Journal of Intelligent and Connected Vehicles
2023, **6**(1): 16–33
https://doi.org/10.26599/JICV.2023.9210002

Research Article

Inferring truck activities using privacy-preserving truck trajectories data

Arnav Choudhry, Sean Qian✉

Carnegie Mellon University, Pittsburgh PA 15213, USA

Received: November 17, 2022; Revised: January 1, 2023; Accepted: January 3, 2023

ABSTRACT: Global Navigation Satellite System (GNSS) data is an inexpensive and ubiquitous source of activity data. Global Positioning System (GPS) is an example of such data. Although there have been several studies about inferring device activity using GPS data from a consumer device, freight GPS data presents unique challenges for example having low and variable frequency, long transmission gaps, and frequent and unpredictable device ID resetting for preserving privacy. This study aims to provide an end-to-end, generic data analytical framework to infer multiple aspects of truck activity such as stops, trips, and tours. We use popular existing methods to construct the data processing pipeline and provide insights into their practical usage. We also propose improved data filters to different aspects of the data processing pipeline to address challenges found in privacy-preserving freight GPS data. We use freight data across four weeks from the greater Philadelphia region with variable transmission frequency ranging from one second to several hours to perform experiments and validate our methods. Our findings indicate that auxiliary information such as land use can be helpful in fine tuning stop inference, but spatio-temporal information contained in timestamped GPS pings is still the most powerful source of false stop identification. We also find that a combination of simple clustering techniques can provide a way to perform fast and reasonable clustering of the same stop.

KEYWORDS: freight activity, GPS processing, stop detection, freight privacy, tour inference

1 Introduction

$12.6 trillion dollars, or 63.55% of the total value, worth of freight in USA is estimated to be moved using trucks in the year 2022 (U.S. Department of Transportation, Bureau of Transportation Statistics, 2022). Trucking also accounts for the highest share of sub-1,000 mile freight movements. Understanding truck freight movements can thus lead to very high impact interventions. Traditionally, truck activity has been studied using logbooks and travel diaries. The increasing deployment of intelligent transportation technologies can provide many advantages over traditional methods such as being inexpensive to process at scale (Choudhry, 2022).

Global Navigation Satellite System (GNSS) data is an inexpensive and ubiquitous source of activity data. Global Positioning System (GPS) is an example of such data. Despite its wide availability, public agencies use it in limited ways (Taghavi et al., 2019). The lack in adoption may be attributed to the confidentiality issues restricting the acquisition of data from commercial vehicles (De Jong et al., 2004). In some cases, the device ID associated with a particular truck may be reset, which can cause long term activity trends to be lost. This can limit the ability of researchers and public agencies to learn about long term trends in truck activity from anonymized data. To address this issue, we propose the use of a device matching algorithm. This algorithm would be able to reconstruct long term activity trends in data by linking together data from different devices that are associated with the same truck, even if the device IDs have been reset. By doing this, the algorithm can provide valuable insights into truck activity without requiring explicit identification information such as registration numbers. As a result, the utility of GPS data for public agencies could be increased.

In addition, issues from hardware or operating conditions in urban environments such as inexact transmission frequency or jitter make the widespread usage of GPS data from multiple source challenging. The combination of these factors leads to problems with applying methods developed for personal mobile GPS to freight GPS trajectories. Using truck GPS data with inaccuracies, such as device resetting, jitter, and variable transmission frequency, in a logistic delivery system that includes both trucks and drones could hinder the coordination between the two modes of transportation and negatively impact the overall efficiency and effectiveness of the system (Qu et al., 2022a). Moreover, improved truck GPS inference could be useful for more accurately modeling the energy consumption of EVs during deliveries, which would allow for better understanding of the energy impact of using EVs in transportation. This information could be valuable in optimizing the use of EVs in delivery operations and assessing their potential for reducing emissions and other environmental impacts (Liu et al., 2021; Qu et al., 2022b).

Several studies on freight GPS data processing focus on individual aspects of the entire pipeline and may not utilize multiple simple approaches to get the result. Moreover, studying the effect of the aforementioned challenges on popular freight GPS processing algorithms is also an open research area. We study the effect of the challenges either created artificially by data providers

✉ Corresponding author.
E-mail: seanqian@cmu.edu

or occurring naturally due to the nature of GPS data collection devices throughout the entire GPS processing pipeline. We then propose methods and provide practical guidance in navigating those issues and provide robust truck activity inference on such real-world GPS datasets. Consequentially, we provide a modular framework of GPS processing which is not predicated on availability of auxiliary information, such as land use, to be useful. If such information is available, it can be used within our framework, if it is not, the rest of it still works.

The rest of the paper is organized as follows. Section 2 provides an overview of existing work in the area of truck activity inference covering the areas of working specifically with anonymized GPS device data and truck activity inference. Section 3 describes the proposed device matching algorithm and the truck activity inference procedure. Section 4 describes the data that is used in this project. Section 5 contains the experiments carried out, and the results of applying the truck activity inference process on the dataset. And finally, Section 6 summarizes the learnings and provides directions further analysis and experimentation.

2 Literature review

2.1 Device matching

To the best of our knowledge, device matching is a novel research area which can help transportation practitioners extend their inference capabilities from GPS data while requiring very minimum change from data vendors to still collect data with the same privacy-preserving principles that they have been using. We employ trajectory and device-based features to perform the matching. This choice was inspired by work on truck activity inference.

2.2 Stop inference

The literature on truck activity inference, particularly stop detection, has focused on various methods including threshold-based techniques and machine learning algorithms. Threshold-based methods often use vehicle speed, stop duration, or acceleration to identify potential stop locations, while machine learning approaches utilize a variety of features to classify stops. While both types of methods have their advantages and limitations, there is a trend towards more generalizable approaches that can be applied universally and are less sensitive to contextual factors such as network topology and device characteristics. In this review, we will examine the existing literature on truck activity inference, highlighting the various methods and techniques that have been proposed and discussing their strengths and weaknesses.

Most of the work on truck activity inference is related to stop detection. Stop detection often involves finding the GPS ping(s) associated with a stop (Table 1), filtering stops to remove false positives (Table 2), merging stops (Table 3), and optionally, inferring the location of the representative stop. Different papers may do one or several of these operations using different methods. The most common method of stop detection is based on using simple thresholds on different aspects of truck operation such as vehicle speed, stop duration, or even acceleration.

Inferred speed thresholds are commonly used to identify potential stop locations (Akter et al., 2018; Aziz et al., 2016; Camargo et al., 2017; Kuppam et al., 2014; Yang et al., 2014; Yang et al., 2022a). The speeds are inferred using distance and time calculation from consecutive GPS pings. Some datasets have access to instantaneous or spot speeds and then the thresholds can be set on spot speeds to find stop locations (Siripirote et al., 2020; Thakur et al., 2015). The closely related are works which utilize a combination of distance and time thresholds to find stops (Gingerich et al., 2016; Chankaew et al., 2018). This is because a threshold based on a combination of distance and time may be argued to create essentially a speed threshold. Another more recent threshold-based technique includes both speed and acceleration (Holguin-Veras et al., 2020).

Speed based thresholds seem to be popular because they directly lend themselves to the idea of a truck being stopped at speed zero. Speed thresholds used in the literature above vary from 3 to 8.5 mph. Thresholds may also be used to filter out stops, through using dwell time thresholds (Akter et al., 2018; Aziz et al., 2016; Camargo et al., 2017; Chankaew et al., 2018; Holguin-Veras et al., 2020; Hwang et al., 2017; Kuppam et al., 2014; Thakur et al., 2015; Yang et al., 2022a; Yang et al., 2022b; You and Ritchie, 2018), or through trip distance thresholds (Chankaew et al., 2018; Thakur et al., 2015). And finally, thresholds may also be used to merge stops together. Luo et al. (2017), Siripirote et al. (2020), and

Table 1 Summary of different approaches on stop detection/identification

Reference	DBSCAN	Threshold	Histogram	KDE	HMM
Luo et al. (2017)	√				
Yang et al. (2014)		√			
Aziz et al. (2016)	√	√			
Hwang et al. (2017)	√				
Thierry et al. (2013)				√	
Gingerich et al. (2016)		√			
Taghavi et al. (2019)					√
Karam et al. (2020)	√				
Siripirote et al. (2020)		√			
Akter et al. (2018)		√			
Camargo et al. (2017)		√			
Holguin-Veras et al. (2020)		√			
Chankaew et al. (2018)		√			
Thakur et al. (2015)		√			
Yang et al. (2022a)		√	√		
Kuppam et al. (2014)		√			

Table 2 Summary of different approaches on stop filtering. This does not include DBSCAN where stops not in a cluster get filtered out in the identification step itself

Reference	Threshold	SVM	Geo-data	Entropy	HMM
Yang et al. (2014)		√			
Aziz et al. (2016)	√				
Hwang et al. (2017)	√				
Gingerich et al. (2016)				√	
Taghavi et al. (2019)					√
You and Ritchie (2018)	√				
Akter et al. (2018)	√				
Camargo et al. (2017)	√				
Holguin-Veras et al. (2020)	√				
Chankaew et al. (2018)	√				
Thakur et al. (2015)	√		√		
Yang et al. (2022b)	√		√		
Yang et al. (2022a)	√		√		
Kuppam et al. (2014)	√				

Note: DBSCAN represents the density-based spatial clustering of applications with noise.

Table 3 Summary of different approaches on stop merging. This does not include DBSCAN where stops in a cluster are considered merged implicitly

Reference	Threshold	Geo-data	Ward linkage
Luo et al. (2017)	√		
You and Ritchie (2018)		√	
Siripirote et al. (2020)	√		
Camargo et al. (2017)	√		
Sharman and Roorda (2011)		√	√

Camargo et al. (2017) use some combination of distance and/or time thresholds to merge identified stopping locations together.

Threshold based methods, as we shall discover in this project, are susceptible to under-fitting the data. That means that a single value of threshold may not work in all situations. Also, since the speeds in most cases are inferred, large gaps in data may make accurate quantification of quantities like speed hard. So the quality of the method may even depend on factors outside of any methodology used in setting the thresholds. Setting the threshold itself may be an art and is subjective to the specific combination of data frequency and network topology.

Some interesting anecdotes from setting thresholds can be found in the literature. Camargo et al. (2017) and Akter et al. (2018) use threshold values without and change, directly citing previous literature. Yang et al. (2014) and Kuppam et al. (2014) posit that the speed threshold chosen can solve contextual problems such as wind canyons and GPS jitter. Aziz et al. (2016) sets the speed threshold on the basis of the device transmission frequency. More recently there is a shift to create more general methods of determining speed thresholds such as in Yang et al. (2022a), a histogram of average speeds between GPS events is used to determine the value of the speed threshold. This paper seeks to continue work in that direction by proposing metrics of interest which can be applied universally irrespective of network topology or device characteristics.

The next set of algorithms that are used to infer stops belong to a general class of machine learning algorithms. These have the benefit of being more generally applicable such as having the ability to provide dynamic decisions based on a variety of factors as opposed to a simple threshold stacking approach (Yang et al., 2014; Taghavi et al., 2019). However, the flexibility comes at the cost of complexity. A more complex model yields more parameters that need to be tuned, sometimes manually. The parameters may require large numbers of labeled data points in order to be tuned and validated with a reasonable degree of confidence.

The most commonly used machine learning algorithm is the density-based clustering algorithm, DBSCAN (Ester et al., 1996). Studies by Luo et al. (2017), Hwang et al. (2017), Aziz et al. (2016), and Karam et al. (2020) use DBSCAN to designate areas of higher density of potential stops as actual stops. Due to its procedure, the cluster of stops found by DBSCAN can easily be merged into a singular stop. While DBSCAN is extremely good at finding clusters of points of unique shapes, it cannot find clusters with no separation between them. This may lead to a situation that all the stops in an urban area are classified as one stop. Hierarchical clustering using Ward's linkage is another cluster algorithm used to merge potential stops (Sharman and Roorda, 2011). However, Ward's method does not work as well on non-spherical clusters and can lead to very large cluster sizes.

