



A Data Augmentation Algorithm for Trajectory Data

Yaksh J. Haranwala

Memorial University of Newfoundland, Department of
Computer Science
St. John's, Canada
yjharanwala@mun.ca

Gabriel Spadon

Dalhousie University, Department of Computer Science
Halifax, Canada
spadon@dal.ca

Chiara Renso

Institute of Science and Technology A. Faedo, National
Research Council of Italy
Pisa, Italy
chiara.renso@isti.cnr.it

Amilcar Soares

Linnaeus University, Department of Computer Science and
Media Technology
Växjö, Sweden
amilcar.soares@lnu.se

ABSTRACT

The growing prevalence of location-based devices has resulted in a significant abundance of location data from various tracking vendors. Nevertheless, a noticeable deficit exists regarding readily accessible, extensive, and publicly available datasets for research purposes, primarily due to privacy concerns and ownership constraints. There is a pressing need for expansive datasets to advance machine learning techniques in this domain. The absence of such resources currently represents a substantial hindrance to research progress in this field. Data augmentation is emerging as a popular technique to mitigate this issue in several domains. However, applying state-of-the-art techniques as-is proves challenging when dealing with trajectory data due to the intricate spatio-temporal dependencies inherent to such data. In this work, we propose a novel strategy for augmenting trajectory data that applies a geographical perturbation on trajectory points along a trajectory. Such a perturbation results in controlled changes in the raw trajectory and, consequently, causes changes in the trajectory feature space. We test our strategy in two trajectory datasets and show a performance improvement of approximately 20% when contrasted with the baseline. We believe this strategy will pave the way for a more comprehensive framework for trajectory data augmentation that can be used in fields where few labeled trajectory data are available for training machine learning models.

CCS CONCEPTS

• **Computing methodologies** → *Classification and regression trees*;
• **Software and its engineering** → *Software libraries and repositories*; • **Information systems** → **Data cleaning**; **Geographic information systems**.

KEYWORDS

Data Augmentation, Trajectory Data, Machine Learning



This work is licensed under a Creative Commons Attribution International 4.0 License.

EMODE '23, November 13, 2023, Hamburg, Germany

© 2023 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-0347-8/23/11.

<https://doi.org/10.1145/3615885.3628008>

ACM Reference Format:

Yaksh J. Haranwala, Gabriel Spadon, Chiara Renso, and Amilcar Soares. 2023. A Data Augmentation Algorithm for Trajectory Data. In *1st ACM SIGSPATIAL International Workshop on Methods for Enriched Mobility Data: Emerging issues and Ethical perspectives 2023 (EMODE '23)*, November 13, 2023, Hamburg, Germany. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3615885.3628008>

1 INTRODUCTION

In recent years, driven by the widespread adoption of location-based devices, we have witnessed a burgeoning interest in research concerning the analysis of movement data [10, 15, 17]. Nonetheless, the abundant availability of such data from tracking product vendors is juxtaposed with the scarcity of publicly accessible real datasets for research purposes. This challenge, compounded by the necessity for extensive labeled databases to train machine learning and deep learning techniques, underscores a pivotal obstacle in the progression of research within this field [4, 5, 14].

Data augmentation has emerged as a powerful technique in machine learning, strengthening model robustness while mitigating overfitting and underfitting issues by generating diverse synthetic data that can be used to overcome the scarcity of labeled data. Data augmentation is the process of generating synthetic data by applying transformations to the original examples [13]. Data augmentation not only increases the efficiency of the machine learning models but they have also been shown to make the machine learning models more robust [9].

In the literature, we have seen different techniques for data augmentation in different domains such as image geometric transformations (e.g., data jittering, cropping, flipping, distortion, and rotation) [7], Fourier transformations (e.g., fast Fourier transform and Gaussian noise injection) [8], time-series augmentation (e.g., time warping, slicing, permutation, and interpolation) [8], Generative Adversarial Networks (GANs) [1] and Encoder-Decoder Networks [11]. Despite its success in other domains, data augmentation's potential remains largely untapped in mobility data analysis, primarily due to the intricate nature and unique format of trajectory data. A trajectory is a spatio-temporal data consisting of temporally and spatially spaced points associated with location information, as detected by location devices.

