Inferring Trip Destinations From Driving Habits Data

Rinku Dewri, Prasad Annadata, Wisam Eltarjaman, Ramakrishna Thurimella
Colorado Research Institute for Security and Privacy
Department of Computer Science, University of Denver
{rdewri,prasad,wisam,ramki}@cs.du.edu

ABSTRACT

The collection of driving habits data is gaining momentum as vehicle telematics based solutions become popular in consumer markets such as auto-insurance and driver assistance services. These solutions rely on driving features such as time of travel, speed, and braking to assess accident risk and driver safety. Given the privacy issues surrounding the geographic tracking of individuals, many solutions explicitly claim that the customer's GPS coordinates are not recorded. Although revealing driving habits can give us access to a number of innovative products, we believe that the disclosure of this data only offers a false sense of privacy. Using speed and time data from real world driving trips, we show that the destinations of trips may also be determined without having to record GPS coordinates. Based on this, we argue that customer privacy expectations in non-tracking telematics applications need to be reset, and new policies need to be implemented to inform customers of possible risks.

1. INTRODUCTION

Many auto-insurance owners are probably familiar with the insurance discounts one can get by enrolling in telematics-based pay-how-you-drive programs. Examples of such programs in North America and Europe include Progressive's Snapshot[1], AllState's Drivewise[2], State Farm's In-Drive[3], National General Insurance's Low-Mileage Discount[4], Travelers' Intellidrive[5], Esurance's Drivesense[6], Safeco's Rewind[7], Aviva's Drive[8], Amaguiz PAYD[9], Insure The Box[10], Coverbox[11], Ingenie[12], MyDrive[13], and others. These programs rely on the collection of *driving habits data* (time of driving, speed, mileage, etc.) during a monitoring period, which is later analyzed to offer a customized discount to the enrollee.

Vehicle telematics based programs offer many advantages to insurers and the consumers. Insurers can offer more accurate pricing to consumers based on their driving habits. This increases affordability for safe drivers, and motivates others to adopt safer driving habits. Given the incentive to drive less, these programs also help reduce road accidents, traffic congestion, and vehicle emissions. Telematics have also proven useful in monitoring driver safety (e.g. the OnStar program), evaluating accident liability, preventing vehicle theft, tracking fleet movement, and routing traffic efficiently.

While few programs disclose that their data collection devices track the driver, most do not (or at least claim not to) track GPS locations, and imply an expectation of privacy that the customer's destinations are not tracked. Privacy policies clearly state what information is collected, as well as the possibility of sharing the data with third-parties, using it for fraud prevention and research, or for compliance with the law.

A significant body of research has gone into understanding the importance of quasi-identifiers in database privacy preservation. Quasi-identifiers are attributes of a database record that are non-identifying by themselves, but can be used to uniquely identify individuals when used in combination. A classic example is the re-identification of Governor William Weld's health records from an anonymized data set, based on a combination of gender, postal code and date of birth [3, 13]. Along similar lines, research has shown how individuals can be identified by their web searches [2], by their social network structures [11], by their movie ratings [10], or by their familial structures [9]. A large fraction of the population is also identifiable from their home and work location pair [4]. While the objective of this work is not to re-identify an individual in an anonymized data set, we do ask a similar question in the context of location privacy preservation: *can the different attributes of a driving habits dataset serve as quasi-identifiers of the destination of a driving trip?*

To answer this question, we develop a location inference attack that executes on real traces of driving habits data, and attempts to identify the destinations of the trips during which the data were collected. Our techniques extract quasi-identifying information such as traffic stops, driving speed and turns from the data, and match them to publicly available map information to determine potential destinations of a trip. We describe the implementation of these techniques and demonstrate that a number of trips can indeed be geographically matched to their destinations using simple driv-

[1] www.progressive.com/auto/snapshot
[2] www.allstate.com/drive-wise.aspx
[3] www.statefarm.com/insurance/auto_insurance/drive-safe-save/inDrive.asp
[4] www.nationalgeneral.com/auto-insurance/smart-discounts/low-mileage-discount.asp
[5] www.travelers.com/personal-insurance/auto-insurance/discounts-advantages/low-mileage-discount.aspx
[6] www.esurance.com/discounts/drivesense-discount
[7] www.rewindprogram.com
[8] www.aviva.co.uk/drive
[9] www.amaguiz.com/assurance-auto/comprendre-tarif-PAYD
[10] www.insurethebox.com/telematics/how-does-it-work
[11] www.coverbox.co.uk
[12] www.ingenie.com
[13] www.mydrivesolutions.com

ing features. Our conclusions are based on a probabilistic ranking of the possible destinations of a trip. Although not a foolproof method, our study shows that the destinations of certain trips can be very easily identified, thereby raising concerns about current expectations of privacy set by the data collection agencies. Of greater concern is the relatively unsophisticated (often common sense) nature of the concepts underlying our inference algorithm. Once the possibility of inference is identified, the techniques can probably be conceptualized by a group of undergraduate computer science students.

1.1 Related work

Location obfuscation is the most extensively researched method for location privacy. By performing spatial and temporal cloaking of locations, users can be provided location anonymity in a typical points-of-interest search application [5]. Cloaking regions can also be created such that the number of still-objects inside it is limited [1], or a minimum level of entropy is maintained in the queries originating from the region [8]. The popularity of public regions can also be used as the privacy level enforced by a method [14]. Unfortunately, anonymity or entropy based metics give inaccurate assessments of the privacy level of an algorithm [12].