Another instance of a machine learning algorithm is using the Support Vector Machine (SVM (Boser et al., 1992)) for stop filtering. Yang et al. (2014) use an SVM to characterize stops as either delivery or non-delivery stops. Thierry et al. (2013) use Kernel Density Estimation (KDE) as a general non-parametric way of determining stops by learning stop densities in space. Entropy of carriers visiting a potential stop has also been used to characterize stops as primary (freight related) or secondary (non-freight related) (Gingerich et al., 2016). Hidden Markov Models (HMMs (Rabiner and Juang, 1986)) have also been used to find traffic stops, activity stops, and non-activity stops (Taghavi et al., 2019).

While geospatial information such as land use is not used directly to find stops, it is used to filter or merge them. You and Ritchie (2018) geocode stops into TACs which are dis-aggregated versions of TAZs and then these TACs are clustered using Mahalanobis distance. Land use has been used to inform the purpose of stops in Karam et al. (2020) and consequently filter them out (Thakur et al., 2015). Yang et al. (2022a) propose filtering out stops using freight POI and urban road network data. Sharman and Roorda (2011) propose combining clusters whose medians lie in the same parcel boundary. Yang et al. (2022b) use Voronoi diagrams constructed using geocoded freight POIs, and use that to filter out stops (as many as 73.2%).

2.3 Trip and tour inference

In contrast to the more prevalent work on identifying trip ends or stops, research on identifying trips is less common. Some approaches to finding trips involve using filters based on speed and location, characterizing trips as occurring between internal and external zones, introducing time-based thresholds, or using criteria based on the circuitousness of trajectories. Inferring tours, or collections of trips, is even less common and is often done as a step towards developing truck tour models. One approach to identifying tours involves the concept of closed tours, which are collections of trips that depart from and return to a depot, and open tours, which do not return to the depot. In this review, we will examine the existing literature on inferring trips and tours, highlighting the various methods and techniques that have been proposed and discussing their strengths and weaknesses.

As compared to work on finding trip ends or stops, work on finding trips is not so prevalent. Hwang et al. (2017), You and Ritchie (2018), Akter et al. (2018), Thakur et al. (2015), Yang et al.

(2022a), and Kuppam et al. (2014) contain examples of trips found between two freight related stops. You and Ritchie (2018) use a speed and location filter to filter out trips. Akter et al. (2018) suggest characterizing trips as not between stops but between internal/external zones. This may make sense depending on the scope of analysis and provide for more flexibility in terms of accuracy needed if the scale of analysis is coarser than the scale at which stops are found. Thakur et al. (2015) and Kuppam et al. (2014) introduce a time-based threshold to break trips. Yang et al. (2022a) use a criteria based on circuitousness of trajectories. It basically tries to ensure that a segement of the GPS trajctory is deemed to be a trip only if the truck travelled between the two trip ends directly. Ma et al. (2016) study the trip characteristics of freight truck such as average trip chains, average trip stops per trip chain, average dwell time, and average trip distance.

Inferring tours, which are collections of trips, of a truck as part of its activity is uncommon and mostly done as a step to prepare input for truck tour models (You and Ritchie, 2018; Kuppam et al., 2014). You and Ritchie (2018) introduce the idea of closed tours as a collection of trips departing from a depot and returning to the same location. An open tour on the other hand seems to be just one that does not return to the depot. They also reset tour numbers when the truck has been waiting for more than 3 h. The authors assume that the depot is the location where a truck is at the end of the day. We present instead a more data-driven framework of inferring the location of the depot involving topographical and trajectory characteristics.

The novel contributions of this project can be summarized as follows:

· We introduce a method to perform a novel type of inference on GPS data, device matching.

· We suggest an updated development of a data pipeline to use a simple inference procedure to infer truck activity.

· We propose novel strategies of finding the depot/hub of operations.

· We implement and test the algorithm on a large real world dataset containing data across four months greater Philadelphia region (~196 M rows).

· We share insights on practical parameter selection for each step in the data processing pipeline, while highlighting the effects of errors in data collection, such as gaps and variable transmission frequency, on data processing.

3 Research methodology

The problem addressed in this research is how to accurately infer truck activity, including stops, trips, tours, and hub of operations, from GPS data that contains resetting device IDs. This problem is important because the resetting of device IDs in commercially available GPS data can obscure long-term trends in truck activity and limit the usefulness of the data to public agencies. To address this problem, we propose a pipeline that includes a device matching algorithm to create longer records of activity for each truck, as well as an activity inference module that uses a variety of methods to infer stops, trips, tours, and hub of operations from the GPS data. The goal of this research is to develop a privacy-preserving method for accurately inferring truck activity from GPS data with resetting device IDs, and to demonstrate the utility of this method for public agencies seeking to understand long-term trends in truck activity.

In this section, we first present in detail the exact methodology adopted by us to perform device matching. This is then followed by a comprehensive introduction of the studied truck activity inference methods using GPS data. Some common notations are first established. We may represent each data point in the GPS way point dataset, $d_t^{(i)}$, as Eq. (1):

$$d_t^{(i)} = \left(x_t^{(i)}, y_t^{(i)}, z_t^{(i)}, m_t^{(i)}, l_t^{(i)}\right) \tag{1}$$

where $i \in \mathcal{I}$ is the device ID, $t \in \{1, \ldots, T\}$ is a time index of the data points for each device. $x_t^{(i)}$, $y_t^{(i)}$, and $z_t^{(i)}$ are the latitude, longitude, and actual timestamp of device i at index t, respectively. $m_t^{(i)}$ and $l_t^{(i)}$ are the Micro Analysis Zone (MAZ) ID and land use category associated with the latitude-longitude position of device i at time t. MAZs are disaggregations of Traffic Analysis Zones (TAZs) created using block level census data to collect and project socioeconomic data at neighborhood level. By using MAZs as the units of clustering analysis, clustering can effectively capture network artifacts like highways without spanning clusters across them. These data points make up the entire dataset for a device, $d^{(i)} = \{d_t^{(i)}\}_{t=1}^{T}$. We know the weight class $w_i \in$ {"Light", "Mediun", "Heavy"} for each device i. We also know other supplemental information such as provider ID and driving profile (locally owned fleet, nationally owned fleet) for each device which is used to characterize and study truck behaviors. There is no fixed transmission frequency and the methodology is modular and flexible enough to work for different levels of data quality.

3.1 Device matching

This section describes the rule-based approach developed to match *eligible* devices with *candidate* devices. The overall process is summarized in Fig. 1. The set of eligible devices, E, represents the devices that we think have reset their IDs. Equation (2) formalizes this definition for datasets in which the device reset happens within a fixed time interval.

$$E = \{i : T_s \le d_T^{(i)} \le T_r\} \tag{2}$$

where T_s and T_r are the start and end time thresholds respectively, between which a device must have its time of last transmission to be an eligible device. T_r can further be understood as the reset time of devices. We find this set of devices by looking at the mode of distribution of device start and end time, and the device transmission frequency. The mode of the distributions shows the time at which the device resets are likely happening, thus informing T_r. The device transmission frequency helps inform how far back from the reset time should we search for the eligible devices, thus informing T_s. For example, if a device resets happen at midnight and the transmission frequency is every 5 min, then we may want to search matches for devices which stop transmitting in the 15 min prior to midnight. We found that this is an essential step to minimize instances of false positives in the device matching algorithm. However, note that not all eligible devices may have a match since they may stop transmitting due to reasons other than a reset, such as simply turning off.

For each eligible device, $j \in E$, we define a set of candidate devices, C_j, which consist of devices that the eligible device may be matched to

$$\begin{aligned} C_j = \{k :\ & k \in \mathcal{I} \\ & \wedge\ d_1^{(k)} - d_T^{(j)} \le \delta \\ & \wedge\ \left(x_1^{(k)}, y_1^{(k)}\right) \in P \\ & \wedge\ |m_j - m_k| \le \Theta \\ & \wedge\ w^{(j)} = w^{(k)} \\ & \wedge\ \left|s_{1,2}^{(k)} - s_{j,k}\right| \le \Delta s \\ & \wedge\ s_{j,k} \le S_{\text{imp}}\} \qquad \forall j \in E \end{aligned} \tag{3}$$

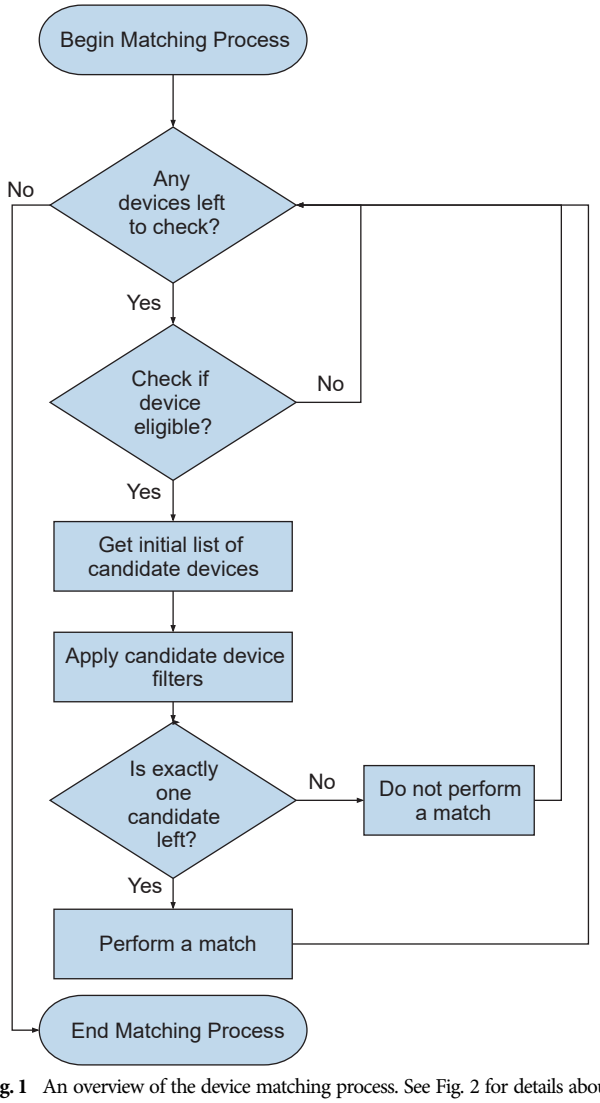


Fig. 1 An overview of the device matching process. See Fig. 2 for details about the candidate device filters.

We now describe all the conditions presented in Eq. (3) as a sequence of filters. The first is a timestamp-based filter where we only keep devices which start transmitting information δ min after the last transmitted data point of the eligible device as shown in Eq. (4):

$$d_1^{(k)} - d_T^{(j)} \leq \delta \tag{4}$$

We set the δ to be equal to 15 min to match up with the value selected for T_s, for the same reasons. Therefore, we have a 15-min interval after the last transmitted data point of the eligible device. This interval may be adjusted based on the transmission frequency of GPS pings in a particular dataset to obtain suitable candidate devices.

Following this step, a few filters are added to match devices accurately, demonstrated in Fig. 3. We apply a second, membership-based filter to the candidate devices. Let P define a bounded set in $\mathbb{R}^2$ such that $\left(x_T^{(j)}, y_T^{(j)}\right)$ are members of this set. The second filter then is shown in Eq. (5):

$$\left(x_1^{(k)}, y_1^{(k)}\right) \in P \tag{5}$$

In this project, we construct P by drawing a circle centered at $\left(x_T^{(j)}, y_T^{(j)}\right)$ with a radius equal to the max speed limit of network, M (70 mph) multiplied by δ (15 min). This ensures that an eligible device is not matched to a candidate device whose location could not have been realistically reached by the eligible device. In this case, Eq. (5) can equivalently be represented by points satisfying the following condition:

$$\left(x_T^{(j)} - x_1^{(k)}\right)^2 + \left(y_T^{(j)} - y_1^{(k)}\right)^2 \leq (M\delta)^2$$

The third filter is based on the direction of travel. The last few pings of the eligible device and the first few pings of the candidate device are used to calculate the direction of travel for the eligible device and the candidate device respectively. The number of pings selected depends on the transmission frequency of the device. Higher the transmission frequency, more pings are selected to find the direction of travel. Instead of selecting a fixed number of pings, this ensures that enough pings are selected for high frequency devices to find the general direction of travel rather than the immediate direction of travel for the device. In order to achieve this, a reasonably large window of duration τ is selected. We would like to introduce notation in Eq. (6) representing a period of time to aid in explaining the methodology.

$$d_{a:b}^{(j)} = \left\{d_t^{(j)} : a \leq z_t^{(j)} \leq b\right\} \tag{6}$$

We use trajectory points $d_{T-\tau:T}^{(j)}$ and $d_{1:1+\tau}^{(k)}$ for eligible and candidate devices to find the direction of travel for those devices respectively. The hyperparameter τ controls the time window to be used for determining the direction of travel. A longer time window may not be appropriate for devices performing shorter range travel as they may frequently change direction of travel. A very short window on the other hand may just show a particular maneuver that a truck might need to make enroute to its destination. Additionally, consideration should be provided to the data frequency while setting τ. For example, in data of frequency 10 min, a time window of 5 min may only have one data point. We use a time window of 5 min in this study.

