Better Route Recommendation for Goods Delivery Based on Local Users' Historical Travel Patterns

Sudha Chaturvedi[1], Tapsi Nagpal[1], Vishnu Shankar Tiwari[2], Praveen Kumar S M[2]

[1]Department of Computer Science and Information Technology, Lingayas University, India
[2]Department of Computer Science and Engineering (AI&ML), PES University, India

ABSTRACT

Route recommendation and route prediction have evolved on a large scale with the proliferation of location-capturing devices and maps. Food, grocery, and goods delivery from online platforms are using route recommendation applications to guide their delivery boys on the route to be followed for delivery. But many times it is observed that recommended routes have many blockers like dead-end, temporary repairing of roads, bad road conditions, poor lighting, and security issues etc. In some extreme scenarios, it is noticed that users are recommended to follow a route that leads to the wrong destination, and then they must search for the path again and drive towards the destination. It is observed that when these items are ordered from local shops in the area, the delivery boys follow the route based on their knowledge and don't face these kinds of issues, as they know the area and roads better. A typical route recommendation system uses static parameters like shortest paths from source to destination, and dynamic road conditions like current traffic congestion, roadblocks, temporary route diversion, etc. User historical travel data is available in abundance, captured during travel using location traces capturing devices like GPS, mobile phones, PDAs, etc. This article presents a route recommendation system that leverages the historical user travel pattern data combined with dynamic attributes to recommend better routes for delivery boys.

Keywords: Route, Recommendation, Prediction, Open Street Map, Travel, Pattern

INTRODUCTION

Route recommendation applications are frequently used by travelers for their route planning for commuting. Travelers try to optimize their route for commuting for short distances to minimize commute time while also avoiding roads with various unwanted scenarios like bad road conditions, poor lighting conditions, security issues, etc. It is observed that local users, like residents and taxi drivers, are more aware of the road conditions and follow a much better route with better planning, but avoid road conditions like poor road quality, security, and poor lighting on the road [1]. These users' travel patterns can be captured using location devices like GPS and cell phones. Over a period, a large volume of location data accumulated on the server, which contains the user travel pattern behavior [2] [3]. This travel pattern behavior can be captured and fed into the prediction systems. A sample route suggested by the automated route recommendation system is shown in Figure 1a, which has poor road conditions, but local users follow a better route for the same source and destination as shown in Figure 1b.

Another example of route recommendation suggested a route that has a dead end. This is often reported in the news media and newspapers, where travelers had to face issues, and then they travel back and take an alternative path. The route suggested by the route recommendation system is shown in Figure 2a, which has a dead end. For the same destination, a better route followed by a local traveler is shown in Figure 2b.

Also in route planning, security is a major concern for travelers, especially when the place they are visiting is new to them. In such scenarios, travelers prefer to avoid the route that has security concerns. Figure 3a shows a scenario where the route suggested by the route panning application has security concerns, but residents and travelers are following better routes, as shown in Figure 3b.

Road conditions can be categorized into two kinds, namely static and dynamic. Scenarios shown in Figure 1, the road has a bad condition, and Figure 2, the road has a dead end, are known as static conditions and don't change frequently. There are other conditions that can lead to bad experiences that change very frequently and for short periods. For example, road digging by municipal authorities for maintenance of sewer lines, Underground cables, clearing blocked underground cables, clearing underground sewer lines, etc.

Figure 1a. Route suggested by the automated route system has poor road conditions

Figure 1b. Local users travel on a better alternative path

Figure 2a. The suggested route has a dead end

Figure 2b. local users follow a better alternative route

Figure 3a. Suggested route has poor security condition

Figure 3b. local users follows better alternative route

DATASET FOR ROUTE RECOMMENDATION

Route prediction and route recommendation are very long-researched areas. The research ranges from simple graph-based algorithms (e.g., Dijkstra and Tree-based searches) to modern Artificial Intelligence and Machine Learning based algorithms. Traditional algorithms model the road network as a graph model, which consists of edges that model road segments and vertices that model important places and road intersections. Weights are assigned to edges of the road network, and search algorithms are run on the graph to compute the route [5]. These algorithms have limitations, don't scale well, and don't consider dynamic route conditions. Modern heuristic search algorithms (e.g., A*, LPA*) [6] [13] overcome the shortcomings of traditional models but still don't utilize the historical travel pattern of users and are based only on static conditions. Route prediction was proposed by Froehlich et. al. [1], which is based on historical user travel patterns. The GPS coordinate logs are broken into smaller units called trips. Trips are then used to cluster them based on similarity score, and then the clustered trips are used to predict the route. These trips in this work are raw GPS coordinates and don't use mapping to the road network, which leads to inaccuracies and clustering and affects route prediction. Historical travel pattern-based models, Prediction by partial match (PPM) [13], Map-Reduce based scalable Lempel-Ziv (LZ) [15], Distributed Context Tree Weighting (CTW) [10], and probabilistic generalized suffix tree (PGST) [17] predict the route followed based on historical GPS data.