As mentioned in the introduction, a number of researchers have shown that privacy cannot be guaranteed simply by avoiding sharing or avoiding the collection of private data. The possibility of linking using quasi-identifiers, or other sophisticated methods, always remain. Large fractions of the population can be identified by a combination of their gender, date of birth and place of stay [3]. People may enter locations, interests, affiliations, etc. in search queries, which makes them unique in a de-anonymized web search database [2]. Knowing the ratings assigned to eight movies is sufficient to identify an individual, even when there is a two week error in obtaining the dates of the ratings [10]. Half of the individuals in the U.S. population can be uniquely determined if their home and work locations are known at the level of a census block [4]. In GPS logs, people can be identified based on the last destination of the day and the most populated cluster of points [6, 7]. It is unfortunate that there is no method to a priori assess the inferences possible on a data set. Similar to the cases in database privacy, this work shows that our places of visit may very well be reflected in the underlying driving features.

The remainder of the paper is organized as follows. Section 2 states the location privacy expectations assumed in this study. Section 3 details our data collection process, followed by an explanation of the inference technique in Section 4. Section 5 presents results of executing the inference algorithm on real world traces of driving habits data. Section 6 concludes the paper.

2. LOCATION PRIVACY MODEL

The advantages of services that rely on the collection of driving habits data are noteworthy. Nonetheless, the threats of location tracking are equally concerning. Location tracking enables inferences about an individual's lifestyle and social circles, most of which may be considered private. Although the decision to share one's location is a personal one, such decisions can only be made when the intent to collect location data is fully disclosed. Therefore, location data collection and sharing practices should be explicitly stated in the privacy policies of pertinent businesses. The difficulty arises when the location information is inferable from other types of seemingly unrelated data, in which case, either the possibility of inference is unknown to the business, or the location data is inferred and used without consumer consent. We make the conservative assumption that if inferences are possible, they will be made.

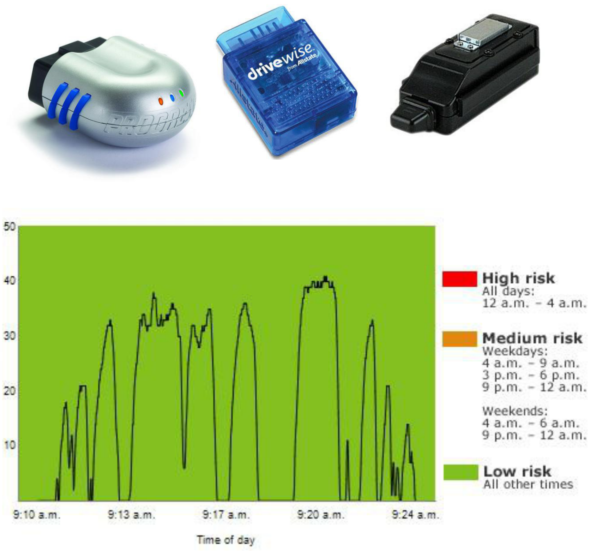

Figure 1: Driving habits data collection devices (OBD-II based) and the GPS tracking device used in this study (top right). Bottom plot shows driving habits data viewed in an online portal.

In our discussion of the related works, we mentioned projects that studied the threat of re-identification in anonymized location data. We study a somewhat different problem in this work, namely, the threat of location inference. Location inference is a deduction about the geographic location of an event from other known facts. We focus on the problem in the context of driving habits data collected *with* the consent of the driver. The collected data has no direct tracking of the user's location. Therefore, the offered privacy guarantee is that the data collection agency, or an adversary with access to the data, is unable to track the driver using this data. Consequently, we assume that obtaining knowledge of the destinations of travel is a clear violation of the location privacy expectations of the driver. This also implies that if a destination can be reached via more than one route, an inference of the correct destination is considered a violation even if the correct route is not inferred. We also assume that the driver has typical driving habits, such as staying within reasonable speed limits and taking best possible routes.

3. DRIVING HABITS DATA

Driving habits data includes features such as time of driving, speed, acceleration/deceleration patterns, distance traveled, braking practices, and others. Unless the associated service explicitly requires customer tracking, collection of location data is avoided for privacy concerns. We explain a typical data collection exercise by using an auto-insurance discount program as an example. Typical auto-insurance discount programs (propelled by driving habits data) are

