IJTE, Vol. 13, No. 4, Spring 2026, pp. 2339-2353
Research Paper

Estimating Urban Travel Times from Sparse GPS Data: A Practical ArcGIS-Python Framework

Alireza Ganjkhanloo[1], Mojtaba Rajabi-Bahaabadi[2,*]

Received: 2025/08/02 *Accepted: 2026/01/28*

Abstract

Accurate estimation of travel time across urban road networks is essential for effective traffic management, efficient route planning, and the development of intelligent transportation systems (ITS). In practice, GPS data is recorded at low sampling frequencies (sparse GPS data) due to limitations such as data storage and transmission costs. The low frequency of data poses significant challenges for traditional travel time estimation techniques, which often rely on continuous trajectories. To address this issue, the present study proposes a practical framework for estimating travel times at both the link and route levels using sparse GPS data. The methodology is implemented using Python scripting within the ArcGIS environment. A case study using GPS data collected from Tehran, Iran, is conducted to assess the performance of the proposed framework. The estimated travel times are validated against high-frequency GPS records, demonstrating that the approach yields accurate and reliable results despite the data sparsity. The integration of Python within ArcGIS enhances automation and makes the proposed framework both effective and accessible for real-world transportation analysis and planning.

Keywords: Low-frequency GPS Data, Map-matching, Path Inference, Travel Time Estimation

[*] Corresponding author. E-mail: mojtaba.rajabi@yazd.ac.ir
[1] School of Civil Engineering, College of Engineering, University of Tehran, Tehran, Iran
[2] Department of Civil Engineering, Yazd University, Yazd, Iran

1. Introduction

Awareness of travel time is essential for planning, designing, and managing transportation networks. Travel time is a critical factor influencing passenger decisions and is regarded as one of the most significant components of travel cost from the traveler's perspective [Gu et al., 2021]. Moreover, accurate travel time information plays a crucial role in the calibration of traffic simulation models, real-time traffic control, incident management, travel time prediction, and analysis of travel time reliability.

Traditionally, travel time has been estimated using fixed-location sensors such as video cameras, loop detectors, and Bluetooth sensors. While these technologies can provide reliable data, they are often associated with high installation and maintenance costs [Shao et al., 2025; Zhang et al., 2025]. Additionally, certain sensor types, such as loop detectors, may interfere with traffic operations during installation or maintenance, potentially causing increased delays or accidents. Most importantly, it is practically unfeasible to deploy these sensors across an entire urban road network, especially in large cities [Rahmani, 2015]. Due to these limitations, there is a growing need for alternative or complementary technologies that can support travel time estimation more efficiently and affordably.

Over the past two decades, GPS-equipped devices, including those embedded in smartphones and in-vehicle units of taxis, delivery vans, and service vehicles, have become a new class of mobile traffic sensors. In contrast to stationary sensors, GPS-equipped vehicles provide spatially distributed coverage across the network. They allow for continuous travel time and speed data collection along any segment of the road traversed by the vehicle, eliminating the need for direct traffic flow measurements [Antoniou et al., 2011; Jeneliusand Koutsopoulos, 2013]. Additionally, compared to fixed infrastructure like cameras or detectors, GPS technology has a lower initial implementation cost [Guo et al., 2021].

Most early studies on GPS-based travel time estimation such as those by Kumar et al. (2011) and Sun et al. (2007) relied on high-frequency GPS data, where data points are recorded at very short time intervals. However, high-frequency GPS data is typically expensive to obtain and often unavailable for large-scale studies [Sanaullah et al., 2016]. Consequently, recent research has shifted towards the use of low-frequency or sparse GPS data, as seen in the studies of Jenelius and Koutsopoulos (2013) and Rahmani et al. (2017). Despite its broader availability, low-frequency data introduces two major challenges. The first challenge stems from the long temporal and spatial gaps between consecutive GPS points, which makes the exact route taken by the vehicle between two points ambiguous. Multiple plausible paths may exist, especially in dense urban networks. The second challenge involves the disaggregation of travel time between consecutive points into the individual links that constitute the inferred path. Since the actual time spent on each link is not directly observed, estimating link-level travel time from sparse data requires effective allocation methods [Rahmani et al., 2017].

To overcome these challenges, most travel time estimation methods based on sparse GPS data are structured into two main components: 1) map-matching and path inference, and 2) travel time allocation and estimation [Rahmani et al., 2017]. In the first step, GPS points are projected onto a digital road network (map-matching), and the most probable route between consecutive points is identified (path inference). In the second step, the travel time between GPS observations is distributed across the links of the inferred path, and travel times are estimated accordingly.

Map-matching algorithms aim to align raw GPS trajectories with the underlying road network and determine the most probable position of a moving entity on the map [Jiang et al., 2022;

Singh et al., 2023]. Based on the characteristics of the available data, map-matching algorithms can be classified into four categories: geometric-based methods, topology-based approaches, probabilistic models , and advanced models [Huang et al., 2021]. Geometric-based methods rely on spatial relationships such as distance and direction between GPS points and road segments, often using techniques like least squares fitting, Frechet distance, or elliptical search regions. Topology-based map-matching algorithms rely on the structural layout of the road network and use graph-based representations to determine the most plausible path. These methods typically integrate local search techniques to identify candidate road segments. Probabilistic methods aim to maximize the likelihood of a correct match by considering temporal and spatial uncertainties through models such as Bayesian inference, fuzzy logic, and probabilistic trees. Finally, advanced models leverage artificial intelligence and machine learning techniques including support vector machines, neural networks, reinforcement learning, and genetic algorithms to handle complex scenarios and improve matching accuracy in sparse GPS datasets [Huang et al., 2021]. Note that geometric map-matching approaches, while simple, perform well only in uncomplicated networks and tend to degrade in dense urban areas with complex intersections and parallel links. As noted by Hashemi and Karimi (2014), these methods are sensitive to GPS noise and low-frequency sampling because they rely solely on spatial proximity and lack temporal or probabilistic modeling.

