# Personalized and On-device Trajectory Mobility Prediction

Cuauhtemoc Anda*
temo.andacastro@ncs.com.sg
NCS
Sinapore

Ning Cao*
ning.cao@ntu.edu.sg
College of Computing & Data Science, NTU
Singapore

Shuai Liu†
shuai.liu@ntu.edu.sg
College of Computing & Data Science, NTU
Singapore

Shaowei Ying
shaowei@ncs.com.sg
NCS
Sinapore

Gao Cong
gaocong@ntu.edu.sg
College of Computing & Data Science, NTU
Singapore

## Abstract

Accurately predicting individual trajectory mobility is critical for various urban applications, including traffic management and personalized services. However, existing deep learning models often suffer from overfitting due to noisy, large-scale trajectory data and struggle to capture the unique movement patterns of minority groups. Additionally, privacy concerns arise when personal trajectory data must be uploaded to cloud servers for processing. To address these challenges, we propose a novel feature-engineering and machine learning-based framework for trajectory prediction. Our method incorporates Time2Vec to capture both periodic and trend-based temporal features and utilizes CatBoost to handle structured, non-sequential trajectory data efficiently. This approach reduces overfitting, enhances privacy by eliminating the need for cloud processing, and achieves competitive performance, ranking in the top 10 among over 100 team submissions. Our framework offers a robust, privacy-conscious solution for individual trajectory prediction, advancing both accuracy and inclusivity in urban mobility applications.

## CCS Concepts

• **Computing methodologies** → **Boosting**.

## Keywords

Catboost, Feature engineering, Time2Vec, Human mobility prediction

**ACM Reference Format:**
Cuauhtemoc Anda, Ning Cao, Shuai Liu, Shaowei Ying, and Gao Cong. 2024. Personalized and On-device Trajectory Mobility Prediction. In *2nd ACM SIGSPATIAL International Workshop on the Human Mobility Prediction Challenge (HuMob'24), October 29-November 1, 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3681771.3699917

*These authors contributed equally to this research.
†Corresponding author.


*HuMob'24, October 29-November 1, 2024, Atlanta, GA, USA*

ACM ISBN 979-8-4007-1150-3/24/10
https://doi.org/10.1145/3681771.3699917

## 1 INTRODUCTION

Understanding, modeling, and predicting human mobility trajectories in urban areas is a critical task with widespread implications across fields such as urban planning, transportation management, and personalized location-based services [1, 16]. Accurate trajectory predictions can improve traffic systems [13], enable real-time routing [5], and drive more personalized recommendations [6]. With the availability of extensive spatio-temporal data from mobile devices, the potential for precise human mobility analysis has grown significantly.

With the increasing availability of large-scale data, deep learning models have recently shown strong potential in revealing population level mobility patterns [9, 11, 17]. However, while these models perform well in analyzing the dynamics of urban populations, they face several challenges when applied to individual-level trajectory prediction. First, large-scale trajectory datasets often suffer from varying data quality and sparsity, introducing significant noise. This noise can cause deep learning models to overfit during training, meaning that these models rely too heavily on patterns specific to the training data and struggle to generalize to new individual trajectories. As a result, although these models can effectively learn generalized movement trends, their performance tends to degrade when predicting individual movement patterns. Secondly, privacy concerns also complicate the widespread application of deep learning models for personal trajectory prediction [14]. Many AI-driven systems require users to upload their personal movement data to cloud servers for centralized processing, raising privacy risks. Privacy-conscious users face a difficult trade-off between sacrificing personal data security and forgoing these personalized services. This underscores the demand for privacy-preserving, on-device models that allow for personal trajectory prediction without the need for cloud-based data uploads.