Data augmentation techniques developed in other domains, like geometric augmentation methods for image processing, cannot

be directly applied to trajectories since they would barely impact the features extracted from a trajectory. Similarly, Fourier transformations, predominantly employed in image and wave analysis, find limited use outside these domains and may need a careful re-design to be used in trajectory analysis. A notable parallel lies in the resemblance between time series and trajectory data, given that trajectories also comprise temporally spaced points. This similarity encourages the adaptation of augmentation techniques from the time-series domain to the trajectory analysis domain.

Our proposal is based on the concept of adding noise to data before feature extraction. In the case of trajectory data, simply adding random noise to trajectory point features may create attributes with inconsistent values, like increasing the object's speed and, at the same time, decreasing the acceleration. To resolve the matter at hand, our approach entails the implementation of a random geographical variance into the original trajectory data. This is accomplished by adjusting the geographical coordinates of the trajectory points within a predetermined circular region. We achieve this by selectively shifting the location of specific points either within the circle or along its perimeter. By utilizing this spatial threshold, we aim to prevent any potential discrepancies from arising.

In summary, this work's contributions are the following:

- We propose two geographical noise strategies that randomly move trajectory points inside a geographical circle (*in-circle*) or on the boundaries of a geographical circle (*on-circle*).
- We propose an algorithm that randomly selects trajectory points and applies geographical noise using two techniques (e.g., in-circle and on-circle) to generate consistent augmented trajectories.
- We test and properly evaluate our proposed method in three datasets, showing that the technique creates augmented trajectories that can improve machine learning methods' performances.

2 METHODOLOGY

The objective of trajectory data augmentation is to enhance a given set of trajectories by introducing geographical noise, thereby generating new synthetic trajectories that offer increased diversity that data mining methods can use. We can apply the strategy to augment these data by moving trajectory points within a limited geographical neighborhood, thus creating consistent data. The proposed methodology for data augmentation for trajectory data is based on choosing a circular spatial buffer around each trajectory point and then moving the point based on two strategies: (1) inside the circle or (2) at the border, detailed below.

2.1 Geographical Noise Strategies

Our first strategy to apply geographical noise in trajectory points is to create a geographical circle in selected points and randomly move this trajectory point **inside** (i.e., in-circle geographical noise) to this geographical boundary. An example of such a technique is seen in Figure 1, where two trajectory points are selected, and two trajectory points are moved inside the circle (red dots).

Our second strategy to apply geographical noise in trajectory points is to create a geographical circle in selected points and randomly move this trajectory point **on the border** (i.e., in-circle

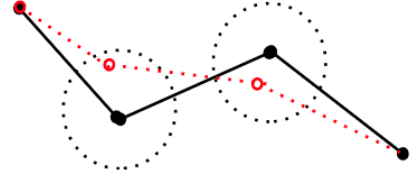


Figure 1: In-Circle Geographical Noise Example

geographical noise) of this geographical boundary. An example of such a technique is seen in Figure 2, where two trajectory points are selected, and two trajectory points are moved on the circle's border (i.e., white dots).



Figure 2: On-Circle Point Geographical Noise Example

2.2 Trajectory Data Augmentation Algorithm

This section details Algorithm 1, which performs the data augmentation in raw trajectory data. The algorithm takes as input several parameters: (i) p_{aug} , representing the percentage of trajectories to be augmented; (ii) n_{gen} , indicating the number of trajectories to be generated for each selected trajectory; (iii) p_{noise} , which specifies the portion of trajectory points to undergo geographical noise; (iv) A chosen geographical noise strategy, offering two distinct options: in-circle or on-circle; and (v) r_{circle} , denoting the radius percentage to be used in the noise strategy.

Upon execution, the algorithm systematically proceeds as follows. Initialize an empty set, aug , which will be used to store the augmented trajectories (line 1). Select a subset of trajectories (line 2) from the input dataset to be augmented, with the subset size determined by the input parameter p_{aug} . For each trajectory t within the chosen subset, we start by conducting n_{gen} iterations to create a specified number of new trajectories based on the selected one (lines 3 to 11). We first duplicate the currently selected trajectory t , storing it as t_{cpy} (line 5). After, we randomly choose a percentage of trajectory points from t_{cpy} according to the parameter p_{noise} (line 6). Then, we apply the designated geographical noise strategy (in-circle or on-circle) on the selected points, generating a set of modified trajectory points new_{pts} (line 7). We then replace the trajectory points in t_{cpy} with the modified points new_{pts} (line 8). Finally, we add the adapted trajectory t_{cpy} to the set of augmented trajectories, aug (line 9). After processing all trajectories in the

selected subset, the algorithm concludes by returning the set of augmented trajectories, *aug* (line 12).

