

# On feature selection and evaluation of transportation mode prediction strategies

Mohammad Etemad

Institute for Big Data Analytics, Dalhousie University  
Halifax, NS, Canada  
etemad@dal.ca

Stan Matwin\*

Institute for Big Data Analytics, Dalhousie University  
Halifax, NS, Canada  
stan@cs.dal.ca

Amílcar Soares

Institute for Big Data Analytics, Dalhousie University  
Halifax, NS, Canada  
amilcar.soares@dal.ca

Luis Torgo

Faculty of Computer Science, Dalhousie University  
Halifax, NS, Canada  
ltorgo@dal.ca

## ABSTRACT

Transportation modes prediction is a fundamental task for decision making in smart cities and traffic management systems. Traffic policies based on trajectory mining can save money and time for authorities and the public. It may reduce the fuel consumption, commute time, and more pleasant moments for residents and tourists. Since the number of features that may be used to predict a user transportation mode can be substantial, finding a subset of features that maximizes a performance measure is worth investigating. In this work, we explore a wrapper and an information retrieval methods to find the best subset of trajectory features for a transportation mode dataset. Our results were compared with two related papers that applied deep learning methods. The results showed that our work achieved better performance. Furthermore, two types of cross-validation approaches were investigated, and the performance results show that the random cross-validation method may provide overestimated results.

## KEYWORDS

Trajectory mining, feature selection, cross-validation

## 1 INTRODUCTION

Trajectory mining is a very hot topic since positioning devices are now used to track people, vehicles, vessels, natural phenomena, and animals. It has applications including but not limited to transportation mode detection [3, 6, 7, 31, 33], fishing detection [4], tourism [8], vessels monitoring [5], and animal behaviour analysis [9]. There are also a number of topics in this field that need to be investigated further such as high performance trajectory classification methods [3, 6, 20, 31, 33], accurate trajectory segmentation methods [28, 30, 34], trajectory similarity and clustering [10, 17], dealing with trajectory uncertainty [15], active learning [29], and semantic trajectories [2, 22, 24]. These topics are highly correlated and solving one of them requires to some extent exploring more than one.

As one of the trajectory mining applications, transportation mode prediction is a fundamental task for decision making in smart cities and traffic management systems. Traffic policies that are designed based on trajectory mining can save money and time

for authorities and the public, may reduce the fuel consumption and commute time, and may provide more pleasant moments for residents and tourists. Since a trajectory is a collection of geolocations captured through time, extracting features that show the behavior of a trajectory is of prime importance. The number of features that can be generated for trajectory data is significant. However, some of these features are more important than others for the transportation mode prediction task. Selecting the best subset of features not only saves processing time but also may increase the performance of the learning algorithm. The features selection problem and the trajectory classification task were selected as the focus of this work. The contributions of this paper are listed below.

- We investigated several classifiers using their default parameters values and selected the one with the best performance.
- Using two distinct feature selection approaches, we investigated the best subset of features for transportation modes prediction.
- After finding the best subset of features for some classifiers, we compare our results with the works of [3] and [6]. The results showed that our approach performed better than the others from the literature.
- Finally, we investigate the differences between the two methods of cross-validation used by the literature on transportation mode prediction. The results show that the random cross-validation method may suggest overestimated results in comparison to user-oriented cross-validation.

The rest of this work is structured as follows. The related works are reviewed in section 2. The basic concepts and definitions are provided in section 3 and the proposed framework is presented in section 4. We provide our experimental results in section 5. Finally, the conclusions and future works are shown in section 6.

## 2 RELATED WORKS

Feature engineering is an essential part of building a learning algorithm. Some of the algorithms artificially extract features using representation learning methods; On the other hand, some studies select a subset from the handcrafted features. Both methods have advantages such as faster learning, less storage space, performance improvement of learning, and generalized models building [18]. These two methods are different from two perspectives. First, artificially extracting features generates a new set of features by learning, while feature selection chooses a subset of existing handcrafted ones. Second, selecting handcrafted features constructs more readable and interpretable models than

\*Institute for Computer Science, Polish Academy of Sciences, Warsaw and Postcode, Poland

artificially extracting features [18]. This work focuses on the handcrafted feature selection task.

Feature selection methods can be categorized into three general groups: filter methods, wrapper methods, and embedded methods [12]. Filter methods are independent of the learning algorithm. They select features based on the nature of data regardless of the learning algorithm [18]. On the other hand, wrapper methods are based on a kind of search, such as sequential, best first, or branch and bound, to find the best subset that gives the highest score on a selected learning algorithm [18]. The embedded methods apply both filter and wrapper [18]. Feature selection methods can be grouped based on the type of data as well. The feature selection methods that use the assumption of i.i.d. (independent and identically distributed) are conventional feature selection methods [18] such as laplacian methods [14] and spectral feature selection methods [32]. They are not designed to handle heterogeneous or auto-correlated data. Some feature selection methods have been introduced to handle heterogeneous data and stream data that most of them working on graph structure such as [11].

