mode assigns the maximum permitted speed for each route arc and is executed when 1) the Normal Driving mode produces vehicles to arrive late, and 2) additionally, fleet managers wish to evaluate the ETA to evaluate future expeditions for other customers or schedule other non-foreseen orders. The steps are as follows:

Algorithm 1 Normal Driving Mode

Require: **Inputs**
 $\mathbf{H_a}$ = list of historical speed values for each arc a
 $\mathbf{MRS_a}$ = maximum real speed for each arc a (obtained by telemetry)
 $\mathbf{MPS_a}$ = maximum permitted speed for each arc a (obtained by OSM API)
 for each arc a of the route **do**
  $s_a = P_{60}(H_a)$ // Let $\mathbf{s_a}$ be the selected speed for each arc a
  if s_a does not exist **then**
   $s_a = MPS_a$
  end if
  $s_a = \min(s_a, MRS_a)$
end for
Find all stops for the journey();
ETA = ComputeETA();
if ETA is within time windows **then**
 return solution
else if ETA is later than time windows **then**
 Apply $P_{100}(H_a)$ or MPS_a only for the first arcs of the route until meet time windows (speed up strategy)
else if ETA is before than the time window **then**
 Iterate over the last arcs of the route by applying gradually the $P_{50}(H_a)$ until meet time windows
end if
Output: Journey with the list of all s_a

1) Start by assigning the maximum speed from the speed history. If it does not exist, use the maximum permitted speed.
2) Compute all speed recommendations for the journey by including stops and estimate the ETA.

B. SPEED SMOOTHING APPROACH

The above procedure returns a recommendation of speed for each arc, which results in a large variability of values along the entire route. These recommendations cannot be communicated to the driver for practical reasons but must be smoothed by calculating the average speed for subsets of them. In addition, it should be noted that the system searches for locations to make mandatory stops, always under the criterion that drivers should drive whenever possible to maximize efficiency.

The criterion used for the smoothing strategy is to partition the arcs into blocks with a size of 10 arcs and calculate the average velocity. On the other hand, the speed changes must be given in increments or decrements of 1 km/h, which makes it necessary to implement another process to adjust the recommendations to the real operation. This prevents the driver from receiving, for example, a recommendation of 74 km/h after a recommendation of 70 km/h. The system gradually increases or decreases the recommendations by 1 km/h.

V. SMART DRIVING ARCHITECTURE

The SDS backend components have been developed, tested through several simulations, and implemented on rapid prototyping for in-vehicle testing. It is assessed in terms of driver experience and performance.

The SDS runs on 2 Ubuntu virtual machines with MongoDB and Java 8. Telemetry data is recorded in the Azure Cosmos DB and a third-party provider supplies tachograph status data for each driver. As described before, Overpass API, OSM elevation API, Graphhopper Directions API and Graphhopper Map Matching are the other components consumed by SDS on the runtime.

The main challenge from a technical point of view was to cope with the difficulty of scalability and the feasibility of the system. When designing the system architecture for an SDS workflow request, it is essential to conduct a manual feasibility analysis. When developing a workload that processes thousands of route requests per day, manual feasibility analysis is clearly untenable. To address this critical issue, the architecture team designed various approaches based on simulations of the inputs and outputs to evaluate the feasibility of the system. The question then is, how does one consider the feasibility of route requests? This can be verified by identifying all the required simultaneous operations for a given request.

The most critical feasibility and performance question lies in the response time because the route request is made from the trucks. The response time is determined by the following microservices:

1) the time of retrieving the status of tachograph.

2) the processing time to calculate the path through the homologated routes network

3) the search of the historical telemetry of the arcs of the route path 3) the processing time of schedule the stops and ETA estimation.

4) the search of the radars.

5) the request to weather (Open Weather API) and traffic incidents.

An event-driving architecture that decouples the services and ensures interoperability was chosen as the design approach. A set of microservices is triggered for every route request (event) that enters the system. This ensures that all microservices operate in response to the event and can process them in parallel with a different purpose. The duration of the route request is approximately 30 seconds. The SDS handles asynchronous requests and is not blocked-in case of their failure. The search for historical telemetry is more than 90% of the time cached (fetched from Azure Cosmos DB), and radars are digitally linked to arcs in the Mongo Database. The status of the tachograph is asynchronously monitored (no need to access it under demand; it is periodically retrieved). The calculation of the path is executed with all homologated route networks in memory (5 GB RAM).

Four software high-level microservices modules are part of the SDS architecture.

- SD-MM: Smart Driving Map Matching. This microservice runs on one virtual machine and periodically retrieves data from the Azure Cosmos DB to perform a map matching workload and store the results on a Mongo database.

- SD-STOP: Smart Driving Stop Location: Stops registered by fleet managers and others derived from telemetry are stored in the Mongo Collection. The microservice provides the nearest stop location for a given coordinate based on driving hours and tachograph status.
- SD-RH: Smart Driving Route Homologated. The module manages, registers, and computes the Dijkstra algorithm to calculate the journey based on the HRN. This is a stand-alone application that uses the JSON format to build routes and compute distances.
- SD-CORE: Smart Driving Core. This is the centralized component. Receive requests for routing and return the response in JSON format. The speed values are calculated, the ETA is estimated, and the vehicle speed is updated to meet the requirements. In addition, other useful information for drivers can be retrieved, such as weather conditions or radar locations.