Algorithm 1 Trajectory Augmentation Algorithm

Require: p_{aug} : Percentage of trajectories to be augmented
 n_{gen} : Number of trajectories to be generated
 p_{noise} : Percentage of trajectory points for geographical noise
 Geographical noise strategy: in-circle, on-circle
 r_{circle} : Radius percentage for circle
Ensure: *aug*: Set of augmented trajectories

```

1: aug = {}
2: subset_trajs = Select  $p_{aug}$  percentage of trajectories to be augmented
3: for all t in subset_trajs do
4:   for i = 1 to  $n_{gen}$  do
5:     tcpy = Create a copy of t
6:     pts = Randomly select percentage  $p_{noise}$  from tcpy
7:     newpts = Apply geographical noise strategy to pts
8:     tcpy = Replace original trajectory points by newpts
9:     aug = Add tcpy to aug
10:  end for
11: end for
12: return aug

```

3 EXPERIMENTAL SETUP AND RESULTS

3.1 Datasets and Feature Extraction

In our experiments, we utilized three distinct datasets to assess the performance of our proposed methods. The first dataset, named Geolife, is a subset of the comprehensive Geolife dataset [18], comprising 36 trajectories for a total of 355,181 trajectory points. These trajectories encompass various transportation modes, including airplanes, boats, subways, and taxis. The second dataset, referred to as Traffic¹, consists of 125 trajectories, encompassing 44,905 trajectory points, specifically involving large and standard vehicles. The third dataset, called Birds, comprises 58 trajectories, incorporating a total of 528,488 trajectory points attributed to geese [6], gulls [16], and vultures [12].

In our experimental setup, we examined the combined impact of the geographical changes introduced by our proposed algorithm on the performance of trajectory classification models, as detailed in Section 3.3.

In the machine learning space around spatio-temporal data, it is usually preferred that the data is in segment-based form where each tuple in the dataset represents an entire trajectory or a subset of trajectory containing the statistical description of the trajectory or the subset. We calculate the Kinematic trajectory features from the entire trajectory and encompass metrics such as average speed and percentiles of direction variation using PTRAIL[2, 3] In total, PTRAIL extracts 72 trajectory features from the trajectory itself. It is important to note that these trajectory features are utilized in both facets of our experiments: as attributes to calculate Euclidean distances and as the feature vector for training the machine

learning models. Due to the limited space to present our contributions, details about these 72 features can be found in the original documentation of PTRAIL².

3.2 Feature Vector Average Distance Analysis

When evaluating strategies for reducing geographical noise, we calculate the average Euclidean distance between the original trajectory features selected for modification and the novel trajectory features modified by our geographical noise reduction strategies. To gain initial insight into the effectiveness of our geographical noise strategies, we randomly chose 30% of the trajectories for testing. We ensure that there are no significant overlaps between the trajectory points in both experiments by using a 50% radius for the geographical circle (denoted by r_{circle} in Algorithm 1). Subsequently, we generate 20 augmented trajectories (n_{gen} in Algorithm 1) for each randomly selected trajectory, aiming at evaluating the average impact in the feature vector from each of our proposed strategies. We modify 20%, 40%, and 60% of trajectory points with geographical noise strategies (p_{noise} in Algorithm 1) to analyze changes in the feature vector. We repeat this experiment 20 times using the decimals of π taken four by four (e.g., 1415, 9265, etc.) as seeds to ensure variability in the random selection of trajectories and the geographical noise application, and the results are reported in Table 1.

