Historical Travel Pattern Knowledge-Based System for Accurate Route Recommendation

Sudha Chaturvedi,
Lingayas University
Department of Computer Science and Information Technology,

Tapsi Nagpal,
Lingayas University
Department of Computer Science and Information Technology,

Vishnu Shankar Tiwari
PES University
Department of Computer Science & Engineering (AI & ML),

Praveen Kumar S M
PES University
Department of Computer Science & Engineering (AI & ML),

Abstract

Route recommendation and route prediction are two of the most frequently used applications in recent days. Popular usage includes personal route suggestions, resource prediction in grid computing, traffic estimation, infrastructure planning, solution to the vehicle routing problem (VRP), etc. 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. In many scenarios, it lands in situations where users face bad experiences like poor lighting conditions, bad road conditions, security issues on the route, narrow roads, 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 local travelers, like taxi drivers and residents in the area, know the best routes from source to destination in their locality and based on their local expertise, they plan their route for traveling. 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 users.

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, cell phone location devices, and cell phone location capturing. Over a period, a large volume of location data is accumulated over the server, which contains the user travel pattern behavior [2] [3]. This travel pattern behavior can be captured and fed into the prediction. A sample route suggested by the automated route recommendation system, as shown in Figure 1a, has poor road conditions, as shown in Figure 1b, but local users follow a better alternative route for the same source and destination shown in Figure 1c.

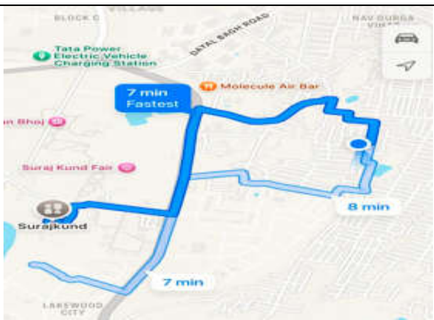

Figure 1a: Route suggested by application

Figure 1b: Suggested route has poor road conditions

Figure 1c: Local users follow a better alternative route

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, and is shown in Figure 2b. The same destination route followed by a local traveler is shown in Figure 2c.

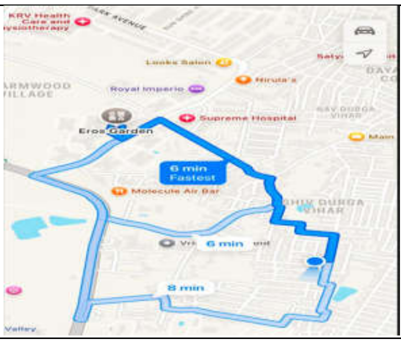

Figure 2a: Route suggested by application

Figure 2b: Suggested route has a dead end

Figure 2c: Local users follow a better alternative route

Road conditions can be categorized into two kinds, namely static and dynamic. Scenarios shown in Figure 1- road has a bad condition, and Figure 2- road has a dead end, are known as static conditions and don't change frequently. Other conditions 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.

The suggested route recommendation system road has a bad condition, and the figure 3a, which has a dead end, is shown in figure 3b. The same destination route followed by local travelers is as shown in Figure 3c.

Figure 3a: Route suggested by application

Figure 3b: Suggested route has poor security conditions

Figure 3c: Local users follow a better alternative route

User 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 the market, and then travels back to home. This makes three trips. For example, the continuous trip can be decomposed in three trips represented as:

$$(x_0, y_0, t^0), (x_1, y_1, t^1). (x_2, y_2, t^2) \dots\dots\dots. (x_m, y_m, t^m).$$
$$(x_m, y_m, t^m), (x_{m+1}, y_{m+1}, t^1), (x_{m+2}, y_{m+2}, t^{m+2}) \dots\dots\dots. (x_p, y_p, t^p).$$
$$(x_p, y_p, t^p), (x_{p+1}, y_{p+1}, t^{p+1}), (x_{p+2}, y_{p+2}, t^{p+2}) \dots\dots\dots. (x_n, y_n, t^n).$$

Decomposed trips are then mapped to digitized road network edges. The sequence of road network edges resultant from trip segmentation is done using a process known as map matching [4]. A prediction model is then developed using the trips as a sequence of road edges. This prediction model contains static and dynamic road conditions, which are then used to recommend the route for the user, given a source and destination pair. This model is based on user travel patterns and the current road segment condition, which appears in the recommended route and avoids roads with bad conditions.