Conventional feature selection methods are categorized in four groups: similarity-based methods like laplacian methods[14], Information theoretical methods [26], sparse learning methods such as [19], and statistical based methods like chi2 [21]. Similarity-based feature selection approaches are independent of the learning algorithm, and most of them cannot handle feature redundancy or correlation between features[21]. Likewise, statistical methods like chi-square cannot handle feature redundancy, and they need some discretization strategies[21]. The statistical methods are also not effective in high dimensional space[21]. Since our data is not sparse and sparse learning methods need to overcome the complexity of optimization methods, and they were not a candidate for our experiments. On the other hand, information retrieval methods can handle both feature relevance and redundancy[21]. Furthermore, selected features can be generalized for learning tasks. Information gain, which is the core of Information theoretical methods, assumes that samples are independently and identically distributed. Finally, the wrapper method only sees the score of the learning algorithm and tries to maximize the score of the learning algorithm.

The most common evaluation metric reported in the related works is the accuracy of the models. Therefore, we use the accuracy metric to compare our work with others from literature. Since the data was imbalanced, we reported the f-score as well to give equal importance to precision and recall. Despite the fact that most of the related work applied the accuracy metric, it is calculated using different methods including random cross-validation, cross-validation with dividing users, cross-validation with mix users and simple division of the training and test set without cross-validation. The latter is a weak method that is used only in [35]. The random cross-validation or the conventional cross-validation was applied in [31], [20], and [3]. [33] mixed the training and test set according to users so that 70% of trajectories of a user goes to the training set and the rest goes to test set. Only [6] performed the cross-validation by dividing users between the training and test set. Because trajectory data is a kind of data with spatial and temporal dimensions, users can also be placed in the same semantic hierarchical structure such as students, worker, visitors, and teachers, a conventional cross-validation method could provide overestimated results as studied in [27].

### 3 NOTATIONS AND DEFINITIONS

*Definition 3.1.* A trajectory point,  $l_i \in L$ , so that  $l_i = (x_i, y_i, t_i)$ , where  $x_i$  is longitude and it varies from  $0^\circ$  to  $\pm 180^\circ$ ,  $y_i$  is latitude and it varies from  $0^\circ$  to  $\pm 90^\circ$ , and  $t_i$  ( $t_i < t_{i+1}$ ) is the capturing time of the moving object, and  $L$  is the set of all trajectory points.

A trajectory point can be assigned by some features that describe different attributes of the moving object with a specific time-stamp and location. The time-stamp and location are two dimensions that make trajectory point *spatio-temporal* data with two important properties: (i) *auto-correlation* and (ii) *heterogeneity* [1]. These features make the conventional cross validation less suitable [27].

*Definition 3.2.* A raw trajectory, or simply a trajectory  $\tau$ , is a sequence of trajectory points captured through time, where  $\tau = (l_i, l_{i+1}, \dots, l_n), l_i \in L, i \leq n$ .

*Definition 3.3.* A sub-trajectory is one of the consecutive subsequences of a raw trajectory generated by splitting the raw trajectory into two or more sub-trajectories.

For example, if we have one split point,  $k$ , and  $\tau_1$  is a raw trajectory then  $s_1 = (l_i, l_{i+1}, \dots, l_k)$  and  $s_2 = (l_{k+1}, l_{k+2}, \dots, l_n)$  are two sub trajectories generated by  $\tau_1$ .

*Definition 3.4.* The process of generating sub-trajectories from a raw trajectory is called *segmentation*.

We used a daily segmentation of raw trajectories and then segmented the data utilizing the transportation modes annotations to partition the data. This approach is also used in [6] and [3].

*Definition 3.5.* A point feature is a measured value  $F^P$ , assigned to each trajectory points of a sub trajectory  $s$ .

$$F^P = (f_i, f_{i+1}, \dots, f_n) \quad (1)$$

Notation 1 shows the feature  $F^P$  for sub trajectory  $s$ . For example, speed can be a point feature since we can calculate the speed of a moving object for each trajectory point. Since we need two trajectory points to calculate speed, we assume the speed of the first trajectory point is equal to the speed of the second trajectory point.

*Definition 3.6.* A trajectory feature is a measured value  $F_t$ , assigned to a sub trajectory,  $s$ .

$$F_t = \frac{\sum f_k}{n} \quad (2)$$

Equation 2 shows the feature  $F_t$  for sub trajectory  $s$ . For example, the speed mean can be a trajectory feature since we can calculate the speed mean of a moving object for a sub trajectory.

The  $F_t^P$  is the notation for all trajectory features that generated using point feature  $p$ . For example,  $F_t^{speed}$  represents all the trajectory features derived from *speed* point feature. Moreover,  $F_{mean}^{speed}$  denotes the mean of the trajectory features derived from the *speed* point feature.

### 4 THE FRAMEWORK

In this section, the sequence of steps of q framework with eight steps are explained (Figure 1). The first step groups the trajectory points by *Trajectory id* to create daily sub-trajectories (segmentation). Sub-trajectories with less than ten trajectory points were discarded to avoid generating low-quality trajectories.