Using the trajectory points above, we find the direction of travel filter in two ways. In the first method, we assume that the least square regression line using all the trajectory points is the direction of travel. In the second method, we take the first and last points in the trajectory interval outlined above and simply join them to find the direction of travel.

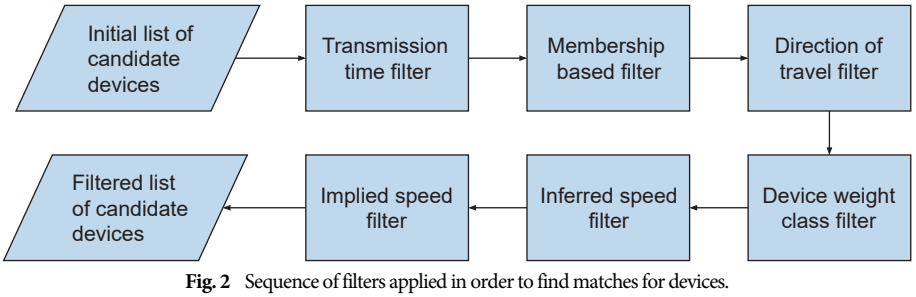


Fig. 2 Sequence of filters applied in order to find matches for devices.

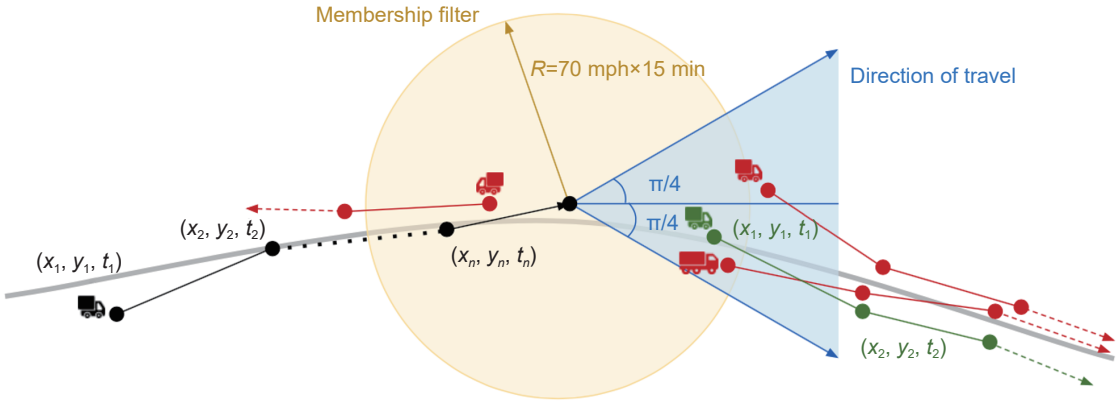


Fig. 3 Application of membership, direction of travel, and weight class filters.

$$|m_j - m_k| \leq \Theta \tag{7}$$

where m_j and m_k are the bearings for eligible and candidate device respectively. Equation (7) shows that the magnitude of their difference must then be less than or equal to Θ. This was based on the observation that freight devices are unlikely to drastically switch the direction of travel. In our study we set the value of $\Theta = 45°$ since we thought it would be unlikely that a truck would switch from a north–south freeway to an east–west freeway. The exact value of the allowable angle between direction of travel of the eligible and candidate devices may be adjusted based on the topography of the road network.

The fourth filter shown in Eq. (8) posits that weight class of the eligible device and candidate device should be the same. This is based on the rationale that the after resetting the device ID, barring any hardware issues, the weight class of the device should not change. For example, a heavy weight truck should remain a heavy weight truck.

$$w^{(j)} = w^{(k)} \tag{8}$$

The fifth filter posits that the inferred speed of the candidate device should be within Δs of the implied speed of candidate device as shown in Eq. (11). The inferred speed, $s_{1,2}^{(k)}$, of the candidate device is calculated using the first two GPS pings of the candidate device as shown in Eq. (9). We also use a more general definition of inferred speed later while inferring truck activity (see Eq. (13)).

$$s_{1,2}^{(k)} = \frac{\text{dist}\left((x_1^{(k)}, y_1^{(k)}), (x_2^{(k)}, y_2^{(k)})\right)}{z_2^{(k)} - z_1^{(k)}} \tag{9}$$

The implied speed, $s_{j,k}$, is calculated by first calculating the distance between the last ping and first ping of the eligible device and candidate device respectively, which is then divided by the time difference between the last and first pings of the eligible and candidate devices. Equation (10) shows this concept mathematically:

$$s_{j,k} = \frac{\text{dist}\left((x_1^{(k)}, y_1^{(k)}), (x_T^{(j)}, y_T^{(j)})\right)}{z_1^{(k)} - z_T^{(j)}} \tag{10}$$

The rationale behind this filter is that the speed of the candidate device should not be changing abruptly. A device may change its speed quickly but to have high confidence in a match, we assume that devices keep travelling at constant speed.

$$\left|s_{1,2}^{(k)} - s_{j,k}\right| \leq \Delta s \tag{11}$$

And finally an implied speed filter is also used (Eq. (12)), in addition to the second filter described above. The benefits of doing so are that it enables us to control the area of search and implied speed during matching, in case the search area in the second filter is required to be different from a circle. This is a simple filter where the implied speed must be less than a threshold S_{imp} :

$$s_{j,k} \leq S_{\text{imp}} \tag{12}$$

An eligible device is then matched to a candidate device if after applying all the six filters, the set of candidate devices for an eligible device just has one member (i.e., $|C_j| = 1$). The reason being that if there are multiple devices which pass through all the filters, it might be hard to programmatically select which of the multiple candidate devices is the correct match. It may require human intervention which may not be possible for large datasets.

3.2 Truck activity inference

This section describes the rule-based approach utilized to identify truck activities such as *stops*, *trips*, and *tours* from GPS data. Stop activity, $\sigma_k^{(i)}$, for a truck i, is when it is performing some freight delivery related functions. The set of all stops is represented by Σ_i.

$$\sigma_k^{(i)} = \left\{d_t^{(i)}\right\}_{t=t_{k,i}^b}^{t_{k,i}^e}$$

$$\Sigma_i = \left\{\sigma_k^{(i)}\right\}_{k=1}^{N}$$

where $k \in \{1, \ldots, N\}$ is an increasing index of the order of stops. $t_{k,i}^b$ and $t_{k,i}^e$ are the time index associated with the beginning and ending of stop k of device i respectively. Some other reasons for a truck to stop, such as being stuck in traffic or taking a rest, which are not relevant to this project, are intended to be excluded.. Trip activity, $\lambda_{k,k+1}^{(i)}$, is when a truck moves from one stop activity to another stop activity.

$$\lambda_{k,k+1}^{(i)} = \left\{d_t^{(i)}\right\}_{t=t_{k,i}^e+1}^{t_{k+1,i}^b-1}$$

A tour activity, $\Lambda_{O,D}^{(i)}$, is a collection of trip and stop activities undertaken by truck i between origin, O, and destination, D. Tours may be characterized as either closed tours, meaning that the truck returns to its hub of operations, $\Lambda_{H,H}^{(i)}$, or as open tours, the truck does not return to a hub, $\Lambda_{H,\bar{H}}^{(i)}$, where H and $\bar{H}$ are just the inferred hub and any place not the hub respectively.

$$\Lambda_{O,D}^{(i)} = \left\{d_t^{(i)}\right\}_{t=t_{O,i}^b}^{t_{D,i}^e}$$

We apply a speed threshold based on Kuppam et al. (2014) to identify stopping activity by trucks since it is the most commonly used technique (Yang et al., 2014; Aziz et al., 2016; Akter et al.,

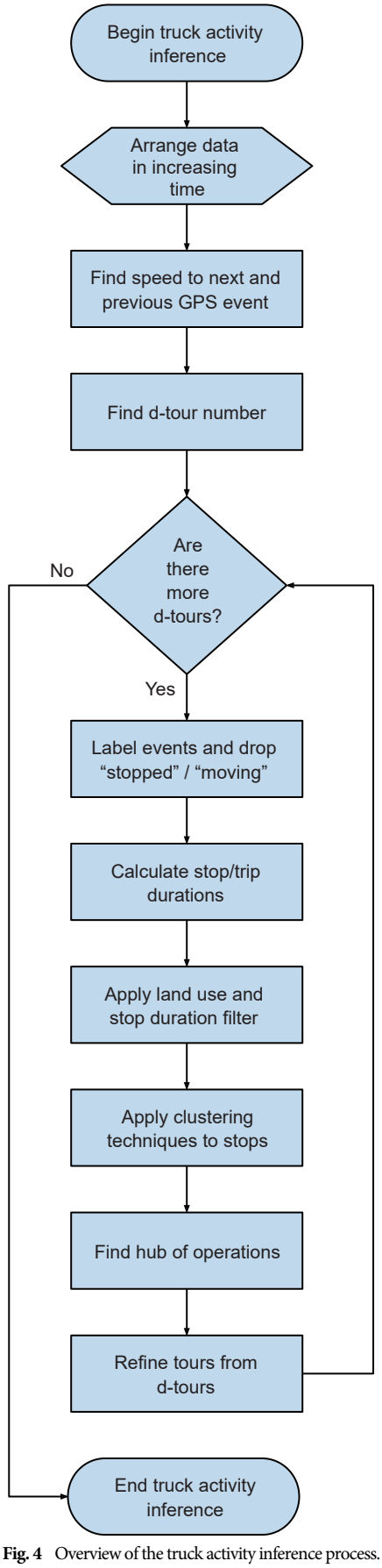


Fig. 4 Overview of the truck activity inference process.

2018; Camargo et al., 2017; Yang et al., 2022a), easiest to understand and fast to apply on large datasets. Another motivation was to create an algorithm that works with the least amount of additional data. The basic idea behind the technique, as illustrated in Fig. 4, is that when we find that a device's inferred speed to fall below a certain threshold we say that the device has stopped, and use a variety of filters and clustering methods to remove and summarize stopping events into representative stops, finally followed by inferring higher order activity characteristics such as inferring the hub of operations, trips undertaken, and finally chains of trips created to finish tours.

3.2.1 *Finding stops and defining device-tours*

The first step towards stop inference is to calculate the inferred space mean speed from the previous GPS ping, $s^{(i)}_{t-1,t}$, and to the next GPS ping, $s^{(i)}_{t,t+1}$, as shown in Eq. (13):

$$s^{(i)}_{t,t+1} = \frac{\text{dist}\left((x^{(i)}_t, y^{(i)}_t), (x^{(i)}_{t+1}, y^{(i)}_{t+1})\right)}{z^{(i)}_{t+1} - z^{(i)}_t} \tag{13}$$

We sort the GPS records in increasing time order and use Eq. (13) to calculate speeds for each device. We use the haversine distance between consecutive GPS events and divide that by the time passed between those consecutive events to obtain the inferred speed to the next GPS event and from the last GPS event.

The next is to find the device-tours (d-tours), $\hat{\Lambda}^{(i)}_k$, for each device (Kuppam et al., 2014). Sometimes there are large gaps in the dataset, for example the device does not transmit data for 7 h at a time, after stopping. These situations may lead to calculation of extremely long stop durations. Inaccurate calculations of stop duration may lead to compounding errors in downstream steps of the data processing pipeline to infer truck activities. Therefore, it might be useful to break down the entirety of the GPS trajectory of a truck into meaningful chunks of analysis. As an added bonus, this ensures that our analysis lines up with the realities of the freight delivery industry such as regulations on how long a driver may drive, so how long a tour can last.

