Extracting Journeys from Truck GPS Traces

Bo Xu, Rohit Gupta, Basel Hashisho, Reinhard Köhn, Sebastian van de Hoef
HERE Technologies
{bo.5.xu,rohit.gupta,basel.hashisho,reinhard.koehn,sebastian.vandehoef}@here.com

ABSTRACT

In this paper we propose a method called START for automatically extracting journeys from truck GPS traces. Each journey represents a movement of a truck from a task place (e.g., a cargo station) to another task place, possibly with multiple stops in between for non-task purposes (resting, eating, refueling, etc.). The START method begins with detecting stops. Then it classifies each detected stop as a task stop or a non-task stop. For stop classification, START utilizes a novel feature called route-detour and combines it with a set of temporal features and place category features. The method further clusters detected stops and applies a majority voting to consolidate the classification of the member stops in each cluster. We evaluate START using real-world data and compare it with a baseline method that uses only temporal and place category features. The results show that START achieves a better stop classification accuracy than the baseline method. Finally, we demonstrate the utility of START via using the journeys extracted by START to build and evaluate an ML-based ETA prediction model.

CCS CONCEPTS

• Information systems→Geographic information systems; Location based services; Clustering; • Computing methodologies→Classification and regression trees

KEYWORDS

GPS traces, stop classification, GIS, routing, ETA

ACM Reference format:

Bo Xu, Rohit Gupta, Basel Hashisho, Reinhard Köhn, and Sebastian van de Hoef. 2022. Extracting Journeys from Truck GPS Traces. In *Proceedings of 15th International Workshop on Computational Transportation Science (IWCTS'22). ACM, Seattle, WA, USA, 10 pages.*

IWCTS '22, November 1, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9539-7/22/11…$15.00
https://doi.org/10.1145/3557991.3567797

https://doi.org/10.1145/3557991.3567797

1 Introduction

Providing accurate Estimated Time of Arrival (ETA) is crucial for many fleet management and logistical operations. In recent years there have been extensive studies that employ machine learning to improve ETA accuracy (see e.g., [1, 2, 3, 12]). However, few of these studies are dedicated to the truck mode. To build machine learning models for truck ETA, data about historical journeys are usually required, where each journey starts with a *task stop* (e.g., loading from a warehouse) and ends with another task stop (e.g., unloading to another warehouse). A truck may perform multiple *rest stops* in between for the purpose of resting, refueling, eating, and sleeping (see Fig. 1.1).

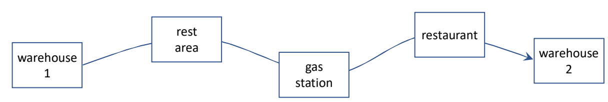


Fig. 1.1: A truck journey starts with a task stop and ends with another and may perform multiple rest stops in between.

In the past, journey information could only be recorded by manual logging. The widespread usage of Global Positioning System (GPS) (and other GNSS[1] type systems) nowadays makes it an attractive source of journey data. However, utilizing this source requires extracting those journeys from the raw GPS data. Extracting journeys can be translated into the problem of detecting and classifying stops from GPS traces. Once we know where a vehicle stops and what reason it stops for, journeys can be straightforwardly constructed.

A stop can be detected by identifying a sequence of GPS points that are stationary or move very slowly in a small radius (see e.g., [4]). A more challenging problem is to identify the purpose of a stop. To this end, a few methods have been developed to classify stops based on temporal features such as stop arrival times and stop durations [4] as well as points of interest (POIs) nearby [5].

In this paper we introduce a novel feature, called *route-detour*, to help classifying stops into task stops and rest stops, and thus to facilitate journey extraction. The intuition behind route-detour is

[1] Global Navigation Satellite System.

as follows: a truck usually only performs rest stops at locations that are on the way to its destination; if a stop requires a significant detour, then this stop is likely to be a task stop. More specifically, let A, B, C be three consecutive stops of a truck. The route detour feature of stop B is defined to be the cost difference between the route from A to C directly and the route from A to C via B. Typical cost measures include travel distance, travel time, etc. The higher the cost difference, the more likely B is a task stop than a rest stop. This idea is illustrated by Fig. 1.2, where B_1 is on the way from A to C, whereas B_2 requires significant detour. Intuitively, B_1 is likely to be a rest stop whereas B_2 task.

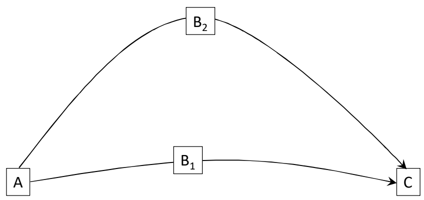


Fig. 1.2: Intuition behind the route detour feature. B_1 is on the way from A to C, whereas B_2 requires significant detour. Thus, B_1 is likely to be a rest stop whereas B_2 task.

We combine the route-detour feature with a set of temporal features and place category features for stop classification using machine learning. Furthermore, we introduce an arbitration procedure to revise the classification of a stop based on the classifications of other stops close by. The arbitration procedure reduces the impact of wrong classifications of individual stops and makes the final classification more accurate.

We call our journey extraction method *STop classification by Arbitration and Route-deTour* (START). In this paper we evaluate START using real-world truck GPS traces and show its advantage over a method that uses only temporal and place category features. Furthermore, we utilize the journeys extracted by START to build and evaluate an ML-based ETA prediction model. In summary, our paper makes the following contributions:

1. Introducing a novel feature called route-detour for classifying truck stops into task stops and rest stops.
2. Combining the route-detour feature with a set of temporal and place category features using machine learning.
3. Introducing an arbitration procedure to consolidate the classifications of individual stops.
4. Evaluating the benefit of START using real-world truck GPS traces.
5. Demonstrating the utility of START by using it to build and evaluate an ML-based ETA prediction model.

The rest of this paper is organized as follows. In Section 2 we discuss relevant work. In Section 3 we give definitions and the problem statement. In Section 4 we present the overall structure of START and the dataset used for its evaluation. In Section 5 we describe and evaluate START in detail. In Section 6 we utilize the journeys extracted by START to build and evaluate an ML-based ETA prediction model. In Section 7 we conclude the paper and discuss future work.