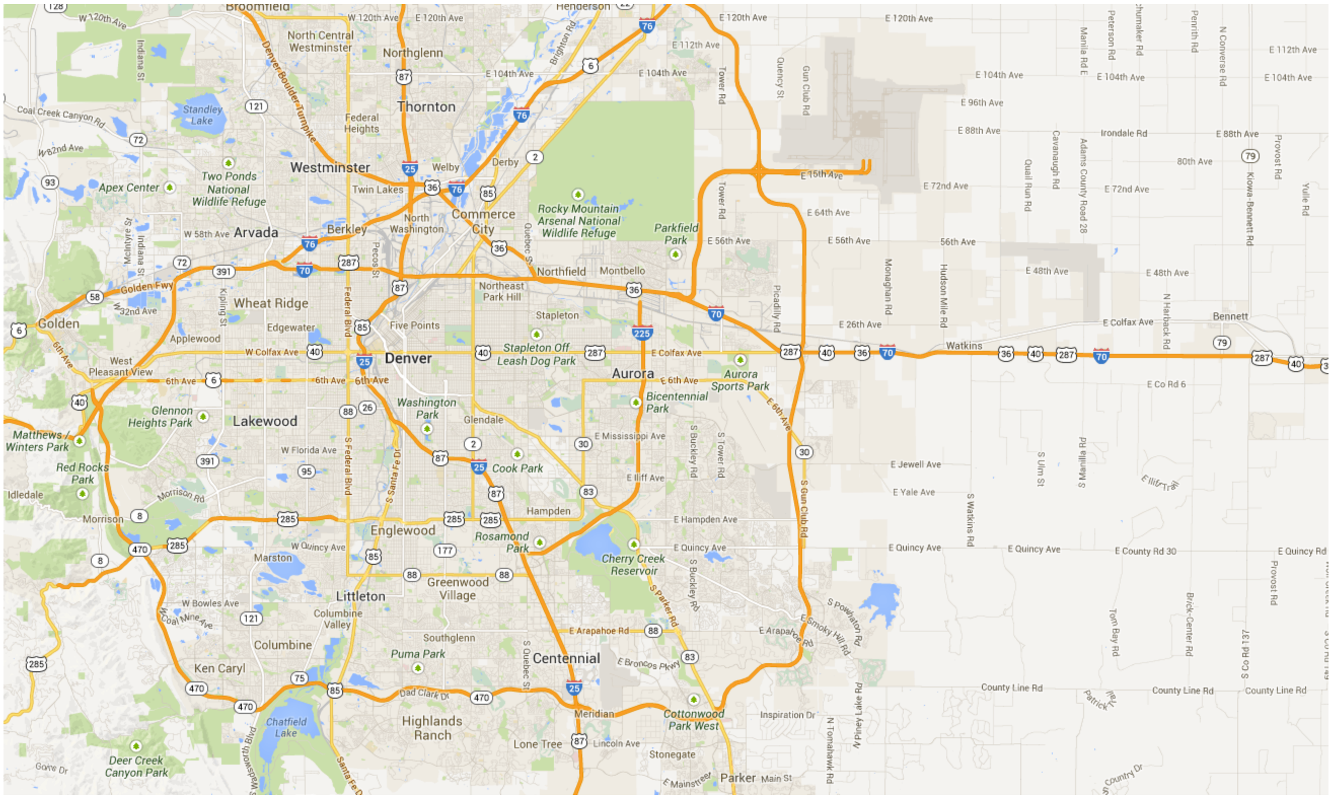

Figure 2: Denver area map (graph) explored during candidate path generation. Map data: Google (2013).

opt-in programs where the driver has to enroll to be evaluated for a discount in her insurance premium. Upon enrollment, the driver receives a data collection device (Fig. 1) that can be plugged into the on-board diagnostic (OBD) port of the vehicle. The device collects driving habits data over a period of several days to few months. Some devices, such as those used by the Progressive Snapshot program, can periodically upload the data to a background server using consumer telecommunication networks. This also enables the driver to see the data in an online access portal (Fig. 1). The device is returned to the agency at the completion of the data collection phase. Based on factors such as distances driven, time when driven, absence of hard brakes, and others, the driver is issued a discount in the insurance premium for the current and future terms.

3.1 Data collection

The motivation for this study came from observing real world plots such as that in Fig. 1 when one of the authors participated in an auto-insurance discount program. Unfortunately, we are not able to access the raw data underlying these plots. Executing our method on data points extracted from a graphical plot also fails to produce interesting results, clearly due to the inaccuracies involved in the extraction. With the ability to read most of the data from the vehicle's on-board computer, the collected raw data is expected to be precise and frequent. Therefore, we used a commodity tracking device (LandAirSea GPS Tracking Key[14]) to collect the raw data pertinent to this study. This battery powered device logs detailed driving data such as vehicle speed and GPS position, which can be later extracted into a computer through a USB connection. Note that a device connected to the OBD port can easily obtain more than ten samples per second; our tracking device operates at a much lower resolution of one sample per second. Although the device collects the GPS location (useful for validation later), the only data fields used in the inference process are: *time stamp* (t), *driving speed* (s), and *distance traveled* (d). We introduce here the term "trip" to mean a subset of the collected data, signifying a drive from one point of interest (e.g. home, office, hospital, store, friend's home, etc.) to another. Each $\langle t, s, d\rangle$ tuple of a trip is a data point of the trip.

We kept the devices in our vehicles for a period of 15 days in order to collect data from regular home-office trips, occasional shopping trips, and visits to infrequent places. We also collected a few trips between random locations at varying distances. During these trips, normal driving habits were maintained.

We use a total of 30 trips in this study. All trips are in the Denver, Colorado area, and includes home to work and work to home drives, visits to the airport, the downtown area, local grocery stores, school drop-offs, social visits, and others. Length of trips range from 1 mile to 25 miles, and spanned interstates, state highways, city roads and residential areas.

3.2 Pre-processing

We pre-process each trip to remove data points that may correspond to driving in traffic conditions. Our inference algorithms currently do not account for slow or "stop-and-go" driving resulting from heavy traffic; removal of data points

[14] www.landairsea.com/gps-tracker/gps-tracking-key

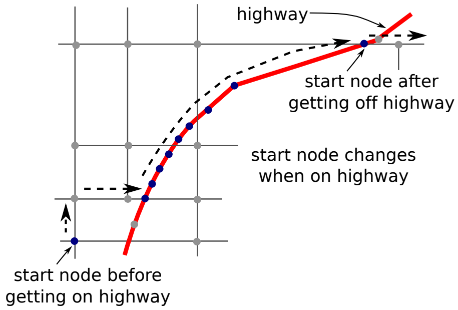

Figure 3: Disabling of shortest path constraint while exploring highway nodes.