Ideally, we would want to perform analysis within a tour since that is the top level of truck activity that we are interested in. But also note that our definition of a tour includes the idea that it is a new tour if the truck returns to its hub of operations. Since the inference of hubs requires the knowledge of stops which requires us to set tours. It creates a circular problem. The concept of d-tour (Eq. (14)) solves this by providing conservative continuous subsets of the GPS trajectory to perform analysis in, which may then be further broken down with the knowledge of hub locations into actual tours.

$$\hat{\Lambda}^{(i)}_k = \left\{ d^{(i)}_t : d^{(i)}_t \notin \{\hat{\Lambda}^{(i)}_j\}^{k-1}_{j=1} \wedge z^{(i)}_{t+1} - z^{(i)}_t \leq \Delta_d \right\} \tag{14}$$

For each device, we say that a new d-tour has started if there is time difference of Δ_d h between any two GPS events for that device. In this study, we define a new d-tour if there is time difference of 8 h between any two GPS events for that device. This threshold is chosen since it reflects the longest allowable continuous driving time for commercial drivers in the USA①.

In the next step, the GPS events are provided a label, $\mathcal{L}^{(i)}_{k,j}$, within each d-tour. A speed threshold value, Δ_s, is selected and each GPS event is given one of "starting", "moving", "stopping", "stopped" event labels. The labels are applied according to the rules

①Federal Motor Carrier Safety Administration. Interstate Truck Driver's Guide to Hours of Service. Available online at https://www.fmcsa.dot.gov/sites/fmcsa.dot.gov/files/docs/Drivers%20Guide%20to%20HOS%202015_508.pdf (accessed on April 1, 2022)

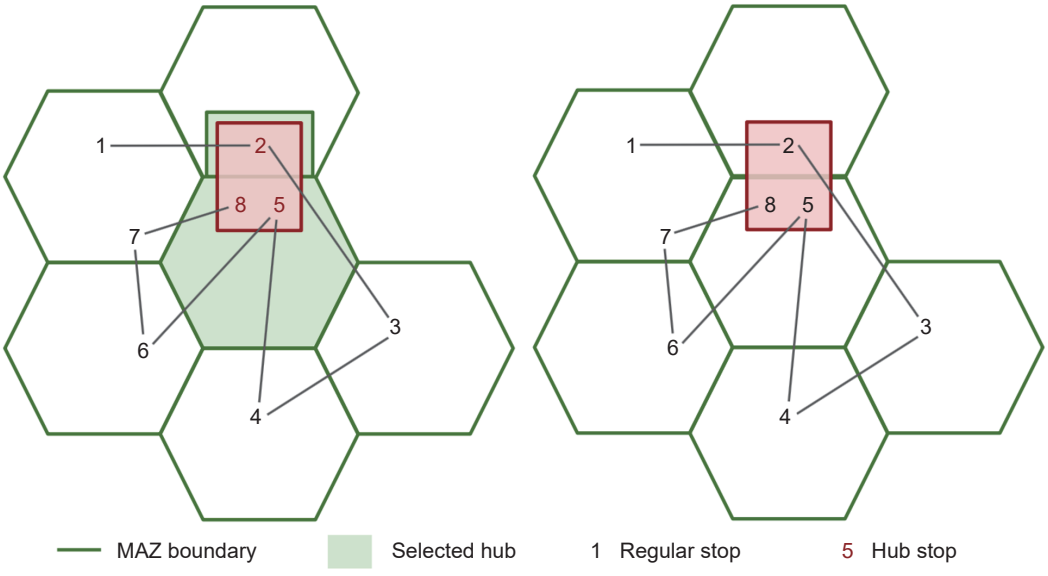


Fig. 5 Demonstration of hub-finding algorithm.

presented in Eq. (15):

$$\mathcal{L}_{k,j}^{(i)} = \begin{cases} \text{starting}, & s_{t-1,t}^{(i)} \leq \Delta_s \ \wedge \ s_{t,t+1}^{(i)} \geq \Delta_s \\ \text{moving}, & s_{t-1,t}^{(i)} \geq \Delta_s \ \wedge \ s_{t,t+1}^{(i)} \geq \Delta_s \\ \text{stopping}, & s_{t-1,t}^{(i)} \leq \Delta_s \ \wedge \ s_{t,t+1}^{(i)} \leq \Delta_s \\ \text{stopped}, & s_{t-1,t}^{(i)} \leq \Delta_s \ \wedge \ s_{t,t+1}^{(i)} \leq \Delta_s \end{cases} \tag{15}$$

where we use the index j for each GPS event in $\hat{\Lambda}_k^{(i)}$. Based on prior research, we searched for a suitable speed threshold in 1–7 miles per hour (Akter et al., 2018; Aziz et al., 2016; Camargo et al., 2017; Kuppam et al., 2014; Yang et al., 2014; Yang et al., 2022a). In order to inform the choice for the speed threshold, we analyzed the distributions of several truck activity metrics, including stop duration (Yang et al., 2022a), trip distance, and number of stops per trip, as well as device transmission frequency. By plotting these distributions and identifying the elbow point, we were able to make informed decisions about the appropriate speed threshold.

However, it is worth noting that in some cases, there may not be a clear elbow point, or the need to adjust spot-checked instances. In such cases, we found that too many stops of very short duration, too many trips of short distance, and a very high number of trips per tour were generally considered indicators of a poor speed threshold, as they would represent unrealistic operating conditions.

Empirically, we found that different speed thresholds may not label the exact same GPS event as the start of a stop, but overall it seems like the same general stop is identified. Additionally, we observed that higher speed thresholds are more tolerant to small changes in speed, such as movements within a parking lot, but may be less tolerant to congestion and stop-and-go conditions on the highway.

Consistent with the methodology in Kuppam et al. (2014), the events labelled "stopped" and "moving" are dropped. To keep things simple, we are going to take advantage of some notation overloading and represent this filtered set of events with $\hat{\Lambda}_k^{(i)}$ as well. This leaves a "stopping" event at time index $t_{k,i}^b$, as the event when a device arrived at a potential stop. And the consecutive "starting" event at time index $t_{k,i}^e$ as when the device left that stop. The stop durations, $\omega_k^{(i)}$ (Eq. (16)), and trip durations, $\omega_{k,k+1}^{(i)}$ (Eq. (17)), are calculated between successive events by taking the difference of timestamps between consecutive events. Similarly, haversine distance between successive events is calculated and stored.

$$\omega_k^{(i)} = z_{t_{k,i}^b}^{(i)} - z_{t_{k,i}^e}^{(i)} \qquad \forall \sigma_k^{(i)} \tag{16}$$

$$\omega_{k,k+1}^{(i)} = z_{t_{k,i}^e+1}^{(i)} - z_{t_{k+1,i}^b-1}^{(i)} \qquad \forall \lambda_{k,k+1}^{(i)} \tag{17}$$

3.2.2 *Applying filters to identified stops*

Since some of the potential stops might not be related to freight activities, we propose applying two stacked filters to potential stops to obtain stops related to freight activities. We also apply clustering techniques to merge stop events located close by, but detected as unique events, possibly due to effects like GPS jitter.

We first propose applying a filter based on invalid land use like in Thakur et al. (2015). The rationale behind this filter being that some inferred stops are associated with land uses which are unlikely to contain a valid freight stop. Examples of land uses that can potentially be used to be filter out nonfreight stops are water, golf course, undeveloped, cemetery, rail right-of-way, highway right-of-way, utility right-of-way and wooded. This filter should help remove temporary stops such as those made in traffic and stops made on neighboring invalid land use due to GPS jitter. The effectiveness of this filter depends on the accuracy of the GPS data and the correctness of the land use data. If the set of invalid land use is written as Ψ, we can formalise this filter as shown in Eq. (18):

$$l_t^{(i)} \notin \Psi \, \forall \, d_t^{(i)} \in \sigma_k^{(i)} \qquad \forall i, k \tag{18}$$

The second filter is based on stop duration (Eq. (19)). These are based on the idea that freight related stops must take some time to be completed. While used widely (You and Ritchie, 2018; Akter et al., 2018; Camargo et al., 2017; Holguin-Veras et al., 2020; Chankaew et al., 2018; Thakur et al., 2015; Yang et al., 2022b, 2022a; Kuppam et al., 2014), we study the effectiveness and interaction of this filter criteria on other aspects of a comprehensive pipeline such as clustering and hub finding. Apart from a visual inspection, we suggest looking at the distributions of the number of stops per trip and the marginal decrease in number of stops on increasing the stop duration threshold, to determine

the stop duration threshold, Δ_t. In general, we found that a short duration filter of 3 min is good enough to remove most traffic stops. The effectiveness of this filter is influenced by the transmission frequency of data since infrequent data can make accurate determination of stop duration hard.

$$\omega_k^{(i)} \geq \Delta_t \qquad \forall i,k \tag{19}$$

As a final step towards reducing external effects of working with GPS data on accurately inferring truck activities, we perform some stop clustering to reduce the number of false trips and get a better estimate of stop/trip duration. We studied clustering by using pre-determined geospatial boundaries (i.e., MAZs) (Sharman and Roorda, 2011), using density of points (i.e., DBSCAN) (Luo et al., 2017; Aziz et al., 2016; Hwang et al., 2017; Karam et al., 2020), and using distances between stops (hierarchical clustering) (Sharman and Roorda, 2011). The goal of clustering is to identify a membership matrix, C, which has M columns representing clusters and N rows corresponding to each stop in a d-tour. Specifically, since the purpose of clustering was to remove short trips within the same stop, the clusters are identified among the stops of a d-tour. In this way, clustering allows us to group stops from a d-tour into clusters based on their characteristics or attributes.

$$C = [\mathbf{c_1}, \ldots, \mathbf{c_M}] \in \mathcal{C} = \left\{ C \in \{0,1\}^{N \times M} \wedge C\mathbf{e} = \mathbf{1} \right\}$$

where $\mathbf{e}$ is a $M\times1$ column of ones, and $\mathbf{1}$ is a $N\times1$ column of ones. Different methodologies populate this matrix in different ways. To identify the best set of parameters for the clustering algorithms and the algorithm itself, we look at distribution of land use within identified clusters, distribution of trip distance, and overlap of clustering results. Ideally, we would want the stops clustered together to agree with the same land use if they are at one destination, they should eliminate short trips between stops at the same facility, and similar clusters should be found through equivalent methods.

We found that MAZ-based clustering (Eq. (20)) is most appropriate if it is desired to apply topological constraints to the clustering results, such as a cluster of stops should not straddle a freeway, or to assert choices from the downstream use of the inferred activities, such as truck trips within an MAZ may not be of concern for truck trip modeling. They can also provide a more direct, and potentially more tedious, way of controlling for sizes of clusters. The main difficulty of this approach is the need for additional information which may not be readily available. Mathematically, this can be represented as Eq. (20):

$$C_{ij} = \begin{cases} 1 & \text{if } m_t^{(i)} = m_t^{(j)} \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

Vanilla DBSCAN was found to be most appropriate for an easy-to-use algorithm with which we can group closer by stops together based on density. This does not require any extra information but there are parameters, the search radius, and the minimum number of points, that need to be tuned. However, a limitation that we ran into quickly was that parameters for DBSCAN need to be set differently for urban and suburban/rural stops. For example, if the parameters are set to be better able to cluster geographically spread out stops on facilities outside dense urban areas, then they create very large urban stop cluster where all stops may even be clustered into a large stop. This could be hard to programmatically achieve. As an aside, we found that utilizing a stop duration threshold to filter out stops was especially helpful if using DBSCAN, as it strips away a lot of very short stops in an urban area as it might drive up the density of stops artificially.

To deal with the extremely large/small sizes of clusters found with DBSCAN for stops with different densities but to preserve the programmatic nature of a clustering algorithm which does not need additional rich data, we tried out Agglomerative clustering with complete and single linkage. We did not try Ward criterion as it does not perform well for non-spherical clusters. Agglomerative clustering uses a bottom-up approach, starting with each observation as its own cluster, the clusters are successively merged based on distance between them.

$$C_{ij} = \begin{cases} 1, & \text{if } \max\{\text{dist}(a,b) : a \in \mathcal{A}, b \in \mathcal{B}\} \leq \Delta_c \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

where $\mathcal{A}$ is the cluster that stop i is a member of, similarly $\mathcal{B}$ is the cluster that stop j is a member of. If we are using the complete linkage criteria for calculating distance, it utilizes the maximum distance between observations of pairs of clusters (Eq. (21)). If we use the single linkage criteria, it utilizes the minimum distance between observations of pairs of clusters. While the single linkage was found to also lead to larger cluster sizes in areas of higher density of stops, the complete linkage criteria was selected in order to control the maximum size of the cluster and compactness of clusters.