Related Work

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] [7] [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. Continuous logs for GPS traces are decomposed into smaller units called trips. These trips, composed of GPS coordinates, are mapped matched to the road network to convert trips to a sequence of road network edges. 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. Proposed work uses both static and dynamic road conditions for route recommendation, which can predict better routes.

Terminologies

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 $G(V, E)$ [9] [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. An example of road network data captured from OSM is shown in Figure 4. Road network data captured from OSM and stored in spatial database is as shown in Figure 5.

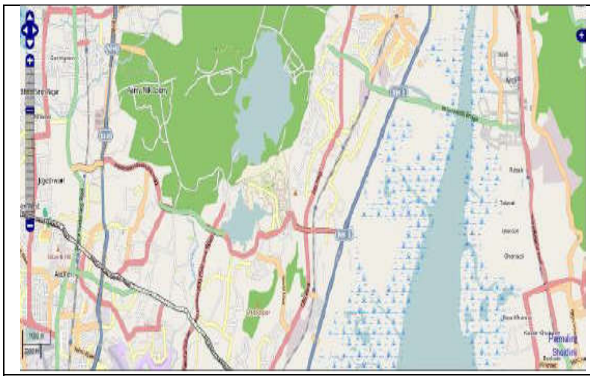

Figure 4. Road network data from OSM

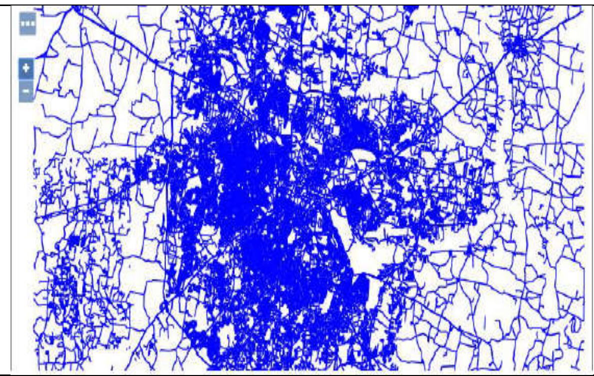

Figure 5. Road network data stored in PostGis

Another set of data used is GPS location traces. Location traces are a continuous sequence of latitude/longitude coordinates captured at discrete intervals represented by $(x_0, y_0, t^0), (x_1, y_1, t^1), (x_2, y_2, t^2) \dots\dots\dots (x_n, y_n, t^n)$. In this work, the location traces used is 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. A sample of raw GPS data is shown in Figure 6. The continuous location traces are decomposed into smaller logical units that express the user's travel pattern. For example, a user travels from home to office and commutes back to home, which consists of two trips, whereas the raw GPS traces this whole traversed path will be one long sequence of location traces. At this stage, the trip is still a sequence of raw location traces, but in a shorter and logical form.

For processing the travel pattern, the two datasets, road network and location traces data, should be mapped to create an association. 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 7.