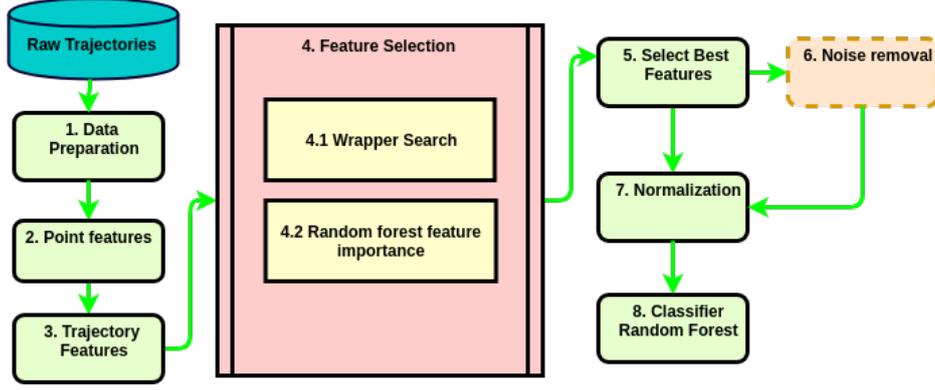


Figure 1: The steps of the applied framework to predict transportation modes.

Point features including speed, acceleration, bearing, jerk, bearing rate, and the rate of the bearing rate were generated in step two. The features speed, acceleration, and bearing were first introduced in [34], and jerk was proposed in [3]. The very first point feature that we generated was duration. This is the time difference between two trajectory points. This feature gives us essential information including some of the segmentation position points, loss signal points, and is useful in calculating point features such as speed, and acceleration. The distance was calculated using the haversine formula. Having duration and distance as two point features, we calculate speed, acceleration and jerk using Equation 3, 4, and 5 respectively. A function to calculate the bearing ( $B$ ) between two consecutive points was also implemented and is detailed in Equation 6, where  $\phi_i, \lambda_i$  is the start point,  $\phi_{i+1}, \lambda_{i+1}$  the end point.

$$S_i = \frac{Distance_i}{Duration_i} \quad (3)$$

$$A_{i+1} = \frac{(S_{i+1} - S_i)}{\Delta t} \quad (4)$$

$$J_{i+1} = \frac{(A_{i+1} - A_i)}{\Delta t} \quad (5)$$

$$B_{i+1} = \text{atan2}(\sin \lambda_{i+1} - \lambda_i \cos \phi_{i+1}, \cos \phi_i \sin \phi_{i+1} - \sin \phi_i \cos \phi_{i+1} \cos \lambda_{i+1} - \lambda_i) \quad (6)$$

Two new features were introduced in [7], named bearing rate, and the rate of the bearing rate. Applying equation 7, we computed the bearing rate.

$$B_{rate(i+1)} = \frac{(B_{i+1} - B_i)}{\Delta t} \quad (7)$$

$B_i$  and  $B_{i+1}$  are the bearing point feature values in points  $i$  and  $i + 1$ .  $\Delta t$  is the time difference. The rate of the bearing rate point feature is computed using equation 8. Since extensive calculations are done with trajectory points, it was necessary an efficient way to calculate all these equations for each trajectory. Therefore, the code was written in a vectorized manner in Python programming language which is faster than other online available python versions of the bearing calculation. It can be possible to gain more performance using other languages like C/C++.

$$Br_{rate(i+1)} = \frac{(B_{rate(i+1)} - B_{rate(i)})}{\Delta t} \quad (8)$$

After calculating the point features for each trajectory, the trajectory features were extracted in step three. Trajectory features were divided into two different types including global trajectory

features and local trajectory features. Global features, like the Minimum, Maximum, Mean, Median, and Standard Deviation, summarize information about the whole trajectory and local trajectory features, percentiles (10, 25, 50, 75, and 90), describe a behavior related to part of a trajectory. The local trajectory features extracted in this work were the percentiles of every point feature. Five different global trajectory features were used in the models tested in this work. In summary, we computed 70 trajectory features (10 statistical measures including five global and five local features calculated for 7 point features) for each sample trajectory. In Step 4, two feature selection approaches were performed, wrapper search and information retrieval feature importance. According to the best accuracy results for development set, a subset of top 19 features was selected in step 5. The code implementation of all these steps is available at <https://github.com/metemaad/TrajLib>.

In step 6, the framework deals with noise in the data optionally. This means that we ran the experiments with and without this step. Finally, we normalized the features (step 7) using the Min-Max normalization method to avoid saturation, since this method preserves the relationship between the values to transform features to the same range and improves the quality of the classification process [13]. Another possible method is Z normalization; however, finding the best normalization method was out of the scope of this work.

## 5 EXPERIMENTS

In this section, we detail the four experiments performed in this work. In this work, we used the GeoLife dataset [34]. This dataset has 5,504,363 GPS records collected by 69 users, and is labeled with eleven transportation modes: taxi (4.41%); car (9.40%); train (10.19%); subway (5.68%); walk (29.35%); airplane (0.16%); boat (0.06%); bike (17.34%); run (0.03%); motorcycle (0.006%); and bus (23.33%). Two primary sources of uncertainty of the Geolife dataset are device and human error. This inaccuracy can be categorized in two major groups, *systematic errors* and *random errors* [16]. The *systematic error* occurs when the recording device cannot find enough satellites to provide precise data. The *random error* can happen because of atmospheric and ionospheric effects. Furthermore, the data annotation process has been done after each tracking as [34] explained in the Geolife dataset documentation. As humans, we are all subject to fail in providing precise information; it is possible that some users forget to annotate the trajectory when they switch from one transportation mode to