Path inference, which identifies the most probable route between two matched GPS points, has also been the subject of extensive research. Brakatsoulas et al. (2005) proposed incremental and global algorithms for low-frequency data, showing that the global method achieved higher accuracy at the cost of computation time. Lou et al. (2009) developed ST-Matching, a spatio-temporal algorithm suitable for data with 2–5-minute intervals. Hunter et al. (2014) used a Bayesian approach to construct and evaluate candidate trajectories. Rahmani and Koutsopoulos (2013) applied the A^* algorithm to identify the shortest and most probable path. Liu et al. (2017) introduced a spatio-temporal inference method and demonstrated improvements over ST-Matching. Ozdemir et al. (2018) framed path inference as an optimization problem, demonstrating its applicability for both high- and low-frequency data.

Following map matching and path inference, observed travel times derived from probe vehicle timestamps are allocated to individual links to estimate link-level travel time along the traversed paths. One of the simplest and most widely used methods assumes constant driving speed between consecutive sampling points, allocating travel time through linear interpolation based on reported positions and timestamps. While this approach performs well in freeway conditions with stable traffic flow, it becomes unreliable in urban environments characterized by frequent stops, traffic signals, and variable speeds [Neumann, 2014]. To address these challenges, more advanced models have been developed [Hofleitnerand Bayen, 2011; Puangprakhonand Narupiti, 2017; Zhengand Van Zuylen, 2013].

Despite advancements in theory and modeling, the practical implementation of travel time estimation methods often requires significant technical effort and specialized tools. In this study, we address this gap by presenting a practical and GIS-integrated approach for travel time estimation using low-frequency GPS data. The method is implemented within the ArcGIS environment, leveraging Python scripting for automation. This integration simplifies the overall process, making it more accessible for researchers and practitioners in transportation planning. The proposed methodology is applied to taxi GPS trajectory data collected in Tehran, Iran, and validated using high-frequency GPS data. The findings confirm the effectiveness of the proposed method in accurately estimating travel times, even in the presence of sparse data. The rest of this paper is organized as follows. Section 2 presents the methodology, followed by a description of the data and study area in Section 3. Section 4 discusses the results. Finally, Section 5 provides conclusions and potential future directions.

2. Methodology

Let $G=(V,E)$ be a directed graph representing the digital road network, where *V* is the set of nodes (intersections) and *E* is the set of directed edges (road links). The road network used in this study was derived from OpenStreetMap (OSM), which includes real-world operational constraints such as one-way streets, prohibited turns, and intersection-level maneuver restrictions. The extracted OSM network was first compared with the actual road conditions, and where discrepancies were identified, the network was corrected to ensure full consistency with the real-world layout.

Each probe vehicle is equipped with a GPS device that periodically transmits its location (latitude, longitude), timestamp, speed, and identifier. A GPS location is referred to as a GPS point, and the sequence of GPS points reported by vehicle *i* is denoted as $P_i=\left\{p_{i,1},p_{i,2},...,p_{i,j},...,p_{i,t}\right\}$. The actual route traversed by the vehicle, composed of an ordered set of links, is called the true path τ_{true}^{i}, which is unknown. The line connecting two consecutive GPS points is termed a line segment, and a GPS trajectory is defined as a set of consecutive segments.

The problem addressed in this study is the estimation of travel time using GPS trajectories from probe vehicles. This process comprises three main stages: (1) map-matching, (2) path inference, and (3) link travel time allocation and estimation. Each of these steps is described below.

2.1. Map-Matching

GPS points are often not located exactly on road links due to inaccuracies in the location data and the underlying digital road network. To address this, the map-matching process is used. This process identifies the vicinity of a GPS point and searches for a set of candidate links within a defined search area to which the GPS point can be reasonably matched.

As illustrated in Figure 1, consider a GPS point $p_{i,j}$ in a simple road network with three links. Within the neighborhood (or search area) of $p_{i,j}$, there are two candidate links: *Link 1* and *Link 2*. Let $x_{i,j}^{1}$ and $x_{i,j}^{2}$ be the matched (projected) points of $p_{i,j}$ on *Link 1* and *Link 2*, respectively. Let $o_{i,j}^{1}$ and $o_{i,j}^{2}$ denote the offsets, which are the distances from the starting node of each link to the matched points $x_{i,j}^{1}$ and $x_{i,j}^{2}$, respectively. These offsets represent how far along the link the projection falls. In the case of bidirectional roads, both directions are considered during the candidate selection. In this study, we implemented the map-matching algorithm proposed in by White et al. (2000), using ArcGIS software and the Python programming language. We adopted this algorithm because of its simplicity, computational efficiency, and its effective integration with ArcGIS geometry functions.

To describe the map-matching method used in this study, let *p* denote a GPS probe and let *C*(*p*) be the set of candidate links identified within an adaptive search region around the probe. In this study, a fixed search radius of 50 meters was used to identify candidate links, following the findings of Rahmani and Koutsopoulos (2013), which indicate that a range of 45–60 meters provides optimal performance for urban networks. Each link $k\in C(p)$ is represented in the digital road network as a polyline composed of ordered vertices $k=\left(S_k(0),S_k(2),\ldots,S_k(n_k)\right)$ where $S_k(0)$ and $S_k(n_k)$ are the upstream and downstream nodes, and the intermediate vertices define the link geometry. For each segment $\left(S_k\left(m\right),S_k\left(m+1\right)\right)$, the projection parameter is calculated as:

$$\lambda=\frac{\left(p-S_k\left(m\right)\right).\left(S_k\left(m+1\right)-S_k\left(m\right)\right)}{\left\|S_k\left(m+1\right)-S_k\left(m\right)\right\|^2} \tag{1}$$

The corresponding projected point on segment $\left(S_k\left(m\right),S_k\left(m+1\right)\right)$ is:

$$x\left(k,m\right)=\begin{cases}S_k\left(m\right)+\lambda\left(S_k\left(m+1\right)-S_k\left(m\right)\right), & 0\le\lambda\le1\\ S_k\left(m\right), & \lambda<0,\\ S_k\left(m+1\right), & \lambda>1.\end{cases} \tag{2}$$

The distance from the probe to the segment is:

$$D_{k,m}=\left\|p-x\left(k,m\right)\right\| \tag{3}$$

Finally, the matched point on link *k* is

$$x^k=\arg\min_m D_{k,m} \tag{4}$$

2.2. Path Inference

The second step is path inference. The aim of this step is to determine the most probable path that connects a sequence of candidate matched points. Since the time interval between consecutive matched points is short, it is reasonable to assume that vehicles follow the shortest path between the matched points. This assumption has been widely adopted in previous studies [Liu et al., 2017; Rahmaniand Koutsopoulos, 2013]. In this study, the shortest path refers to the path with the minimum expected travel-time cost rather than the shortest geometric distance. The use of travel-time–based link costs implicitly incorporates intersection delays, signal timing, and typical traffic conditions. Accordingly, the shortest path between each pair of consecutive matched points is considered the admissible path.