In this project, due to the availability of the MAZ layer data, we find two separate cluster memberships of a stop. One through MAZ-based clustering, and the other using the complete linkage Agglomerative clustering. The goal is to be able to use the benefits of both clustering methodologies. This way we can use the topological constraints provided by the MAZ layer to find sensible clusters, and take care of situations where a stop may straddle two MAZs. At this point, the trips and stops durations are also updated to reflect the filtering away of some stops. This finishes the characterization of trips and stops.

3.2.3 *Finding the hub and inferring tours*

The subsequent step in the pipeline involves identifying the truck's hub, which enables the inference of the final activity of trucks, namely tours. The strategy to find a hub has three parts to it. The first part is to create initial estimates of which GPS stop(s) may be the hub. In case the criteria to find initial estimate of the hub yields multiple potential hubs, the second part of the strategy helps tie break to find which stop can be designated the hub. And finally, the third part of hub finding strategy helps extend the identified hubs to take advantage of the benefits afforded by a different clustering strategy. Therefore, we can mix two different clustering strategies.

There are two main decisions that need to be made to inform all parts of the hub finding strategy. The first decision relates to deciding whether to use the cumulative stop duration at a potential hub or the number of unique visits to a potential hub as the better indicator of which stop is the hub. This decision corresponds with assuming whether a hub is the location at which the truck spends most time or whether a hub is the location which the truck visits most often. If the two criteria were correlated, that the number of visits to a location leads to higher overall stop duration, then the decision becomes trivial and any one of the two may be chosen. It may be hard to see a clear correlation between the two quantities due to data issues such as gaps in data. In our case, since gaps in data impact inferred stop duration more than the number of unique visits, we select the number of unique visits to a potential hub as the main criteria. We however, use the

cumulative stop duration at a potential hub as the tiebreaker.

The second decision is to decide which clustering methodology to use to calculate unique visits or cumulative stop duration. Empirically we get similar results using either of MAZ-based or Agglomerative clustering. If there is a rich geospatial information such as MAZs available, then using those might mean making more meaningful assumptions. This is since a distance based cluster however defined, may include cases where stops on either side of a highway are clustered together.

Combining the two decisions, we use the following three-part strategy for finding hubs. For each device, we find the number of unique visits to an MAZ. We then designate the MAZ associated with the mode of the distribution of number of unique visits as the hub. In cases where there is a tie, we break the tie by looking at which agglomerative cluster has the highest cumulative stop duration. We then extend the identified hub to also include stops using its associated agglomerative cluster. That means that if a certain MAZ had membership of a particular cluster found through agglomerative methods, then other MAZs which are also a member of the same cluster would also be designated as MAZ hubs (Fig. 5). In our trials we test out eight other common sense hub strategies which are listed in Table 4. We then introduce a new column indicating if a stop, $\sigma_k^{(i)}$, is a hub or not,

$$h_k^{(i)} = \begin{cases} 1 & \text{if hub} \\ 0 & \text{otherwise} \end{cases}$$

Finally, after finding the hub, we are now able to finish inferring the tour activity for trucks. For each device-tour, we find a closed tour (You and Ritchie, 2018) if a continuous subsequence of trips within it starts and ends at the inferred hub. For every subsequence of trips which does not satisfy the criteria that it starts and ends at a hub, that is designated as an open tour. This finishes the methodology employed to infer truck activities from GPS data.

4 Data

We used anonymized truck GPS trajectory data for this project. There are about 196 million GPS pings from 383,647 devices. These data are from four weeks selected between 15 October 2017 through 21 July 2018 covering different seasons. The transmission frequency of data transmitted for devices varies from one second to more than two days. The average transmission frequency however is closer to 2 min. The data is provided in the World Geodetic System 1984 (WGS-84) coordinate system. Each GPS data point contains timestamp information, latitude, longitude, and device ID.

Additionally, we have information related to the weight class (light-, medium-, or heavy-weight) and a unique identifier for the data provider for each device. We use the weight class data and provider information to group the data for processing and analysis. We can see the breakdown of the number records, number of devices by week and weight class in Tables 5 and 6. The median and maximum gap between data points broken down by weight class and week is provided in Table 7.

We also use MAZ and Land Use information provided as geospatial layers. MAZs are sub-divisions of Traffic Analysis Zones (TAZs) which allow for a higher spatial resolution for modeling. These are informed using block level census data and can allow for collecting and projecting socioeconomic data at neighborhood level. There are 40 different land use categories informing on the purpose designated for a particular tract of land such as commercial, industrial, institutional, residential, transportation and their sub-categories among other things.

5 Calculations and results

5.1 Device matching

This section describes the different experiments undertaken to identify, understand, and resolve the different challenges encountered while developing a device matching methodology. We use truck data from January to construct the figures and tables in this section as we found that it was representative of the challenges present in data from other time periods.

5.1.1 Device reset time

To ensure that we are not matching devices which are not in need of matching, it is essential to first validate if the device IDs reset. An interesting followup to that question is to identify when do the device IDs reset. That can also help inform efforts to match the devices by narrowing the search window. We found that data from only one of the providers was amenable for the matching algorithm.

We established this by looking at the distribution of the first and last time of data transmission for that provider. Fig. 6 shows that most of the devices start transmitting information around midnight and finish transmitting near midnight. We can also see from Fig. 7 that there are two peaks for device duration. One of the peaks happens just short of 8 h and could be representative of the standard working day for a truck. The other peak is the actual mode happening at the 24-h mark. This means that there are many devices which persist for a day and no longer. If we plot the start and end time specifically for the devices which persist for 20–25 h, the start and end time get even more concentrated around midnight.

Hence, we can conclude that although not all devices from this provider reset their ID at midnight, there is a significant number of devices which end and start transmissions around midnight.

Table 4 Summary of different practical hub strategies tried out. N/A = Not Applied

Index	Initial strategy	Extending strategy	Tie-breaking strategy
1	Starting MAZ	N/A	N/A
2	Mode MAZ	N/A	N/A
3	Mode MAZ	N/A	Total stop duration from MAZ
4	Mode MAZ	N/A	Mode distance cluster
5	Mode MAZ	N/A	Total stop duration from distance cluster
6	Mode MAZ	Distance cluster	Total stop duration from distance cluster
7	Cumulative stop duration from MAZ	N/A	N/A
8	Mode distance	N/A	N/A
9	Cumulative stop duration from MAZ	N/A	Mode MAZ

These devices may be good candidates to be matched.

5.1.2 Using temporal criteria for device matching

Temporal features can be an important source of information to provide accurate device matching, however they may not be very straightforward to use due to noise and uncertainty. The first temporal feature we utilize to match devices is their first or last time of data transmission. From Fig. 6, it can be seen that most devices stop transmitting just before midnight. Since 23.43% of the devices are covered in the last 15 min, we set that as the threshold to construct the set of eligible devices.

Next, if we plot the cumulative distribution of the transmission frequency of pings and devices from this provider, in Fig. 8, we see that 88.95% of devices have a 95-percentile transmission frequency of up to 5 min. We argue that the maximum length of time between the new device ID and old device ID can be modelled based on the following scenario. The device resets just as it is about to record a new data point, 5 min after its last one. And then it must wait another 5 min before recording the next data point. So, the candidate devices for an eligible must start transmitting information within two times the transmission gap plus device resetting time, amount of time. We approximate this quantity to be three times the device transmission frequency.

This is an attempt to balance having a large enough window to be able to find the match, with having a small enough window to reduce the chance of false positive matches. Therefore, the set of candidate devices for an eligible device must then satisfy the constraint that it should start transmitting information within 15 min of the last transmission of the eligible device. Interestingly, we observed that only 42% of the devices have the same transmission frequency in the first and last 10 min of the time that it is recording information. This led us believe that directly using the transmission frequency as a criterion for device may not be appropriate.

We observed that if we just use a temporal filter as described above and improve it by adding constraints such as the weight class of the devices should match and that the implied speed of the device should be less than 70 miles per hour, we obtain a lot of candidate devices for every eligible device. Multiple matches such as these are not desirable as they can be seen as a proxy for estimating false positives. By adding the filters described in Section 3.1, we can decrease the number of multiple matches from 2,882 to 4 in January.

5.1.3 Synthetic dataset

In the dataset, the ground truth of which device reset to become another device is unknown. To validate our device matching procedure, we create synthetic dataset to test it on. The synthetic

Table 5 Number of devices in dataset by month and weight class

Month	# devices			
	Light	Medium	Heavy	Total
January	1,214	32,820	58,285	92,319
April	1,419	37,305	59,185	97,909
July	1,523	39,446	58,837	99,806
October	1,030	36,811	55,772	93,613
Total	5,186	146,382	232,079	383,647

Table 6 Number of data points in dataset by month and weight class

Month	# pings			
	Light	Medium	Heavy	Total
January	2,572,492	21,738,581	22,410,326	46,721,399
April	2,781,923	24,477,479	24,159,324	51,418,726
July	3,014,320	24,843,564	25,034,782	52,892,666
October	1,937,633	22,648,435	20,339,965	44,926,033
Total	10,306,368	93,708,059	91,944,397	195,958,824

Table 7 GPS data frequency of devices in dataset by month and weight class. Here s = seconds, m = minutes, and h = hours

Month	Light		Medium		Heavy		Total	
	Median	Max	Median	Max	Median	Max	Median	Max
January	6 s	7 h 59 m 22 s	1 m 29 s	17 h 53 m 22 s	1 m	8 h 02 m 42 s	1 m	17 h 31 m 42 s
April	6 s	17 h 52 m 31 s	1 m 30 s	17 h 24 m 28 s	1 m	23 h 39 m 50 s	1 m	23 h 39 m 50 s
July	6 s	15 h 08 m 52 s	1 m 30 s	20 h 41 m 29 s	1 m	10 h 16 m 36 s	1 m	20 h 41 m 29 s
October	6 s	9 h 04 m 40 s	1 m 30 s	20 h 53 m 32 s	1 m	21 h 39 m 28 s	1 m	21 h 39 m 28 s

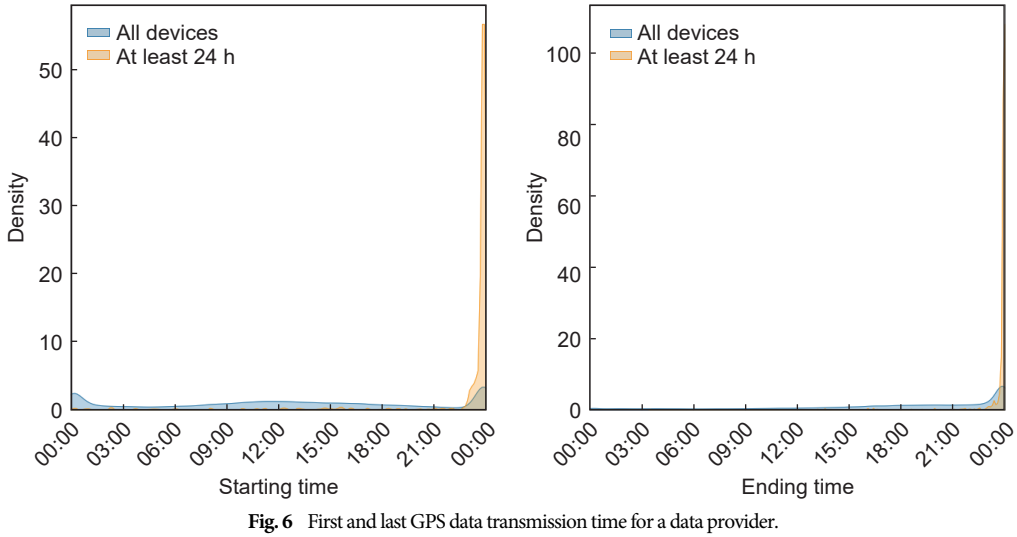


Fig. 6 First and last GPS data transmission time for a data provider.