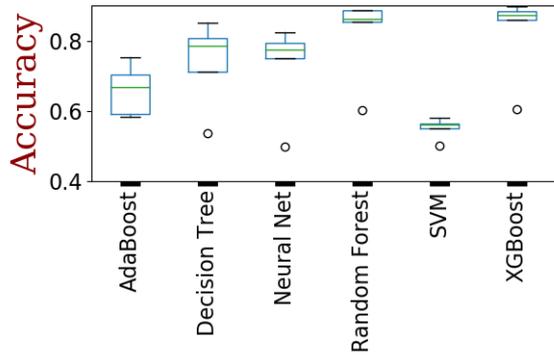


Figure 2: Among the trained classifiers random forest achieved the highest mean accuracy.

another. For example, the changes in the speed pattern might be a representation of human error.

Moreover, we divide data into two folds: development set and validation set. These two folds divided in a way that each user can be either in development set or validation set. Therefore, there is no overlap in terms of users. This division is applied for user-oriented cross validation. We divide the validation fold to five folds to do the cross validation and using this fold to compare our results with related work.

The best classifier using their default input parameters (Section 5.1) was found in our first experiment (check scikit-learn documentation<sup>1</sup> for the classifiers default parameters values). Tuning the classifiers parameters may lead to find a better classifier, but doing a grid search is expensive and does not change the framework. In our second experiment (Section 5.2), the wrapper and information theoretical methods are used to search the best subset of our 70 features for the transportation modes prediction task. The third experiment (Section 5.3) is a comparison between [6] and [3] and our implementation. In the last experiment (Section 5.4), the type of cross validation was investigated.

In order to avoid using non-parametric statistical tests, we repeat the experiments with different seeds and collect more than 30 samples for performing the statistical tests. According to central limit theorem, we can assume these samples follow the normal distribution. Therefore, t-test results are reported.

## 5.1 Classifier selection

In this experiment, we investigated among six classifiers, which classifier is the best. The experiment settings use to conventional cross-validation and to perform the transportation mode prediction task showed on [3]. XGBoost, SVM, decision tree, random forest, neural network, and adaboost are six classifiers that were applied in the reviewed literature [7, 31, 33, 35].<sup>2</sup> The dataset is filtered based on labels that have been applied in [3] (e.g., walking, train, bus, bike, driving) and no noise removal method was applied. The classifiers mentioned above were trained, and the accuracy metric was calculated using random cross-validation similar to [20], [31], and [3]. This experiment was repeated for eight randomly selected seeds (8, 65, 44, 7, 99, 654, 127, 653) to generate more than 30 result samples that make safe to assume a normal distribution for results based on central limit theorem. The results of cross validation accuracy, presented in figure 2,

<sup>1</sup>[https://scikit-learn.org/stable/supervised\\_learning.html#supervised-learning](https://scikit-learn.org/stable/supervised_learning.html#supervised-learning)

<sup>2</sup>available on <https://github.com/metemaad/trajpred>

show that the random forest performs better than other models ( $\mu_{accuracy} = 0.8189, \sigma = 0.10\%$ ) on the development set.

The results of cross validation f-score, presented in figure 3, show that the random forest performs better than other models ( $\mu_{f_1} = 0.8179, \sigma = 0.12\%$ ) on the development set.

The second best model was XGBoost ( $\mu_{accuracy} = 0.8245, \sigma = 0.11\%$ ). The XGBoost was ranked the second because a paired T-Test indicated that the random forest classifier results were not statistically significantly higher than the XGBoost classifier results, but since it has a higher variance than random forest, we decided to rank random forest as first. In the other hand, paired t-tests indicated that the random forest classifier results were statistically significantly higher than the SVM, decision tree, Neural Network, and Adaboost classifiers results.

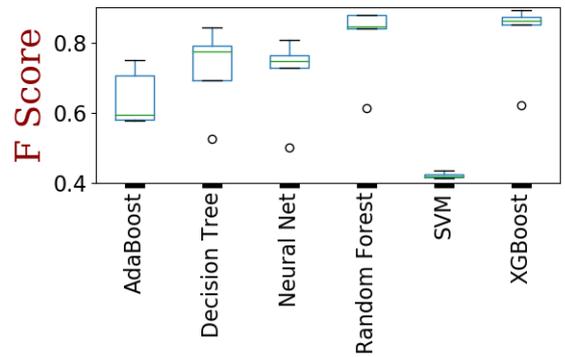


Figure 3: Among the trained classifiers random forest achieved the highest mean F-score.

## 5.2 Feature selection using wrapper and information theoretical methods

The second experiment aims to select the best features for transportation modes prediction task for the Geolife dataset.

We select one method from filter category which is information theoretical method to see the effect of the heterogeneity of data on feature selection method. Another method was selected from wrapper category which is the full search wrapper method. Filter methods suffer from having i.i.d assumption, while wrapper methods do not. Therefore, comparing these two methods shows the importance of taking into account the heterogeneity of features of trajectory data.

We selected the wrapper feature selection method because it can be used with any classifier. Using this approach, we first defined an empty set for selected features. Then, we searched all the trajectory features one by one to find the best feature to append to the selected feature set. The maximum accuracy score was the metric for selecting the best feature to append to selected features. After, we removed the selected feature from the set of features and repeated the search for union of selected features and next candidate feature in the feature set. We selected the labels applied in [6] and the same cross-validation technique.