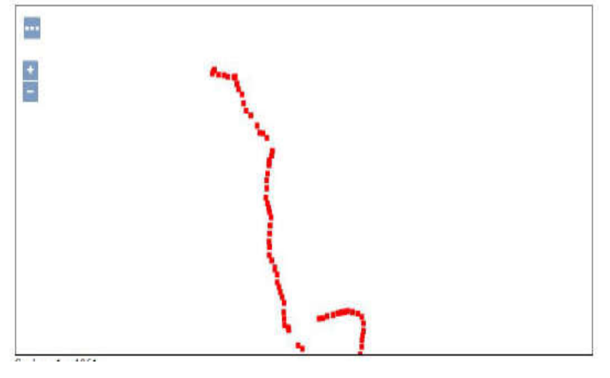

Figure 6. User trip in GPS coordinates

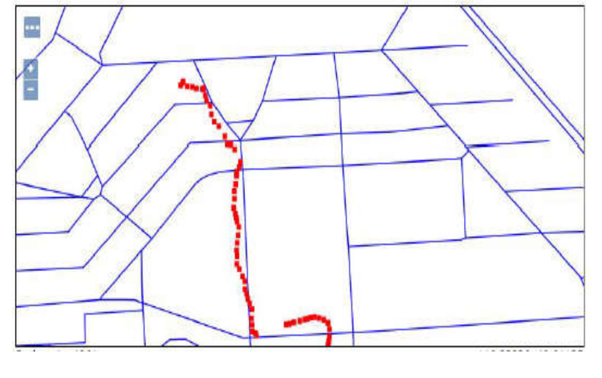

Figure 7. User trip mapped to road network

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.

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

Where $(x_{i,}y_{i,}t^i)$ is the location coordinate of the user and $(e_0,e_1,e_2,\ldots\ldots.e_n)$ 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.

Given a set of edges $\sum = \{e_0,e_1,e_2,\ldots\ldots.e_k\}$ trip (T) is a contiguous sequence of edges $T = e_0,e_1,e_2,\ldots\ldots.e_k \ni \sum$. A substring (S) of trip is a contiguous sequence of edges within a Trip T. Given a trip $e_1 \rightarrow e_2 \rightarrow e_{3,}$ the substrings of the trip are $\{e_1 \rightarrow e_2, e_2 \rightarrow e_3,\ e_1 \rightarrow e_2 \rightarrow e_3\}$. A $prefix$ of a trip is a contiguous sequence of edges belonging to a set of substrings and occurs at the beginning of the trip. In this example $prefix$ of the trip are $\{e_1 \rightarrow e_2,\ e_1 \rightarrow e_2 \rightarrow e_3\}$. Similarly, an $suffix$ of a trip 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 $\{e_2 \rightarrow e_3,\ e_1 \rightarrow e_2 \rightarrow e_3\}$. Prediction model can be constructed using either prefix or suffix of the trip segments. For example, let's consider the user has made following trips. with prefixes and suffixes.

$$Trip_1 = e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_5$$
$$Trip\ Prefixes = e_1 \rightarrow e_2, e_1 \rightarrow e_2 \rightarrow e_3, e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4, e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_5$$
$$Trip\ Suffixes = e_4 \rightarrow e_5, e_3 \rightarrow e_4 \rightarrow e_5,\ e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_5, e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_5$$

$$Trip_2 = e_3 \rightarrow e_4 \rightarrow e_5 \rightarrow e_6 \rightarrow e_7$$
$$Trip\ Prefixes = e_3 \rightarrow e_4, e_3 \rightarrow e_4 \rightarrow e_5, e_3 \rightarrow e_4 \rightarrow e_5 \rightarrow e_6, e_3 \rightarrow e_4 \rightarrow e_5 \rightarrow e_6 \rightarrow e_7$$
$$Trip\ Suffixes = e_6 \rightarrow e_7, e_5 \rightarrow e_6 \rightarrow e_{7,}\ e_4 \rightarrow e_5 \rightarrow e_6 \rightarrow e_7, e_3 \rightarrow e_4 \rightarrow e_5 \rightarrow e_6 \rightarrow e_7$$