Several conclusions and insights can be extracted from the results on the average Euclidean distances and standard deviations between original data points and points with injected noise. The average Euclidean distances generally tend to increase as the percentage of trajectory points selected is increased from 20% to 60%, suggesting adding more noise to trajectory points results in greater distance (i.e., higher differences in the trajectory features) compared to the original trajectory. The choice of injection method (on-circle vs. in-circle) also has a noticeable impact on the average distances. In many cases, the on-circle method tends to result in more considerable distances than the in-circle method, indicating that injecting noise on the outer boundary of the original data points results in more significant perturbations. One may choose on-circle or in-circle strategies depending on the specific application and goals. The on-circle method may be suitable when introducing more spread-out noise and potentially significantly impacting the trajectory feature. In contrast, the in-circle approach may be preferred when you wish noise closer to the original data points. It is also noticeable that each dataset exhibits its own characteristics regarding how noise affects the average distances. For instance, the Geolife dataset generally shows more considerable distances than the other datasets, indicating that noise significantly impacts this dataset.

The Geolife dataset, particularly with the on-circle method, shows higher standard deviations at 20% noise, indicating more variability in how noise affects this dataset at that specific noise level and injection method. Standard deviations for both the on-circle and in-circle methods are relatively consistent across different noise levels for the Traffic dataset. The highest standard deviation in this dataset is 0.4933 for the on-circle method at 20% noise. The Bird dataset generally exhibits lower standard deviations than the

¹<https://zen-traffic-data.net/english/outline/dataset.html>

² Available at: <https://ptrail.readthedocs.io/en/latest/>

	on-circle 20%		on-circle 40%		on-circle 60%		in-circle 20%		in-circle 40%		in-circle 60%	
	avg. dist	std	avg. dist	std	avg. dist	std	avg. dist	std	avg. dist	std	avg. dist	std
Geolife	1.1519	0.5922	0.5956	0.4430	1.3101	0.5452	0.7459	0.4500	1.3716	0.5283	0.8128	0.4511
Traffic	1.0214	0.4933	0.9917	0.4571	1.1052	0.4642	1.0675	0.4145	1.1370	0.4545	1.0960	0.4062
Bird	1.0439	0.2250	0.8237	0.2125	1.0764	0.2233	0.9082	0.2273	1.1057	0.2316	0.9645	0.2477

Table 1: Average Euclidean distances for different percentages of trajectory points selected to be geographically modified by our strategies. In the table, bold values indicate the smallest value in each column. For this experiment, the lower the values, the better.

Geolife and Traffic datasets. This suggests that noise has a more consistent impact on the Bird dataset. This dataset’s standard deviations are relatively low, indicating that the distances between original and noisy data points are relatively consistent. Even at 60% noise, the standard deviations remain relatively low (0.2233 for on-circle and 0.2477 for in-circle)

3.3 Classification Performance Analysis

This experiment evaluates the impact of introducing augmented trajectories in a trajectory classification problem. We establish a baseline model for the trajectory classification performance evaluation by splitting the original data set into 80% for training and 20% for testing and repeating such split 20 times using the decimals of π taken four by four. We then average these performances to establish such a baseline value. After, we start testing the impact of introducing augmented trajectories in the training data. To train a model, from the training set, we randomly select 30% of the trajectories (designated as p_{aug} in Algorithm 1) to test our proposed strategies. We test our algorithm with a 50% radius (r_{circle} in Algorithm 1) for the geographical circle in both experiments to ensure we do not have significant overlaps between trajectory points. We generate 20 augmented trajectories (n_{gen} in Algorithm 1) for each subselected one. We again test the values of 20%, 40%, and 60% of trajectory points to be modified (p_{noise} in Algorithm 1) by our geographical noise strategies to evaluate the impact in classification performances for distinct classification models. Aiming at evaluating the impact of the newly created augmented trajectories in trajectory classification problems, we evaluate the impact on the F-scores of training supervised machine learning models with and without augmented trajectories in the training set. We use three supervised machine learning models named Random Forest, Extra Trees, and XGBoost. The results of the experiments are detailed in Table 2.

Using the ExtraTreesClassifier in the Geolife dataset, in most cases, F-scores are below the baseline, indicating that the augmentation techniques (on-circle and in-circle) generally result in no improvements over the baseline data with no augmentation. The only exception is the on-circle strategy applied to 20% of the points, leading to a marginal gain of 0.29%. For the GradientBoostingClassifier, the F-scores are more consistently above the baseline, with gains ranging from 4.72% to 2.27%, suggesting that the augmentation techniques also improve performance for this model. Similarly, the F-scores for the RandomForestClassifier are consistently above the baseline, with gains ranging from 1.67 to 1.33%, indicating that

the strategies generally improve model performance on the Geolife dataset.