Since multiple candidate matched points may exist for each GPS point, the number of possible paths between a given origin–destination pair can be more than one. To handle this, a candidate graph is constructed in which candidate matched points are represented as nodes, arranged in layers corresponding to their timestamps, and edges connect nodes belonging to consecutive layers. Each edge represents the shortest-time subpath between two candidate matched points, consistent with the approach of Rahmani and Koutsopoulos (2013), and its cost is defined as the expected travel time of that subpath. Edges whose expected travel times exceed the observed time interval between consecutive GPS points (within a tolerance) are removed to ensure temporal feasibility. Following Rahmani and Koutsopoulos (2013), global spatial consistency is evaluated using measures such as the ratio of the inferred path length to the GPS trajectory length or the Fréchet distance between these paths. Similarly, global temporal consistency is assessed through the ratio of expected travel time on the inferred path to the actual observed travel time between GPS points. The most probable vehicle path is then selected by identifying the least-cost path in this candidate graph. More specifically, for every pair of candidate matched points, we compute the least-cost subpath, where the cost of each link is defined as its expected travel-time value. Figure 2 shows four GPS points recorded by vehicle *i*. The GPS points are shown by circles. The GPS points are first projected onto the network (i.e., map matching). The matched points are represented by multiplication signs. As can be seen in Figure 2, there exists one candidate matched point for GPS points $p_{i,1}$ and $p_{i,4}$. Furthermore, there are two candidate matched points for GPS points $p_{i,2}$ and $p_{i,3}$. Based on the candidate matched points, a candidate graph consisting of two admissible paths is constructed (i.e., Path 1 and Path 2). Finally, a path should be selected from the candidate graph. In this example, if we consider the length of the path as a criterion for selecting the most probable path, then Path 1 is selected because it is shorter than Path 2.

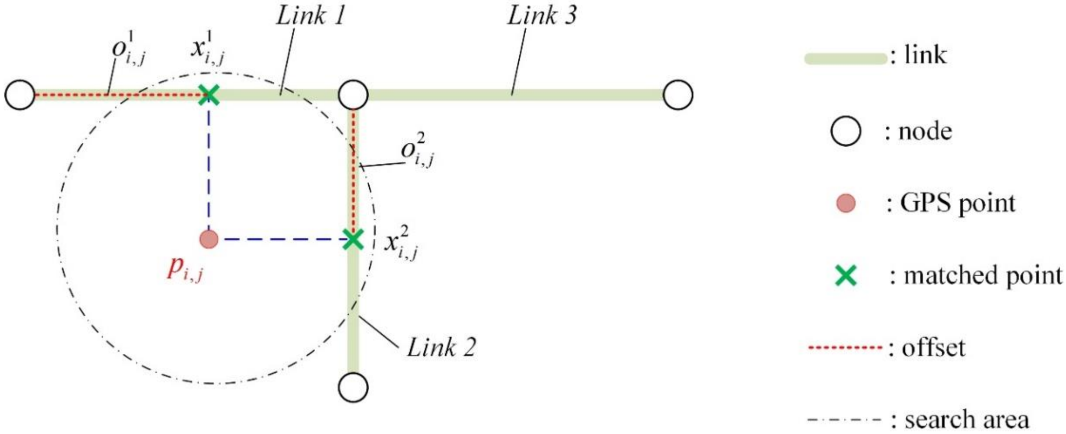


Figure 1. Map-matching of a GPS Point to Two Links

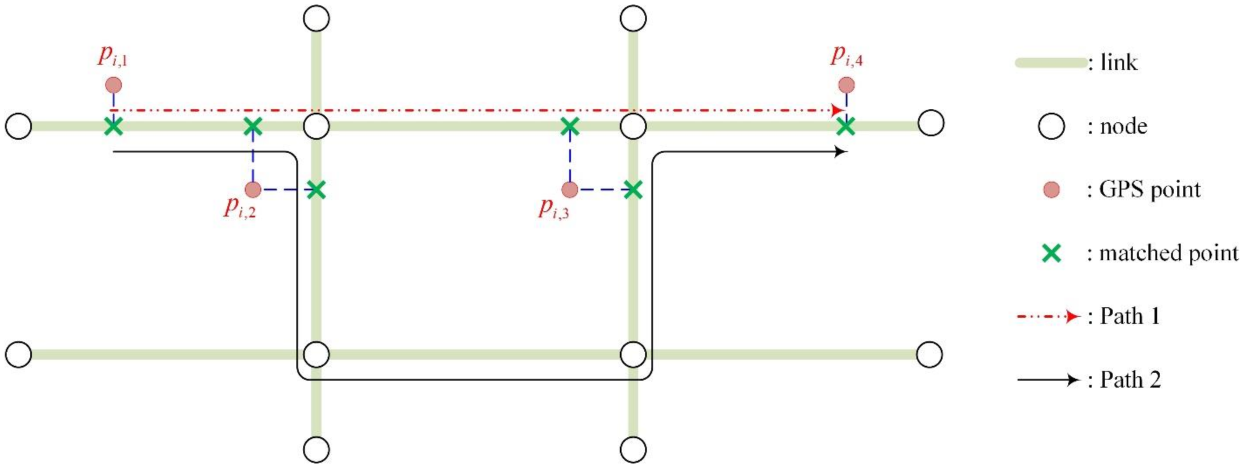


Figure 2. An Example of a Candidate Graph with Two Paths

In this study, path inference was performed in ArcGIS using Python and the ArcPy library. Matched GPS points were used to construct a candidate graph, and shortest paths were computed through Python scripts, which also automated the selection of the most probable path based on path cost.