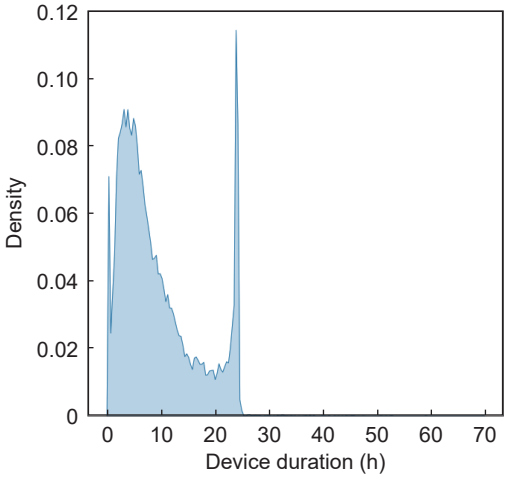


Fig. 7 Total device data transmission duration for a data provider.

dataset consists of devices that are known to have a match. Since we found that we could only reliably match devices from a specific provider, we decided to only use data from that provider while constructing the synthetic dataset.

Next we needed to select a reset time at which we will artificially reset the device ID. To ensure that we have a lot of data points to test, we selected 2 pm as the reset time since that time has one of the highest number of data points. Consequently, for a device to be a part of the synthetic dataset, we say that it needs to have at least 50 datapoints between 1–2 pm and 2–3 pm. To ensure that the artificial dataset mimics the actual conditions, we also add the constraint that the device should have a duration of at least 8 h.

We find 3,612 devices in January which satisfy these constraints. On using the parameters recommended in the methodology section, we were able to identify 3,599 (99.64%) devices as eligible for matching. We were then able to find a single matched device for 742 (20.62%) of the eligible devices. We find that 465 (62.67%) of the singly matched devices were matched correctly. Therefore, it validated our design goal of identifying conservative parameters to balance matching with reducing the incidence of false positives. There are limitations to the synthetic data approach to validate the algorithm. Primarily, they are that we may not know if the external conditions impact the data collection in an unknown way during the day as compared to the night.

5.2 Truck activity inference

In this section we describe the numerical experiments undertaken to identify, understand, and resolve issues encountered while developing a truck activity inference procedure. Like Section 5.1, we again use the truck data from January to construct the figures and tables in this section. The section is organized in the order of when the specific challenge would have been encountered in the activity inference pipeline.

5.2.1 Event type inference

One of the first issues that we noticed while attributing event types based on the thresholds on speed to the next GPS event and speed form the last GPS event is that there were some (very rare) consecutive starting and stopping events. On manual inspection we found that there was usually a large gap in time when the device did not transmit any information between two consecutive starting/stopping events. Hence this would be a missing data problem, which can be dealt with by assigning the events to different tours. In case it would not make sense to assign the events to different tours, care should be taken while calculating the stop/trip duration before and after the consecutive events to account for this uncertainty.

In another issue, we noticed that the device used a very long time to travel between a pair of starting and then stopping events, relative to the distance between the two events. Basically, the device has an unrealistically slow speed while moving between two locations. As an example, spot inspection yielded a device that took 8 h to travel 53.23 miles. In another extreme example, a device was inferred to take about 20 h for a trip of distance 0.27 miles. Manual inspections of these instances show that the devices transmit information very infrequently, once in 7 h, so it escapes the common sense filter for finding d-tours. However, it might be possible to fix this by breaking d-tours using a different aadaptive criterion (refer to Table 8 for the results obtained using different criteria to infer d-tours).

Related to the last issue, we also found instances where the distance between a consecutive pair of inferred stopping and then starting events was very high. A spot-checked example showed that the distance between a pair of consecutive stopping and starting events to be 9.62 miles. Overall, in January data, we found 11,000 (6.5% of all potential stops) instances where the starting location and the previous stopping location was more than 1 mile away, out of which 5,500 instances where the two locations were

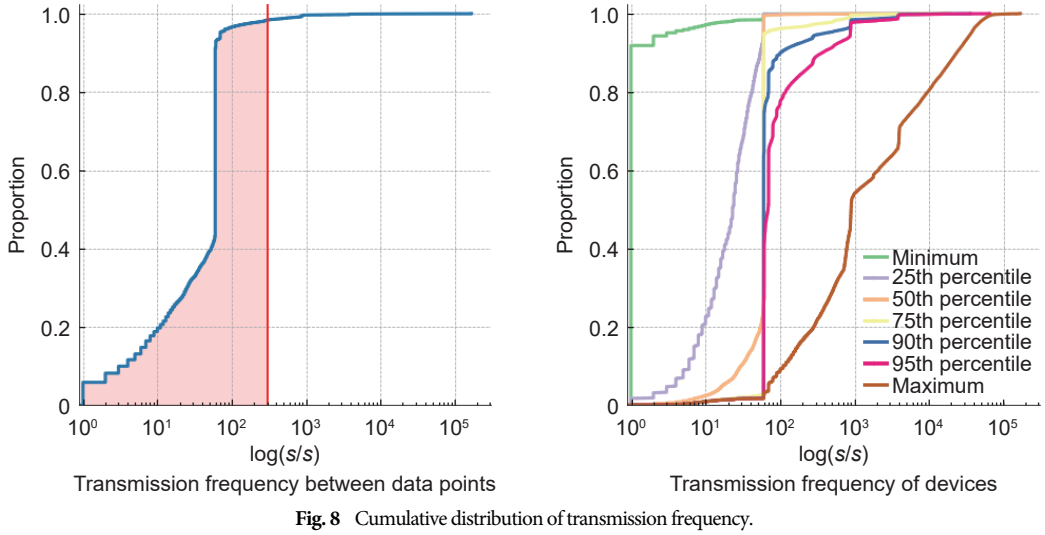


Fig. 8 Cumulative distribution of transmission frequency.

more than 10 miles apart. This was also identified to be an artefact of infrequent device data transmission. Like the previous issue, we can potentially perform better in these situations by using some sort of more complex tour breaking criteria based on data uncertainty.

5.2.2 *Device activity characteristics*

This set of issues stem primarily from the observation that the trip or stop duration are very low, or that the number of trips per tour or the number of stops per tour are very high for some combination of weight class and speed threshold used for stop inference.

One of the ideas we explored was application of different speed thresholds to infer stops depending on the weight class of the truck. Based on examples in the literature, we experimented with various speed thresholds ranging from 2 to 8 mph (see Table 9 for the results of using these different speed thresholds). Ultimately, we use the six mph threshold to infer stops and perform further analysis. The actual speed threshold may change some stops, but the stop duration filters and clustering efforts were more effective in dealing with false stops.

Another idea we explored to deal with implausible stop durations or unexpectedly large number of stops in a tour was to change the definition of d-tours. We tested 8 h and a change of date and 2 h (Kuppam et al., 2014) as the gap between two events which will break the trajectory into a new d-tour (Table 8). However, it did not completely take care of several of the longest stops/trips. Since the trajectory only gets broken up if it does not transmit information for more than 8 h. Ultimately, we stuck with 8 h since it had practical meaning and was conservative. Setting a hub to break d-tours into actual tours later was found to be more useful. It might be possible to test smarter ways of setting the threshold such as by using percentiles of the device data transmission frequency.

One of the first filters we tried was the stop duration filter to drop stops also suggested in Kuppam et al. (2014) (Table 10). To select the value of the stop duration filter, we checked the distribution of the length of trips, particularly, the percentage of trips of shorter duration. In doing so, we use the length of trips as a proxy for detecting false positive stops. The underlying assumption is that a higher number of false stops would yield a higher number of shorter duration trips. In the subset of data from January, we found that there are 1.75M (53.40%) stops with a stop duration less than 1 min. Additionally, 61.82% of trips are shorter than 5 min if no stop duration filter is used. We found that by using a 3 min stop duration filter, this proportion drops to 34.12%. On increasing the value of the stop duration filter to 5 min only a very small decrease (~7%) was observed, hence we stick with a 3 min threshold.

5.2.3 *Filters using land use*

We explored using land use data as another criteria to filter out inferred stops which may not be actual freight related stops. The motivation behind explicitly utilizing land use to filter out stops was due to an observation that some inferred stops were assigned land uses it would not make sense for private fleet vehicles to stop in. We found that applying land use as a filter decreases the average number of stops per tour and provides a good common sense criteria filter to remove stops which would be improbable (Table 11).

However, doing so was not straightforward. The primary reason behind the difficulty of doing so is that the specific event recognized as a stop by the algorithm may not be in the location characterized by the correct land use code for that stop. This is

Table 8 Impact of different device-tour methodologies on truck activity metrics. Here mi = minutes and m = meters

Method	Weight class	Stop duration (mi)	Trip duration (mi)	# Stops per tour	Trip distance (m)
No d-tour	Light	22.75	2.12	224.02	1,998.34
	Medium	73.85	9.77	53.53	9,070.20
	Heavy	21.33	8.96	32.68	10,408.54
8 h	Light	6.30	2.10	48.69	1,988.81
	Medium	24.35	8.95	17.69	8,347.89
	Heavy	10.64	8.10	28.70	9,679.04
Date change and 120 mi (Kuppam et al., 2014)	Light	11.90	2.10	67.63	1,985.82
	Medium	28.98	9.10	19.12	8,491.78
	Heavy	11.75	8.29	29.47	9,789.53

Table 9 Impact of different speed thresholds on truck activity metrics, using an 8-h threshold for finding d-tours. Here mi = minutes and m = meters

Speed threshold	Weight class	Stop duration (mi)	Trip duration (mi)	# Stops per tour	Trip distance (m)
2 mph	Light	8.93	4.29	31.30	3,199.47
	Medium	25.64	11.23	16.35	8,597.19
	Heavy	10.80	9.03	28.02	9,608.30
4 mph	Light	7.14	2.72	41.57	2,357.06
	Medium	24.74	9.81	17.22	8,424.05
	Heavy	10.69	8.55	28.28	9,738.34
6 mph	Light	6.30	2.10	48.69	1,988.81
	Medium	24.35	8.95	17.69	8,347.89
	Heavy	10.64	8.10	28.70	9,679.04
8 mph	Light	5.86	1.76	53.57	1,781.20
	Medium	24.13	8.32	18.02	8,289.52
	Heavy	10.67	7.69	29.20	9,534.90

Table 10 Impact of different stop duration thresholds on truck activity metrics, using an 8-h threshold for finding devicetours and 6 mph speed threshold for finding GPS ping labels. Here mi = minutes and m = meters

Stop duration threshold	Weight class	Stop duration (mi)	Trip duration (mi)	# Stops per tour	Trip distance (m)
0 mi	Light	6.30	2.10	48.69	1,988.81
	Medium	24.35	8.95	17.69	8,347.89
	Heavy	10.64	8.10	28.70	9,679.04
3 mi	Light	43.87	17.45	8.21	13,124.71
	Medium	54.56	19.82	8.80	17,464.19
	Heavy	64.14	41.26	6.31	46,801.81
5 mi	Light	60.88	24.63	6.29	17,431.41
	Medium	66.22	24.25	7.53	20,675.36
	Heavy	74.56	46.45	5.74	52,000.98

exacerbated when considering that we can cluster several stop events at a facility, but they may lie on different parcels and hence have different land uses. This adds another step in inferring the actual land use of the stop.

Table 12 shows the share of land use for inferred stops in January. In most cases, the majority of the stops were inferred to be on land uses which are considered valid. There were stops that were inferred to be on invalid land uses, but simply filtering them out may not be the best. We now present some observations from examples of stops inferred on an invalid land use code:

· Stops on water land use corresponded predominantly with devices on bridges crossing water bodies. These stops may have been inferred incorrectly, or may be a result of trucks breaking down on bridges. Some stops were on islands. The stops on islands may be a result of trucks stopping at facilities on the island.

· Stops on golf courses were made on roads running through or around golf courses. These stops may correspond to stores and facilities next to the golf course. Some of the stops were inferred next to facilities located on a golf course.

· Several stops inferred on undeveloped land use, based on satellite imagery[②], appeared to actually be on developed facilities. The age of development varied, with some developments seemingly being present for a while whereas others being more recent.