The F-scores are all below the baseline for the ExtraTreesClassifier in the Traffic dataset, indicating that the modification techniques lead to no improvement for this model on the Traffic dataset. The F-scores for GradientBoostingClassifier show improvements for all percentages using the on-circle strategy, ranging from 2.86% to 1.59%. For the RandomForestClassifier, the F-scores are consistently above the baseline, suggesting that the techniques result in improvements for this model on the Traffic dataset and a maximal gain of 2.58%. Therefore, for the Traffic dataset, GradientBoostingClassifier with the on-circle method and RandomForestClassifier with any strategy consistently benefit from the modification techniques, while GradientBoostingClassifier shows mixed results.

In the Birds dataset, the ExtraTreesClassifier shows F-scores substantially above the baseline in all cases, with gains ranging from 19.61% to 7.85%. The GradientBoostingClassifier produced F-scores below the baseline, suggesting that the modification techniques do not improve performance for this model on the Bird dataset. The F-scores are mostly above the baseline for the RandomForestClassifier, indicating improvements, except 60% of trajectory points modified by the in-circle strategy. For the Bird dataset, ExtraTreesClassifier and RandomForestClassifier show improvements with the augmentation techniques, while GradientBoostingClassifier struggles to benefit from them.

In summary, the analysis in relation to the baseline suggests that the impact of the data augmentation algorithm techniques varies depending on the dataset and the machine learning model, and our proposed algorithm indeed achieves performance improvements. For the Geolife and Traffic datasets, the techniques often lead to improvements when using GradientBoostingClassifier and RandomForestClassifier. For the Bird dataset, ExtraTreesClassifier and RandomForestClassifier benefited the most from the techniques, while GradientBoostingClassifier showed no improvement.

4 CONCLUSIONS AND FUTURE WORKS

In this study, we introduced a new algorithm for enhancing trajectory data by incorporating geographical noise, employing two distinct strategies – in-circle and on-circle. Our fundamental assertion is that the augmentation of trajectory data should precede feature extraction in trajectory classification tasks. Applying random noise to the trajectory features may lead to data inconsistencies. Hence, our algorithm is devised to randomly select trajectory data points

	Model	Baseline	on-circle 20%	on-circle 40%	on-circle 60%	in-circle 20%	in-circle 40%	in-circle 60%
Geolife	ExtraTreesClassifier	63.47%	63.76%	58.73%	63.28%	63.04%	63.36%	61.57%
	GradientBoostingClassifier	61.23%	65.95%	64.91%	63.50%	63.55%	61.04%	59.63%
	RandomForestClassifier	66.62%	68.20%	65.13%	67.70%	68.29%	67.95%	66.03%
Traffic	ExtraTreesClassifier	83.68%	83.40%	83.40%	83.30%	83.41%	83.29%	83.41%
	GradientBoostingClassifier	69.60%	71.19%	72.46%	71.49%	67.80%	69.42%	68.01%
	RandomForestClassifier	79.68%	81.84%	82.26%	80.48%	81.77%	81.20%	80.47%
Bird	ExtraTreesClassifier	76.86%	90.59%	84.71%	96.47%	96.47%	95.63%	94.45%
	GradientBoostingClassifier	66.47%	59.75%	59.75%	58.63%	58.63%	62.30%	50.53%
	RandomForestClassifier	59.75%	61.18%	61.18%	61.18%	61.18%	66.47%	58.82%

Table 2: Baseline and average F-scores for different percentages of trajectory points selected to be geographically modified by our strategies.

and introduce geographical noise to these points, thereby generating augmented trajectories. This augmentation enables feature extraction to be conducted subsequently, ensuring data consistency and reliability. In our experiment, we evaluated how the incorporation of augmented trajectories affects the performance of machine learning algorithms in trajectory classification tasks. We observe that introducing augmented trajectories via our algorithm can positively impact the performance of machine learning models in many instances, thus paving the way for more robust strategies in trajectory data augmentation.

As future work, we envision this algorithm as the cornerstone for a comprehensive and robust framework for augmenting trajectory datasets. While our initial results have demonstrated promising outcomes even with a completely random selection of trajectories and trajectory points for noise injection, we recognize the importance of systematically controlling this noise to enhance its applicability in classification problems. Furthermore, we believe that innovating new techniques for ranking trajectories and trajectory points presents an exciting possibility for augmenting trajectories.