These software components run in parallel in a microservice architecture based on real-time information obtained not only from on-board sensors but also from external services such as tachograph status and weather services. The Android application installed on the driver's smartphone makes requests for routing through a REST Ful client connector. The request is processed by the LSP web server that resends the request to SD-CORE.

As shown in Figure 6, an SDS workflow request starts with a transport order generated by a fleet manager. A transport order (origin, destination, load weight, and time windows) is linked to a specific vehicle and driver. First, SD-CORE make a call to the tachograph service to obtain the status of the driving and resting hours for the driver. In parallel, SD-RH finds the route by using the HRN. The route is passed back to SD-CORE that interacts with SD-STOP to find stop locations and to compute speed values in a synchronous operation. Finally, the route is retrieved back to the driver, and the smartphone application reproduces the speed recommendations in map and with voice modes.

VI. SMARTPHONE APPLICATION

Drivers receive speed recommendations on a customized-smartphone application, as shown in Figure 7. This GUI has followed the guidelines of the drivers and fleet managers. The terminal displays the results of the SD-CORE service. In addition to the recommended speed, the application displays the total driving distance, total driving time, ETA at the destination, radar locations, and weather forecasts.

Every 5 minutes or once the driver has started the vehicle engine, the application makes a new request for routing based on the actual location of the vehicle. The response can be similar to the previous one, although it may be slightly different because of the updated driving times and some delays in timing. In the case of traffic jams, SD-CORE finds that vehicles are slower than expected and updates the recommended speeds to higher values.

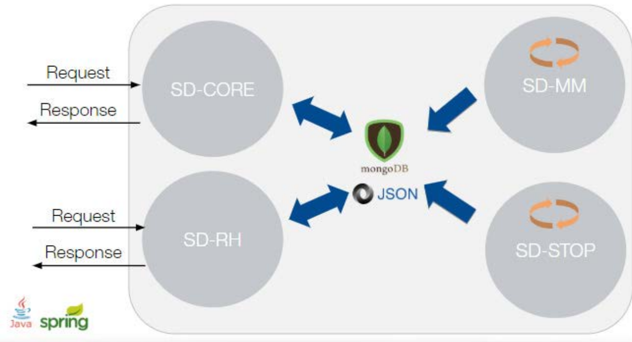


FIGURE 5. The four building blocks of the Smart Driving Architecture. SD-MM and SD-STOP run automatically and periodically to update historical speeds and stops. SD-RH is a stand-alone application used by fleet managers to manage the homologated route network. SD-CORE receives a request for routing and returns the list of speed recommendations.

VII. SMART DRIVING CONTROL PANEL

While drivers are on the roads, fleet managers can control and visualize what is happening with their vehicles and interact with drivers to manage exceptions and satisfy customer requirements. Tailored dashboards have been designed and implemented to rapidly offer managers real-time information on all the data and processes.

For example, the most valuable dashboard plots the homologated route returned by the SDS. In addition to visualizing the recommended speeds, fleet managers can evaluate internally the ETA, stop locations, resting time periods, and weather forecasts. Fleet managers can anticipate unexpected events and perform manual adjustments along the route.

Other dashboards have been built to address the management of map matching calculations, stop location identification by processing telemetry, control and adjust recommendation algorithms, test speed smoothing systems, and request management. All these graphical components are more developer-IT oriented and work in concordance with SDS functionalities.

A. ROUTE PERFORMANCE INDEX

Another key challenge of SDS is to assess how good the recommendations have been and whether the drivers have complied with them. Of course, there are many situations in which it is impossible to comply with the rules, but it is necessary to have a feedback report to detect the behavior of the drivers and what to do differently to drive better.

To address this issue, a Route Performance Index (RPI) is designed to evaluate the four indicators separately:

1) Real arrival time to destination.
2) Evaluation of the driving by comparing the recommended speeds with the real speed at checkpoints located every 200 km.
3) Evaluate whether the recommended stops have been followed and determine why drivers did not stop at them.