2 Relevant Work

In the literature, researchers have explored various features for classifying stops detected from GPS traces. The authors of [4] utilize temporal features including stop arrival times and stop durations. Their method assumes that each stop place is visited a sufficiently large number of times in the analyzed GPS traces such that a histogram of stop arrival times and a histogram of stop durations can be constructed for each stop place. The two histograms are concatenated to form a vector in an (A+D) dimensional space, where A is the number of bins in the stop arrival time histogram and D is the number bins in the stop duration histogram. The method then categorizes the vectors of different stops by K-means clustering to obtain K representative vectors. Finally, these representative vectors are manually matched to a few functional types (e.g., eating, refueling, resting) based on their temporal characteristics. One disadvantage of this method is that it requires that the GPS traces include a fair number of visits to each stop place in order to build histograms. With the dataset we use, the average number of visits to a stop place is 4, which is insufficient for the method presented in [4].

The methods proposed in [5] and [6] utilize both temporal features and place category features. For place category features they look at the types of POIs near a stop, namely, whether near the stop there are places related to paid work, daily shopping, recreation, service, and so on. The method in [6] further considers the transition probabilities between various stop types. There are two limitations with these methods. First, a stop may have various POI types nearby, in which case it is difficult to associate the stop with the correct type. Second, many POIs can be both a task stop or a rest stop, depending on the purpose of visit. For example, a gas station is typically a rest stop; but it is a task stop when a truck delivers gasoline to it. The route-detour feature introduced in our START method helps solving ambiguity in such cases.

The arbitration procedure in START is similar to that in K-nearest neighbor (KNN) classification [14]. The difference is that the arbitration procedure in KNN classification is used for classifying a new instance whereas the arbitration procedure in START is used for consolidating existing classifications.

Finally, there have been extensive studies on ETA prediction (see e.g., [1, 2, 3, 12]), which use historical trip data to build ETA models. However, these studies are not dedicated to truck ETAs, and therefore in their case a trip can essentially start or end at any point in a road network. Thus, there is no need to distinguish between task stops and rest stops. For truck ETA prediction, on the other hand, it is important for the historical trip data to contain journeys rather than arbitrary trips so that movement patterns specific to trucks, such as the resting behaviors, can be reflected in ETA models. This is a primary motivation of START.

3 Definitions and Problem Statement

We introduce the following definitions in order to formulate a formal problem statement.

Definition 1: A *GPS trace T* is a sequence $<(x_1, y_1, t_1), (x_2, y_2, t_2),\ldots(x_n, y_n, t_n)>$ where $t_1< t_2<\ldots< t_n$, indicating that a truck is at geolocation (x_1, y_1) at time t_1, at geolocation (x_2, y_2) at time t_2, and so on. Each point (x_i, y_i, t_i) is called a *GPS point* and denoted p_i. □

Definition 2: Denote by d_i the great circle distance between geolocations (x_i, y_i) and (x_{i-1}, y_{i-1}). The *speed* of a GPS point (x_i, y_i, t_i), denoted v_i, is equal to $\frac{d_i}{t_i - t_{i-1}}$. □

Definition 3: A *real stop* is an uninterrupted time period during which a truck is stationary. The start (or end) point of this time period is called the *arrival* (or *departure*) time of this real stop. □

Due to random GPS errors, the GPS points sampled during a real stop vary slightly and therefore the derived speed is seldom zero. Instead, the truck usually appears to move very slowly within a small spatial range. Thus, we have the following definition to capture a real stop indicated in a GPS trace.

Definition 4: A *trace-stop* is a longest consecutive subsequence of T such that the speed of each GPS point in the trace-stop is lower than a threshold v_{low} and the total distance traveled in the trace-stop is smaller than a threshold d_{low}. Formally, a subsequence $S = <p_h, p_{h+1}, \ldots, p_{h+k}>$ is a trace-stop if the following three conditions hold:

1. $v_i < v_{low}$ for every $h \le i \le h+k$ (low speed criterion)
2. $d_{h+1} + d_{h+2} + \ldots + d_{h+k} < d_{low}$ (short distance criterion)
3. $v_{h-1} \ge v_{low}$ and $v_{h+k+1} \ge v_{low}$ (longest subsequence criterion)

GPS point p_h is called the *head* of S and p_{h+k} the *tail*. The centroid of the geolocations in S is called the *location* of S. □

In the rest of this paper, unless otherwise specified, the term *stop* refers to a trace stop.

Definition 5: A *place* is a geographic area with certain functionality. □

For example, a rest area is a place, and so is a restaurant, a grocery store, a gas station, and a warehouse.

Definition 6: A stop is called a *task stop* if the truck performs the stop connected to its primary purpose of transporting goods such as loading/unloading cargo. A stop is called a *rest stop* otherwise, for instance, for the driver to rest. □

Warehouses are typical places where task stops occur. Rest areas, restaurants, gas stations, and hotels are typical places where rest stops occur.

Definition 7: A *journey* is a subsequence of T that starts at the tail of a task stop and ends at the head of a task stop, and all the stops in between if any are rest stops.

Fig. 3.1 shows a GPS trace with 18 GPS points $p_1, \ldots, p_{18}$. There are 4 stops $<p_1, p_2, p_3, p_4>$, $<p_7, p_8, p_9>$, $<p_{11}, p_{12}, p_{13}>$, and $<p_{16}, p_{17}, p_{18}>$. $<p_1, p_2, p_3, p_4>$ and $<p_{11}, p_{12}, p_{13}>$ are task stops, performed at place A and place C respectively. $<p_7, p_8, p_9>$ and $<p_{16}, p_{17}, p_{18}>$ are rest stops, both performed at place B. The GPS trace in Fig. 3.1 has one journey: $<p_4, \ldots, p_{11}>$. The journey goes from place A to place C, with a rest stop performed at place B.

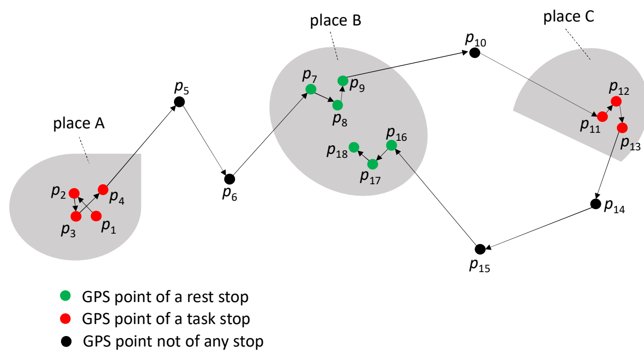


Fig. 3.1: An example GPS trace.