Road Network Data

The route recommendation system proposed in this work makes use of two sets of data - digitized road network and GPS location traces. Road network in graph form is composed of a set of Vertices (V) representing road junctions and important places. The relationship between the vertices is represented by Edges (E). For example, if it's possible to reach directly from one vertex to another, then there is an edge between them. It may be possible that there is no edge directly between the vertices, but through transitivity of edges, another vertex can be reachable, which is known as a path. The overall road network graph (G) is represented by $G(V, E)$ [12]. In this work, the road network data is obtained from Open Street Map (OSM). OSM is an open-source platform that hosts various kinds of geographical data, like road networks, water bodies, and domestic and international boundaries of the world [8] [14]. Data can be visualized on the provided web-based interface and can be downloaded and stored in spatial-enabled databases like PostGIS. Data rendering tools like GeoServer can connect to spatial databases and render the road network. Road network data captured from OSM and stored in a spatial database is shown in Figure 4.

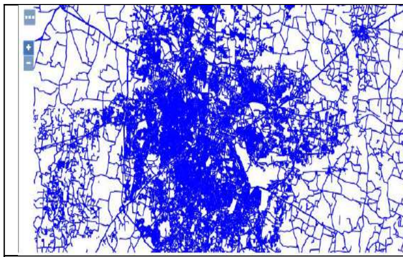	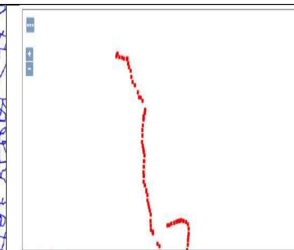	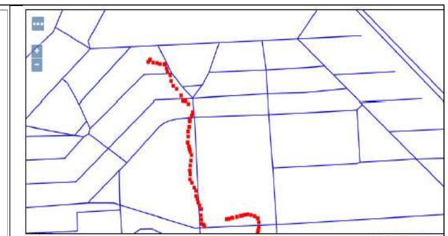
Figure 4. Road network data stored in PostGIS	Figure 5. User trip in GPS coordinates	Figure 6. User trip mapped to the road network

Travel History Data

Location traces are a continuous sequence of latitude/longitude coordinates captured at discrete intervals represented by $(x_0, y_0, t^0), (x_1, y_1, t^1), (x_2, y_2, t^2) \dots \dots \dots . (x_n, y_n, t^n)$. Users' travel history is a continuous log of location data in the form of (x_i, y_i, t^i) where x_i represents latitude and y_i represents longitude at time t^i. The log entry represents the user travel sequence$(x_0, y_0, t^0), (x_1, y_1, t^1), (x_2, y_2, t^2) \dots \dots \dots . (x_n, y_n, t^n)$. This continuous log represents multiple trips made by users, for example, a user travels from home to office, then to market, and then travels back to home. This makes three trips. In this work, the location traces used are from Microsoft Research's Geolife project [15] [16] [11]. The project captured data from various parts of the world over a period of more than 5 years for hundreds of users to capture their travel pattern behavior.

For this work, data were collected from various parts of the Bengaluru area in India, and a majority of the data was collected from local vendors who deliver groceries, food items, and other goods to various parts of Bengaluru and the Delhi area in India. A sample of raw GPS data is shown in Figure 5.

Continuous logs for GPS traces are decomposed into smaller units called trips. These trips, composed of GPS coordinates, are mapped and matched to the road network to convert trips to a sequence of road network edges [4]. The process of mapping location traces to road network data is known as map-matching [18]. Trip as a sequence of location triplet data (x_i, y_i, t^i) is mapped to road network edges. A trip as a sequence of location traces data map matched to road network edges is as shown in Figure 6.

Then these trips, composed of road network edges, are used to develop prediction models which can then be used for route prediction and route recommendation. These models are purely based on static road attributes from the historical location traces corpus and don't use dynamic road conditions like temporary roadblocks, temporary digging of roads, etc., which leads to the recommendation of infeasible routes. The proposed work uses both static and dynamic road conditions for route recommendation, which can predict better routes.