2.3. Allocation and Estimation

The final stage is the estimation of link-level travel times. This process involves three sub-steps: travel time allocation, scaling, and aggregation.

Let $p_{i,j}$ and $p_{i,j+1}$ be two consecutive GPS points with timestamps $t_{i,j}$ and $t_{i,j+1}$, and matched to links k_j and k_{j+1}, respectively. The offsets from the start of the respective links to the matched points are $o_{i,j}$ and $o_{i,j+1}$. The observed travel time between the points is $\Delta T_i(j, j+1) = t_{i,j+1} - t_{i,j}$. Let $\eta(j \rightarrow j+1)$ be the path inferred between these points, and let l_k be the length of link *k*. The distance r_{jk}^i that vehicle *i* travels on link $k \in \eta(j \rightarrow j+1)$ is defined as:

$$r_{jk}^i = \begin{cases} o_{i,j+1} - o_{i,j}, & k = k_j = k_{j+1}, \\ l_k - o_{i,j}, & k = k_j \neq k_{j+1}, \\ o_{i,j+1}, & k = k_{j+1} \neq k_j, \\ l_k, & k \in \eta(j \rightarrow j+1) \setminus \{k_j, k_{j+1}\}, \\ 0, & \textit{otherwise.} \end{cases} \tag{5}$$

The allocated travel time $\hat{T}_k^i$ for each link $k \in \eta(j \rightarrow j+1)$ is calculated proportionally to the distance traveled:

$$\hat{T}_k^i = \Delta T_j \cdot \frac{r_{jk}^i}{\sum_{e \in \eta(j \rightarrow j+1)} r_{je}^i} \tag{6}$$

If the link was only partially traversed (i.e., $r_{jk}^{i} < l_k$), the travel time is scaled to estimate the full-link travel time:

$$\tilde{T}_k^i = \hat{T}_k^i \frac{l_k}{r_{jk}^i} \tag{7}$$

In Equation (7), the scaling assumes a uniform travel speed along each road link. While this assumption is widely used in sparse GPS-based studies such as Rahmani et al. (2013), it may not fully capture real-world variations such as acceleration and deceleration near intersections, stop-and-go behavior, traffic signal delays, or mid-link congestion. It should also be noted that several advanced techniques have been developed to explicitly address this limitation, most notably the approaches presented by Shi et al. (2017). In the present study, we adopt a practical approach in which each road link is subdivided into multiple smaller segments (sub-links) based on detailed road geometry derived from GPS trajectories and Google Maps traffic data. Each sub-link corresponds to a portion of the original link between consecutive vertices of the road polyline, and travel time is allocated proportionally along these sub-links. This method allows a more realistic modeling of speed variations and traffic conditions along the link, improving the accuracy of link-level travel time estimation.

In addition, during a given observation period, individual trajectories may only partially traverse a link, leading to varying coverage lengths across different observations. Consequently, the contribution of each observation to the overall link travel time estimation should be weighted according to the extent of coverage. To address this, the average travel time for a link is computed as a weighted mean of the scaled travel time estimates as follows:

$$\bar{T}_k^i = \frac{\sum_i \tilde{T}_k^i r_{jk}^i}{\sum_i r_{jk}^i} \tag{8}$$

This approach ensures that observations with greater coverage exert a larger influence on the final estimate, thereby improving the accuracy and robustness of travel time estimation.

2.4. Implementation in ArcGIS–Python

The proposed travel time estimation framework was implemented within the ArcGIS environment using Python scripting and the ArcPy library. The framework integrates map-matching, path inference, and travel-time allocation into a single automated workflow, as illustrated in Figure 3.

Raw GPS trajectory data were initially provided in CSV format. During preprocessing, ArcPy data management tools were employed to convert the GPS records into point feature classes and store all data within a file geodatabase. Simultaneously, the digital road network was prepared using OpenStreetMap (OSM) data. Topological and connectivity inconsistencies inherent in the raw OSM network, such as broken links, incorrect node connections, and missing or invalid turn restrictions, were identified and corrected to ensure a consistent and operational network. Furthermore, mid-journey stops can be identified by detecting GPS points exhibiting minimal spatial displacement coupled with large temporal gaps.

Map-matching was performed using ArcPy geometry and spatial analysis functions. For each GPS point, candidate road links were identified within a predefined search radius, and GPS points were projected onto candidate polylines to obtain matched locations and along-link offsets.

Path inference was conducted using ArcPy in combination with the ArcGIS Network Analyst extension. Matched GPS points were loaded as stops in the network dataset, and shortest-time paths between consecutive observations were computed using travel-time–based link costs while enforcing one-way restrictions and prohibited turns. The resulting sequences of traversed links were extracted and stored for subsequent processing.

Travel-time allocation was implemented using Python scripts. Observed travel times between consecutive GPS points were allocated to the links of the inferred paths in proportion to the distance traveled on each link. Allocated values were scaled to estimate full-link travel times. Outlier travel times, those deviating significantly from the majority of observations, were identified and removed using standard statistical criteria (e.g., mean ± 3σ). The remaining link-level travel times were then scaled to estimate full-link travel times, accounting for partial link coverage. Multiple observations were aggregated across all trajectories to generate final link-level and route-level travel-time estimates. All intermediate and final results were stored in the geodatabase, allowing direct visualization and further analysis within ArcGIS.