· Stops inferred to be on cemetery land use were located on road bordering cemeteries. Due to their location in urban areas, it is possible that these stops were inferred incorrectly and belong to freight activity at stores and facilities adjacent to the cemetery.

· Stops on the rail right-of-way were mostly inferred on roads running parallel to railroad tracks.

· Stops on highway right-of-way were by far the most prevalent type of stops on an invalid land use category. These stops may be a result of traffic conditions, truck repairs, rest stops or simply actual freight stops for locations next to highways.

· All the stops inferred on utility right-of-way are located on roads. In the dataset, a large number of inferred stops are along the New Jersey Turnpike. The land use designation seems to be a result of utility lines located close to these roads.

· Stops inferred to be on wooded land use were located on roads or facilities in developed areas with trees nearby. However, the stops were not observed to lie inside the clusters of trees.

5.2.4 *Clustering stops*

Clustering methods were employed to arrive at more accurate estimates of stop duration and trip duration. This is accomplished by merging stops which may represent the same destination during a specific interval of time. We tried out several different types of clustering algorithms such as topology based (MAZ clustering), density based (DBSCAN), and distance based (Agglomerative clustering). Table 13 shows the results of applying these clustering methodologies as a filter on truck activity metrics.

Somewhat unsurprisingly, we found that clustering by MAZ leads to the greatest number of clusters with the same land use. We also observe that MAZ based clustering provided the most meaningful clusters respecting network topology. A potential disadvantage of using MAZ-based clustering is that the spatial resolution of the clustering is essentially at a census block group level. The median area of an MAZ in the dataset is about 7.6 acres, and the median bounding box dimensions are about 870 feet × 890 feet. Therefore, MAZ clustering may not be appropriate if activity inference is required at a finer spatial resolution.

We found that DBSCAN makes large clusters for light truck data in urban areas. Fig. 9 shows a scenario where all the stops for a light truck were identified in the same distance cluster when they are clearly spread out across the city. This results in all of them being identified as a hub. Consequently, every trip is inferred as a tour as it ends at a hub. This is clearly undesirable behavior. Essentially, different densities of stops in suburban stop locations and urban stop locations requiring to be merged makes it hard for vanilla DBSCAN to perform well. Towards that end, we found it helpful to apply the stop duration-based filter to make the urban stop density lower for DBSCAN to work better.

Finally, we utilized Agglomerative clustering with complete and single linkage. We find that the complete linkage criteria is able to reasonably handle cases such as that presented above without growing the cluster to a really large size. We can obtain compact clusters while controlling for max size of clusters. This can be helpful for cluster size of a specific spatial resolution. We also utilize this fact later for using hierarchical clustering for extending inferred hub for devices. For the distance measure, experiments using haversine distance and Euclidean distance on Mercator projected points yield similar results. Additionally, using Agglomerative clustering does not require stops to be filtered for stop duration as it is not density based.

5.2.5 *Identifying hubs*

Finding a hub is an important part of the algorithm to infer tours. As described in Section 3.2, we employed different methods to create and test various hub finding strategies. We first test two very simple hub finding strategies: the first MAZ a device visits is its hub, and the MAZ that a device visits most often is its hub. Using just the MAZ however led to problems, for example we

②Available at https: //historicaerials.com

Table 11 Impact of applying land use filters on truck activity metrics, using an 8-h threshold for finding d-tours, 6 mph speed threshold for finding GPS ping labels and a 3-min stop duration filter. Here mi = minutes and m = meters

Land use	Weight class	Stop duration (mi)	Trip duration (mi)	# Stops per tour	Trip distance (m)
Without filter	Light	43.87	17.45	8.21	13,124.71
	Medium	54.56	19.82	8.80	17,464.19
	Heavy	64.14	41.26	6.31	46,801.81
With filter	Light	51.05	17.33	7.43	13,007.14
	Medium	65.22	19.80	7.78	17,420.77
	Heavy	77.24	41.41	5.71	47,309.40

Table 12 Breakdown of land use by inferred stops for January. Although unknown land use is invalid, stops with unknown land use are not filtered out

	Light truck		Medium truck		Heavy truck	
	#Stops	%age	#Stops	%age	#Stops	%age
Valid land use						
Residential	12,989	33.96%	155,667	22.19%	9,309	3.06%
Industrial	658	1.72%	63,623	9.07%	75,189	24.68%
Transportation	650	1.70%	25,260	3.60%	32,595	10.70%
Utility	122	0.32%	21,740	3.10%	2,803	0.92%
Commercial	13,019	34.04%	176,949	25.23%	51,660	16.96%
Institutional	3,232	8.45%	46,392	6.61%	6,362	2.09%
Military	38	0.10%	641	0.09%	414	0.14%
Recreation	726	1.90%	11,513	1.64%	1,382	0.45%
Agriculture	112	0.29%	10,356	1.48%	2,533	0.83%
Mining	1	0.00%	2,044	0.29%	200	0.07%
Cumulative	31,547	82.48%	514,185	73.31%	182,447	59.90%
Invalid land use						
Highway row	3,413	8.92%	67,055	9.56%	29,469	9.67%
Wooded	377	0.99%	18720	2.67%	6,695	2.20%
Water	14	0.04%	449	0.06%	178	0.06%
Undeveloped	1,023	2.67%	24519	3.50%	10,568	3.47%
Unknown	1875	4.90%	76435	10.90%	75,254	24.70%
Total	38,249	100.00%	701,363	100.00%	304, 611	100.00%

noticed for a device that sometimes when the truck returns to an adjacent MAZ, it is mislabeled as not having returned to the hub, whereas the facility that it returned to is just straddling two MAZs.

As a specific example, for a device, MAZ ID 4709 was identified as the hub based on the most frequently visited criteria. MAZ ID 4609 is adjacent to MAZ ID 4709, and distance-based clustering methodologies identify points in both MAZs together. However, when the device returns to MAZ ID 4609, it is not counted as the end of the previous tour. This leads to one of the tours of the device lasting for 2 days, when the tour number was not reset until it returned to MAZ ID 4709. This example demonstrated that we could use distance-based clusters as an alternative or supplementary way to identify hubs.

In several instances, we found that while trying to find the MAZ which was most visited by a device, there were instances where two MAZs were both visited the same number of times. This necessitated the need for a tie-breaking strategy. Consequently, we tried out three ideas to tie break. One of the ideas was to look at the cumulative stop duration from the MAZ cluster, or we could use the most visited associated distance cluster or use the cumulative stop duration from associated distance cluster.

The stop duration-based approaches above yielded results similar to each other. An additional assumption had to be made for using the distance cluster based rules: for cases in which an MAZ ID gets associated with two distance clusters, for example, say that MAZ ID 70732 is associated with distance cluster IDs 10 and 38, then the mode for the hub is found by counting instances of both 10 and 38. Also, as illustrated in the example of the facility straddling two MAZs above there is some need to "extend" identified hubs.

Since we use MAZ based methods for finding the initial hub locations, we decide to use a different (distance-based) cluster method to extend the hub location to incorporate a different source of information and use the advantage of distance-based methodologies to find things which are close by. Finally, we utilize these ideas to come up with the final strategy of using MAZ to find initial hub locations, tie-break with cumulative stop duration from distance clusters, and extend hubs using the distance cluster membership. Table 14 shows the result of hub finding methodologies on truck activity metrics.

5.2.6 Mean versus median

Selecting the measure of central tendency when evaluating metrics is an important decision in any data-driven project. Table 15 shows the mean and median values of the different truck activity metrics using a simple threshold-based baseline and using the entire inference pipeline with default values. The distributions of the truck activity metrics used to check the quality of truck activity inference itself are very skewed. While in traditional settings, we would use the median as the preferred measure of central tendency for a skewed distribution, in this project we use the

Table 13 Impact of applying clustering methods as a filter on truck activity metrics, using an 8-h threshold for finding device-tours, 6 mph speed threshold for finding GPS ping labels and a 3 min stop duration filter. The number in brackets denotes the distance threshold specific to each clustering algorithm. Here mi = minutes and m = meters

Clustering methodology	Weight class	Stop duration (mi)	Trip duration (mi)	#Stops per tour	Trip distance (m)
None	Light	43.87	17.45	8.21	13,124.71
	Medium	54.56	19.82	8.80	17,464.19
	Heavy	64.14	41.26	6.31	46,801.81
MAZ	Light	43.68	18.95	8.04	13 420.98
	Medium	54.26	24.76	8.32	18,638.61
	Heavy	60.92	62.28	5.46	55,391.02
DBSCAN (500 feet)	Light	43.79	19.35	7.98	13,527.68
	Medium	54.11	24.34	8.37	18,492.71
	Heavy	59.75	58.13	5.67	53,092.51
DBSCAN (0.25 mile)	Light	44.89	22.15	7.57	14,339.60
	Medium	55.27	29.95	7.75	20,126.95
	Heavy	61.06	68.93	5.19	58,539.50
Agglomerative (500 feet)	Light	43.78	19.15	8.01	13,483.79
	Medium	54.17	23.01	8.51	18,165.09
	Heavy	60.81	54.66	5.79	51,883.37
Agglomerative (0.25 mile)	Light	44.46	21.19	7.72	14,047.37
	Medium	54.65	27.12	8.08	19,263.27
	Heavy	60.81	64.87	5.37	56,565.79

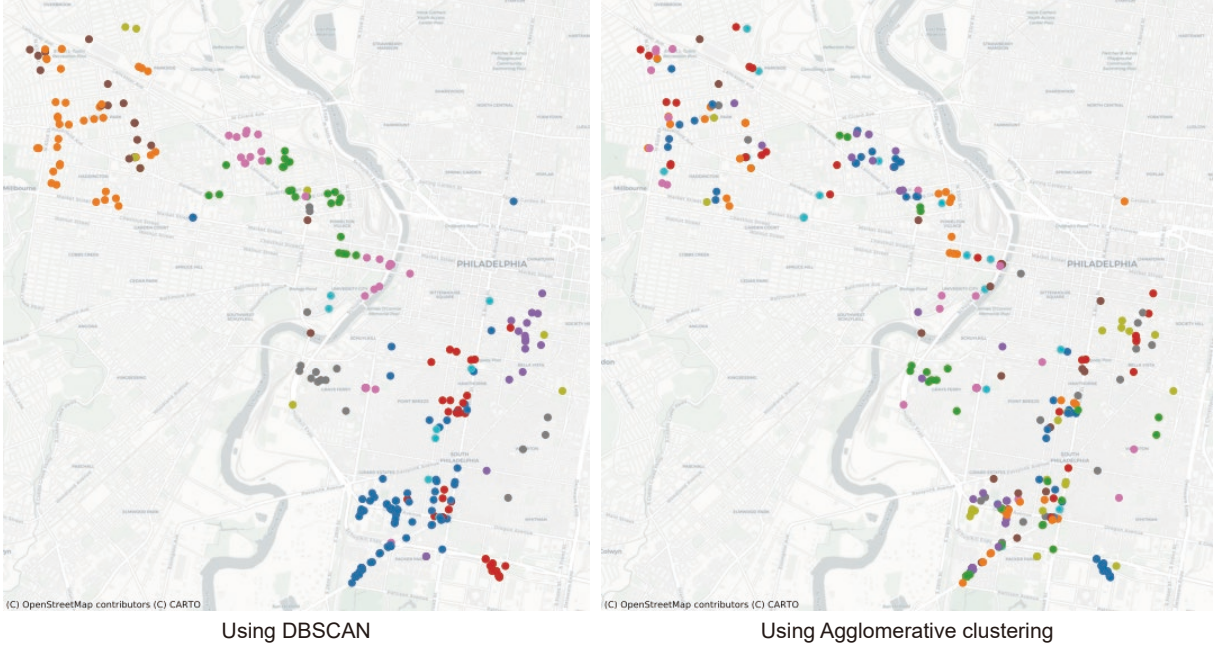


Fig. 9 An example of DBSCAN creating larger clusters in urban areas. Agglomerative clustering enforces a maximum size constraint on clusters, creating more compact clusters. Nearby stops of one color belong to the same cluster.

mean instead.

The median is preferred while analyzing skewed distributions due to its robustness to outliers. However, the advantage of using the median becomes a disadvantage in this scenario since the difficulties in truck activity inference do not lie in the usual cases but in the messy, outlier cases. The mean, being driven by outliers, gives a better sense of the performance of the pipeline on the harder to infer cases.