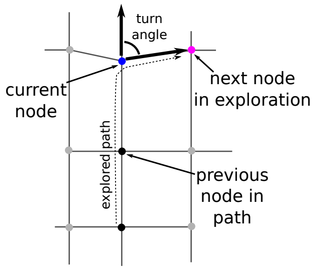

Figure 4: Turns along an explored path.

collected during such conditions help infer locations accurately in more number of trips.

Two steps are performed in this process. In the first step, we identify the data points where the driving speed is zero (possible stop in traffic). Thereafter, all data points between two zero-speed data points (inclusive) are removed if the total distance traveled between those two points is less than a threshold (half a mile used in this study). In the next step, consecutively time stamped zero-speed data points are removed if they do not span a time interval of at least 3 seconds.

After the traffic pre-processing, we note the distance (unique) values corresponding to the remaining data points with a zero speed value. We refer to these distances as *stop-points*, possible distances from the beginning of trip where the driver had to halt due to traffic stops at signals and intersections.

4. LOCATION INFERENCE METHOD

Our location inference method works under the hypothesis that the stop-points of a trip can be used as a set of quasi-identifiers for the destination of the trip. Therefore, if the start-location of the trip is known, we can search a map of the area for paths that begin at the start-location, and have traffic stops at distances given by the stop-points. The assumption of a known start-location is not unrealistic, since the data collectors are typically aware of the street address where the vehicle is parked overnight. Start-locations in subsequent trips can be obtained from the destinations of previous trips. Unless the roadways in the area are very regular, it is expected that a relatively smaller number of paths will satisfy the constraint to match every stop-point. The end-points of these *candidate paths* are potential destinations of the trip. We will employ a ranking process when multiple candidate paths are identified. In the following, we give a step-by-step account of the inference process as executed by us.

4.1 Area map as a graph

The first step to identifying candidate paths is to obtain a reliable map of the area. We obtained the map data available from the crowd-sourced OpenStreetMap project[15]. The map data from the project comes in the form of XML formatted .osm files. We processed these files to generate a graph with 323928 nodes, and 639395 directed edges representing motorways, trunks, primary/ secondary/ tertiary/ residential roads, and corresponding link roads. Nodes are typically placed at intersections. Nodes are also placed between two intersections if the road in between is curved. Therefore, the length of a road segment can be accurately computed by aggregating the distances between successive nodes placed on the road segment. Each node is labeled with its latitude and longitude coordinates. Each edge is labeled with the geodesic distance between the two nodes of the edge. Distances are computed using the Vincenty inverse formula for ellipsoids, available as part of the gdist function in the Imap R package. Edges are also annotated with a road type extracted from the downloaded XML files. This map data[16] covers an area of more than 1500 sq. miles in Denver, Colorado and its suburbs (Fig. 2), spanning between latitudes $39.41015^o N$ and $39.91424^o N$, and longitudes $105.3150^o W$ and $104.3554^o W$.

We also assigned speed limit values to the edges of the graph. Since it was difficult to obtain the legal speed limit on all roadways, we assigned numbers based on the road type indicated in the XML data. A capable adversary can obtain more accurate speed limit data from commercial sources.

4.2 Generating candidate paths

Candidate paths are generated by performing a standard depth-first search (DFS) of the map graph. The DFS starts at a node corresponding to the start-location of a trip and outputs all paths that satisfy the list of constraints discussed next.

Stop-point matching. During the DFS traversal, we keep track of the length of the path from the start node. This constraint requires that, at any stage of the traversal, the current path must have an intersection node (3-way or more) at all stop-points less than the current length of the path. However, since traffic stops often happen a few feet away from the signal (the exact coordinates of the intersection), we allow for a *slack* while matching the path length to a stop-point. The slack is set to 500 feet in this study. Stop-point matching is not performed for the last stop-point, since the last stop-point appears due to the vehicle being parked, rather than due to a traffic stop.

Shortest path. The second constraint requires that, at any stage of the traversal, a path to a node must always be the shortest one (within a slack of 0.1 miles) from the start node to that node. The constraint is motivated by typical driving behavior where a shortest path is preferred when traveling short distances inside the city. In such cases, shortest paths are often fastest paths too. This is a reasonable

[15] wiki.openstreetmap.org

[16] crisp.cs.du.edu/datasets

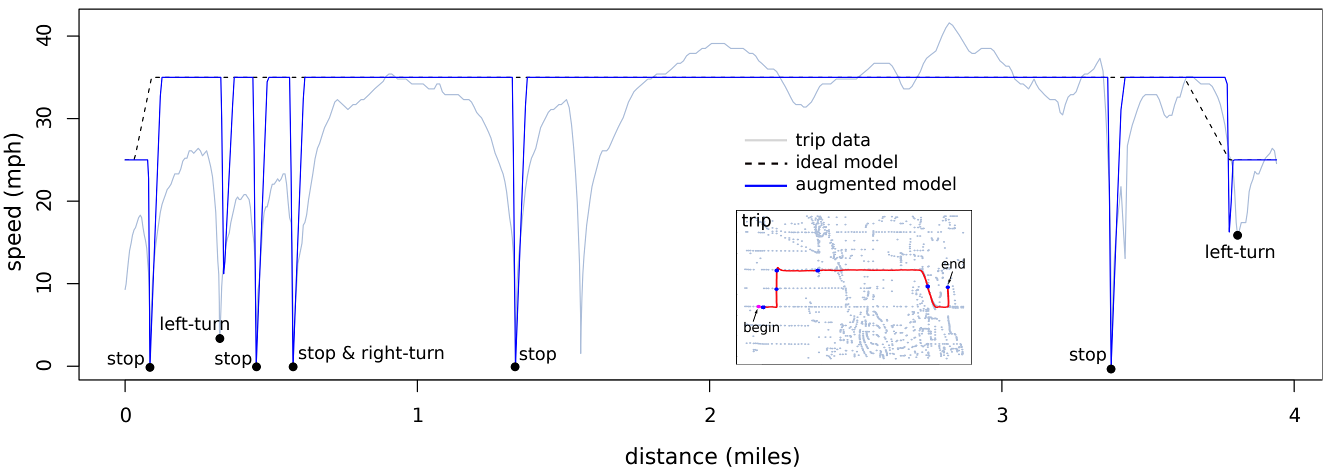

Figure 5: Speed profile for a trip, along with that generated from the ideal and the augmented models for a different path (differs in the first mile).