To address the issues of overfitting, we propose SoloPath, a method based on feature engineering and machine learning. Specifically, we utilize Time2Vec [8] to capture both the periodicity and trends in temporal features. At the same time, we apply feature engineering to convert trajectory data into tabular format, transforming the

original spatiotemporal sequences into structured, non-sequential feature data. To further enhance model performance, we adopt CatBoost, a machine learning algorithm optimized for handling categorical features. CatBoost [10] is known for its robustness to noise and overfitting, and in our method, it demonstrates excellent performance in personalized trajectory prediction. What's more, our method is specifically optimized to address privacy concerns. Since our model runs entirely on local devices, there is no need to upload users' trajectory data to the cloud for processing, effectively preventing potential privacy breaches. In conclusion, our contributions are:

- By leveraging Time2Vec and utilizing CatBoost, our proposed SoloPath effectively reduces overfitting, particularly in scenarios with noisy or sparse data, thereby improving the accuracy of individual trajectory predictions.
- Our approach eliminates the need for cloud-based data processing, ensuring that personal trajectory data remains local to the device, thereby safeguarding user privacy.
- Our method has achieved top 10 performance among over 100 team submissions in trajectory prediction tasks, demonstrating its competitive advantage in real-world applications.

## 2 PROBLEM Formulation

### 2.1 Data Description

We use datasets sourced from the HuMob Challenge 2024, provided by Yahoo Japan Corporation [15], focusing on predicting individual human mobility across four metropolitan areas in Japan, designated as cities A, B, C, and D. Each area is divided into 500x500-meter grid cells on a 200x200 grid, and the movement data spans 75 days, with each day divided into 48 intervals of 30 minutes. The task involves forecasting the locations of individuals based on their movements recorded within these grid cells.

City A contains full trajectory data for 100,000 individuals over 75 days, while cities B, C, and D provide partial trajectories (days 1 to 60) for 25,000, 20,000, and 6,000 individuals, respectively. The objective is to predict the movements of 3,000 individuals in cities B, C, and D during days 61 to 75, where the future locations are masked with cell values '999'. Additionally, point-of-interest (POI) data, represented as 85-dimensional vectors per grid cell, is provided for optional use.

### 2.2 Problem Objective

The objective of the task is to predict the next location of 3,000 individuals each in cities B, C, and D during days 61 to 75. Participants can utilize the full movement data from city A (days 1-75), as well as the available movement data from cities B, C, and D (days 1-60). The use of cross-city data, such as incorporating city A's movement patterns to predict movements in cities B, C, and D, is optional and may or may not improve prediction accuracy.

**Notation:** Let $G = \{A, B, C, D\}$ represent the set of cities. $X_i^t = (x_i^t, y_i^t)$ denotes the grid cell coordinates for individual $i$ at time $t$. $T = \{1, \ldots, 75\}$ represents the set of time intervals. $P \in \mathbb{R}^{85}$ denotes the 85-dimensional POI vector for each grid cell. $D_i^{\text{train}} = \{X_i^t \mid t \in [1, 60]\}$ represents the historical trajectory data for individual $i$ in cities B, C, and D. $D^{\text{test}} = \{X_i^t \mid t \in [61, 75]\}$ denotes the data to be predicted for the individuals during days 61 to 75.

The goal is to learn a function: $f : D_i^{\text{train}} \times P \rightarrow X_i^t \quad \text{for} \quad t \in [61, 75]$, predicting the next location $X_i^t$ for each individual based on their historical movements and optional POI data.

## 3 METHODOLOGY

Our framework, denoted as SoloPath, addresses the challenge of personal trajectory prediction by modeling human mobility based solely on temporal features, discarding the spatial dependencies tied to cities or other external factors. The novelty of SoloPath lies in transforming sequential trajectory data into tabular format, thus enabling the use of decision-tree-based models like CatBoost, which are known to perform competitively on tabular data compared to deep learning models [3, 4, 12].

### 3.1 Feature Engineering

In SoloPath, we focus on converting temporal sequences into feature-rich tabular data by encoding time-based features that can effectively capture both linear and periodic patterns in human mobility. A critical component of this is Time2Vec [8], which models both time-of-day and day-of-week's linear and periodic behaviors simultaneously. Time2Vec is particularly useful for trajectory data because it offers the flexibility to model bothperiodic (e.g., daily or weekly routines) and non-periodic (e.g., one-time events) patterns. It is defined as follows:

$$t(x) = [x, sin(w_1 x), sin(w_2 x), \cdots, sin(w_k x)] \tag{1}$$

where x represents the time input. Time2Vec offers three key advantages: it models both regular and irregular temporal behaviors, is invariant to time rescaling, and is simple enough to integrate with other features. By capturing these temporal nuances, Time2Vec allows SoloPath to effectively model the recurring and non-recurring patterns in individual mobility data. Besides Time2Vec, we also introduce another feature—whether it is weekend or not.