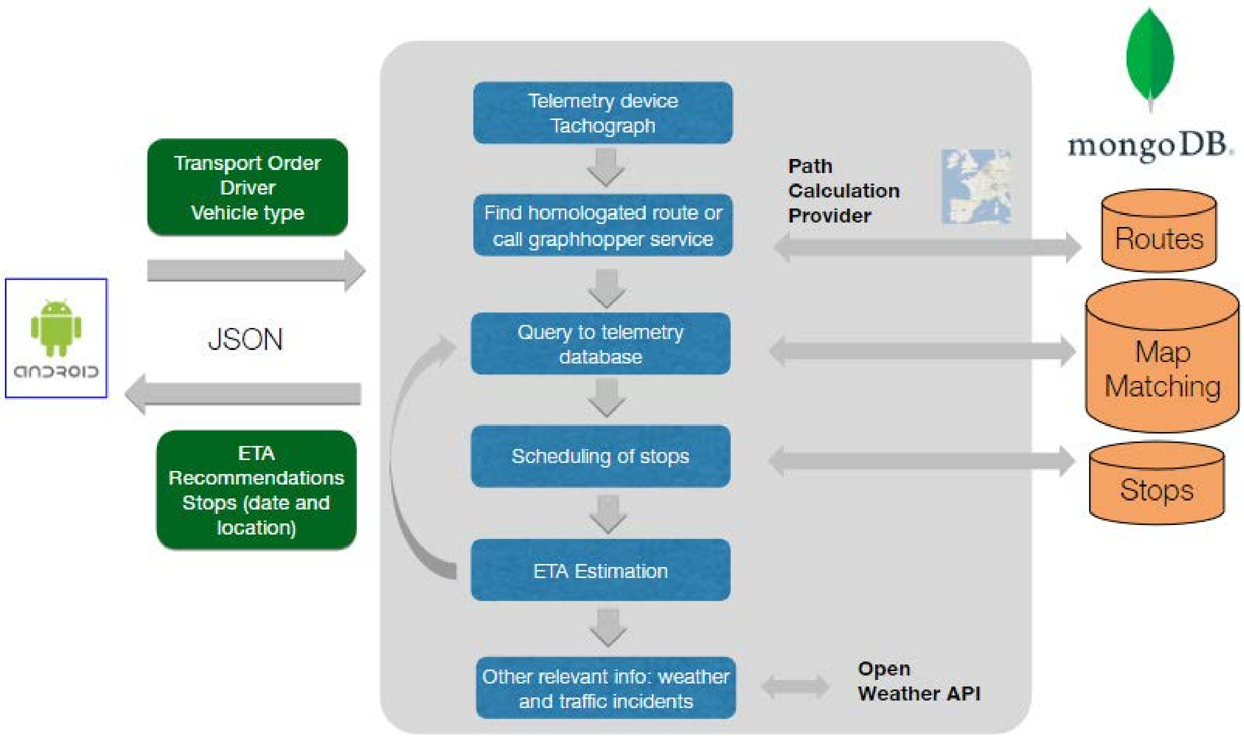


FIGURE 6. Smart driving workflow for each request.

FIGURE 7. Screenshot of the smart driving application installed on driver's mobile phone.

4) Evaluate whether the resting times have been as short as possible.

Each indicator has a relative importance (weight) of 25%. 48 hours after the vehicle has completed the route, the SD-CORE evaluates the above-mentioned four indicators separately and builds a dashboard to graphically plot their values and obtain conclusions. Fleet managers identify the personal and non-personal driving factors affecting the driving experience to adapt the system's decision-making with respect to a driver's progress and responses to recommendations.

FIGURE 8. Details of track recommended by Smart Driving Service. Blue icons represent stops and yellow icons are the recommended speed values. The start and end of the route and weather forecasts are plotted.

VIII. DISCUSSION

The main objective of SDS is to reduce fuel consumption. Consumption is not only reduced by driving the best, but

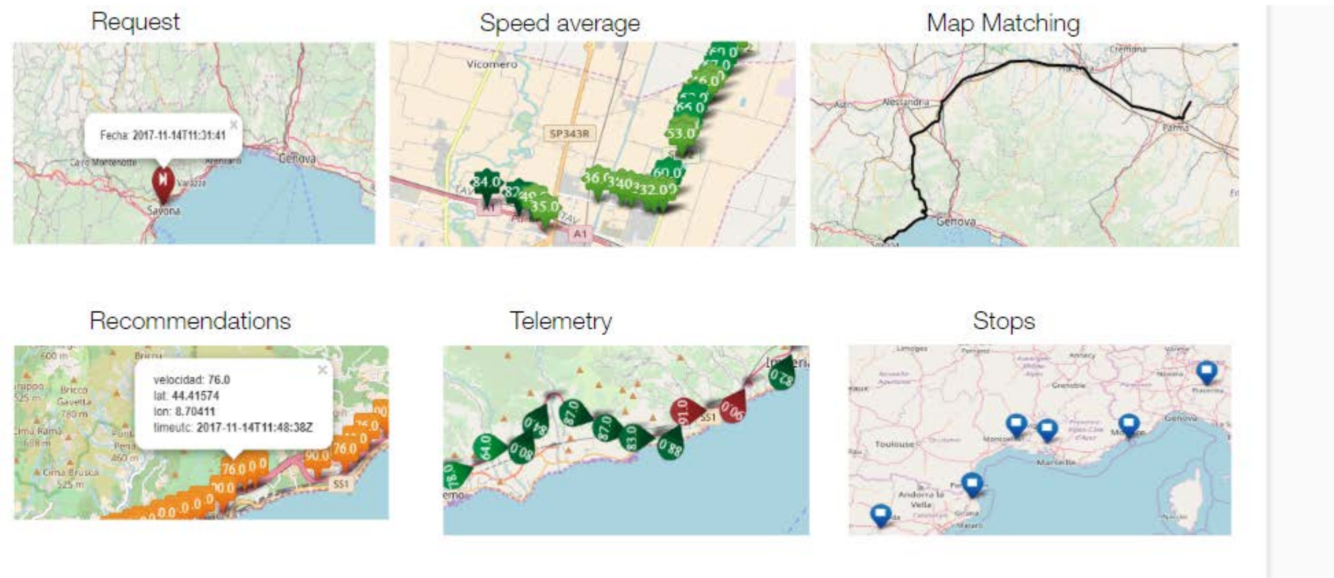


FIGURE 9. Screenshots of fleet managers' dashboards. From top to bottom and from left to right. Location of the new routing request. Real average speed of the extracted vehicle. The route followed by a vehicle is derived after matching observations with OpenStreetMap. Recommendations with timestamps for a particular truck. Telemetry details Locations of scheduled stops in a particular expedition.