Problem Statement: Given a set of GPS traces, identify all the journeys for each trace.

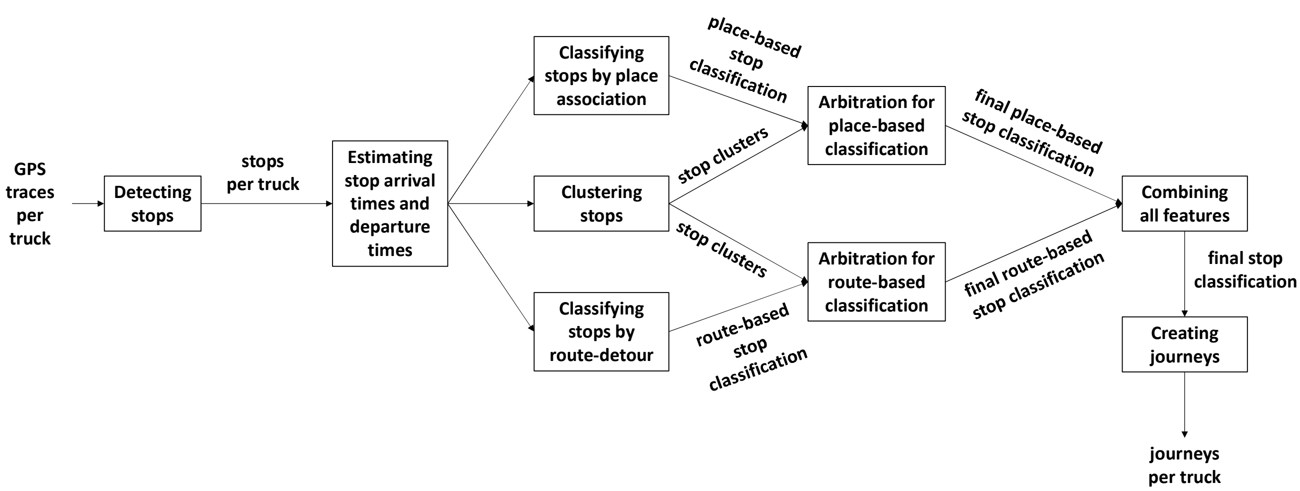


Fig. 4.1: The overall structure of START

4 Overall Structure and Dataset Used

4.1 Overall Structure

The overall structure of START is shown in Fig. 4.1. As the figure shows, START takes as input a set of GPS traces and process them in the following steps.

1. Detect stops for each GPS trace.
2. For each detected stop, estimate the stop arrival time and the stop departure time.
3. Classify each stop based on place association and route-detour, respectively. As a result, each stop is assigned a place-based classification and a route-based classification.
4. Cluster all the stops such that each cluster contains stops that are likely to belong to a common place.
5. Within each cluster, finalize the place-based classification for all member stops by a majority voting. Apply the same procedure for route-based classification.
6. Classify each stop as a task stop or rest stop based on multiple features, including place-based classification, route-based classification, stop arrival time, stop departure time, and stop duration.
7. Create journeys for each GPS trace based on stop classification results.

4.2 Dataset Used

The dataset used in this paper consists of about 3000 GPS traces in multiple European countries. Each trace records the movement of a truck for consecutive 31 days. The GPS sample interval is approximately 10 minutes. There are totally around 15 million GPS points.

5 Detailed Description and Evaluation of START

In this section we elaborate the steps outlined in subection 4.1. For each step we also present related evaluation results.

5.1 Stop Detection

Stops are detected for each trace according to Definition 4 defined in Section 3. For the evaluation in this paper we set the speed threshold v_{low}=1 km/h and the distance threshold d_{low}=1 km. Observe that due to the 10-minute sample interval in the dataset, this step may miss stops that are shorter than 10 minutes.

5.2 Estimating Stop Arrival/Departure Time

Since a GPS trace is a sampling of the real movement of a truck, the start/end time of a detected stop can be different than the arrival/departure time of the real stop. To see this, consider Fig. 5.1, in which the red solid circles are GPS points that constitute a stop <p_h, p_{h+1}, p_{h+2}>. Point p_h is the head, with t_h=8:00am. The point preceding p_h is p_{h-1}, with t_{h-1}=7:50am. But the real stop starts at 7:56am (at location q). However, due to the 10-minute sample interval, q is missed, incurring a 4-minute error if we estimate the stop arrival time to be p_h. Similarly, there can be an error if we estimate the stop departure time to be t_{h+2}.

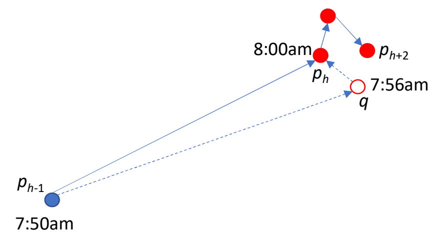


Figure 5.1: The arrival time reflected by the head of a stop may be unreliable

In the Fig. 5.1 example, a more reasonable estimate of the stop arrival time is 7:50am plus the typical travel time from p_{h-1} to p_h. In general, let p_h be the head of a stop and p_{h-1} be p_i's predecessor. We estimate the stop arrival time to be $t_{h-1}+T$ where T is the typical truck travel time from (x_{h-1}, y_{h-1}) to (x_h, y_h) if a truck departs at t_{h-1}. Symmetrically for the estimation of the stop departure time. In this paper, we obtain T via the HERE Routing API [7]. We denote the estimated stop arrival time of a stop S by $t_{arr}(S)$ and the estimated stop departure time by $t_{dep}(S)$, and define $t_{dep}(S)-t_{arr}(S)$ to be the *stop duration* of S.