The results are shown in figure 4. The results of this method suggest that the top 19 features get the highest accuracy. Therefore, we selected this subset as the best subset for classification purposes using the random forest algorithm.

Information theoretical feature selection is one of the methods widely used to select essential features. Random forest is a

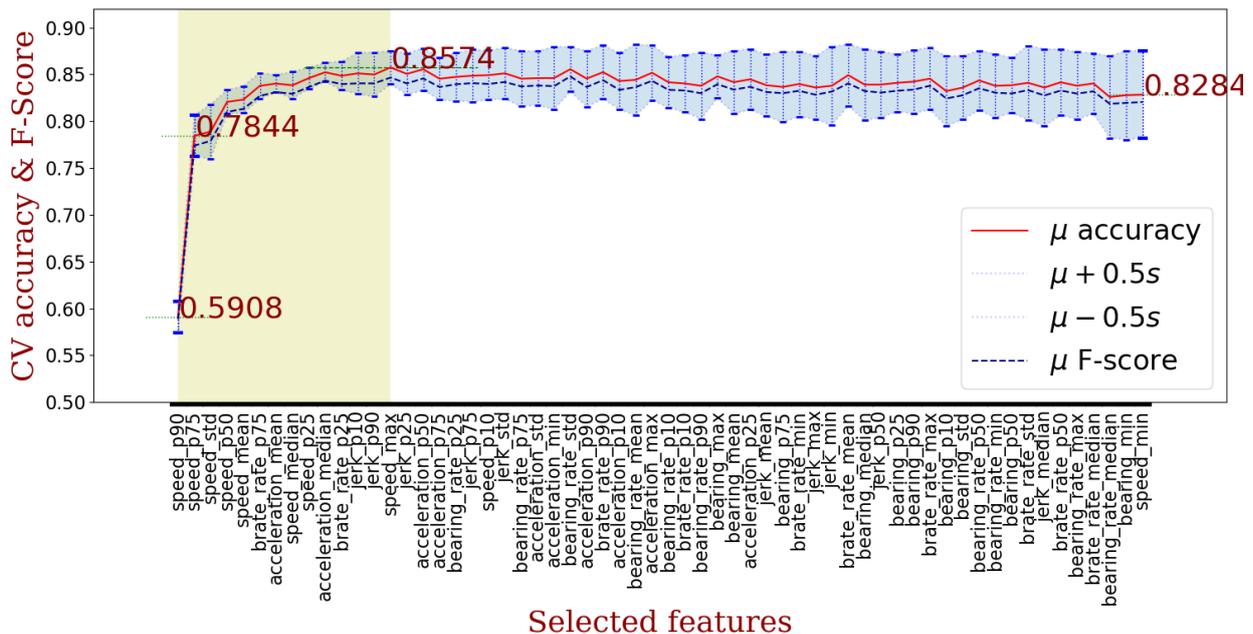


Figure 4: Accuracy of random forest classifier for incremental appending features ranked by random forest feature importance.

classifier that has embedded feature selection using information theoretical metrics. We calculated the feature importance using random forest. Then, each feature is appended to the selected feature set and calculating the accuracy score for random forest classifier. The user-oriented cross-validation was used here, and the target labels are similar to [6]. Figure 5 shows the results of cross-validation for appending features with respect to the importance rank suggested by the random forest. We chose the wrapper approach results since it produces statistically significant higher accuracy score.

### 5.3 Comparison with the related work

In this third experiment, we filtered transportation modes which have been used by [6] for evaluation. We divided the validation fold into the training and test folds in a way that each user can appear only either in the training or test fold. The top 19 features were selected to be used in this experiment which is the best features subset mentioned in section 5.2. Therefore, we approximately divided 80% of the data as training and 20% of the data as the test set.

We selected [6] because this is the only paper that divided the dataset in a way that isolated users in training and test set. Moreover, This research applied the handcrafted features and interpretable classifiers, while [3] did not isolated users and used representation learning features. Therefore, these two research are in the two ends and spectrum and comparing our results with theirs and may provide insights for validating our results.

We assume the *bayes error* is the minimum possible error and human error is near to the *bayes error* [23]. Avoidable bias is defined as the difference between the training error and the human error. Achieving the performance near to the human performance in each task is the primary objective of the research. The recent advancements in deep learning lead to achieving some performance level even more than the performance of doing the

task by human because of using large samples and scrutinizing the data to fine clean it. However, “we cannot do better than *bayes error* unless we are overfitting”. [23]. Having noise in GPS data and human error, as we discussed, suggest that the avoidable bias is more than five percent. This ground truth was our base to exclude papers that reported more than 95% of accuracy.

Thus, we compare our accuracy per segment results, repeated for 8 different seeds, against [6] mean accuracy, 67.9%. A one-sample T-test indicated that our accuracy results (70.97%) are higher and statistically significantly better than [6]’s results (67.9%),  $p=0.0182$ .