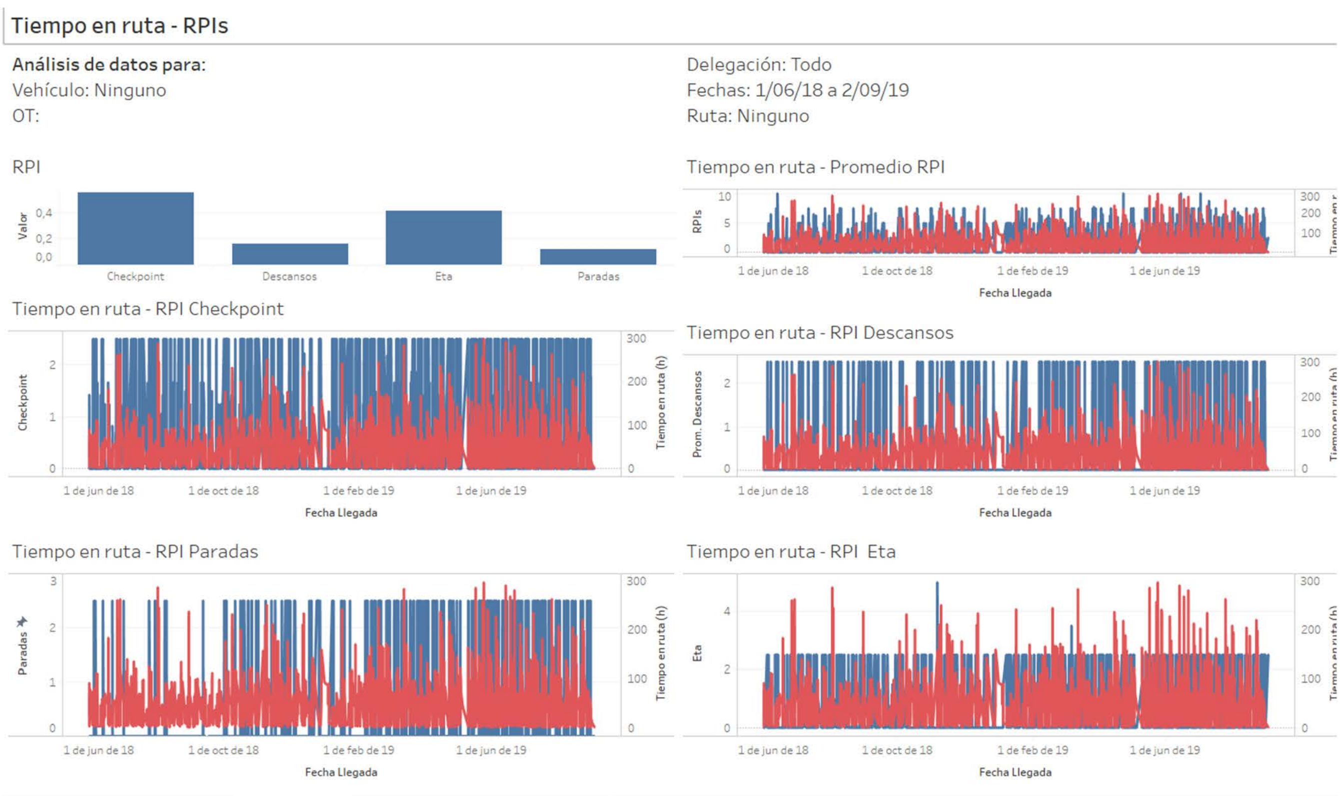


FIGURE 10. Dashboard provides an at-a-glance view of the RPI indicators for a given time period. Fleet managers monitor driver behavior, identify inefficiencies, and adjust Smart Driving Service accordingly.

also by reducing the kilometers in empty trucks. The first criteria are met by the recommended speeds, and the second is related to driver experience and satisfaction. As an increasing number of drivers use the application and more feedback is submitted, the SDS learns more and more efficiency is achieved in terms of kilometers.

The dashboard in Figure 11 is built weekly using a business intelligence tool. This helps fleet managers and