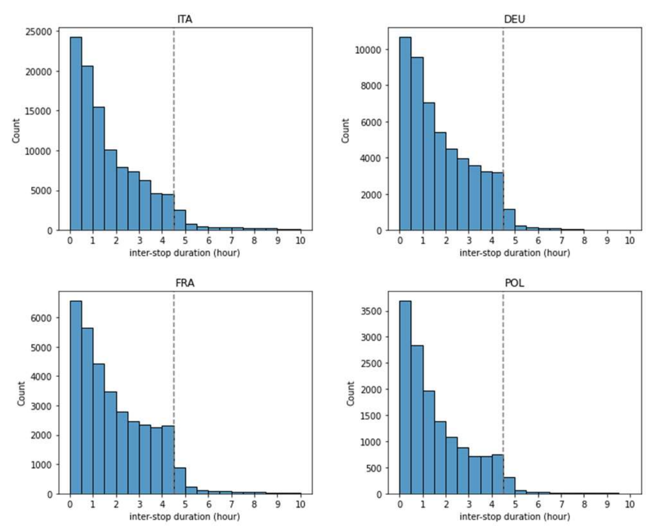


Figure 5.2: Histogram of inter-stop duration for four EU countries

As a sanity check of the presented stop detection and stop arrival/departure time estimation methods, we analyze the time durations between each pair of consecutive stops, which are referred to as *inter-stop durations*. Specifically, given two consecutive stops S_1 and S_2 of a trace, the inter-stop duration is equal to $t_{arr}(S_2)-t_{dep}(S_1)$. Presumably, an inter-stop duration represents a period of continuous driving. Fig. 5.2 shows the histogram of inter-stop durations within four European Union (EU) countries. From all the plots in Fig. 5.2, we see a drastic drop of count at the 4.5-hour inter-stop duration (marked by a vertical dashed line). This is consistent with the EU regulation that a driver must take a rest after driving for 4.5 hours [11]. The durations beyond 4.5 hours could be due to the algorithm missing short stops (see subsection 5.1), violations of the regulation, and stop detection errors.

5.3 Stop Classification

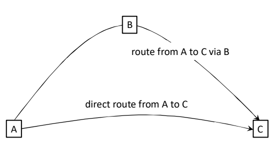
5.3.1 Stop Classification by Route-Detour

Stop classification by route-detour builds on the intuition that drivers tend to take rest at places that are on the way to their next stop. If they detour a lot to perform a stop, then that stop is likely to be a task stop. Let *A*, *B*, *C* be three consecutive stops traveled by a driver (see Fig. 5.3(a)). Denote by $|\overline{ABC}|$ the length of a typical route from *A* to *C* via *B* and by $|\overline{AC}|$ the length of the typical direct route from *A* to *C*. $|\overline{ABC}| - |\overline{AC}|$ is referred to as the *route-detour*. We use HERE Routing API [7] to compute the typical routes.

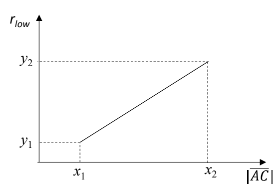
If the route-detour is longer than a certain threshold r_{low}, we classify stop *B* as "detour-task"; otherwise "detour-rest". The threshold r_{low} is dynamically determined based on the value of $|\overline{AC}|$ as shown in Fig. 5.3 (b):

$$r_{low} = \begin{cases} y_1 & |\overline{AC}| < x_1 \\ y_1 + \dfrac{y_2 - y_1}{x_2 - x_1}(|\overline{AC}| - x_1) & x_1 \le |\overline{AC}| \le x_2 \\ y_2 & |\overline{AC}| > x_2 \end{cases}$$

where x_1, x_2, y_1, and y_2 are configurable parameters of the algorithm. For the evaluation in this paper, we set x_1=10km, x_2=400km, y_1=2km, y_2=8km.



(a) A, B, C are consecutive stops.



(b) Dynamic thresholding

Fig. 5.3: Route-based stop classification

5.3.2 Stop Classification by Place Association

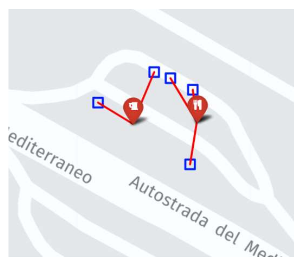
This step searches for places of relevant categories near a stop and classifies the stop based on the search results. We use the HERE Geocoding & Search API (see [8]) for place searching. The relevant categories include: "rest-area", "fueling-station", "parking", "ferry", "hotel-motel", "eat-drinking", and "cargo-transportation". The category "cargo-transportation" refers to a facility that handles some aspect of the transportation of cargo freight, such as a cargo center, a courier station, a loading dock, or a delivery entrance. Thus, "cargo-transportation" indicates a task stop; all the other relevant categories indicate a rest stop.

A stop may belong to a place that is of a category that is not included in the relevant categories. For example, a supermarket is of categories "grocery" and "pharmacy". In this case, the stop is classified as "place-undetermined".

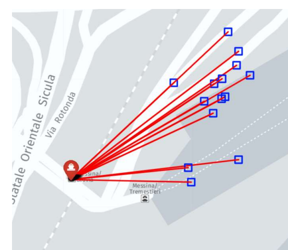
The complete procedure of place-based stop classification is as follows.

1. Search "rest-area", "fueling-station", "hotel-motel", "eta-drinking", "cargo-transportation" places within 150 meters from the stop location; search "parking", "ferry" places within 200 meters from the stop location. "parking" and "ferry" use a larger search radius because they usually occupy a larger space.
2. If no place is found at step 1, then classify the stop as "place-undetermined".
3. If one or more places are found at step 1, find the place that is closest to the stop. If this place is "cargo-transportation", then classify the stop as "place-task"; otherwise classify it as "place-rest".

Fig. 5.4 shows a few examples of place association.



(a) Stops (blue squares) in a rest area associated with a gas station and a restaurant



(b) Stops (blue squares) associated with a ferry terminal

Fig. 5.4: Examples of place association

Fig. 5.5 shows the stop history of a truck. Each row shows the timeline of a day (24 hours) of March 2022. Each bar represents the duration of one stop, colored by the place category they are associated with. The gaps between bars represent driving. The truck appears to have a regular pattern that it drives during daytime and rests during nighttime. Furthermore, it rests a lot during weekends (thick lines) and never drives for more than 9 hours following the EU working time regulations [11]. Fig. 5.5 also shows that many stops are place-undetermined. This does not necessarily mean that these stops do not belong to any places. More likely it means that they belong to a category that is not included in the relevant categories. For these stops, we let the classification be decided by the route-detour feature and temporal features. For details see Section 5.3.4.