### 3.2 CatBoost for Trajectory Prediction

After transforming the sequential data into a tabular format, we utilize CatBoost [10], a gradient boosting decision tree algorithm, to model each individual's trajectory. CatBoost is well-suited for this task due to its ability to handle mixed data types (categorical and numerical features), its robustness against overfitting, and its scalability. The key advantages of using CatBoost in SoloPath are:

**Categorical Feature Handling:** CatBoost natively supports categorical features without requiring manual encoding techniques like one-hot encoding. This is critical as our dataset contains categorical features such as grid cell locations and the day of the week, which play an essential role in modeling mobility.

**Overfitting Prevention:** One of the primary challenges in modeling individual trajectories is the risk of overfitting, especially given the limited data for each user. CatBoost employs Ordered Boosting, a technique that dynamically adjusts the data used for tree construction to prevent overfitting and target leakage, making it ideal for our framework, where individual user data may be sparse or highly variable.

**Efficiency and Scalability:** SoloPath requires training a separate model for each individual, making the efficiency of the algorithm crucial. CatBoost's optimized implementation ensures fast training times, even when modeling the trajectories of thousands of users simultaneously. This scalability is vital for handling the large number of individuals in the dataset without compromising model performance.

**Interpretability:** An additional benefit of CatBoost is its interpretability, which allows us to generate feature importance metrics. These insights enable us to better understand the contribution of temporal and spatial features in mobility prediction, offering explainability in a domain where understanding movement behavior is as important as achieving high accuracy.

**Security:** CatBoost's local computation provides added security benefits. Since the SoloPath framework runs models directly on user devices, all data processing and predictions occur locally, reducing the need to upload data to the cloud and, consequently, minimizing the risk of data breaches. This enhances the privacy protection of the model and reduces potential security vulnerabilities associated with processing data on remote servers. Moreover, CatBoost's local execution ensures that personal data remains on the user's device throughout the entire process, offering a highly secure prediction environment, especially suitable for privacy sensitive applications.

In summary, SoloPath leverages feature engineering techniques to convert sequential trajectory data into a format suited for decision tree models. By combining linear temporal encoding with Time2Vec and utilizing CatBoost's efficient, scalable, and interpretable modeling capabilities, SoloPath achieves accurate, personalized trajectory predictions without the need for deep learning or transfer learning from other users or cities.

# 4 EXPERIMENTS

In this section, we evaluate the performance of our framework, SoloPath, using two widely adopted metrics: GEO-BLEU and Dynamic Time Warping (DTW). These metrics are chosen to capture the spatial and temporal accuracy of predicted trajectories, providing a comprehensive evaluation of our model's performance in human mobility prediction.

## 4.1 Experimental Setup

We conduct our experiments using data from city D, focusing on users with complete trajectory data across all 75 days. The first 60 days are used as the training set, and the remaining 15 days serve as the test set for evaluating the predicted trajectories. We compare the performance of SoloPath with two baselines:

**Multi-layer GRU [2]:** A simple deep learning model based on Gated Recurrent Units, which is often used for sequential data modeling.

**C-MHSA [7]:** A sophisticated neural network model that integrates spatio-temporal contexts for next location prediction. It incorporates multiple sources of contextual information, including the time of visit, activity duration, and land use data.