6 Conclusions

The overall goal of this project was to setup and apply a simple and modular truck activity inference method on a real-world GPS dataset. The real-world dataset is composed of multiple providers with challenges presented by its size, variable frequency of transmission, long gaps in data transmission, and assorted auxiliary information. These challenges restrict the application of algorithms in literature to be directly employed on datasets other than those they were created for. We study the effects of these challenges encountered in real world datasets and propose practical tips for using simple methods on such data effectively. The different aspects of truck activity that we identified were stops made by a truck, hubs of its operations, and consequently the tours and trips made by a truck. To achieve this goal, we undertook two additional activities. First, a device matching/chaining method was developed to deal with the issue of resetting

Table 14 Impact of hub finding methodologies on truck activity metrics using an 8-h threshold for finding d-tours, 6 mph speed threshold for finding GPS ping labels and a 3-min stop duration filter. The number in brackets denotes the distance threshold for agglomerative clustering. Here mi = minutes and m = meters

Method	Weight class	Stop duration (mi)	Trip duration (mi)	#Stops per tour	Trip distance (m)
No hub finding	Light	43.87	17.45	8.21	13,124.71
	Medium	54.56	19.82	8.80	17,464.19
	Heavy	64.14	41.26	6.31	46,801.81
Hub finding (500 feet)	Light	43.68	18.95	3.59	13,420.98
	Medium	54.26	24.76	3.61	18,638.61
	Heavy	60.92	62.28	2.69	55,391.02
Hub finding (0.25 mile)	Light	43.68	18.95	3.42	13,420.98
	Medium	54.26	24.76	3.39	18,638.61
	Heavy	60.92	62.28	2.57	55,391.02

Table 15 Mean and median values of truck activity metrics using just thresholds as baseline approach and using the entire pipeline with filter, clustering and hub finding. Here mi = minutes and m = meters

Method	Weight class	Stop duration (mi)		Trip duration (mi)		# Stops per tour		Trip distance (m)	
		Mean	Median	Mean	Median	Mean	Median	Mean	Median
Baseline	Light	22.75	0.55	2.12	0.85	224.02	130.5	1,998.34	521.03
	Medium	73.85	2.07	9.77	3.00	53.53	29.0	9,070.20	1,426.81
	Heavy	21.33	0.25	8.96	4.35	32.68	27.0	10,408.54	3,888.26
Pipeline	Light	43.68	9.12	18.95	9.77	3.59	2.0	13,420.98	4,754.35
	Medium	54.26	13.63	24.76	11.23	3.61	2.0	18,638.61	5,247.79
	Heavy	60.92	17.00	62.28	28.58	2.69	2.0	55,391.02	18,445.67

device identifiers. Second, effective clustering methods for stops to remove false positive stops and infer hubs of operations were studied and developed.

We used threshold methods for their simplicity and employed calibrated dwell time filters to remove non-freight stops. We then utilized clustering methods based on MAZs and complete linkage Agglomerative clustering to further consolidate stops to yield meaningful stop/trip durations. Agglomerative clustering helped in getting clusters which were close in distance together while applying the constraint on how far the farthest point of a cluster of stops can be. We experimented with and suggested novel strategies of finding the hub/depot of operations, which finally led to finding tours. We showed how the parameter selection for these steps be performed in a general project while highlighting the effect of gaps in data, and variable data frequency.

The utilized threshold methods are inherently inhibited by their simplicity may be under-fitting the data. That means that may be unable to pick up trends or signals within the dataset. Additionally, their practical application is impeded by their heuristic nature. A direction of future analysis could be to utilize more methods based in machine learning so that those methods are capable of learning more complicated artefacts of the dataset and require less manual tuning. Although literature contains several examples of such methods for individual aspects of truck activity inference, their interaction in a data processing pipeline, such as the effect of using Support Vector Machines or Deep Sequence Models to identify stops on identifying trips or tours, could be an interesting direction.

Acknowledgements

This research is funded in part by a Pennsylvania Infrastructure Technology Alliance grant and Mobility 21, and a national University Transportation Center on mobility funded by U.S. Department of Transportation. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein.

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

Akter, T., Hernandez, S., Diaz, K.C., Ngo, C., 2018. Leveraging open-source GIS tools to determine freight activity patterns from anonymous GPS data. In: American Association of State Highway and Transportation Officials (AASHTO) GIS for Transportation Symposium, 55–69.

Aziz, R., Kedia, M., Dan, S., Basu, S., Sarkar, S., Mitra, S. et al., 2016. Identifying and characterizing truck stops from GPS data. In: Industrial Conference on Data Mining. Cham: Springer, 168–182.

Boser, B. E., Guyon, I. M., Vapnik, V. N., 1992. A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Annual Workshop on Computational LearningTheory, New York: ACM, 144–152.

Camargo, P., Hong, S., Livshits, V., 2017. Expanding the uses of truck GPS data in freight modeling and planning activities. Transportation Research Record, 2646, 68–76.

Chankaew, N., Sumalee, A., Treerapot, S., Threepak, T., Ho, H. W., Lam, W. H. K., 2018. Freight traffic analytics from national truck GPS data in Thailand. Transp Res Procedia, 34, 123–130.

Choudhry, A., 2022. Smart mobility: challenges and opportunities for the next generation of transportation. XRDS Crossroads ACM Mag Stud, 28, 14–19.

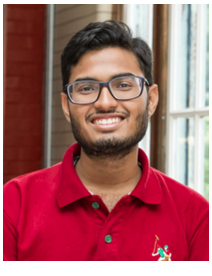
DE JONG, G., Gunn, H., Walker, W., 2004. National and international freight transport models: An overview and ideas for future development. Transp Rev, 24, 103–124.

Ester, M., Kriegel, H. P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, 226–231.

Gingerich, K., Maoh, H., Anderson, W., 2016. Classifying the purpose of stopped truck events: An application of entropy to GPS data. Transp Res C Emerg Technol, 64, 17–27.

Holguín-Veras, J., Encarnación, T., Pérez-Guzmán, S., Yang, X. S., 2020.

Mechanistic identification of freight activity stops from global positioning system data. Transportation Research Record, 2674, 235–246.
Hwang, S., Evans, C., Hanke, T., 2017. Detecting Stop Episodes from GPS Trajectories with Gaps. In: Seeing Cities Through Big Data. Cham: Springer, 427–439.
Karam, A., Illemann, T. M., Reinau, K. H., Vuk, G., Hansen, C. O., 2020. Towards deriving freight traffic measures from truck movement data for state road planning: A proposed system framework. ISPRS Int J Geo Inf, 9, 606.
Kuppam, A., Lemp, J., Beagan, D., Livshits, V., Vallabhaneni, L., Nippani, S., 2014. Development of a tour-based truck travel demand model using truck GPS data. In: Transportation Research Board 93rd Annual Meeting, Washington DC.
Kuppam, A., Lemp, J., Beagan, D., Livshits, V., Vallabhaneni, L., Nippani, S., 2014. Development of a tour-based truck travel demand model using truck GPS data. In: Transportation Research Board 93rd Annual Meeting.
Liu, Y., Zhang, Q., Lyu, C., Liu, Z., 2021. Modelling the energy consumption of electric vehicles under uncertain and small data conditions. Transp Res A Policy Pract, 154, 313–328.
Luo, T., Zheng, X., Xu, G., Fu, K., Ren, W., 2017. An improved DBSCAN algorithm to detect stops in individual trajectories. ISPRS Int J Geo Inf, 6, 63.
Ma, X., Wang, Y., McCormack, E., Wang, Y., 2016. Understanding freight trip-chaining behavior using a spatial data-mining approach with GPS data. Transportation Research Record, 2596, 44–54.
Qu, X., Zeng, Z., Wang, K., Wang, S., 2022a. Replacing urban trucks via ground–air cooperation. Commun Transp Res, 2, 100080.
Qu, X., Zhong, L., Zeng, Z., Tu, H., Li, X., 2022b. Automation and connectivity of electric vehicles: Energy boon or bane? Cell Rep Phys Sci, 3, 101002.
Rabiner, L., Juang, B., 1986. An introduction to hidden Markov models. IEEE ASSP Mag, 3, 4–16.
Sharman, B. W., Roorda, M. J., 2011. Analysis of freight global positioning system data. Transportation Research Record, 2246, 83–91.
Siripirote, T., Sumalee, A., Ho, H. W., 2020. Statistical estimation of freight activity analytics from Global Positioning System data of trucks. Transp Res E Logist Transp Rev, 140, 101986.
Taghavi, M., Irannezhad, E., Prato, C. G., 2019. Identifying truck stops from a large stream of GPS data via a hidden Markov chain model. In: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). Auckland, New Zealand: IEEE, 2265–2271.
Thakur, A., Pinjari, A. R., Zanjani, A. B., Short, J., Mysore, V., Tabatabaee, S. F., 2015. Development of algorithms to convert large streams of truck GPS data into truck trips. Transportation Research Record, 2529, 66–73.
Thierry, B., Chaix, B., Kestens, Y., 2013. Detecting activity locations from raw GPS data: A novel kernel-based algorithm. Int J Health Geogr, 12, 14.
U.S. Department of Transportation, Bureau of Transportation Statistics, 2022. Freight facts and figures: Moving Goods in the United States. https://data.bts.gov/stories/s/Moving-Goods-in-the-United-States/bcyt-rqmu (accessed on 2022-08-13)
Yang, X., Sun, Z., Ban, X. J., Holguín-Veras, J., 2014. Urban freight delivery stop identification with GPS data. Transportation Research Record, 2411, 55–61.
Yang, Y., Jia, B., Yan, X. Y., Jiang, R., Ji, H., Gao, Z., 2022a. Identifying intracity freight trip ends from heavy truck GPS trajectories. Transp Res C Emerg Technol, 136, 103564.
Yang, Y., Jia, B., Yan, X. Y., Li, J., Yang, Z., Gao, Z., 2022b. Identifying intercity freight trip ends of heavy trucks from GPS data. Transp Res E Logist Transp Rev, 157, 102590.
You, S. I., Ritchie, S. G., 2018. A GPS data processing framework for analysis of drayage truck Tours. KSCE J Civ Eng, 22, 1454–1465.



Arnav Choudhry is a Ph.D. candidate specializing in Advanced Infrastructure Systems under the guidance of Prof. Sean Qian. His research focuses on using state-of-the-art machine learning and optimization techniques to design and enhance cyber-physical infrastructure. He is working on using data to inform decision making for the next generation of last-mile delivery infrastructure through prototyping new robotic delivery platforms, performing stochastic risk assessment of robots, and developing robust and verifiable big data pipelines. He earned his B.Tech. degree in Civil Engineering with a minor in Operations Research from IIT Madras.

Sean Qian is a Professor jointly appointed at the Department of Civil and Environmental Engineering (major), Heinz College of Information Systems and Public Policy (minor), and the Department of Electrical and Computer Engineering (courtesy) at Carnegie Mellon University (CMU). He directs the Mobility Data Analytics Center (MAC) at CMU. His research interest lies in large-scale dynamic network modeling and large-scale data analytics for multi-modal transportation systems, in development of intelligent transportation systems (ITS) and in understanding infrastructure system interdependency. His research has been supported by a number of public agencies and private firms, such as U.S. NSF, U.S. DOE, U.S. DOT, Pennsylvania Department of Transportation (PennDOT), Maryland Department of Transportation (MDOT), Pennsylvania Department of Community and Economic Development (DCED), IBM, Honda R&D, Fujitsu Inc., Benedum Foundation, and Hillman Foundation. He serves as an Associate Editor for *Transportation Research Part C*: *Emerging Technologies*, *Transportation Science*, *Transportmatrica B*, and *Journal of Public Transportation*, and an editorial board editor for *Transportation Research Part B*: *Methodological*, and is an active member of the Network Modeling Committee of TRB and AI committee of ASCE. He is the recipient of the NSF CAREER award in 2018 and Greenshields Prize from the Transportation Research Board in 2017. He was a postdoctoral researcher in the Department of Civil and Environmental Engineering at Stanford University from 2011 to 2013, and received his Ph.D. degree in Civil Engineering at the University of California, Davis in 2011 and his M.S. degree in Statistics at Stanford University in 2012.