assumption in lieu of traffic conditions data at the time of the trip. However, the assumption fails when traveling long distances, where the driver is likely to take a faster (not necessarily shorter) route through the highway. Nonetheless, we can make the assumption that the driver would take the shortest route up to the highway, and then again from the point of exit on the highway to the destination. We incorporate this assumption by changing the start node to be the currently explored node, if the current node is part of a highway segment. As a result, the shortest path constraint remains disabled as long as the exploration continues on the highway nodes; the constraint is enabled when the exploration enters non-highway nodes, although the start node now is the last highway node (point of exit) on the path (Fig. 3).

Turn feasibility. The third constraint requires a path to always satisfy feasible speed limits at points of right and left turns. At every point of the exploration, we compute the angle by which a vehicle would have to turn when moving from the current node to the next node (Fig. 4). An angle higher than 60^o is considered a turn, in which case we consult the trip data to ensure that the speed at that point of time was under 25 mph. We use the current length of the path to extract the closest data point from the trip, and use the speed in that data point as the current driving speed.

Length. The length constraint terminates the exploration along a particular path when the path length exceeds the trip length. The path is then a candidate path if all stop-points (except the last one) have been matched in the path. When multiple candidate paths to the same end node are discovered, we retain the one with the least number of turns.

The nodes in our map graph correspond to points on roadways. However, the initial few data points (and the last few as well) of a trip may correspond to driving on a parking lot or a driveway. We used the GPS coordinates logged by the tracking device to manually discard some of these initial data points such that the first data point of a trip always corresponds to a node of the map graph. This processing is not required when more elaborate map data is used to generate the graph; many online services (e.g. Google Maps) already use commercial maps with data for parking areas, bikeways, and pedestrian paths.

4.3 Candidate ranking

The DFS traversal for a given trip outputs the candidate paths that satisfy the four constraints discussed above. We process the candidates through a ranking procedure to arrive at the top inferred destinations of a trip. The ranking procedure makes use of information on typical speed limits along the candidate paths to find ones that best match the speed changes observed in the trip data points. We begin by first creating an ideal speed model for each candidate, then augment the model with driving behavior typically seen when making turns, and then compute a probability for the observed trip data to have been generated from the model. The candidates are ranked based on decreasing order of the probabilities.

Ideal speed model. The ideal speed model of a path P is a representation of the speeds that an ideal driver would follow when driving along the path under ideal conditions. An ideal driver is considered to be one who drives at exactly the speed limit, and ideal conditions imply no acceleration or decelerations in the driving speed. The model can be formally expressed as a function M of distance d and a path P. The output of such a function is the legal speed limit at distance d from the beginning of path P (assuming speed limit is same along both directions of travel).

$$M(d, P) = s^{limit}$$

In a discrete representation, the ideal speed model is an array of distance and speed pairs at points where the speed limit changes along the path.

Augmenting the model. An ideal speed model can be improved by correcting the output speed in parts of the path where the vehicle would be performing a turn. Even an ideal driver in ideal conditions will decelerate to a reasonable speed to make a right or a left turn. A turn is assumed to happen exactly at the node joining the two edges that make the turn. We assume that all left turns happen at a speed of 15 mph and all right turns happen at 10 mph. The augmented model, denoted by M_{aug}, gradually reduces the output speed to the turning speed over a distance that depends on the acceleration and deceleration capabilities of the vehicle. Similarly, the model also incorporates the required acceleration behavior after the turn is complete. For all vehicles in this study, we use a fixed deceleration rate of 25 $feet/s^2$ ($= 7.8m/s^2 = 0.8g$, g being the acceleration of gravity), and a fixed acceleration rate of 6.5 $feet/s^2$ ($= 2m/s^2$).

The augmented model also incorporates the information that the vehicle must have come to a complete halt at all stop-points. Similar to the turns, the output speed is corrected around the vicinity of the stop-points as well. Fig. 5 compares the speed values from a trip, and the values generated from the ideal speed model and the augmented model along a similar path to the same destination.