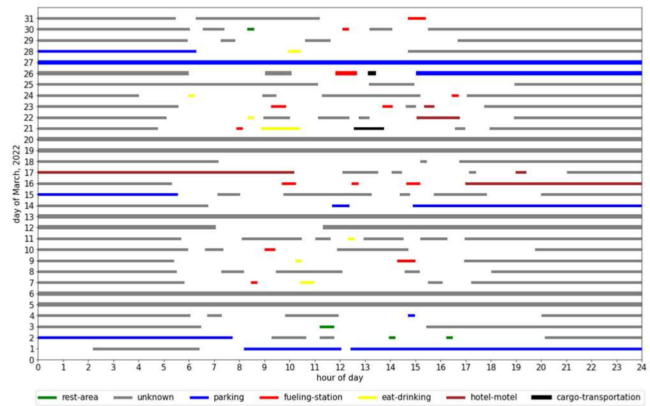


Fig. 5.5: Stop history of a truck constructed by place association

5.3.3 Clustering and Arbitration

To cluster all stops and places, we apply the OPTICS algorithm proposed in [9] and implemented in [10]. For each cluster, we conduct a plurality-voting based on the route-based classification of each member stop, and then assign the voting result to all the member stops of the cluster. For example, if the majority of the stops in the cluster are detour-task, then all the stops in the cluster are assigned as detour-task. Similarly, we conduct a plurality-voting based on place-based classification and assign the voting result to all the member stops of the cluster. As a result, each stop is assigned a final route-based classification and a final place-based classification.

Fig. 5.6 shows the cumulative distribution of cluster size. From the figure it can be seen that 90% of the clusters have no more than 5 stops, and only 5% of the clusters have 10 or more stops. Such cluster sizes are not sufficient for building temporal histograms as proposed in [4] (see Section 2 for a discussion of [4]). On the other hand, 15% of the clusters have 3 or more stops; these clusters can benefit from arbitration.

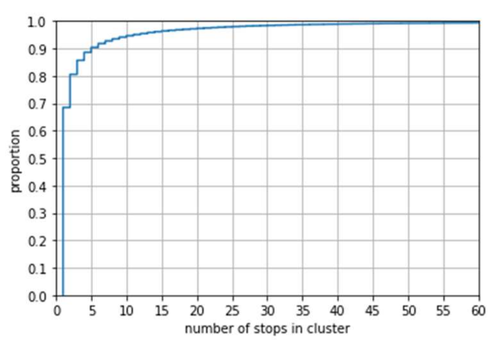


Fig. 5.6: Cumulative distribution of cluster size

Fig. 5.7 shows a stop cluster at a cargo. Each rectangle is a member stop colored by its stop duration. From this figure it can be seen that most stops lasted no more than 2 hours, which is expected for delivering or picking up at a cargo station. On the other hand, at the lower right corner many stops lasted for 12 hours or more. It appears that drivers took a long rest during these stops. This observation suggests that, unlike one would expect, a long stop duration by itself is not always a sufficient indicator of a rest stop.

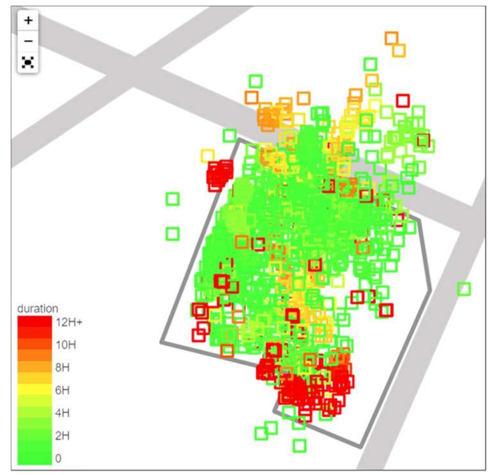


Fig. 5.7: A stop cluster at a cargo station. The gray polygon is the cargo station. Each rectangle is a member stop. Notice that each member stop per se is an aggregation of multiple GPS points which are not shown in the figure.

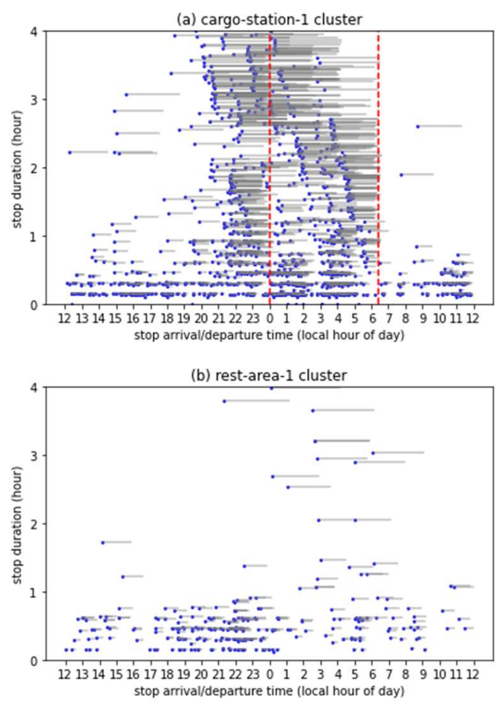


Fig. 5.8: Stop duration vs stop arrival time/departure for two clusters.

Fig. 5.8 shows stop duration vs stop arrival/departure time for a cargo station cluster and a rest area cluster, respectively. Each blue dot in a plot corresponds to one stop. Each horizontal gray line starts at the stop arrival time and ends at the stop departure time. From subplot (a), it appears that cargo-station-1 has a few fixed departure times such as midnight and 6am, as highlighted by the red dashed lines. The rest-area (subplot (b)), understandably, does not have fixed departure schedules. The study in subsection 5.3.4 shows that indeed stop departure time is a useful feature for identifying task stops.

5.3.4 Combined Stop Classification

We take an ML approach to combine the following features for stop classification.