The label set for [3]’s research is walking, train, bus, bike, taxi, subway, and car so that the taxi and car are merged and called driving. Moreover, subway and train merged and called the train class. We filtered the Geolife data to get the same subsets as [3] reported based on that. Then, we randomly selected 80% of the data as the training and the rest as test set, we applied five-fold cross-validation and repeated this for 8 different seeds. The best subset of features was the same as the previous experiment (Section 5.2). Running the random forest classifier with 50 estimators, using SKlearn implementation [25], results on a mean accuracy of 87.16% for the five-fold cross-validation. A one-sample T-test indicated that our accuracy results (87.16%) are higher and statistically significantly better than [3]’s results (84.8%),  $p=2.27e-12$ .

We avoided using the noise removal method in the above experiment because we believe we do not have access to labels of the test dataset and using this method only increases our accuracy unrealistically.

### 5.4 Effects of types of cross-validation

To visualize the effect of type of cross-validation on transportation modes prediction task, we set up a controlled experiment. We used the same classifiers and same features to calculate the

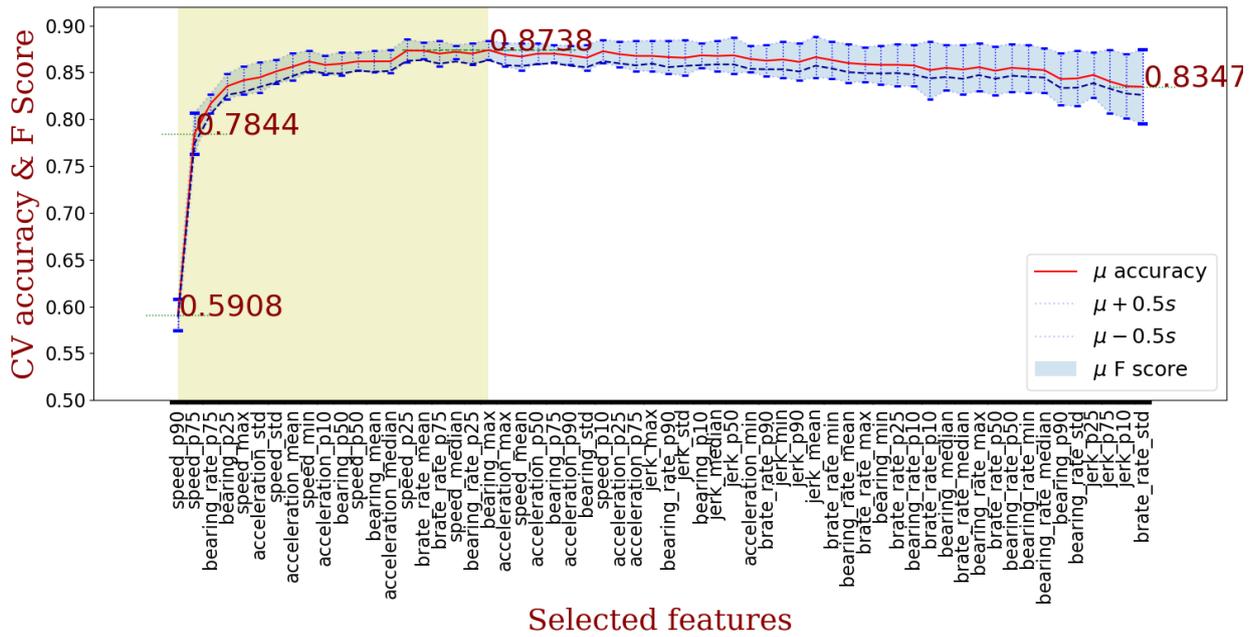


Figure 5: Accuracy of random forest classifier for incremental appending best features

cross-validation accuracy on the whole dataset. Only the type of cross-validation is different in this experiment, one is random, and another is user-oriented cross-validation. Figure 6 shows that there is a considerable difference between the cross-validation accuracy results of user-oriented cross-validation and random cross-validation.

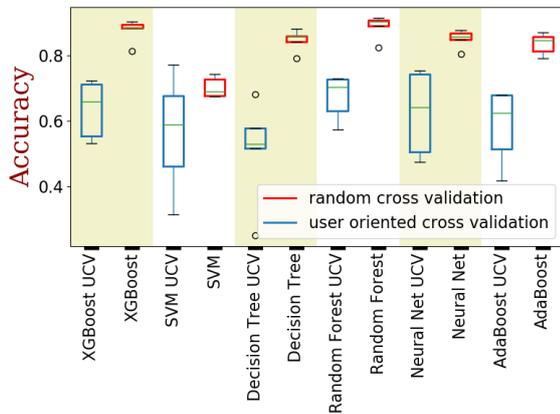


Figure 6: The accuracy cross-validation results for user oriented cross-validation and random cross-validation

Furthermore, figure 7 shows that there is a considerable difference between the cross-validation f-score results of user-oriented cross-validation and random cross-validation.

These results indicate that random cross-validation provides overestimated accuracy and f-score results. Since the correlation between user-oriented cross-validation results is less than random cross-validation, proposing a specific cross-validation method for evaluating the transportation mode prediction is a topic that needs attention.