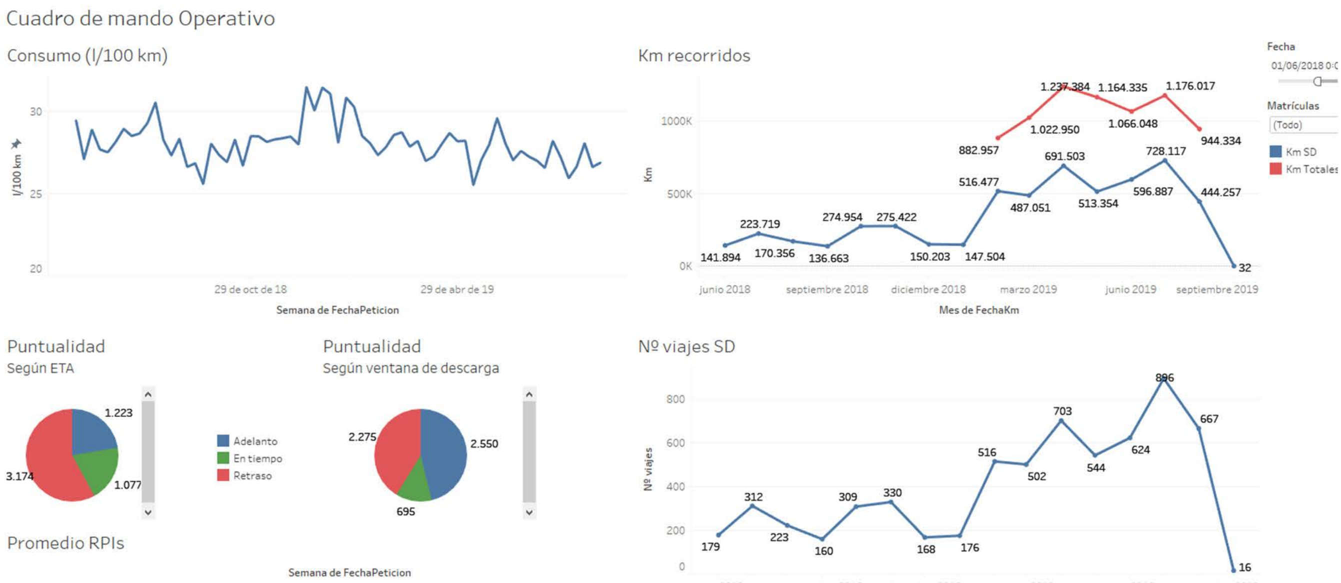


FIGURE 11. Control panel used by fleet and customer managers. It reports the total kilometers covered by SDS trucks and fuel consumption. It also reports the number of vehicles using SDS and ETA compliance.

officers evaluate the kilometers traveled, consumption in liters/100 km, timing or punctuality of drivers, and usage of the SDS. After running in a production stage for 6 months, the reduction in consumption was 2 liters/100 km. Considering that the company runs more than 100M km per year, the savings in fuel are very relevant, apart from the impact environment reduction. Some figures at the time of writing this paper are as follows:

- Number of homologated routes (mainly in Spain, France and Germany): 182
- Number of trucks with telemetry equipment: 905
- Number of trucks equipped with SDS application: 145
- Number of requests / day / trucks:
- Number of stops registered: 231
- Percentage of vehicles arriving within time windows: 57%
- Fuel savings: 2 liters / 100km.
- Average number of speed values for each arc: 16.

Project development has been an enormous challenge that has involved many more issues for fleet managers and drivers. Software integration of all resources used to create a unified single system has been costly. Cloud consultants have provided valuable feedback on data management and many IT lines have been tested to validate data integrity and driver experience. The smartphone application follows the guidelines provided by the final users, and much of the SDS success is due to their involvement.

The innovations of SDS compared to those found in the literature are as follows:

- Generation and maintenance of a proprietary routing system with the daily experience of drivers and fleet managers. This is a completely new block not found in any other fleet-management company. It is common to use third-party systems for agility and flexibility, but building and integrating a proprietary system to homogenize routes gives the company more control over its services.
- Integration of the tachograph status in real time with the search for rest areas to optimize driving times. Current systems assume tachographs with full daily, weekly, and monthly availability. The service areas are selected by the LSP from its own parking pool.
- Telemetry processing as an input source to update speeds based on fuel consumption merged with Open Data and tachograph data. The literature reflects the usage of FCD for 1) building maps 2) identifying trajectories 3) identify traffic patterns 4) Simulate new driving modes. On the other hand, studies dealing with driving assistance are based on real-time data from GPS to optimize the vehicle's speed profile according to the sensors. However, all the components tied together are not found in the state-of-the-art. The contribution of SDS to the literature is that FCD is combined with GPS data in real time in conjunction with the status of the tachograph to achieve fuel reduction and homogenize the way drivers perform the work.
- Design and implementation of an event-driving architecture that provides a system in production. The literature is science-oriented, with pilots testing the results. However, SDS is an essential tool for daily operations in the LSP marketplace.

Further research and innovation activities foreseen by the authors include vehicle-to-vehicle (V2V) communication to improve the service level and analyze the influence of driverless autonomous trucks. Here, the role of the driver is also in question, as the driver's responsibilities will change and eventually, the way the long-distance is conducted.

Driver feedback has been considered from the beginning of the project. In the early stages of the project, recruitment was carried out with drivers with years of experience in road transport and, most importantly, knowledge of Europe's main road transport routes. They emphasized that it had to be an application with little interaction (low visual-manual tasks) but, at the same time, useful for new drivers with low experience joining the company. There is considerable driver turnover in Europe and a shortage at certain times of the year. To motivate the use of the SDS application, drivers requested to know the location of the speed cameras and the ETA at any time. This provides trust and transparency.

Cargo owners positively value the use of the application. It allows them to know the ETA and the location of the vehicles, which makes it easier for them to plan the next activities in depots in the short term. Drivers' familiarity with the network, their previous experience in developing the ability to attend to and process speed recommendations, and the absence of congestion are elements that determine their behavior. KPIs determine user satisfaction and support driver feedback. Drivers highlight that the application does not have a detrimental effect on driver safety.