RECOMMENDATION MODEL AND ROUTE RECOMMENDATION

Route recommendation is the process of suggesting the route to the end user based on optimization of criteria like shortest path, good road conditions, road with better security, no roadblocks, etc., for better user experience [19]. The criteria for optimization can be of two types: static and dynamic conditions. Examples of static conditions are – shortest path, good road conditions, better security, and examples of dynamic conditions are temporary roadblocks, excavation of roads for maintenance, etc. Many existing route recommendation systems are only based on static road conditions and don't leverage dynamic road conditions, resulting in poor road conditions [20]. Additionally, they depend on only current parameters, like the shortest path, and don't include the historical travel patterns of users. But it is noticed that local users know the conditions better, both static and dynamic conditions, and follow a better path for the same source and destination, which is better than suggested by route recommendation systems. The proposed system utilizes the historical travel pattern of the user to recommend a better path. This section deals with model building for prediction and the process to utilize the model for route recommendation.

Trips as a sequence of location traces converted to a sequence of road network edges using map matching are as described in the previous section.

$$T((x_{0,}y_{0,}t^0), (x_{1,}y_{1,}t^1), (x_{2,}y_{2,}t^2) \dots \dots \dots (x_{n,}y_{n,}t^n)) \rightarrow T(e_0, e_1, e_2, \dots \dots . e_n)$$

Where $(x_{i,}y_{i,}t^i)$ is the location coordinate of the user and $(e_0, e_1, e_2, \dots \dots . e_n)$ are edges of the road network to which the location coordinates are mapped [20]. Trips as a sequence of road network edges are used to construct the prediction model. For each trip source and destination, the path traversed is recorded.

A prefix tree is a TRIE data structure that leverages the common prefixes in the user trips to calculate the frequency with which a road segment is traveled. Each node of the tree represents a sequence of edges traveled by the user in the road network graph. Edges of the tree are labeled with road edge ID and the frequency with which the edge is traversed in historical user travel. While traversing the tree, the first edge defines the start road segment for the user trip and the second edge defines which edge is taken from the child node and so on. The algorithm for model construction is as in $Algorithm\ I$.

Algorithm I: *Route recommendation model construction*

1. *Instantiate an empty graph* $\boldsymbol{G}\ (\boldsymbol{V}, \boldsymbol{E})$
2. *For each* $\boldsymbol{Trip_i} \in \boldsymbol{Trip_1}, \boldsymbol{Trip_2} \dots . \boldsymbol{Trip_n}$
3. *For each sub-path* $\boldsymbol{s_j} \in \boldsymbol{Trip_i}$
4. *If* $\boldsymbol{s_j} \in \boldsymbol{G}$ *then*
5. *for each edge* $\boldsymbol{e_i} \in \boldsymbol{s_j}$ *increase weight of* ***edge*** $\boldsymbol{W(e_i) = W(e_i) + 1}$
6. ***else***
7. *for each edge* $\boldsymbol{e_i} \in \boldsymbol{s_j}$ *insert* e_i *in* G *and set weight* $\boldsymbol{W(e_i) = 1}$
8. *For each edge* $\boldsymbol{e_i} \in \boldsymbol{E}$ $where\ E\ is\ set\ of\ edges\ in\ graph\ \boldsymbol{G}$
9. *If edge* $\boldsymbol{e_i}$ *has a blocker, then set edge weight* $\boldsymbol{W(e_i)} = -\infty$

An example of a model constructed from a user travel data corpus is shown in Figure 8. Searching in a model based on a prefix tree is extremely fast and the number of comparison steps to be performed in the tree is the same as the number of edges that appeared in the prefix path in the user trip. This model is hence suitable for handling a large graph and performance is pretty fast, which makes it suitable for practical implementation.

The prefix tree model handles common route prefixes efficiently, but the common shared parts are still stored separately in the model. For example, for trips $e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_6 \rightarrow e_7$ & $e_1 \rightarrow e_8 \rightarrow e_4 \rightarrow e_6 \rightarrow e_7$ the suffix part $e_4 \rightarrow e_6 \rightarrow e_7$ will be stored separately and further can be minimized for storage and faster retrieval. A prefix tree model is an acyclic deterministic finite automata (DFA) for which there exists an algorithm to minimize the number of nodes in the model. Minimizing the number of nodes reduces the size of the data structure, resulting in less storage memory and faster retrieval. The minimized DFA for the graph in Figure 7 is shown in Figure 8.