Probability of a candidate path. Given a trip $\mathcal{T}$ with n data points, $\langle t_i, d_i, s_i \rangle; i = 1, ..., n$, and a path P, we obtain the speed values generated by the augmented model along path P at distances $d_1, ..., d_n$. We denote these values by $s_1^{'}, ..., s_n^{'}$. The probability we seek is

$$Pr\left[\mathcal{T} | M_{aug}(d_i, P) = s_i^{'}; i = 1, ..., n\right].$$

We assume independence of speed values across time and distance, which gives us the probability as

$$\prod_{i=1}^{n} Pr\left[\langle t_i, d_i, s_i \rangle | M_{aug}(d_i, P) = s_i^{'}\right].$$

Therefore, for each time instant t_i, we seek to compute the probability of observing speed s_i when the speed should have been $s_i^{'}$ at distance d_i along the path. The probability is computed from speed variation models based on standard Gaussian distributions. For speed value $s_i^{'}$, the distribution used is

$$f = \begin{cases} \mathcal{N}(s_i^{'} + \frac{s_i^{'}}{10}, \frac{s_i^{'}}{30}) & , s_i^{'} \geq 20\text{mph} \\ \mathcal{N}(s_i^{'}, 1) & , otherwise \end{cases},$$

where $\mathcal{N}(\mu, \sigma)$ signifies a Gaussian distribution with mean μ and standard deviation σ. The distribution implies that, for speed limits of 20 mph or more, the mean driving speed is 10% higher, and 99.7% of the drivers drive between speeds of $s_i^{'}$ and $s_i^{'} + s_i^{'}/20$. For example, in a road with speed limit 60 mph, most drivers are assumed to drive at speeds between 60-72 mph, with 66 mph being the mean. For lower speed limits, we assume that drivers are more likely to stay close to the limit. The probability is then computed as

$$Pr\left[\langle t_i, d_i, s_i \rangle | M_{aug}(d_i, P) = s_i^{'}\right] = \int_{s_i - \epsilon}^{s_i + \epsilon} f(x) dx,$$

where ϵ is a negligible number (10^{-5}). To avoid issues of precision, we take the sum of the logarithm of the probabilities instead of the product of the probabilities at different time instances. The ranking is not affected because of this transformation.

5. EMPIRICAL OBSERVATIONS

We applied the inference algorithm to the data from 30 trips. Inference correctness depends on factors such as stop-points, abidance to the shortest path assumption, ability to drive at speed limits, and the correctness of the map data. The algorithm was unable to generate any path leading to the actual destination in 12 out of the 30 trips. However, in 16 of the remaining 18 trips, the actual destination was always in the top three destinations (in fact the first one in 11 of them) generated after the ranking. The number of candidate paths ranged between 4 and 196 across the trips. Table 1 lists the trip length, number of candidate paths, and rank of actual destination for the 18 trips with successful inference. We are unable to find a correlation between the number of candidate paths and the ranking performance.

trip length (miles)	**number of candidates**	**rank of actual destination**
1.48	12	1
1.59	12	1
2.60	50	1
3.23	15	1
3.78	11	2
3.85	23	1
3.93	52	1
3.93	49	1
3.95	37	3
5.47	11	2
5.89	18	1
5.84	20	1
7.95	196	2
9.42	26	4
13.15	37	3
14.10	53	1
14.57	68	1
24.10	42	13

Table 1: Rank of actual trip destination from amongst the candidate paths.

5.1 Illustrative example

Fig. 6 shows five candidate paths identified for one of the trips. A total of 196 candidate paths were found for this trip. All candidate paths match the four stop-points of the trip (7.95 miles in length). Candidate path 118 is also the actual route taken during the trip. The last plot in the figure shows the end nodes (destinations) of all candidate paths. Irrespective of the large number of candidate paths identified for this trip, most destination nodes cluster around a small number of localities. This is worth noting, since only four stop-points are involved over a distance of 7.95 miles in this trip; yet the ways to match them to an actual map are quite limited!

Fig. 7 compares the speed profiles of the actual trip and that generated by the augmented model for a path. It is clear that the more similar the speed limits and turns along a path are to that of the actual route, the higher is the ranking. Candidate paths 9, 32 and 118 progressively cover more of the highway, thereby increasing the match probability.

5.2 Ranking performance

The ranking method is found to be robust in identifying the actual destination of a trip. If the destination is the end point of a candidate path, the path is often found in the three most likely paths that match the speed profile of the trip. Note that the ranking procedure does a point-by-point