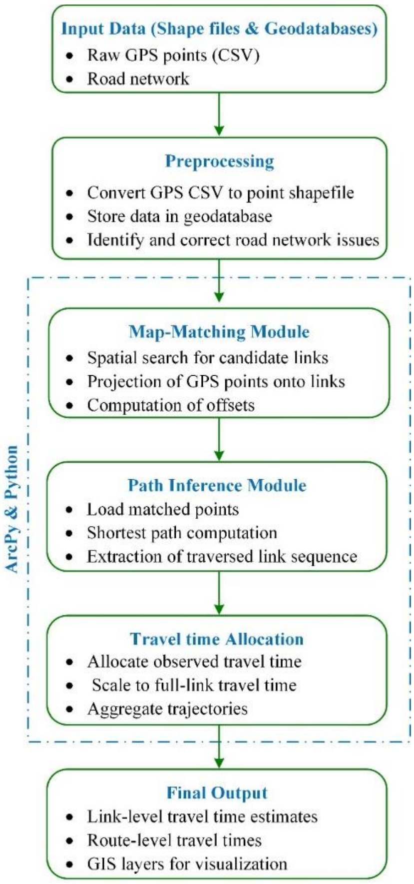


Figure 3. Flowchart of the ArcGIS–Python Framework for Travel Time Estimation

3. Data Description

This section presents the case study area and the dataset utilized in the research. The study was conducted on three major routes located in Tehran. These routes were selected due to their high traffic volumes and their importance in connecting key urban locations within the city. The first route (Route 1) has a total length of approximately 4.8 kilometers and consists of three links. The second route (Route 2) covers 7.2 kilometers and comprises four links. The third route (Route 3) has a total length of 11.3 kilometers and consists of six links. Figure 4 shows the selected routes, which encompass a diverse range of streets in terms of functionality, including major arterials, minor arterials, and collector roads. This functional diversity enables the analysis to capture a comprehensive spectrum of traffic conditions and network characteristics, ensuring that the findings of the study are generalizable beyond any single road type within the urban context. Table 1 summarizes the key characteristics of the three study routes, encompassing their length, number of links, functional class, and the number of intersections along each route.

Table 1. Key Characteristics of the Routes

Route	Length (*km*)	Number of Links	Functional Class
Route 1	4.8	3	Arterial & Collector
Route 2	7.2	4	Arterial & Collector
Route 3	11.3	6	Arterial & Collector

The data for this study were collected using GPS-enabled smartphones. A custom mobile application was installed on several smartphones, which were subsequently placed inside a number of taxis operating along the selected routes. The application recorded the geographic location of each taxi at a frequency of 1 Hz (i.e., one position per second), generating high-resolution trajectory data. Data collection was carried out during the morning peak hours, from 7:00 AM to 9:00 AM on working days, between August 26, 2017, and September 15, 2017.The high-frequency GPS data collected in this manner provide detailed spatial and temporal information on vehicle movement along the study routes. As such, travel times estimated from this dataset are considered highly accurate and are used as ground-truth travel time values in the evaluation process.

To simulate real-world conditions in which probe-vehicle data are often sparsely sampled, we down-sampled the high-frequency GPS dataset to a temporal resolution of 120 seconds, following Sanaullah et al. (2013) and Zheng and Van Zuylen (2013). However, this procedure does not fully represent the complexities of naturally sparse GPS data, such as irregular or non-random missing points. Using the resulting low-frequency dataset, we conducted the travel-time estimations and compared them with travel times derived from the original high-frequency trajectories, which served as ground-truth reference data for evaluating the proposed method. This comparison was performed using the full dataset as the objective was to directly evaluate estimation accuracy under reduced sampling frequency.

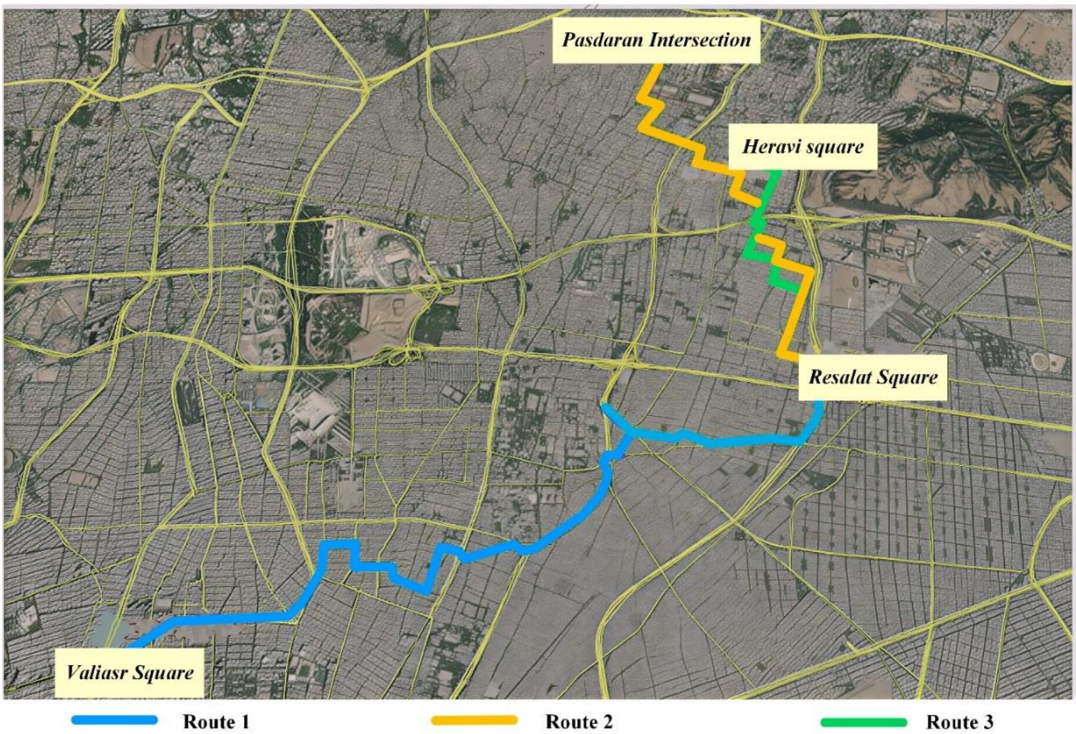


Figure 4. Selected Routes in the Study Area

4. Results

This section presents the results of travel time estimation at both link and route levels. First, the outcomes of the map-matching and path inference steps are discussed. Subsequently, the travel time estimation results for the studied routes and their constituent links are presented separately.