Table 5.1: Features used for combined classification

Feature	Description
route-based classification	stop classification determined by route-detour as discussed in subsection 5.3.1
place-based classification	stop classification determined by place association as discussed in subsection 5.3.2
stop arrival hour	hour of day of the stop arrival time, ranging from 0 to 23
stop departure hour	hour of day of the stop departure time, ranging from 0 to 23
stop duration	stop duration in hours
overnight	whether the stop period crosses midnight

To prepare a dataset for ML training and test, we manually label 506 stops by human inspection of satellite images and street views, out of which 314 are labeled as task stops and 192 rest. We split the labeled stops into a training set of 288 stops and a test set of 218 stops. Using the J48 implementation [13] of the C4.5 decision tree algorithm [15] we machined-learn the classification rules shown in Fig. 5.9.

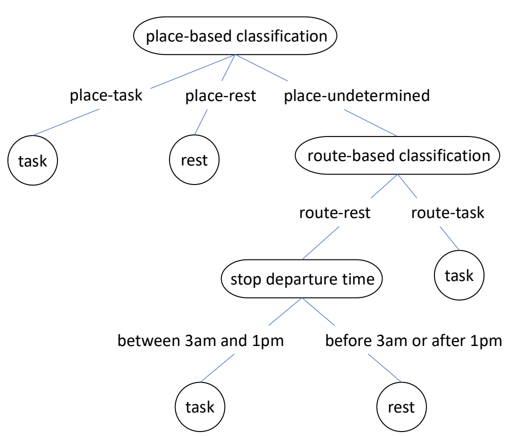


Fig. 5.9: Stop classification rules learned with the C4.5 decision tree algorithm

The rules in Fig. 5.9 suggest the following:

1. place-classification is relatively reliable when an association to a relevant category can be made;
2. when an association cannot be made, route-classification is most useful, followed by stop departure time; and
3. stop arrival time and overnight features are the least useful for classification.

5.3.5 Stop Classification Performance Evaluation

We apply the rules learned in the previous subsection to the test set and obtain an accuracy of 0.79. We also compare START with a method that uses all the features in Table 5.1 except the route-detour feature. Since this method uses only temporal and place features, we call it *Temporal+Place*. Temporal+Place reaches an accuracy of 0.74. Fig. 5.10 shows the confusion matrices of the two methods. From the figure it can be seen that START classifies many more rest stops correctly compared with Temporal+Place. Thus, the accuracy advantage of START mainly comes from the fact that it performs much better than Temporal+Place on classifying rest stops.

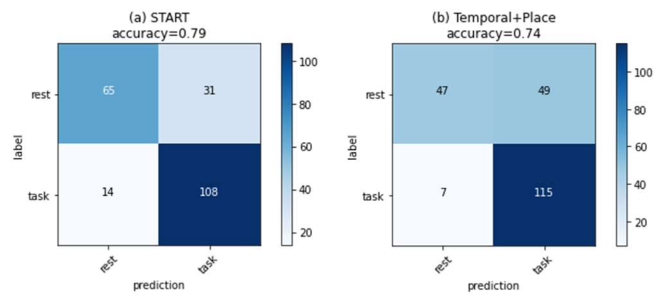


Fig. 5.10: Confusion matrices for START and Temporal+Place

5.4 Journey Creation

Once stops are classified, journeys can be straightforwardly constructed according to the journey definition (see Definition 7 in Section 3). Let J=<p_i, p_{i+1},…, p_j> be a journey, S_i be the stop that p_i belongs to, and S_j be the stop that p_j belongs to[2]. S_i and S_j are referred to as the *source stop* and the *destination stop* of J, respectively. $t_{dep}(S_i)$ and $t_{arr}(S_j)$ are referred to as the *journey departure time* and the *journey arrival time* of J, respectively. $t_{arr}(S_j)-t_{dep}(S_i)$ is referred to as the *journey duration* of J. (x_i, y_i) and (x_j, y_j) are referred to as the *source location* and the *destination location* of J, respectively.

We apply the rules in Fig. 5.9 to all the detected stops. Based on the stop classification results, we create about 162,000 journeys out of the 3000 GPS traces. The cumulative distribution of journey durations is shown in Fig. 5.11. The figure shows that 70% of the journeys are shorter than 4 hours and 5% are longer than 24 hours.

[2] Notice that each of p_i and p_j must belong to a task stop according to the definition of journey.

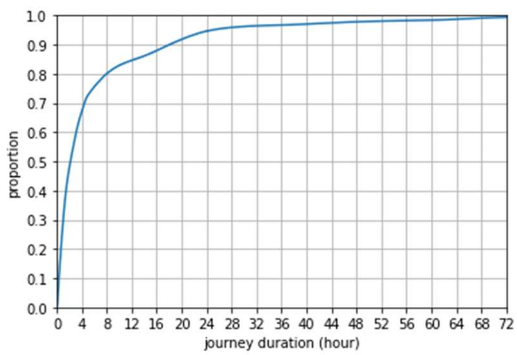


Fig. 5.11: Cumulative distribution of journey durations

6 ML-based ETA Model

In this section we apply the journeys produced by START in Section 5 to build an ML-based model for truck ETA prediction and compare it with two baseline methods.

6.1 Compared ETA Prediction Methods

Given a journey *J*, an *ETA query* asks for an estimated arrival time at the destination location of *J*. Since the estimated arrival time is equal to the journey departure time plus an estimated journey duration, answering an ETA query is equivalent to estimating the journey duration. For this reason, in the rest of this paper we will use the two terms, ETA and estimated journey duration, interchangeably.

We compare the following three ETA prediction methods:

- **Driving Time Prediction (DTP)**. This method predicts ETA by predicting the driving time from the journey source location to the journey destination location without accounting for rest times.
- **Rest-Rule (RR).** This method first uses DTP to predict a driving time. Then it adds upon the driving time a rest time based on an EU rest time regulation. This regulation requires that a truck driver must rest for at least 45 minutes after a driving period of 4.5 hours. Thus, the RR method adds 45 minutes to ETA for every 4.5 hours of the driving time returned by DTP.
- **XGB.** This method builds an XGBoost [16] model based on journeys extracted by START. The model uses the following features of a journey:
  - o DTP-ETA: the driving time in hour predicted by DTP;
  - o start-hour: the hour of day of the journey departure time, ranging from 0 to 23;
  - o source-duration: the duration in seconds of the source stop of the journey;
  - o day of week: the day of week of the journey departure time, ranging from 0 (Monday) to 6 (Sunday).