Finally, to stimulate drivers' behavior and usage of the application, gamification policies based on this approach, such as competition, learning goals, and awards, are also a matter of study.

REFERENCES

[1] J. M. Rowell, "Applying map databases to advanced navigation and driver assistance systems," *J. Navigat.*, vol. 54, no. 3, pp. 355–363, Sep. 2001, doi: 10.1017/S0373463301001503.

[2] S. Messelodi, C. M. Modena, M. Zanin, F. G. B. De Natale, F. Granelli, E. Betterle, and A. Guarise, "Intelligent extended floating car data collection," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 4213–4227, Apr. 2009, doi: 10.1016/j.eswa.2008.04.008.

[3] Z. Ni, L. Xie, T. Xie, B. Shi, and Y. Zheng, "Incremental road network generation based on vehicle trajectories," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 10, p. 382, Sep. 2018, doi: 10.3390/ijgi7100382.

[4] J. A. Jang, H. S. Kim, and H. B. Cho, "Smart roadside system for driver assistance and safety warnings: Framework and applications," *Sensors*, vol. 11, no. 8, pp. 7420–7436, Jul. 2011, doi: 10.3390/s110807420.

[5] E. Gilman, A. Keskinarkaus, S. Tamminen, S. Pirttikangas, J. Röning, and J. Riekki, "Personalised assistance for fuel-efficient driving," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 681–705, Sep. 2015, doi: 10.1016/j.trc.2015.02.007.

[6] G. Bifulco, F. Galante, L. Pariota, and M. Spena, "A linear model for the estimation of fuel consumption and the impact evaluation of advanced driving assistance systems," *Sustainability*, vol. 7, no. 10, pp. 14326–14343, Oct. 2015, doi: 10.3390/su71014326.

[7] Y. Wang and A. Boggio-Marzet, "Evaluation of eco-driving training for fuel efficiency and emissions reduction according to road type," *Sustainability*, vol. 10, no. 11, p. 3891, Oct. 2018, doi: 10.3390/su10113891.

[8] D. Ruscio, G. Caruso, L. Mussone, and M. Bordegoni, "Eco-driving for the first time: The implications of advanced assisting technologies in supporting pro-environmental changes," *Int. J. Ind. Ergonom.*, vol. 64, pp. 134–142, Mar. 2018, doi: 10.1016/j.ergon.2018.01.009.

[9] D. Wijayasekara, M. Manic, and D. Gertman, "Driving behavior prompting framework for improving fuel efficiency," in *Proc. 7th Int. Conf. Hum. Syst. Interact. (HSI)*, New York, NY, USA, Jun. 2014, pp. 55–60.

[10] N. Muslim, A. Keyvanfar, A. Shafaghat, M. Abdullahi, and M. Khorami, "Green driver: Travel behaviors revisited on fuel saving and less emission," *Sustainability*, vol. 10, no. 2, p. 325, Jan. 2018, doi: 10.3390/su10020325.

[11] J. Fleming, X. Yan, C. Allison, N. Stanton, and R. Lot, "Driver modeling and implementation of a fuel-saving ADAS," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, New York, NY, USA, Oct. 2018, pp. 1233–1238, doi: 10.1109/SMC.2018.00216.

[12] I. Galanis, I. Anagnostopoulos, P. Gurunathan, and D. Burkard, "Environmental-based speed recommendation for future smart cars," *Future Internet*, vol. 11, no. 3, p. 78, Mar. 2019, doi: 10.3390/fi11030078.

[13] T. Levermore, M. Sahinkaya, Y. Zweiri, and B. Neaves, "Real-time velocity optimization to minimize energy use in passenger vehicles," *Energies*, vol. 10, no. 1, p. 30, Dec. 2016, doi: 10.3390/en10010030.

[14] P. Ping, W. Qin, Y. Xu, C. Miyajima, and K. Takeda, "Impact of driver behavior on fuel consumption: Classification, evaluation and prediction using machine learning," *IEEE Access*, vol. 7, pp. 78515–78532, 2019, doi: 10.1109/ACCESS.2019.2920489.

[15] M. Masikos, K. Demestichas, E. Adamopoulou, and M. Theologou, "Machine-learning methodology for energy efficient routing," *IET Intell. Transp. Syst.*, vol. 8, no. 3, pp. 255–265, May 2014, doi: 10.1049/iet-its.2013.0006.

[16] E. Gilman, G. V. Georgiev, P. Tikka, S. Pirttikangas, and J. Riekki, "How to support fuel-efficient driving?" *IET Intell. Transp. Syst.*, vol. 12, no. 7, pp. 631–641, Sep. 2018, doi: 10.1049/iet-its.2016.0280.

[17] T. Gruden, K. Stojmenova, J. Sodnik, and G. Jakus, "Assessing drivers' physiological responses using consumer grade devices," *Appl. Sci.*, vol. 9, no. 24, p. 5353, Dec. 2019, doi: 10.3390/app9245353.

[18] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Sensor technologies for intelligent transportation systems," *Sensors*, vol. 18, no. 4, p. 1212, Apr. 2018, doi: 10.3390/s18041212.

[19] M. Baek, D. Jeong, D. Choi, and S. Lee, "Vehicle trajectory prediction and collision warning via fusion of multisensors and wireless vehicular communications," *Sensors*, vol. 20, no. 1, p. 288, Jan. 2020, doi: 10.3390/s20010288.