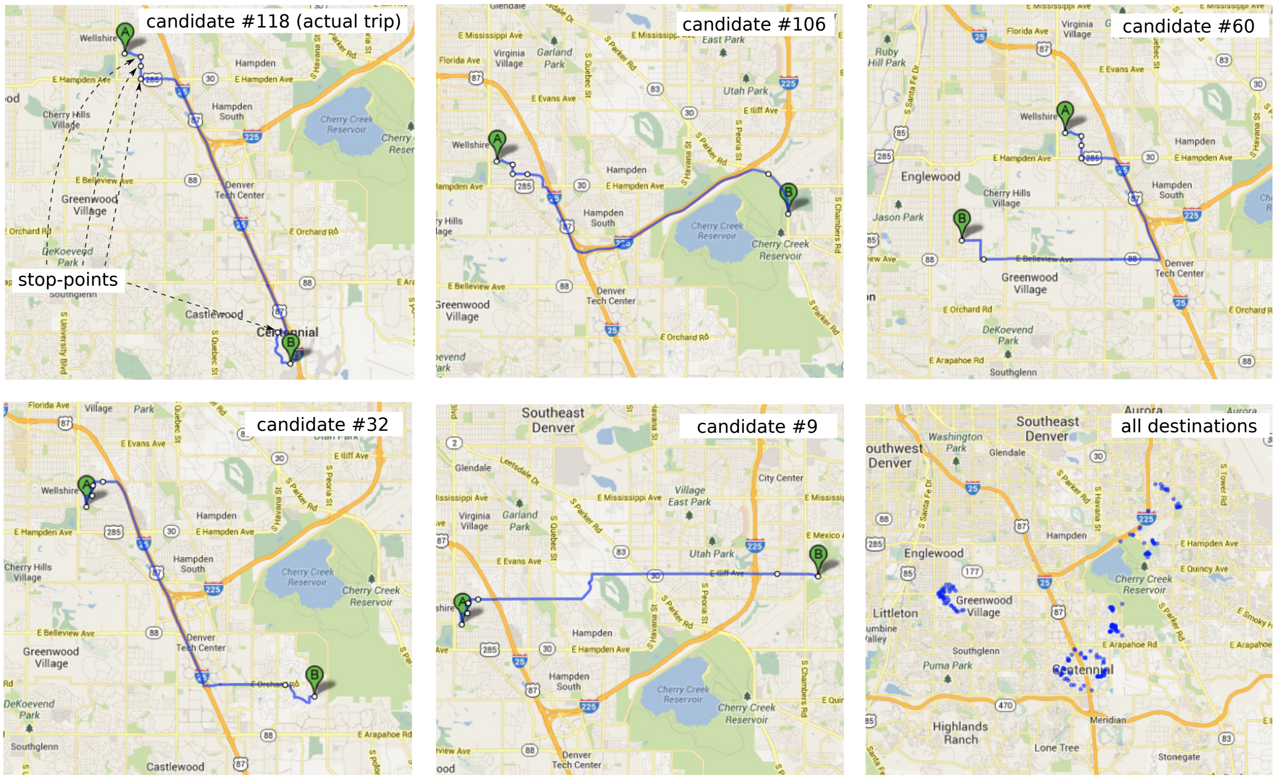

Figure 6: Sample candidate paths generated for a trip. Candidate path 118 is the actual route taken during the trip. The bottom right plot shows the destinations of all (196) candidate paths generated for this trip. A: start node; B: end node. Map data: Google (2013).

probabilistic comparison of the speed values observed in the trip and that along an entire path. Therefore, although we are not interested in the actual route followed during a trip, the obtained paths often represent the exact driving route. An interesting observation is that, even if the top ranked destination is not the actual one, they are usually very close (within 0.5 miles) to each other. Therefore, the locality of the destination can be inferred almost always! The ranking method suffers when speed limits are not reasonably followed, either due to excessive speeding or slow movement in traffic, and another candidate path matches this noisy speed profile.

5.3 Failed inferences

We also manually analyzed the 12 trips to understand why a path to the actual destination was not discovered during the DFS. For 4 out of the 12 paths, the trip involved a route that is not the shortest one. For most others, a stop was made for a significantly long amount of time in the middle of the road due to heavy traffic. Note that our traffic pre-processing looks for more than one stop within a small distance; if a single stop is made due to heavy traffic, we will instead interpret it as a stop-point. In one case, the search was unsuccessful due to errors in the map data. The shortest path issue can be resolved by allowing a larger slack on the constraint, although doing so may increase the number of candidate paths. Identifying single stops in the middle of a road segment due to traffic conditions is more difficult. An alternative is to allow a maximum number of violations of the stop-point matching constraint. We believe that understanding the factors underlying a failed inference is the key to creating a privacy-preserving technique for telematics data collection. For example, an auto-insurance data collection device that intermittently perturbs the detected speed of the vehicle for short durations can make the task of inference more prone to noise.

5.4 Summary

We summarize our observations in this study in the following points.

- Although multiple candidate paths may satisfy the stop-points and turn feasibility constraints, the number of neighborhoods where the paths end can still be limited.
- A robust ranking method can easily identify candidate paths that do not conform with the speed profile of the trip, possibly leaving behind ones that end near the true destination.
- The speed attribute in the collected data is a crucial component in the inference process. It is worth exploring how the data collection process can be modified to introduce noise in this attribute, of course, without affecting its intended use.
- Finally, it is possible to infer the destination (often the full route) of a trip from driving habits data such as speed and distance traveled. It is crucial that agencies

