MobApp: A Data Visualization Tool for Trajectory Analysis

Michael O. Cruz[1], Fernando Neto[1], Luciano Barbosa[1]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50732-970 – Recife – PE – Brazil

michaeloc9@gmail.com.br,fernando.barros@ferreiracosta.com.br
,luciano@cin.ufpe.br

Abstract. *In this paper, we present a demo called MobApp, a data visualization application to facilitate spatio-temporal trajectories analysis. It aims to help domain analysts and practitioners/scientists to analyze and get insights from real-world trajectories. The tool supports: (1) exploratory analysis of trajectories, which allows users to visualize selected trajectories on a map and provides some statistics about trajectories' points; (2) visualization of anomalous trajectories and the regions where the anomalies occur, detected by an anomaly detection model; and (3) evaluation of anomaly detection models to compare their performance.*

1. Introduction

With the advances in mobile computing and GPS devices, a massive volume of mobility data has been produced. For example, companies such as Uber and Waze have collected a great volume of trajectories from users' movements, and NASA archives more than 4TB of stars and asteroids' movements daily[1]. To deal with such a deluge of data, there is a need for tools to help the understanding of the data in order to find useful insights and possible anomalies, caused by erroneous readings or actual anomalous mobility behaviors [Belhadi et al. 2020].

In the literature, there are few tools to help understand GPS data for EDA analysis. For example, [Bose 2009] present a desktop application called V-Analytics to analyze spatio-temporal data. V-Analytics has many functionalities, however, it does not provide an approach to detect anomalies and pinpoint the anomaly regions. [Pappalardo et al. 2019], in turn, propose a Python library called Scikit-mobility to analyze mobility data and simulate human trajectory behavior. The library provides a robust tool to analyze trajectories. However, the library does not present anomaly detection algorithms. To the best of our knowledge, there is no online approach to fill those gaps.

This demo presents MobApp[2], a prototype data-driven open-source web application developed to help domain analysts and practitioners/scientists to analyze real-world trajectories. MobApp supports three different visual analyses. The first one allows analysts to visualize trajectories on a map in a specific period of time along with a plot showing information about the trajectories' points (e.g., speed and acceleration). In the second analysis, the user can observe anomalous trajectories and the region where the anomaly occurs, detected by an anomaly detection algorithm. The last allows users to visually compare on the map the output of different anomaly detection models on the

[1]https://www.nasa.gov/open/data.html
[2]https://github.com/Mobapp-Dashboard

same trajectory. Different users might be interested in these functionalities. For instance, data scientists might use MobApp to inspect trajectories, looking for issues in the data, or the result of anomaly detection models, useful for model diagnostics[Buja et al. 2009]. Transit authorities can use MobApp to automatically identify anomalous bus trajectories, which might indicate traffic congestion or driver misconduct.

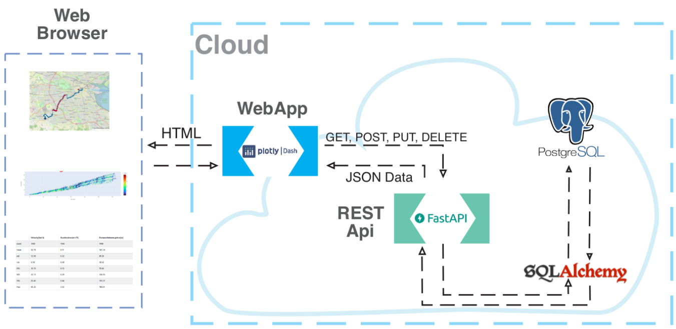


Figure 1. MobApp Architecture.

For this demonstration, we will present to the attendees MobApp running on a Web browser. The demo uses a dataset from the city of Dublin[3] with 1,699.022 points collected from Jan 01, 2013, to Jan 07, 2013. In this demo, we show the following use cases:

- An exploratory data analysis based upon speed and days of the week;
- Insights from the results of the anomaly detection visualization;
- The visual comparison of different anomaly detection models.

2. MobApp Architecture

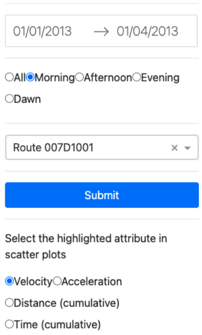
Figure 1 shows the architecture of MobApp. The application comprises two modules: (i) REST API and (ii) WebApp. The first receives requests from clients, and returns the results of the selected analysis. The second module, the WebApp, is the system's user interface (UI), which interacts to the REST API to provide the system's functionalities to end users.

2.1. WebApp

Our WebApp module is the application's front-end. It was developed using the library Dash[4], which is a Web-based Python UI framework for data science apps. The MobApp presents three web screens: (i) Exploratory Data Analysis (EDA), (ii) Anomaly Trajectory Filter, and (iii) Model Evaluation.