ACKNOWLEDGMENTS

This research was partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), Grant number RGPIN-2022-03909; the Institute for Big Data Analytics (IBDA) and the Ocean Frontier Institute (OFI) at Dalhousie University, Halifax - NS, Canada; and further funded by the Canada First Research Excellence Fund (CFREF), the Canadian Foundation for Innovation MERIDIAN cyberinfrastructure³, H2020-MSCA MASTER GA 777695. We thank all funding sources.

REFERENCES

- [1] GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A., AND BENGIO, Y. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).
- [2] HAIDRI, S., HARANWALA, Y. J., BOGORNY, V., RENSO, C., DA FONSECA, V. P., AND SOARES, A. Ptrail—a python package for parallel trajectory data preprocessing. *SoftwareX* 19 (2022), 101176.
- [3] HARANWALA, Y. J., HAIDRI, S., TRICCO, T. S., DA FONSECA, V. P., AND SOARES, A. A dashboard tool for mobility data mining preprocessing tasks. In *2022 23rd IEEE Int. Conf. on Mobile Data Management (MDM)* (2022).

- [4] JÚNIOR, A. S., RENSO, C., AND MATWIN, S. Analytic: An active learning system for trajectory classification. *IEEE computer graphics and applications* 37, 5 (2017), 28–39.
- [5] KARPATNE, A., EBERT-UPHOFF, I., RAVELA, S., BABAIE, H. A., AND KUMAR, V. Machine learning for the geosciences: Challenges and opportunities. *IEEE Transactions on Knowledge and Data Engineering* 31, 8 (2019), 1544–1554.
- [6] KÖLZSCH, A., MÜSKENS, G., GLAZOV, P., KRUCKENBERG, H., AND WIKELSKI, M. Data from: Goose parents lead migration v, 2020.
- [7] MAHARANA, K., MONDAL, S., AND NEMADE, B. A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings* 3, 1 (2022), 91–99. International Conference on Intelligent Engineering Approach(ICIEA-2022).
- [8] MUMUNI, A., AND MUMUNI, F. Data augmentation: A comprehensive survey of modern approaches. *Array* 16 (2022), 100258.
- [9] REBUFFI, S.-A., GOWAL, S., CALIAN, D. A., STIMBERG, F., WILES, O., AND MANN, T. A. Data augmentation can improve robustness. *Advances in Neural Information Processing Systems* 34 (2021), 29935–29948.
- [10] RENSO, C., SPACCAPIETRA, S., AND ZIMÁNYI, E. *Mobility data*. Cambridge University Press, 2013.
- [11] RONNEBERGER, O., FISCHER, P., AND BROX, T. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical image computing and computer-assisted intervention* (2015), 234–241.
- [12] SHERUB, S., WIKELSKI, M., FIEDLER, W., AND DAVIDSON, S. Data from: Behavioural adaptations to flight into thin air, 2016.
- [13] SHORTEN, C., AND KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *J. Big Data* 6 (2019), 60.
- [14] SOARES, A., ROSE, J., ETEMAD, M., RENSO, C., AND MATWIN, S. VISTA: A visual analytics platform for semantic annotation of trajectories. In *Proceedings of the 22nd International Conference on Extending Database Technology (EDBT)* (2019).
- [15] SPADON, G., FERREIRA, M. D., SOARES, A., AND MATWIN, S. Unfolding ais transmission behavior for vessel movement modeling on noisy data leveraging machine learning. *IEEE Access* 11 (2022), 18821–18837.
- [16] SPENCER, N. C., GILCHRIST, H. G., AND MALLORY, M. L. Annual movement patterns of endangered ivory gulls: the importance of sea ice. *PLoS One* 9, 12 (2014), e115231.
- [17] WANG, S., BAO, Z., CULPEPPER, J. S., AND CONG, G. A survey on trajectory data management, analytics, and learning. *ACM Comput. Surv.* 54, 2 (2022), 39:1–39:36.
- [18] ZHENG, Y., AND XIE, X. Learning transportation mode from raw gps data for geographic application on the web. In *Proceedings of the 17th World Wide Web Conference* (April 2008).

³<https://meridian.cs.dal.ca/>