$$Trip_3 = e_4 \rightarrow e_5 \rightarrow e_6 \rightarrow e_7 \rightarrow e_2$$
$$Trip\ Prefixes = e_4 \rightarrow e_5,\ e_4 \rightarrow e_5 \rightarrow e_6, e_4 \rightarrow e_5 \rightarrow e_6 \rightarrow e_7, e_4 \rightarrow e_5 \rightarrow e_6 \rightarrow e_7 \rightarrow e_2$$
$$Trip\ Suffixes = e_7 \rightarrow e_2, e_6 \rightarrow e_7 \rightarrow e_2, e_5 \rightarrow e_6 \rightarrow e_7 \rightarrow e_2, e_4 \rightarrow e_5 \rightarrow e_6 \rightarrow e_7 \rightarrow e_2$$

$$Trip_4 = e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_6 \rightarrow e_7$$
$$Trip\ Prefixes = e_2 \rightarrow e_3, e_2 \rightarrow e_3 \rightarrow e_4, e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_6, e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_6 \rightarrow e_7$$
$$Trip\ Suffixes = e_6 \rightarrow e_7, e_4 \rightarrow e_6 \rightarrow e_7,\ e_3 \rightarrow e_4 \rightarrow e_6 \rightarrow e_7, e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_6 \rightarrow e_7$$

$$Trip_5 = e_3 \rightarrow e_4 \rightarrow e_2 \rightarrow e_5 \rightarrow e_6$$
$$Trip\ Prefixes = e_3 \rightarrow e_4, e_3 \rightarrow e_4 \rightarrow e_2, e_3 \rightarrow e_4 \rightarrow e_2 \rightarrow e_5, e_3 \rightarrow e_4 \rightarrow e_2 \rightarrow e_5 \rightarrow e_6$$
$$Trip\ Suffixes = e_5 \rightarrow e_6, e_2 \rightarrow e_5 \rightarrow e_6,\ e_4 \rightarrow e_2 \rightarrow e_5 \rightarrow e_6, e_3 \rightarrow e_4 \rightarrow e_2 \rightarrow e_5 \rightarrow e_6$$

All Trips Prefix paths and Suffix Paths for trips- $Trip_1, Trip_2, Trip_3, Trip_4, Trip_5$ are as shown in Table 1. It can be noticed that all prefix paths starting from start node of the trip are captured correctly but other paths in suffix paths are not captured. These are the drawbacks of all prefixes-based models like PGST [17]. For example, if route is queried for source e_1 and destination e_4 then model will recommend $e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4$ as it is captured in trip prefix for $Trip_1$.But trip prefix path based model cannot recommend path for source e_4 and destination e_7 but this information is contained in trip data as a trip sub path traversed during traveling $Trip_2$. This can be answered by suffix path based model as $e_4 \rightarrow e_5 \rightarrow e_6 \rightarrow e_7$ from trip sub path contained in $Trip_2$. Similarly, all suffix paths ending in destination node of the trip is 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 source node to all other nodes are captured correctly and all suffix-based routes starting from all nodes to destination nodes are captured and recommendation can be made for all queries contained in either trip prefix or trip suffix. But there are still some scenarios like query is made for source e_2 and destination e_4. There is path traveled $e_2 \rightarrow e_3 \rightarrow e_4$ in $Trip_1$ but is not captured in either of trip prefix or trip suffix.

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 model and edges are labeled with the frequency of the edges travelled in the past. There can also be scenarios where in past the route was traversed by users but due to some dynamic conditions like roadblock users are not traversing the path in current scenario. Then the edge travel frequency is labelled with negative weight. Algorithm for model construction is as in algorithm I.

Algorithm I: *Route recommendation model construction*
Input: *User trips as sequence of road network edges*
Output: *Graph based route recommendation model*
Algorithm:

1. *Instantiate an empty graph* $\boldsymbol{G}\ (\boldsymbol{V}, \boldsymbol{E})$
2. *For each* $\boldsymbol{Trip_i} \in \boldsymbol{Trip_1}, \boldsymbol{Trip_2} \ldots. \boldsymbol{Trip_n}$
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* $\boldsymbol{W(e_i)} = -\infty$

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, the edge with negative weight is excluded, and the next best route is recommended.
This is an interesting situation we landed in, a typical real-world problem with real data captured from a location where we must put in place well-established Graph theory algorithms. Below are the scenarios representing the usage of model.