4.1. Map-Matching and Path Inference Results

The original dataset consisted of GPS points recorded at a frequency of 1 second. However, since this study focuses on estimating travel time using low-frequency GPS data, a subset of the data was extracted such that the time interval between consecutive GPS points was 120 seconds. The map-matching and path inference methods were then applied to this reduced dataset.

To quantitatively evaluate the performance of the map-matching and path inference, the percentage of correctly identified links per trip was calculated using the following equation:

$$T(\%) = 100\frac{|L_m \cap L_a|}{|L_a|} \quad (9)$$

where T is the proportion of accurately identified links, L_m is the set of links identified by the map-matching and path inference methods, and L_a is the set of actual links traveled by the vehicle, determined from the high-frequency data. Figure 5 shows the distribution of the correctly identified link percentages. The results demonstrate that the applied methods correctly identified more than 88% of the traveled links.

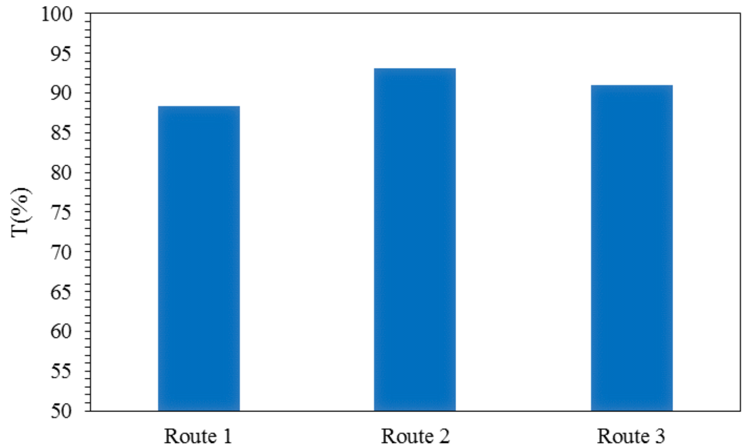


Figure 5. Percentage of Correctly Identified Links Using the Applied Method

4.2. Results of Link Travel Time Estimations

Travel times for each taxi on each link were estimated using the methods described in Section 3. To assess the accuracy of these estimates, the results were compared with true travel times derived from the high-frequency data. Figure 6 depicts the correlation between estimated and actual travel times for all links across the three routes. The high correlation coefficient indicates that link-level travel times were accurately estimated. The estimation errors were also analyzed to evaluate the accuracy of the proposed method. In this study, estimation error was defined as the relative difference between the estimated and actual travel times, expressed as a percentage. For a total of n =244 link-level travel time observations, 85% of travel time estimates exhibited an error of less than 10%, while 7% had errors greater than 20%.

Further evaluation of estimation accuracy was conducted using the Mean Absolute Percentage Error (*MAPE*) and the Normalized Root Mean Square Error (*NRMSE*), defined as follows:

$$MAPE(\%) = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{T_i^R - T_i^E}{T_i^R}\right| \quad (10)$$

$$NRMSE(\%) = 100 \times \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(T_i^R - T_i^E)^2}}{\frac{1}{n}\sum_{i=1}^{n}T_i^R} \quad (11)$$

where T_i^R and T_i^E are the actual and estimated travel times for link i, respectively, and n represents the number of observations.

These metrics were selected because *MAPE* offers an intuitive, percentage-based measure that allows comparison of estimation accuracy across links with different travel time values. In contrast, *NRMSE* provides a normalized measure of absolute error and assigns greater weight to larger deviations.

The calculated *MAPE* and *NRMSE* values were 6.34% and 10.34%, respectively. These relatively low values confirm the high accuracy of the implemented travel time estimation methods at the link level.

4.3. Results of Route Travel Time Estimations

After estimating travel times at the link level, the overall travel times for the three routes were also derived using low-frequency GPS data. Figure 7 presents a comparison between the estimated and actual route-level travel times for individual trips. The high coefficient of determination (R^2) and the slope of the regression line being close to 1 confirm the strong agreement between estimated and actual values, suggesting that the proposed approach reliably estimates travel times at the route level. Table 2 summarizes the *MAPE* and *NRMSE* values computed for each route. The small error values across all three routes further validate the accuracy of the methodology.

It is important to note that all computational steps of the proposed framework, including map-matching, path inference, and link travel time estimation, were executed on a standard desktop workstation equipped with an Intel Core i7 processor and 16 GB of RAM running Windows 10. For the three selected routes, the average processing time was less than1 minute. These results demonstrate the practical feasibility of the workflow for moderate-scale datasets. We note that scaling the framework to city-wide networks would require additional computational resources or parallel/distributed processing to maintain efficiency.