Similar to a prefix tree, there is another version $suffix$ of a trip that is a contiguous sequence of edges belonging to a set of substrings and occurs at the end of the trip. In this example, $suffix$ of the trip are $\{e_2 \rightarrow e_3,\ e_1 \rightarrow e_2 \rightarrow e_3\}$. A prediction model can be constructed using either the prefix or the suffix of the trip segments.

It can be noticed that all prefix paths starting from the start node of the trip are captured correctly, but other paths in the suffix paths are not captured. These are the drawbacks of all prefix-based models like PGST [17]. For example, if route is queried for source e_1 and destination $e_{4,}$ then the model will recommend $e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4$ as it is captured in the trip prefix for $Trip_1$. But the trip prefix path-based model cannot recommend a path for source e_4 and destination $e_{7,}$ but this information is contained in trip data as a trip subpath traversed during traveling $Trip_2$. This can be answered by a suffix path-based model as $e_4 \rightarrow e_5 \rightarrow e_6 \rightarrow e_7$ from the trip subpath contained in $Trip_2$. Similarly, all suffix paths ending in the destination node of the trip are captured correctly, but other paths in prefix paths are not captured. These are the drawbacks of all suffix-based models like LZ prediction model [15], Prediction by partial match (PPM) [13], Context Weighted Tree (CTW) [10], etc. If both prefix and suffix models are merged, then all prefix-based routes starting from the source node to all other nodes are captured correctly, and all suffix-based routes starting from all nodes to destination nodes are captured, and recommendations can be made for all queries contained in either the trip prefix or trip suffix.

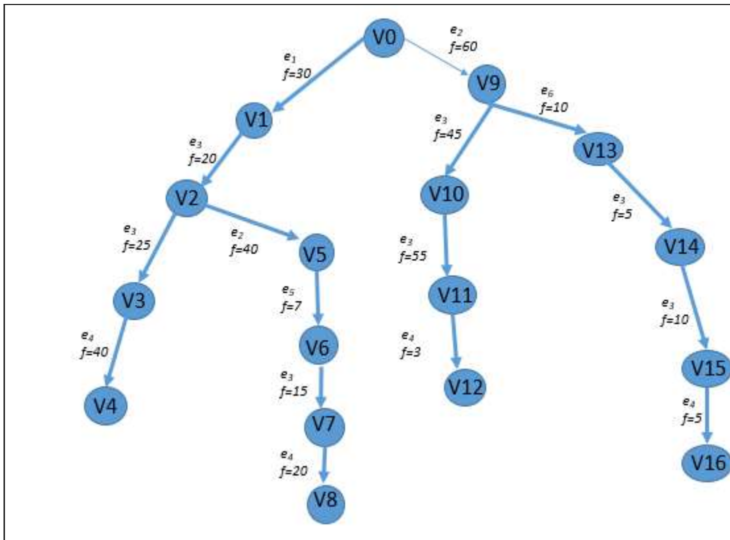

Figure 7. Route prediction model based on route prefixes

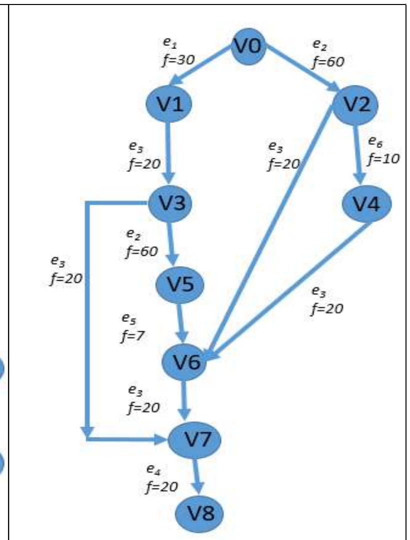

Figure 8: Minimized DFA prediction model

The proposed model combines all prefix paths, all suffix paths and all intermediate paths and captures all scenarios. This model contains all paths traveled and can answer all the queries, including all the scenarios discussed in the above section. The paths from all the nodes to all the nodes are captured in the model and edges are labeled with the frequency of the edges traveled in the past. There can also be scenarios where, in the past, the route was traversed by users, but due to some dynamic conditions like a road block, users are not traversing the path in the current scenario. Then the edge travel frequency is labelled with a negative weight.

While querying for the recommended path from a source to a destination, the resultant model is traversed from the source node to the destination node with optimization on the frequency of the edges traversed in the past. Given this, when the model is traversed for a recommended route, then the edge with negative weight is excluded, and the next best route is recommended. Below are the scenarios for the recommended path for the model constructed in Figure 8.