- Path appears in trip prefix sub path & Path appears in trip suffix sub path, but does not solve the problem – Spotted from ground all prefix representing people traveling from East Bengaluru as shown in Image 1. The proposed solution solves this problem by identifying the heavy crowd area and suggesting alternate paths.

- Path appears in trip suffix sub path & Path appears in trip prefix sub path, but does not solve the problem – Spotted from the ground, all suffixes represent people traveling from West Bengaluru and gathering in a single place, as shown in Image 2. This can be avoided by using the proposed solution, which suggests alternate paths

- The query path has a blockage If the proposed model is used, all gates should be opened, and could have avoided reported deaths in the Bengaluru Cricket celebration. A snapshot of closed/dead end is shown in $Image$ 3.

		
Image 1: Crowd traveling to a single spot	Image 2: Crowd traveling to a single spot from another direction	Image 3: The Crowd reached the stadium gate, where the gate was blocked

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 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.

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] Wei Tian, Jieming Shi, Siqiang Luo, Hui Li, Xike Xie, and Yuanhang Zou. Effective and efficient route planning using historical trajectories on road networks. Proceedings of the VLDB Endowment, 16(10):2512--2524, 2023.

[3] J. Jiang, C. Xu, J. Xu, M. Xu, N. Zheng and K. Kong, "Route planning for locations based on trajectory segments", Proc. 2nd ACM SIGSPATIAL Workshop Smart Cities Urban Analytics, pp. 1-8, Oct. 2016.

[4] Greenfeld, Josh. (2002). Matching GPS Observations to Locations on a Digital Map, Proceedings of the 81st annual meeting of the transportation, Research Board, Washington

[5] Shiming Zhang, Zhipeng Luo, Li Yang, Fei Teng, Tianrui Li, A survey of route recommendations: Methods, applications, and opportunities, Information Fusion, Volume 108, 2024, 102413, ISSN 1566-2535

[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] Q. Gao, F. Zhou, K. Zhang, F. Zhang and G. Trajcevski, "Adversarial Human Trajectory Learning for Trip Recommendation," in IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 4, pp. 1764-1776, April 2023, doi: 10.1109/TNNLS.2021.3058102.

[8] Hosseini, R., Tong, D., Lim, S., Sohn, G., & Gidófalvi, G. (2025). A framework for performance analysis of OpenStreetMap 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] Zheng Y, Zhang L, Xie X, Ma W (2009) Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of international conference on World Wild Web (WWW 2009). ACM Press, Madrid Spain, pp 791–800.

[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] Pung, J., D'Souza, R.M., Ghosal, D. et al. A road network simplification algorithm that preserves topological properties. Appl Netw Sci 7, 79 (2022). https://doi.org/10.1007/s41109-022-00521-8

[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] Klinkhardt, C., T. Wörle, L. Briem, M. Heilig, M. Kagerbauer, and P. Vortisch. 2021. "Using OpenStreetmap as a Data Source for Attractiveness in Travel Demand Models." Transportation Research Record: Journal of the Transportation Research Board 2675 (8): 294–303.

[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] 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.

[19] Shiming Zhang, Zhipeng Luo, Li Yang, Fei Teng, Tianrui Li, A survey of route recommendations: Methods, applications, and opportunities, Information Fusion, Volume 108, 2024, 102413, ISSN 1566-2535, https://doi.org/10.1016/j.inffus.2024.102413.

[20] Cheng, X. A travel route recommendation algorithm based on interest theme and distance matching. EURASIP J. Adv. Signal Process. 2021, 57 (2021). https://doi.org/10.1186/s13634-021-00759-x