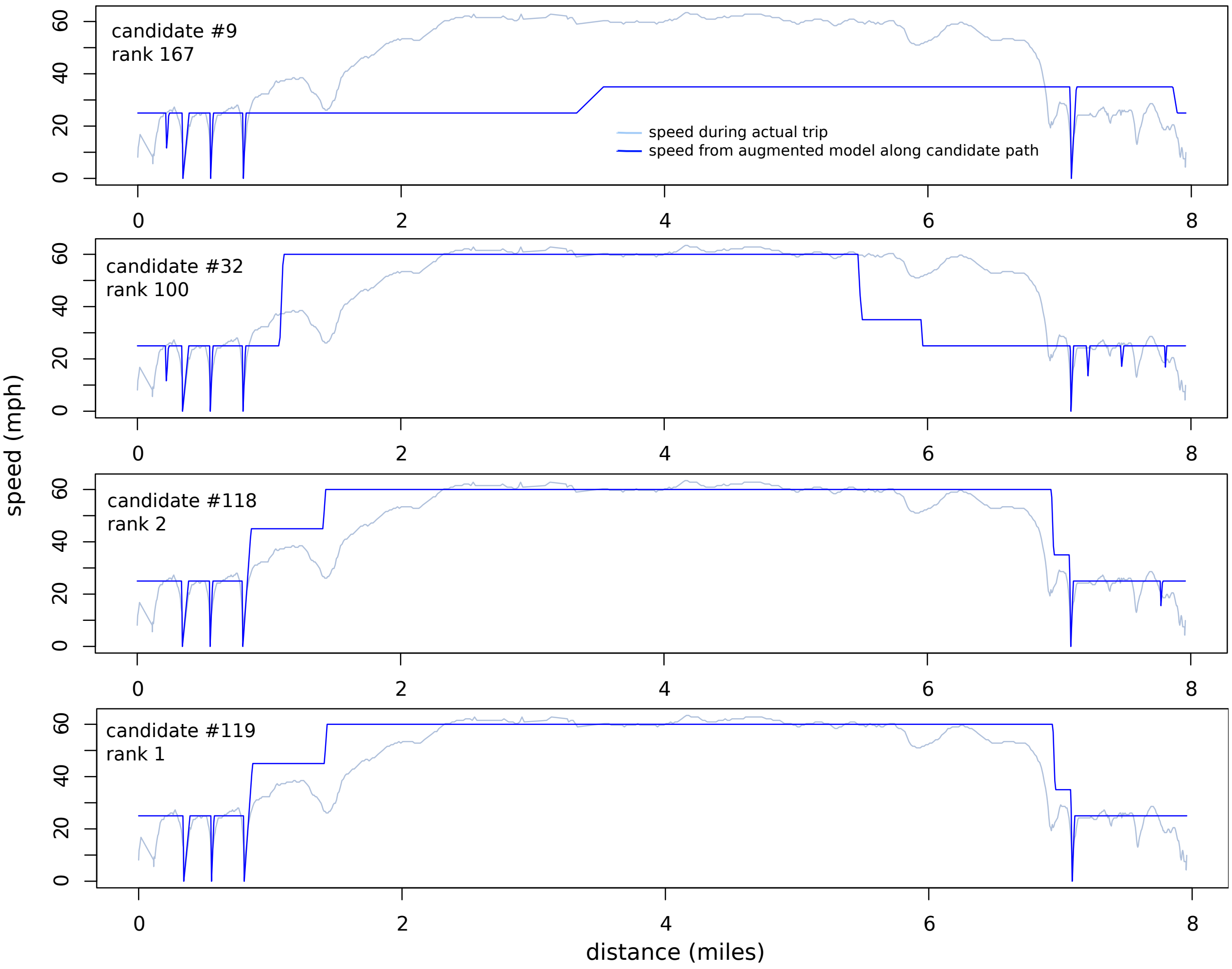

Figure 7: Speed profile during actual trip and that generated by augmented model for sample paths.

that collect such data acknowledge this fact and inform their customers about it.

6. CONCLUSIONS

In this paper, we studied the threat of location inference in vehicle telematics applications that collect driving habits data. We developed an inference algorithm to demonstrate that inferring the destinations of driving trips is possible with access to simple features such as driving speed and distance traveled. The algorithm does fail in some cases. However, we believe that communicating the existence of this threat to privacy is a priority to perfecting the algorithm. Privacy advocates have presumed the existence of location privacy threats in non-tracking telematics data collection practices; our work shows that the threats are real. It is unfortunate, but the difficulties in data collection/sharing due to quasi-identifiers is very much present in this domain as well. The design of privacy-preserving techniques for telematics data collection is open to research. In the meantime, enough information should be conveyed to consumers so that an informed decision can be made.

7. ACKNOWLEDGEMENT

This work is funded in part by the National Science Foundation under Grant No. DUE-0911991. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect those of the National Science Foundation.

8. REFERENCES

[1] Bamba, B., Liu, L., Pesti, P., and Wang, T. Supporting Anonymous Location Queries in Mobile Environments with Privacy Grid. In *Proceedings of the 17th International World Wide Web Conference* (2008), pp. 237–246.

[2] Barbaro, M., and Zeller, T. A Face is Exposed for AOL Searcher No. 4417749: New York Times. http://www.nytimes.com/2006/08/09/technology/09aol.html, 2006.

[3] Golle, P. Revisiting the Uniqueness of Simple Demographics in the US Population. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society* (2006), pp. 77–80.

[4] Golle, P., and Partridge, K. On the Anonymity of Home/Work Location Pairs. In *Proceedings of the 7th International Conference on Pervasive Computing* (2009), pp. 390–397.

[5] Gruteser, M., and Grunwald, D. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services* (2003), pp. 31–42.

[6] Hoh, B., Gruteser, M., Xiong, H., and Alrabady, A. Enhancing Security and Privacy in Traffic-Monitoring Systems. *IEEE Pervasive Computing 5*, 4 (2006), 38–46.

[7] Krumm, J. Inference Attacks on Location Tracks. In *Proceedings of the 5th International Conference on Pervasive Computing* (2007), pp. 127–143.

[8] Liu, F., Hua, K. A., and Cai, Y. Query l-Diversity in Location-Based Services. In *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware* (2009), pp. 436–442.

[9] Malin, B. Re-identification of Familial Database Records. In *AMIA Annual Symposium Proceedings* (2006), pp. 524–528.

[10] Narayanan, A., and Shmatikov, V. Robust De-Anonymization of Large Sparse Datasets. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy* (2008), pp. 111–125.

[11] Narayanan, A., and Shmatikov, V. De-Anonymizing Social Networks. In *Proceedings of the 2009 IEEE Symposium on Security and Privacy* (2009), pp. 173–187.

[12] Shokri, R., Theodorakopoulos, G., Boudec, J.-Y. L., and Hubaux, J.-P. Quantifying Location Privacy. In *Proceedings of the 32nd IEEE Symposium on Security and Privacy* (2011), pp. 247–262.

[13] Sweeney, L. Weaving Technology and Policy Together to Maintain Confidentiality. *Journal of Law, Medicine and Ethics 26*, 2-3 (1997), 98–110.

[14] Xu, T., and Cai, Y. Feeling-Based Location Privacy Protection for Location-Based Services. In *Proceedings of the 16th ACM Conference on Computer and Communications Security* (2009), pp. 348–357.