- **Scenario I:** Route queried is with source vertex v_0 and target vertex v_8, recommends traveling on edges $e_2 \rightarrow e_3 \rightarrow e_{4,}$ the reason is that most of the people are following that path.
- **Scenario II:** Path through vertex v_0 and v_4 that is edge e_6 is never suggested as the edge has blockage and no one is traveling.
- **Scenario III:** Recommended path between vertices v_0 and v_{16} is $e_2 \rightarrow e_6 \rightarrow e_3 \rightarrow,$ with a warning that is only the edge connecting these two vertices, but no one is traveling on edge e6.
- **Scenario IV:** No path is suggested between vertices v_0 and v_4 as the edge between v_{10} and v_{11} has a road block, which is a dynamic road condition.

Results and Conclusion

Existing models for route recommendation systems like Krum [1] are based on route recommendation, which captures only end-to-end trips, and intermediate routes are missing, but in this model, this is not an issue. Models like PGST [17], LZ [15], PPM [13], and CTW [10] capture intermediate paths as well, but don't capture all the intermediate routes. The proposed model captures all intermediate paths and hence performs better in accuracy, but the number of sub-paths to be computed is larger in number and hence takes a larger computation time. Additionally, existing models only use static conditions and hence computation time is lesser, whereas the proposed model resolves the dynamic conditions as well and hence takes a longer computation time, but the accuracy obtained is better than existing models. Model computation time is captured in Figure 8, and accuracy is captured in Figure 9.

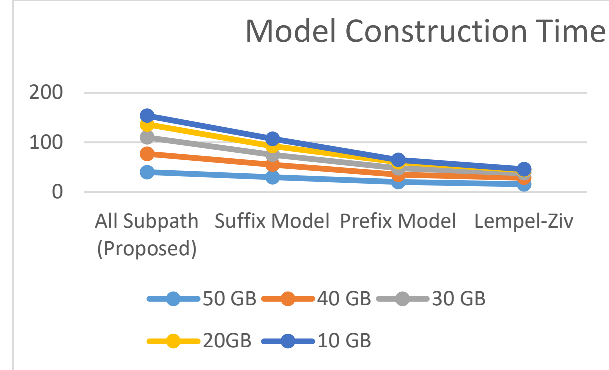

Figure 8: Model construction time

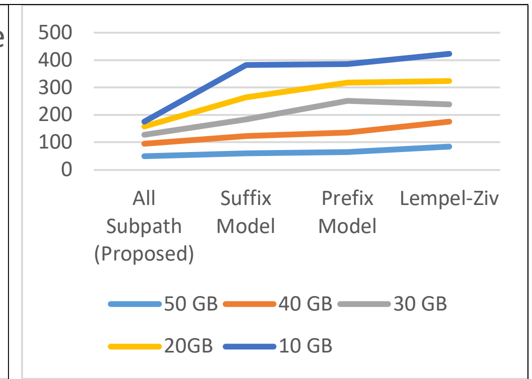

Figure 9: Model accuracy

Future Work

This work addresses the issues in the existing methodology for route recommendation. The proposed model uses the historical travel pattern to leverage the knowledge hidden in the location traces patterns. The solution is not only end-to-end trip prediction but also models the intermediate sequences of paths in travel patterns. Also includes the dynamic road conditions to predict better paths. But this strategy needs to process a huge corpus of data because of intermediate path sequences and processing of dynamic attributes and takes a long time to construct the model. The model construction can be extended to computing on a horizontally scalable platform. Data can be stored on distributed data storage like Hadoop, and computation can be extended to execute in distributed computation models like the MapReduce framework.