To distinguish from the features used for stop classifications, we call the above features *ETA-features*.

6.2 Performance Metric

We use mean absolute percentage error (MAPE) as the performance metric. MAPE is defined to be $\frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{\hat{y}_i-y_i}{y_i}\right|$ where y_i and $\hat{y}_i$ are the true ETA and the predicted ETA respectively for the i-th ETA query and *N* is the number of ETA queries participating in MAPE evaluation.

6.3 Experiment Setup

We apply the following criteria to choose the journeys that are to be used for ETA evaluation.

1. If rest stop durations are subtracted from the journey duration, then the absolute percentage error of DTP is less than 50%. The objective of this criterion is to isolate the two error sources of truck ETA, namely the error of driving time prediction and the error of rest time prediction. The criterion makes the evaluation focus on the rest time prediction error which is particularly important to truck ETA.
2. The journey departure time and the journey arrival time fall into a single calendar day. This criterion removes excessively long journeys that are harder to predict. We leave these journeys to future work.
3. The duration of the source stop is longer than 11 hours. This criterion is to make sure that the driver has taken sufficient rest before starting a journey so that the new rests are independent of the driving time and the rest time before the journey. The 11-hour threshold is based on the EU regulation that a driver must rest for at least 11 consecutive hours per day [11].

Applying the above criteria, we obtain a dataset of ~23,000 journeys for ETA evaluation. We call this dataset the *ETA-set*. We apply an XGB regression model on the ETA-set with repeated K-Fold cross validation. 2/3 of the journeys in the ETA-set are used for training and 1/3 for testing.

6.3 Experiment Results

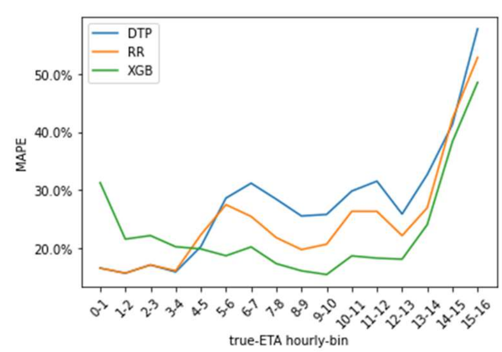


Fig. 6.1: MAPE comparison for each true-ETA hourly-bin

Fig. 6.1 shows the MAPE comparison for each hourly-bin of true ETA. The 0-1 hourly-bin contains the journeys with true ETA lower than 1 hour, the 1-2 hourly-bin contains the journeys with true ETA greater than 1 hour and smaller than 2 hours, and so on. From the figure we can make the following observations:

1. RR outperforms DTP for journeys that are longer than 5 hours. This shows the benefit of RR adding rest times. RR and DTP perform very similarly for journeys that are shorter than 5 hours. This is because RR only adds rest times for journeys estimated to be longer than 4.5 hours.
2. XGB outperforms DTP and RR for journeys longer than 5 hours but underperforms them for journeys shorter than 5 hours.

The reason why XGB performs poorly for relatively short journeys is possibly that the ETA features we have right now might be insufficient for predicting with higher accuracy.

This can be especially true for the 0-1 hourly-bin since this bin contains the largest number of journeys (see Fig. 6.2). A more complex model (resulting from richer features) can possibly perform better on this bin and consequently bring down the MAPE error.

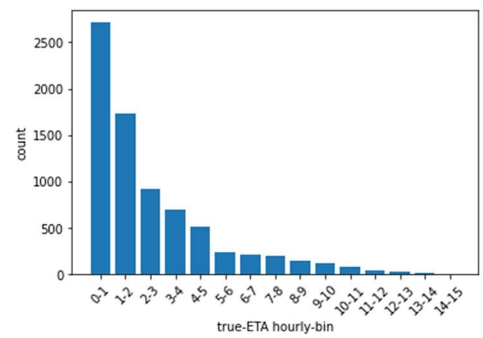
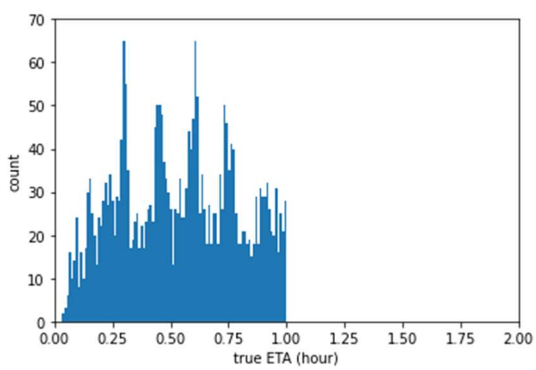
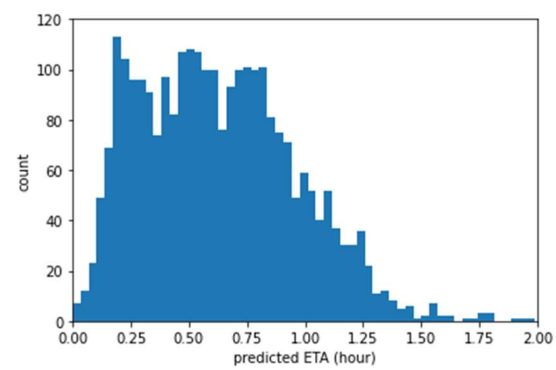


Fig. 6.2: Number of journeys in different true-ETA hourly-bins

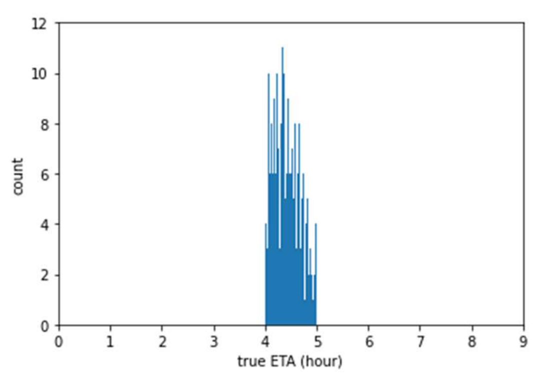
In order to understand the performance on the shorter journeys we examine the distribution of the true ETAs and the predicted ETAs (see Fig. 6.3).