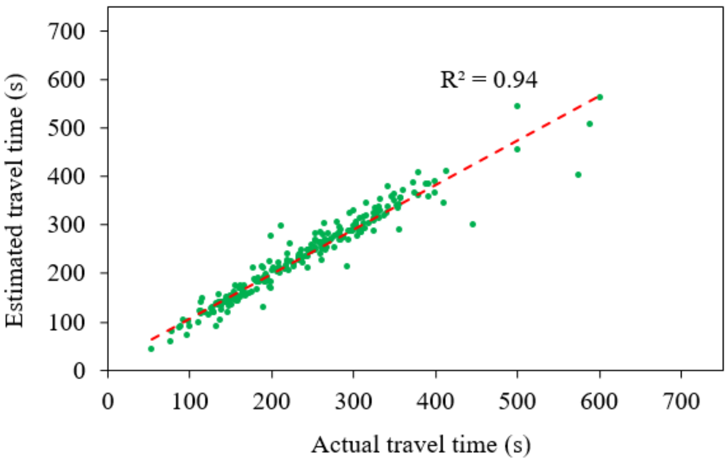
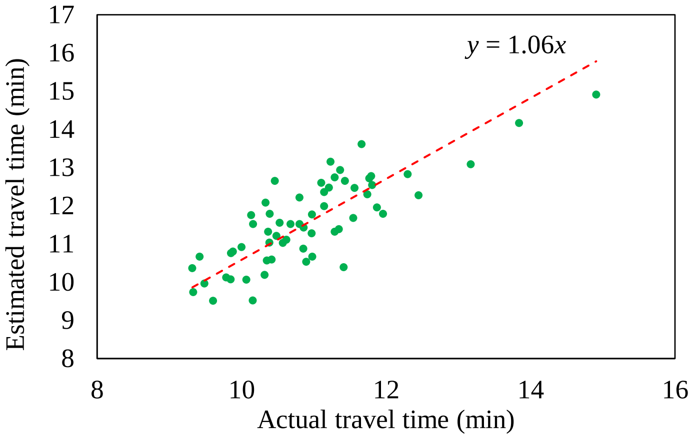
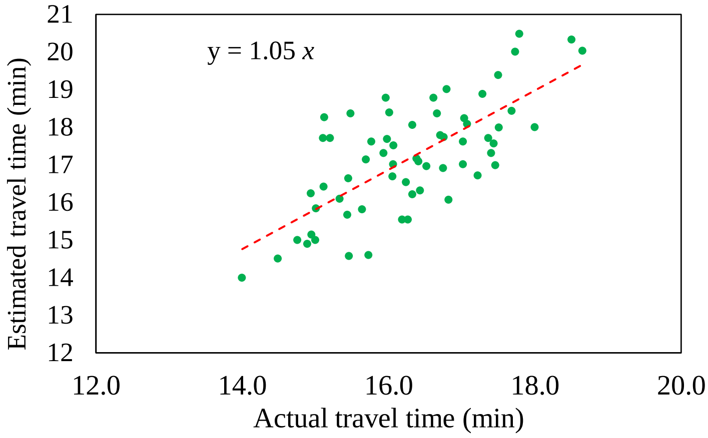


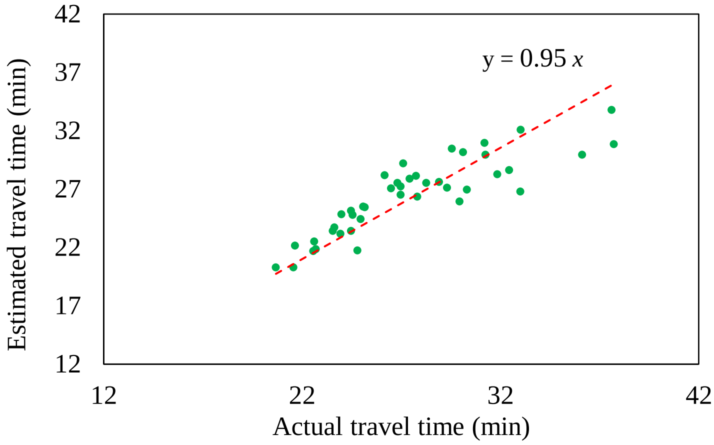
Figure 6. Estimated vs. Actual Link Travel Times



a) Route 1



b) Route 2



c) Route 3

Figure 7. Estimated vs. Actual Route Travel Times

Table 2. *NRMSE* and *MAPE* Values

Route	Number of observations	NRMSE(%)	MAPE(%)
Route 1	42	8.38	6.76
Route 2	62	9.88	6.92
Route 3	60	9.23	5.99

5. Conclusion

Travel time is a critical metric for evaluating the efficiency of transportation networks and traffic management systems. Traditional methods of collecting travel time data, such as Bluetooth sensors or loop detectors, can be costly and are typically limited to instrumented road segments. In contrast, GPS technology offers a more affordable and widely available data source. However, due to constraints related to data transmission costs, memory, and computational complexity, urban traffic management systems often rely on low-frequency GPS data. Therefore, accurate travel time estimation methods using sparse data are essential.

Since implementing travel-time estimation algorithms based on sparse GPS data typically requires advanced programming skills, many practitioners face considerable challenges in applying such methods in practice. To reduce these challenges and facilitate the implementation of travel-time estimation techniques, this study proposed an integrated workflow within the ArcGIS environment. By relying on built-in ArcGIS capabilities and requiring only minimal Python scripting, the workflow enables the full execution of map matching, path reconstruction, and travel-time estimation procedures. This approach allows transportation analysts and GIS users, even those with limited programming experience, to implement reliable and effective travel-time estimation methods within a familiar and operational platform. Therefore, the primary value of this research lies in its accessibility, ease of use, and seamless integration into practical workflows, rather than the introduction of a new algorithmic framework. Therefore, the primary contribution of this research lies in its accessibility, ease of implementation, and compatibility with practical GIS-based workflows, rather than in proposing a new algorithmic framework.

This study proposed a GIS-integrated approach to estimate travel times at both link and route levels using low-frequency GPS data. The methodology was implemented in the ArcGIS software environment using Python scripting. Results from the map-matching and path inference stages demonstrated the robustness of the applied methods, with over 88% of traveled links correctly identified. Most misidentifications occurred in network segments featuring short, parallel links arranged in grid-like patterns, where adjacent links were occasionally confused.

The travel time estimation results confirmed the high accuracy of the proposed approach. At the link level, 85% of travel times were estimated with less than 10% error. At the route level, small *MAPE* and *NRMSE* values further indicated reliable travel time predictions.

For future research, it is recommended to expand the analysis to rural road networks. It is also important to test the proposed tool on larger and more complex road networks. This should be supported by larger and more diverse datasets to evaluate its scalability, strength, and applicability in diverse traffic situations. While the current implementation effectively handles moderate-scale datasets, future applications to citywide or regional networks may require distributed or cloud-based computing architectures to ensure scalability and computational efficiency. Integrating such architectures would enable parallel processing of large GPS datasets and facilitate travel time estimation across extensive road networks. In addition, incorporating advanced map-matching techniques such as Hidden Markov Model-based approaches may improve the map-matching algorithm used in this study, particularly when dealing with sparse or noisy GPS data in dense urban environments. Moreover, the limited scale of the empirical dataset, consisting of only three representative routes, did not allow for a statistically robust spatial or temporal analysis of estimation errors. While the overall accuracy was high, identifying where the largest errors occur or whether they follow systematic patterns requires a richer dataset with more diverse link geometries and traffic conditions. Future work can incorporate larger and more heterogeneous GPS datasets to enable detailed error mapping and to better understand the method's limitations under varying traffic and network configurations.