[20] S. Ahmad, S. Malik, D.-H. Park, and D. Kim, "Design of lightweight driver-assistance system for safe driving in electric vehicles," *Sensors*, vol. 19, no. 21, p. 4761, Nov. 2019, doi: 10.3390/s19214761.

[21] S. Yang, W. Wang, F. Zhang, Y. Hu, and J. Xi, "Driving-style-oriented adaptive equivalent consumption minimization strategies for HEVs," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9249–9261, Oct. 2018, doi: 10.1109/TVT.2018.2855146.

[22] P. Brosch, "A service oriented approach to traffic dependent navigation systems," in *Proc. IEEE Congr. Services*, Los Alamitos, CA, USA, Jul. 2008, pp. 269–272, doi: 10.1109/SERVICES-1.2008.88.

[23] L. Yu, H. Jiang, and L. Hua, "Anti-congestion route planning scheme based on Dijkstra algorithm for automatic valet parking system," *Appl. Sci.*, vol. 9, no. 23, p. 5016, Nov. 2019, doi: 10.3390/app9235016.

[24] O. Altintasi, H. Tuydes-Yaman, and Y. K. Tuncay, "Monitoring urban traffic from floating car data (FCD): Using speed or a los-based state measure," in *Directions of Development of Transport Networks and Traffic Engineering* (Lecture Notes in Networks and Systems), vol. 51. Cham, Switzerland: Springer, 2018, doi: 10.1007/978-3-319-98615-9_15.

[25] O. Altintasi, H. Tuydes-Yaman, and K. Tuncay, "Detection of urban traffic patterns from floating car data (FCD)," *Transp. Res. Proc.*, vol. 22, pp. 382–391, Dec. 2017, doi: 10.1016/j.trpro.2017.03.057.

[26] V. Astarita, V. Giofré, D. Festa, G. Guido, and A. Vitale, "Floating car data adaptive traffic signals: A description of the first real-time experiment with 'connected' vehicles," *Electronics*, vol. 9, no. 1, p. 114, Jan. 2020, doi: 10.3390/electronics9010114.

[27] A. H. Jamson, D. L. Hibberd, and N. Merat, "Interface design considerations for an in-vehicle eco-driving assistance system," *Transp. Res. C, Emerg. Technol.*, vol. 58, pp. 642–656, Sep. 2015, doi: 10.1016/j.trc.2014.12.008.

[28] A. D'Amato, C. Pianese, I. Arsie, S. Armeni, W. Nesci, and A. Peciarolo, "Development and on-board testing of an ADAS-based methodology to enhance cruise control features towards CO_2 reduction," in *Proc. 5th IEEE Int. Conf. Models Technol. Intell. Transp. Syst. (MT-ITS)*, New York, NY, USA, Jun. 2017, pp. 503–508.

[29] *Regulation (EC) No 561/2006 of the European Parliament and of the Council on the Harmonisation of Certain Social Legislation Relating to Road Transport and Amending Council Regulations (EEC) No 3821/85 and (EC) No 2135/98 and Repealing Council Regulation (EEC) No 3820/85*, Mar. 2006. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32006R0561&from=ES

[30] A. Janecek, D. Valerio, K. A. Hummel, F. Ricciato, and H. Hlavacs, "The cellular network as a sensor: From mobile phone data to real-time road traffic monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2551–2572, Oct. 2015, doi: 10.1109/TITS.2015.2413215.