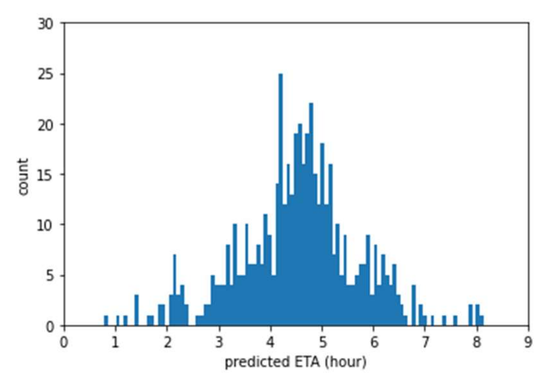
(a) Distribution of true ETAs for the 0-1 true-ETA hourly-bin



(b) Distribution of predicted ETAs for the 0-1 true-ETA hourly-bin



(c) Distribution of true ETAs for the 4-5 true-ETA hourly-bin



(d) Distribution of predicted ETAs for the 4-5 true-ETA hourly-bin

Fig. 6.3: Distribution of true ETAs and predicted ETAs for different hourly-bins.

As we see from subplots (a) and (b) in Fig. 6.3, the true ETAs between 0 and 1 hour seem to have distinct peaks which are predicted to some extent correctly by XGB. Now compare these subplots with subplots (c) and (d) where XGB does better on MAPE (true ETAs between 4 and 5 hours respectively). One thing stands out – For longer journeys XGB is able to identify a distribution that has a single peak which lies closer to the mid-point of the true ETAs. Nevertheless, the concrete reason why XGB performs not well for short journeys needs more investigation.

7 Conclusions and Future Work

In this paper we propose a method called START for automatically extracting journey information from truck GPS traces. The design of START and our experimental analysis suggest some notable strengths of our method:

- START achieves more accurate stop classification compared with a method that uses only temporal features and place features, thanks to the addition of the route-detour feature.
- By clustering and arbitration, START utilizes repeated stop visits for more accurate stop classification.
- On the other hand, the operation of START does not rely on repeated visits unlike the method proposed in [4]. This makes START less sensitive to data density in terms of repeated visits.
- START increases journey duration accuracy by rectifying the stop arrival/departure time errors caused by sparse GPS sampling.
- START enables development and evaluation of ML-based truck ETA models.

For future work, there are several design options worth exploring. For example, instead of having a route-based classifier and a place-based classifier separately and combining their outputs by another classifier, an alternative is to combine route-based features and place-based features directly and build a single classifier that takes all these features as input. This alternative would also eliminate the need for the thresholding of detour in route-based classification. Furthermore, we will study how to improve the XGB-based ETA model and make it perform better on short journeys. We will also explore other ML-based ETA models with richer ETA features.

REFERENCES

[1] Derrow-Pinion, J. She, D. Wong, O. Lange, T. Hester, L. Perez, M. Nunkesser, S. Lee, X. Guo, B. Wiltshire, P. W. Battaglia, V. Gupta, A. Li, Z. Xu, A. Sanchez-Gonzalez, Y. Li, and P. Velickovic. 2021. Eta prediction with graph neural networks in google maps. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM'21), Oct 2021. https://doi.org/10.1145/3459637.3481916

[2] X. Hu, T. Binaykiya, E. Frank, and O. Cirit. 2022. DeeprETA: An ETA Post-processing System at Scale. arXiv preprintarXiv:2206.02127. https://doi.org/10.48550/arXiv.2206.02127

[3] Xu, D. Jonieta, R. Gupta, A. Soleymani, K. Malm, and R. Köhn. 2022. STAMP: An Approach to ETA Prediction by Spatio-temporal Discretization and Machine Learning. To appear in *Proceedings of the 25th IEEE International Conference on Intelligent Transportation Systems (ITSC'22)*, Macau, China, Oct. 8-12, 2022.

[4] R. Aziz, M. Kedia, S. Dan, S. Basu, S. Sarkar, S. Mitra, and P. Mitra. 2016. Identifying and Characterizing Truck Stops from GPS Data. In: Perner, P. (eds) *Advances in Data Mining, Applications and Theoretical Aspects (ICDM'16)*. Lecture Notes in Computer Science, vol 9728. Springer, Cham. https://doi.org/10.1007/978-3-319-41561-1_13

[5] T. Feng and H. J. Timmermans. 2015. Detecting activity type from GPS traces using spatial and temporal information. *European Journal of Transport and Infrastructure Research*, 15(4). https://doi.org/10.18757/ejtir.2015.15.4.3103

[6] L. Liao, D. Fox, and H. Kautz. 2007. Extracting Places and Activities from GPS Traces Using Hierarchical Conditional Random Fields. *The International Journal of Robotics Research*, 26(1), pp. 119-134. https://doi.org/10.1177/0278364907073775

[7] https://developer.here.com/documentation/routing-api/dev_guide/topics/use-cases/duration-typical-time-of-day.html.

[8] https://developer.here.com/documentation/geocoding-search-api/dev_guide/index.html

[9] M. Ankerst, M. M. Breunig, H. P. Kriegel, J. Sander. 1999. OPTICS: Ordering Points To Identify the Clustering Structure. *ACM SIGMOD International Conference on Management of Data (SIGMOD'99)*, Philadelphia, Pennsylvania, June, 1999, pp. 49-60. https://doi.org/10.1145/304181.304187

[10] https://www.npmjs.com/package/density-clustering

[11] https://europa.eu/youreurope/business/human-resources/transport-sector-workers/road-transportation-workers/index_en.htm

[12] T. Fu and W. Lee. 2019. DeepIST: Deep Image-based Spatio-Temporal Network for Travel Time Estimation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge* Management *(CIKM'19)*, November 2019, pp. 69-78. https://doi.org/10.1145/3357384.3357870

[13] https://weka.sourceforge.io/doc.dev/weka/classifiers/trees/J48.html

[14] Z. Zhang. Introduction to machine learning: k-nearest neighbors. 2016. *Ann Transl Med*. 2016 Jun;4(11):218. https://doi.org/10.21037/atm.2016.03.37

[15] J. R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers. https://doi.org/10.1007/BF00993309

[16] T. Chen, C. Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August, 2016, pp. 785-794. https://doi.org/10.1145/2939672.2939785