## 4.2 Evaluation Metrics

**GEO-BLEU:** GEO-BLEU is a similarity measure for geospatial sequences that extends the BLEU score by incorporating spatial proximity. It compares sequences of locations using geospatial n-grams, where each location is represented by its coordinates. The similarity between two n-grams Eq. 6 $g_v = (v_1, \ldots, v_n)$ and $g_w = (w_1, \ldots, w_n)$ is defined as:

$$s(g_v, g_w) = \prod_{k=1}^{n} \exp(-\beta \times d(v_k, w_k)) \tag{2}$$

where $d(v_k, w_k)$ is the Euclidean distance between locations $v_k$ and $w_k$, and $\beta$ is a scaling factor. The final GEO-BLEU score is calculated by taking the weighted geometric mean of the modified precision scores, incorporating spatial similarity, with the brevity penalty applied, as follows:

$$E_{all_h} = E_{XY_h} || E_{T_h} || E_{D_h} || E_{Dur_h} \in \mathbb{R}^{M \times (d_{xy} + d_t + d_d + d_{dur})} \tag{3}$$

$$E_{XY_c} || E_{T_c} || E_{D_c} || E_{Dur_c} = E_{all_c} \in \mathbb{R}^{N \times (d_{xy} + d_t + d_d + d_{dur})} \tag{4}$$

$$E_{all_h} = E_{XY_h} || E_{T_h} || E_{D_h} || E_{Dur_h} \in \mathbb{R}^{M \times (d_{xy} + d_t + d_d + d_{dur})} \tag{5}$$

$$E_{XY_c} || E_{T_c} || E_{D_c} || E_{Dur_c} = E_{all_c} \in \mathbb{R}^{N \times (d_{xy} + d_t + d_d + d_{dur})} \tag{6}$$

$$\text{GEO-BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log q_n\right) \tag{7}$$

This method effectively captures local spatial features, allowing for flexible comparison of geospatial sequences.

**DTW:** Dynamic Time Warping (DTW) is a well-established algorithm used to measure the similarity between two time series, even when they vary in speed or length. Given two sequences $X = (x_1, x_2, \ldots, x_M)$ and $Y = (y_1, y_2, \ldots, y_N)$, DTW finds the optimal alignment by minimizing the distance between corresponding points under specific constraints. The alignment is represented as a warping path $P = \{(i_1, j_1), (i_2, j_2), \ldots, (i_L, j_L)\}$, where $L = \max(M, N)$. The cost of alignment is calculated as:

$$\text{cost}(P) = \sum_{l=1}^{L} d(x_{i_l}, y_{j_l}) \tag{8}$$

where $d(x, y)$ is typically the Euclidean distance. DTW minimizes this cost subject to boundary conditions, monotonicity, and step size constraints, ensuring the time-ordering of the sequences. The final DTW distance is the minimum alignment cost:

$$\text{DTW}(X, Y) = \min_{P} \text{cost}(P) \tag{9}$$

## 4.3 Results

Table 1 presents a performance comparison between SoloPath and the baseline models, GRU and C-MHSA, on data from city D. The test set includes users with IDs ranging from 1500 to 3000, evaluated over the last 15 days using two metrics: GEO-BLEU and DTW.

**Table 1: Performance Comparison of SoloPath and Baseline Models (cityD)**

| | GEO-BLEU | DTW |
|---|---|---|
| GRU | 0.1449 | 71.78 |
| C-MHSA | 0.3291 | 31.62 |
| SoloPath | **0.3431** | **28.73** |

SoloPath outperformed both baseline models across these two metrics.

First, in terms of the GEO-BLEU metric, SoloPath achieved a score of 0.3431, surpassing C-MHSA's 0.3291 and GRU's 0.1449. GEO-BLEU measures the spatial accuracy of the predicted trajectories. GRU's low score indicates poor performance in capturing the spatial patterns of user trajectories. C-MHSA, by integrating complex spatiotemporal context (such as points of interest and a multi-head self-attention mechanism), shows a significant improvement in spatial accuracy. However, despite C-MHSA's use of a more complex deep learning architecture, SoloPath slightly outperforms it by relying on efficient feature engineering and the decision-tree-based CatBoost model, demonstrating superior spatial precision while maintaining a simpler model structure.