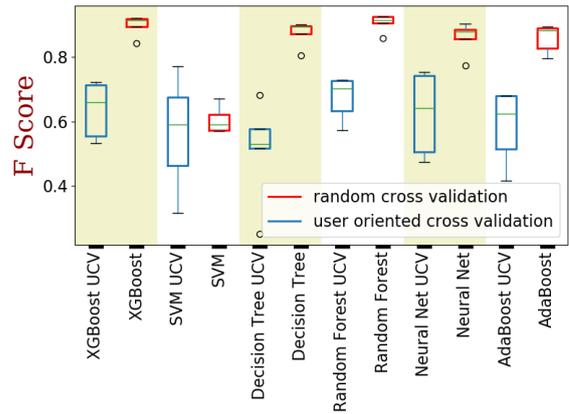


Figure 7: The F-score cross-validation results for user oriented cross-validation and random cross-validation

## 6 CONCLUSIONS

In this work, we reviewed some recent transportation modes prediction methods and feature selection methods. We proposed a framework for transportation modes prediction and four experiments were conducted to cover different aspects of transportation modes prediction.

First, the performance of six recently used classifiers for the transportation modes prediction was evaluated. The results showed that the random forest classifier performs the best among all the evaluated classifiers. The SVM was the worst classifier, and the accuracy result of XGBoost was competitive with the random forest classifier.

In the second experiment, the effect of features using two different approaches, the wrapper method and information theoretical method were evaluated. The wrapper method shows that we can achieve the highest accuracy using the top 19 features.

Both approaches suggest that the  $F_{p90}^{speed}$  (the percentile 90 of the speed as defined in section 3) is the most essential feature among all 70 introduced features. This feature is robust to noise since the outlier values do not contribute to the calculation of percentile 90.

In the third experiment, the best model was compared with the results showed in [6] and [3]. The results show that our suggested model achieved a higher accuracy. Our applied features are readable and interpretable in comparison to [6] and our model has less computational cost.

Finally, we investigate the effects of user-oriented cross-validation and random cross-validation in the last experiments. The results showed that random cross-validation provides overestimated results in terms of the analyzed performance measures.

We intend to extend this work in many directions. The spatiotemporal characteristic of trajectory data is not taken into account in most of the works from literature (e.g. autocorrelation and heterogeneity). Fine tuning the classification models with grid search and automatic (e.g. Genetic Algorithms, Racing algorithms, and meta-learning) methods. We also intend to deeply investigate the effects of cross-validation and other strategies like holdout in trajectory data. Finally, space and time dependencies can also be explored to tailor features for transportation means prediction.

## ACKNOWLEDGMENTS

The authors would like to thank NSERC (Natural Sciences and Engineering Research Council of Canada) for financial support.