[31] A. Le Rhun, F. Bonnans, G. De Nunzio, T. Leroy, and P. Martinon, ''A stochastic data-based traffic model applied to vehicles energy consumption estimation,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 3025–3034, Jul. 2020, doi: 10.1109/TITS.2019.2923292.
[32] M. Asadi, M. Fathy, H. Mahini, and A. M. Rahmani, ''An evolutionary game approach to safety-aware speed recommendation in fog/cloud-based intelligent transportation systems,'' *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7431–7440, Jul. 2022, doi: 10.1109/TITS.2021.3069850.
[33] A. Comi and A. Polimeni, ''Forecasting delivery pattern through floating car data: Empirical evidence,'' *Future Transp.*, vol. 1, no. 3, pp. 707–719, Nov. 2021, doi: 10.3390/futuretransp1030038.
[34] F. Karagulian, G. Messina, G. Valenti, C. Liberto, and F. Carapelluci, ''A simplified map-matching algorithm for floating car data,'' in *Proc. Int. Conf. Adv. Inf. Netw. Appl.*, vol. 227. Cham, Switzerland: Springer, 2021, doi: 10.1007/978-3-030-75078-7_70.
[35] D. Chen, X. Yan, S. Li, X. Liu, and L. Wan, ''Long short-term memory neural network for traffic speed prediction of urban expressways using floating car data,'' in *Green Connected Automated Transportation and Safety* (Lecture Notes in Electrical Engineering), vol. 775. Singapore: Springer, 2022, doi: 10.1007/978-981-16-5429-9_58.
[36] A. Bracci, C. Colombaroni, G. Fusco, and N. Isaenko, ''Investigation and modeling on Drivers' route and departure time choices from a big data set of floating car data,'' in *Proc. 7th Int. Conf. Models Technol. Intell. Transp. Syst. (MT-ITS)*, Jun. 2021, pp. 1–7, doi: 10.1109/MT-ITS49943.2021.9529304.
[37] G. Cantisani, G. D. Serrone, and P. Peluso, ''Reliability of historical car data for operating speed analysis along road networks,'' *Science*, vol. 4, no. 2, p. 18, Apr. 2022, doi: 10.3390/sci4020018.
[38] J. Fleming, X. Yan, C. Allison, N. Stanton, and R. Lot, ''Real-time predictive eco-driving assistance considering road geometry and long-range radar measurements,'' *IET Intell. Transp. Syst.*, vol. 15, no. 4, pp. 573–583, Apr. 2021, doi: 10.1049/itr2.12047.
[39] J. F. González, A. G. Izquierdo, F. Commans, and M. Carlos, ''Fuel-efficient driving in the context of urban waste-collection: A Spanish case study,'' *J. Cleaner Prod.*, vol. 289, Mar. 2021, Art. no. 125831, doi: 10.1016/j.jclepro.2021.125831.
[40] R. Gouribhatla and S. S. Pulugurtha, ''Drivers' behavior when driving vehicles with or without advanced driver assistance systems: A driver simulator-based study,'' *Transp. Res. Interdiscipl. Perspect.*, vol. 13, Mar. 2022, Art. no. 100545, doi: 10.1016/j.trip.2022.100545.
[41] M. N. Tahir, M. Katz, and M. S. Saud, ''Implementation of a cooperative intelligent transport system utilizing weather and road observation data,'' in *Autonomous and Connected Heavy Vehicle Technology: Intelligent Data-Centric Systems*. New York, NY, USA: Academic, 2022, ch. 14, pp. 264–284, doi: 10.1016/B978-0-323-90592-3.00016-1.
[42] F. Rempe, P. Franeck, and K. Bogenberger, ''On the estimation of traffic speeds with deep convolutional neural networks given probe data,'' *Transp. Res. C, Emerg. Technol.*, vol. 134, Jan. 2022, Art. no. 103448, doi: 10.1016/j.trc.2021.103448.

DAVID ESCUÍN is currently pursuing the Ph.D. degree in mechanical engineering and computer engineering with the University of Zaragoza.

He is also working as a Technical Project Manager with the Digital Technologies Department, Instituto Tecnológico de Aragón (ITAINNOVA). He drives digital innovation in the supply chain by deploying software solutions in logistics and transportation within the lines of data analytics, optimization algorithms, and in the design of blockchain architecture-based solutions. He is a AWS Certified Cloud Practitioner and works as a Professor with the Universidad Internacional de La Rioja (UNIR) in the Master of Software Engineering and Computer Systems and the Master of Development and Operations (DevOps). He collaborates actively in national and European research and innovation projects with several articles published in specialized journals.

LORENA POLO is currently pursuing the Ph.D. degree in mechanical engineering and industrial engineering with the University of Zaragoza.

She is also a Master in Supply Chain Management and Technologies at Ramon Llull University. In 2016, she won the CEL-University finalist award for her doctoral thesis ''Replenishment models with demand forecasting considerations and supply chain collaboration.'' She also works as a Technology Researcher with the Digital Technologies Department, Aragon Institute of Technology (www.itainnova.es), where she takes part in international, national, and regional research and innovation projects. She also works as a Professor with the Universidad Internacional de la Rioja (UNIR) in the Master of Logistics and the Universitat Oberta de Catalunya (UOC) in the Master of Logistics and in the Master of Data Science. She has also collaborated with the European Commission in the processes of proposal evaluation and projects review and has articles published in specialized journals. Her main research interests include inventory management, production planning, and demand forecasting, and she also has experience in modeling and optimization of logistics processes, supply chain management, and ICT for logistics.

DAVID CIPRÉS received the Ph.D. degree in industrial engineering from the University of Zaragoza. He has been working at the Aragon Institute of Technology (ITAINNOVA) for more than 20 years on projects related to the application of virtual simulation models in industrial and logistic processes. He is currently an Associate Professor with the Department of Business Management and Organization, University of Zaragoza. He is also the Co-ordinator of the Group of Digitization and Automation of Logistics Processes, LOGISTOP Association, where he leads the working group of digital twin of this association. He is also responsible for the research line of the IoT, blockchain, and data analytics applied to industrial and logistic processes. He has participated in many national and European logistics research and development projects.

CARLOS MILLÁN is currently pursuing the Ph.D. degree in industrial engineering with the University of Zaragoza.

He has been working as an Assistant Director at the Instituto Tecnológico de Aragón (ITAINNOVA), since 2019. Previously, he had been working for more than 20 years, leading logistics and transportation research and development projects within the line of data analytics, optimization algorithms, and in the design of blockchain architecture-based solutions. He has published different articles in specialized journals on transportation optimization within cities, long-haul transportation, and integrated supply chain and transportation issues. He collaborates actively in national and European research and innovation projects with different transportation and logistics companies, with special participation in the European Technology Platform ALICE and with the Spanish mirror, LOGISTOP.

JORGE CARCAS has been working with Grupo Sese Logistico for 15 years in the transport IT area. He is currently a Senior Technician in computer applications development. He is also a Manager of IT transport. In these 15 years, he has participated in all the development projects and I+D projects nationals and internationals.