References

[1] Froehlich, J. and Krumm, J., "Route Prediction from Trip Observations," SAE Technical Paper 2008-01-0201, 2008, https://doi.org/10.4271/2008-01-0201.
[2] Lídia Montero, Xavier Ros-Roca, Using GPS tracking data to validate route choice in OD trips within dense urban networks, Transportation Research Procedia, Volume 47, 2020, Pages 593-600, ISSN 2352-1465, https://doi.org/10.1016/j.trpro.2020.03.136.
[3] Yinuo Huang, Xin Jin, Miao Fan, Xunwei Yang, Fangliang Jiang, 2024, Personalized Route Recommendation Based on User Habits for Vehicle Navigation IDST '24: Proceedings of the 2024 International Conference on Intelligent Driving and Smart Transportation Pages 28 - 32.
[4] Ha Yoon Song, Jae Ho Lee, A map matching algorithm based on modified hidden Markov model considering time series dependency over larger time span, Heliyon, Volume 9, Issue 11, 2023, e21368, ISSN 2405-8440, https://doi.org/10.1016/j.heliyon.2023.e21368.
[5] Cheng, X. A travel route recommendation algorithm based on interest theme and distance matching. EURASIP J. Adv. Signal Process. 2021, 57 (2021).
[6] S. Wang, J. Cao and P. S. Yu, "Deep Learning for Spatio-Temporal Data Mining: A Survey," in IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 8, pp. 3681-3700, 1 Aug. 2022, doi: 10.1109/TKDE.2020.3025580.
[7] de Oliveira e Silva, R.A., Cui, G., Rahimi, S.M. *et al.* Personalized route recommendation through historical travel behavior analysis. *Geoinformatica* **26**, 505–540 (2022).
[8] Hosseini, R., Tong, D., Lim, S., Sohn, G., & Gidófalvi, G. (2025). A framework for performance analysis of Open Street Map data in navigation applications: the case of a well-developed road network in Australia. Annals of GIS, 1–18. https://doi.org/10.1080/19475683.2025.2468184
[9] Yan Zhang, Yunhuai Liu, Genjian Li, Yi Ding, Ning Chen, Hao Zhang, Tian He, and Desheng Zhang. 2019. Route Prediction for Instant Delivery. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 3, 3, Article 124 (September 2019), 25 pages.
[10] Tiwari, Vishnu & Arya, Arti. (2018). Distributed Context Tree Weighting (CTW) for route prediction. Open Geospatial Data, Software and Standards. 3. 10.1186/s40965-018-0052-9.
[11] Zheng Y, Xie X, Ma W (2010) GeoLife: a collaborative social networking service among user, location and trajectory. IEEE Data Eng Bull 33(2):32–40
[12] Yu Zhao, Zhengchao Chen, Zhujun Zhao, Cong Li, Yongqing Bai, Zhaoming Wu, Degang Wang, Pan Chen, A deeply supervised vertex network for road network graph extraction in high-resolution images, International Journal of Applied Earth Observation and Geoinformation, Volume 133, 2024, 104082, ISSN 1569-8432
[13] Tiwari, V.S., Arya, A. & Chaturvedi, S. Scalable prediction by partial match (PPM) and its application to route prediction. Appl Inform 5, 4 (2018). https://doi.org/10.1186/s40535-018-0051-z
[14] Curran, Kevin & Fisher, Gavin & Crumlish, John. (2012). Open Street Map. International Journal of Interactive Communication Systems and Technologies. 2. 69-78. 10.4018/ijicst.2012010105.
[15] Arya, Arti & Chaturvedi, Sudha & Tiwari, Vishnu. (2022). Map reduce-based scalable Lempel-Ziv and application in route prediction. International Journal of Big Data Intelligence. 1. 1. 10.1504/IJBDI.2022.10047952.
[16] Zheng Y, Li Q, Chen Y, Xie X, Ma W (2008) Understanding mobility based on GPS data. In: Proceedings of ACM conference on Ubiquitous Computing (UbiComp 2008). ACM Press, Seoul, Korea, pp 312–321.
[17] Tiwari, Vishnu & Arya, Arti. Horizontally scalable probabilistic generalized suffix tree (PGST) based route prediction using map data and GPS traces. Journal of Big Data. 4, 2017 10.1186/s40537-017-0085-4.

[18] Saki, S., Hagen, T. A Practical Guide to an Open-Source Map-Matching Approach for Big GPS Data. SN COMPUT. SCI. 3, 415 (2022). https://doi.org/10.1007/s42979-022-01340-5.
[19] S. Nakajima, D. Kitayama, Y. Sushita, K. Sumiya, N. P. Chandrasiri and K. Nawa, "Route recommendation method for car navigation system based on estimation of driver's intent," 2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012), Istanbul, Turkey, 2012, pp. 318-323, doi: 10.1109/ICVES.2012.6294305..
[20] Xinyue Xu, Xiaoran Wang, Ziyang Ye, Anzhong Zhang, Jun Liu, Linqi Xia, Zinuo Li, Benxiang Feng, Route recommendation method for frequent passengers in subway based on passenger preference ranking, Expert Systems with Applications, Volume 252, Part A, 2024, 124216, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2024.124216.