6. References

- Antoniou, C., Balakrishna, R., & Koutsopoulos, H. N. (2011). A synthesis of emerging data collection technologies and their impact on traffic management applications. European Transport Research Review, 3(3), 139-148.

- Brakatsoulas, S., Pfoser, D., Salas, R., & Wenk, C. (2005). On map-matching vehicle tracking data. Proceedings of the 31st international conference on Very large data bases.

- Gu, J., Li, M., Yu, L., Li, S., & Long, K. (2021). Analysis on link travel time estimation considering time headway based on urban road RFID data. Journal of advanced transportation, 2021(1), 8876626.

- Guo, J., Liu, Y., Yang, Q., Wang, Y., & Fang, S. (2021). GPS-based citywide traffic congestion forecasting using CNN-RNN and C3D hybrid model. Transportmetrica A: transport science, 17(2), 190-211.

- Hashemi, M., & Karimi, H. A. (2014). A critical review of real-time map-matching algorithms: Current issues and future directions. Computers, Environment and Urban Systems, 48, 153-165.

- Hofleitner, A., & Bayen, A. (2011). Optimal decomposition of travel times measured by probe vehicles using a statistical traffic flow model. 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC).

- Huang, Z., Qiao, S., Han, N., Yuan, C. a., Song, X., & Xiao, Y. (2021). Survey on vehicle map matching techniques. CAAI Transactions on Intelligence Technology, 6(1), 55-71.

- Hunter, T., Abbeel, P., & Bayen, A. (2014). The path inference filter: model-based low-latency map matching of probe vehicle data. IEEE Transactions on Intelligent Transportation Systems, 15(2), 507-529.

- Jenelius, E., & Koutsopoulos, H. N. (2013). Travel time estimation for urban road networks using low frequency probe vehicle data. Transportation Research Part B: Methodological, 53, 64-81.

- Jiang, L., Chen, C., Chen, C., Huang, H., & Guo, B. (2022). From driving trajectories to driving paths: a survey on map-matching algorithms. CCF Transactions on Pervasive Computing and Interaction, 4(3), 252-267.

- Kumar, S. V., Vanajakshi, L., & Subramanian, S. C. (2011). A model based approach to predict stream travel time using public transit as probes. 2011 IEEE Intelligent Vehicles Symposium (IV).

- Liu, X., Liu, K., Li, M., & Lu, F. (2017). A ST-CRF map-matching method for low-frequency floating car data. IEEE Transactions on Intelligent Transportation Systems, 18(5), 1241-1254.

- Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., & Huang, Y. (2009). Map-matching for low-sampling-rate GPS trajectories. Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems.

- Neumann, T. (2014). Accuracy of distance-based travel time decomposition in probe vehicle systems. Journal of advanced transportation, 48(8), 1087-1106.

- Ozdemir, E., Topcu, A. E., & Ozdemir, M. K. (2018). A hybrid HMM model for travel path inference with sparse GPS samples. Transportation, 45(1), 233-246.

- Puangprakhon, P., & Narupiti, S. (2017). Allocating Travel Times Recorded from Sparse GPS Probe Vehicles into Individual Road Segments. Transportation Research Procedia, 25, 2208-2221.

- Rahmani, M. (2015). Urban Travel Time Estimation from Sparse GPS Data: An Efficient and Scalable Approach KTH Royal Institute of Technology].

- Rahmani, M., & Koutsopoulos, H. N. (2013). Path inference from sparse floating car data for urban networks. Transportation Research Part C: Emerging Technologies, 30, 41-54.

- Rahmani, M., Koutsopoulos, H. N., & Jenelius, E. (2017). Travel time estimation from sparse floating car data with consistent path inference: A fixed point approach.

Transportation Research Part C: Emerging Technologies, 85, 628-643.

- Sanaullah, I., Quddus, M., & Enoch, M. (2016). Developing travel time estimation methods using sparse GPS data. Journal of Intelligent Transportation Systems, 20(6), 532-544.

- Sanaullah, I., Quddus, M. A., & Enoch, M. P. (2013). Estimating link travel time from low-frequency GPS data Transportation Research Board 92nd Annual Meeting, Washington DC, United States.

- Shao, M., Wang, Z., & Peng, J. (2025). Estimation of Average Travel Speed on Urban Streets Based on Travel Time Distribution Characteristics. Transportation Research Record.

- Shi, C., Chen, B. Y., & Li, Q. (2017). Estimation of travel time distributions in urban road networks using low-frequency floating car data. ISPRS International Journal of Geo-Information, 6(8), 253.

- Singh, S., Singh, J., Goyal, S., El Barachi, M., & Kumar, M. (2023). Analytical review of map matching algorithms: analyzing the performance and efficiency using road dataset of the indian subcontinent. Archives of Computational Methods in Engineering, 30(8), 4897-4916.

- Sun, D., Luo, H., Fu, L., Liu, W., Liao, X., & Zhao, M. (2007). Predicting bus arrival time on the basis of global positioning system data. Transportation Research Record, 2034(1), 62-72.

- White, C. E., Bernstein, D., & Kornhauser, A. L. (2000). Some map matching algorithms for personal navigation assistants. Transportation Research Part C: Emerging Technologies, 8(1-6), 91-108.

- Zhang, C., Zhou, Y., Zhang, M., Wang, B., & Nie, Y. (2025). Review and prospect of floating car data research in transportation. Journal of Traffic and Transportation Engineering (English Edition).

- Zheng, F., & Van Zuylen, H. (2013). Urban link travel time estimation based on sparse probe vehicle data. Transportation Research Part C: Emerging Technologies, 31, 145-157.