## REFERENCES

- [1] Gowtham Atluri, Anuj Karpatne, and Vipin Kumar. 2017. Spatio-Temporal Data Mining: A Survey of Problems and Methods. *arXiv arXiv:1711.04710* (2017).
- [2] Vania Bogorny, Chiara Renso, Artur Ribeiro de Aquino, Fernando de Lucca Siqueira, and Luis Otavio Alvares. 2014. Constant—a conceptual data model for semantic trajectories of moving objects. *Transactions in GIS* 18, 1 (2014), 66–88.
- [3] Sina Dabiri and Kevin Heaslip. 2018. Inferring transportation modes from GPS trajectories using a convolutional neural network. *Transportation Research Part C: Emerging Technologies* 86 (2018), 360–371.
- [4] Erico N de Souza, Kristina Boerder, Stan Matwin, and Boris Worm. 2016. Improving fishing pattern detection from satellite AIS using data mining and machine learning. *PLoS one* 11, 7 (2016), e0158248.
- [5] Renata Dividino, Amílcar Soares, Stan Matwin, Anthony W Isenor, Sean Webb, and Matthew Brousseau. 2018. Semantic Integration of Real-Time Heterogeneous Data Streams for Ocean-related Decision Making. In *Big Data and Artificial Intelligence for Military Decision Making*. STO. <https://doi.org/10.14339/STO-MP-IST-160-S1-3-PDF>
- [6] Yuki Endo, Hiroyuki Toda, Kyosuke Nishida, and Akihisa Kawanobe. 2016. Deep feature extraction from trajectories for transportation mode estimation. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 54–66.
- [7] Mohammad Etemad, Amílcar Soares Júnior, and Stan Matwin. 2018. Predicting Transportation Modes of GPS Trajectories using Feature Engineering and Noise Removal. In *Advances in AI: 31st Canadian Conf. on AI, Canadian AI 2018, Toronto, ON, CA, Proc. 31*. Springer, 259–264.
- [8] Shanshan Feng, Gao Cong, Bo An, and Yeow Meng Chee. 2017. POI2Vec: Geographical Latent Representation for Predicting Future Visitors. In *AAAI*.
- [9] Sabrina Fosssette, Victoria J Hobson, Charlotte Girard, Beatriz Calmettes, Philippe Gaspar, Jean-Yves Georges, and Graeme C Hays. 2010. Spatio-temporal foraging patterns of a giant zooplanktivore, the leatherback turtle. *Journal of Marine Systems* 81, 3 (2010), 225–234.
- [10] Andre Salvaro Furtado, Laercio Lima Pilla, and Vania Bogorny. 2018. A branch and bound strategy for Fast Trajectory Similarity Measuring. *Data Knowledge Engineering* 115 (2018), 16 – 31. <https://doi.org/10.1016/j.datak.2018.01.003>
- [11] Quanquan Gu and Jiawei Han. 2011. Towards feature selection in network. In *Proceedings of the 20th ACM ICIKM*. ACM, 1175–1184.
- [12] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of ML research* 3, Mar (2003), 1157–1182.
- [13] Jiawei Han, Jian Pei, and Micheline Kamber. 2011. *Data mining: concepts and techniques*. Elsevier.
- [14] X He, D Cai, and P Niyogi. 2005. Laplacian Score for Feature Selection, *Advances in Neural Information Processing Systems*. (2005).
- [15] Sungsoo Hwang, Cynthia VanDeMark, Navdeep Dhatt, Sai V Yalla, and Ryan T Crews. 2018. Segmenting human trajectory data by movement states while addressing signal loss and signal noise. *International Journal of Geographical Information Science* (2018), 1–22.
- [16] Jungwook Jun, Randall Guensler, and Jennifer Ogle. 2006. Smoothing methods to minimize impact of global positioning system random error on travel distance, speed, and acceleration profile estimates. *Transportation Research Record: Journal of the TRB* 1, 1972 (2006), 141–150.
- [17] Hye-Young Kang, Joon-Seok Kim, and Ki-Joune Li. 2009. Similarity measures for trajectory of moving objects in cellular space. In *SIGAPP09*. 1325–1330.
- [18] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. 2017. Feature selection: A data perspective. *CSUR* 50, 6 (2017), 94.
- [19] Zechao Li, Yi Yang, Jing Liu, Xiaofang Zhou, Hanqing Lu, et al. 2012. Unsupervised feature selection using nonnegative spectral analysis. In *AAAI*, Vol. 2.
- [20] Hongbin Liu and Ickjai Lee. 2017. End-to-end trajectory transportation mode classification using Bi-LSTM recurrent neural network. In *Intelligent Systems and Knowledge Engineering (ISKE), 2017 12th International Conference on*. IEEE, 1–5.
- [21] Huan Liu and Rudy Setiono. 1995. Chi2: Feature selection and discretization of numeric attributes. In *Tools with artificial intelligence, 1995. proceedings., seventh international conference on*. IEEE, 388–391.
- [22] B. N. Moreno, A. Soares Júnior, V. C. Times, P. Tedesco, and Stan Matwin. 2014. Weka-SAT: A Hierarchical Context-Based Inference Engine to Enrich Trajectories with Semantics. In *Advances in Artificial Intelligence*. Springer International Publishing, Cham, 333–338. [https://doi.org/10.1007/978-3-319-06483-3\\_34](https://doi.org/10.1007/978-3-319-06483-3_34)
- [23] Andrew Ng. 2016. Nuts and bolts of building AI applications using Deep Learning. NIPS.
- [24] Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady Andrienko, Natalia Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, Jose Macedo, Nikos Pelekis, Yannis Theodoridis, and Zhixian Yan. 2013. Semantic Trajectories Modeling and Analysis. *ACM Comput. Surv.* 45, 4, Article 42 (Aug. 2013), 32 pages.
- [25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *MLR* (2011).
- [26] Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence* 27, 8 (2005), 1226–1238.
- [27] David R Roberts, Volker Bahn, Simone Ciuti, Mark S Boyce, Jane Elith, Gurutzeta Guillera-Aroita, Severin Hauenstein, José J Lahoz-Monfort, Boris Schröder, Wilfried Thuiller, et al. 2017. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography* 40, 8 (2017), 913–929.
- [28] A. Soares Júnior, B. N. Moreno, V. C. Times, S. Matwin, and L. A. F. Cabral. 2015. GRASP-UTS: an algorithm for unsupervised trajectory segmentation. *International Journal of Geographical Information Science* 29, 1 (2015), 46–68.
- [29] A. Soares Júnior, C. Renso, and S. Matwin. 2017. ANALYTIC: An Active Learning System for Trajectory Classification. *IEEE Computer Graphics and Applications* 37, 5 (2017), 28–39. <https://doi.org/10.1109/MCG.2017.3621221>
- [30] A. Soares Júnior, V. Cesario Times, C. Renso, S. Matwin, and L. A. F. Cabral. 2018. A Semi-Supervised Approach for the Semantic Segmentation of Trajectories. In *2018 19th IEEE International Conference on Mobile Data Management (MDM)*. 145–154.
- [31] Xiao. 2017. Identifying Different Transportation Modes from Trajectory Data Using Tree-Based Ensemble Classifiers. *ISPRS* 6, 2 (2017), 57.
- [32] Zheng Zhao and Huan Liu. 2007. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on Machine learning*. ACM, 1151–1157.
- [33] Yu Zheng, Yukun Chen, Quannan Li, Xing Xie, and Wei-Ying Ma. 2010. Understanding transportation modes based on GPS data for web applications. *TWEB* 4, 1 (2010), 1.
- [34] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. 2008. Understanding mobility based on GPS data. In *UbiComp 10th*. ACM, 312–321.
- [35] Qihui Zhu, Min Zhu, Mingzhao Li, Min Fu, Zhibiao Huang, Qihong Gan, and Zhenghao Zhou. 2018. Transportation modes behaviour analysis based on raw GPS dataset. *International Journal of Embedded Systems* 10, 2 (2018), 126–136.