In terms of the DTW (Dynamic Time Warping) metric, SoloPath also performed best with a score of 28.73, lower than C-MHSA's 31.62 and GRU's 71.78. DTW measures the temporal alignment between the predicted and actual trajectories, with lower scores indicating better performance in preserving the temporal sequence. GRU's high DTW score suggests that it struggled with capturing the temporal dynamics of user movements. C-MHSA, due to its ability to model complex spatiotemporal features, showed a significant improvement in temporal sequence accuracy. However, SoloPath, despite not using an advanced deep learning architecture, was able to capture temporal patterns more precisely through effective time feature engineering and the CatBoost model.

Overall, SoloPath excelled in both spatial accuracy (GEO-BLEU) and temporal sequence preservation (DTW), outperforming the baseline models. These results demonstrate that SoloPath is highly competitive in individual trajectory prediction tasks. It not only delivers superior performance but also maintains computational efficiency and reduced model complexity, making it a viable alternative to deep learning models.

## 5 Conclusion

In this work, we presented SoloPath, a novel framework for individual trajectory prediction that leverages feature engineering and decision-tree-based modeling, specifically using CatBoost, to achieve high accuracy in predicting human mobility. SoloPath's approach effectively addresses two key challenges faced by traditional deep learning models: overfitting and privacy concerns. By transforming sequential trajectory data into a tabular format and using Time2Vec to capture both linear and periodic temporal patterns, SoloPath outperforms baseline models, including GRU and the more complex C-MHSA, in both spatial accuracy (GEO-BLEU) and temporal alignment (DTW).

## References


[1] Serina Chang, Emma Pierson, Pang Wei Koh, Jaline Gerardin, Beth Redbird, David Grusky, and Jure Leskovec. Mobility network models of covid-19 explain inequities and inform reopening. *Nature*, 589(7840):82–87, 2021.

[2] Rahul Dey and Fathi M Salem. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pages 1597–1600. IEEE, 2017.

[3] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34:18932–18943, 2021.

[4] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520, 2022.

[5] Chenjuan Guo, Bin Yang, Jilin Hu, and Christian Jensen. Learning to route with sparse trajectory sets. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 1073–1084. IEEE, 2018.

[6] Ngai Lam Ho, Roy Ka-Wei Lee, and Kwan Hui Lim. Btrec: Bert-based trajectory recommendation for personalized tours. *arXiv preprint arXiv:2310.19886*, 2023.

[7] Ye Hong, Yatao Zhang, Konrad Schindler, and Martin Raubal. Context-aware multi-head self-attentional neural network model for next location prediction. *Transportation Research Part C: Emerging Technologies*, 156:104315, 2023.

[8] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*, 2019.

[9] Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. A survey on deep learning for human mobility. *ACM Computing Surveys (CSUR)*, 55(1):1–44, 2021.

[10] Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

[11] Can Rong, Jie Feng, and Yong Li. Deep learning models for population flow generation from aggregated mobility data. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, pages 1008–1013, 2019.

[12] Ravid Shwartz-Ziv and Amitai Armon. Tabular data: Deep learning is not all you need. *Information Fusion*, 81:84–90, 2022.

[13] Zachary Vander Laan, Mark Franz, and Nikola Marković. Scalable framework for enhancing raw gps trajectory data: application to trip analytics for transportation planning. *Journal of big data analytics in transportation*, 3(2):119–139, 2021.

[14] Chunnan Wang, Xiang Chen, Junzhe Wang, and Hongzhi Wang. Atpfl: Automatic trajectory prediction model design under federated learning framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6563–6572, 2022.

[15] Takahiro Yabe, Kota Tsubouchi, Toru Shimizu, Yoshihide Sekimoto, Kaoru Sezaki, Esteban Moro, and Alex Pentland. Yjmob100k: City-scale and longitudinal dataset of anonymized human mobility trajectories. *Scientific Data*, 11(1):397, 2024.

[16] Takahiro Yabe, Yunchang Zhang, and Satish V Ukkusuri. Quantifying the economic impact of disasters on businesses using human mobility data: a bayesian causal inference approach. *EPJ Data Science*, 9(1):36, 2020.

[17] Zefang Zong, Jie Feng, Kechun Liu, Hongzhi Shi, and Yong Li. Deepdpm: Dynamic population mapping via deep neural network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 1294–1301, 2019.