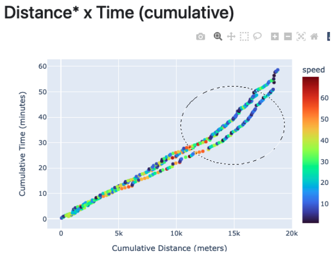
Exploratory Data Analysis (EDA). The EDA screen shows general analyses of the trajectories. Using this functionality, users can: (1) apply filters to select trajectories in the sidebar to select trajectories based on the period of time (Figure 2a); (2) visualize on a map the chosen trajectories (Figure 2b), a cumulative scatter plot (Figure 2c), and a summarization table (Figure 2c). In addition, the MobApp summarizes statistic measures like average, min, max, and standard deviation, according to the selected filter (Figure 2d).

[3]https://data.gov.ie/dataset/dublin-bus-gps-sample-data-from-dublin-city-council-insight-project
[4]https://plotly.com/dash/



(a) Sider-bar filter

(b) Trajectories map



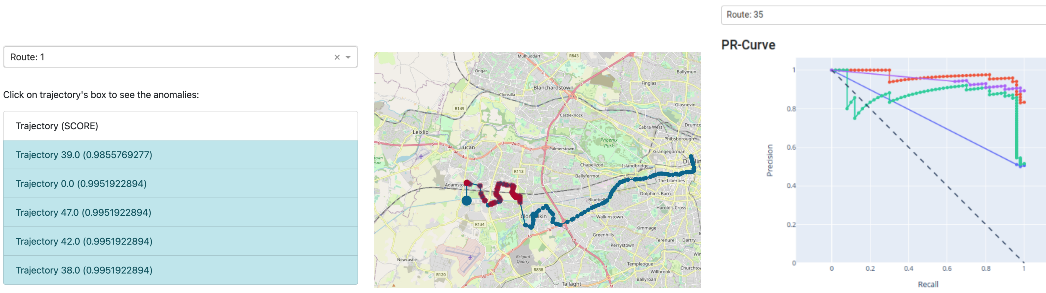
(c) Dispersion graph

Summary

	Velocity(km/h)	Acceleration(m/s^2)	Distance between points(m)	Time between points(s)
count	406	406	406	406
mean	22.32	0.31	134.58	24.79
std	15.08	0.23	87.56	17.77
min	1.07	0.00	10.26	8.00
25%	9.58	0.11	63.94	19.00
50%	19.61	0.26	121.63	20.00
75%	32.77	0.46	194.79	22.00
max	69.87	1.41	648.87	240.00

(d) Basics statistics

Figure 2. WebApp Exploratory Data Analysis.

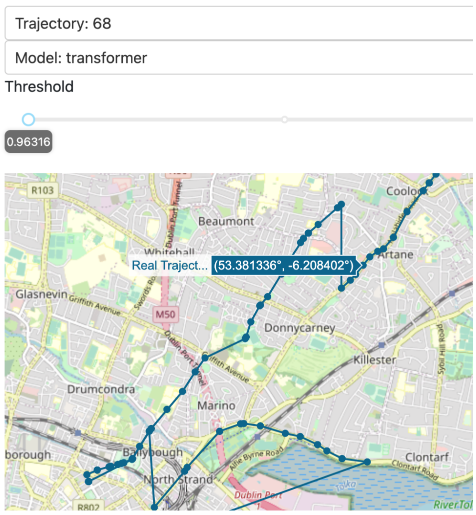


(a) Top-5 (b) Anomalous region (c) Precision-Recall

Figure 3. Web App Anomaly Trajectory Scores and Evaluation.

Anomaly Trajectory Filter. This screen allows users to visualize anomalous trajectories for a given route. As shown in Figure 3, it returns the top-k anomalous trajectories (id trajectory and score). Note that the application allows users to choose a route and click on each returned trajectory to analyze the anomaly behavior on the map. For that, MobApp shows the anomaly region (red points) on the map, which helps users to visually localize where the anomaly likely occurred.

Model Evaluation. This screen tries to facilitate the comparison between anomaly detection models. It is divided into two parts: routes and trajectories. The route screen allows users to compare the performance of anomaly detection models on trajectories in the same route. For this purpose, the application provides a drop-down component for users to select routes, visualize precision, and recall curves. Figure 3c shows the mentioned components and the four available algorithms used at this demo: GMVSAE [Liu et al. 2020], STOD [Cruz and Barbosa 2020], iBOAT [Chen et al. 2013], and RioBusData [Bessa et al. 2016]. The second part of this screen allows users to choose a particular trajectory, an anomaly detection algorithm, and a given anomaly threshold. Figure 4 depicts a map with the selected trajectory and a summary table composed of an anomaly score, threshold, precision, recall, and a result pointing if the trajectory is anomalous or not regarding the selected threshold.



(a) Trajectory and model selection

Route	35
Trajectory	68
Model	transformer
Anomaly Direction	Below threshold
Threshold	0.9631578900000001
Precision / Recall (on route)	0.89285714 / 1.0
Score	0.8051947951
Result	Anomaly DETECTED

(b) Table results

Figure 4. Detection on real-world trajectories.

2.2. REST API

MobApp's back-end has a REST API that provides endpoints for the analysis. It is implemented using the Python FastAPI framework [5]. The API's data is stored in a Postgres [6] SGBD due to its support to spatial and geographic objects, i.e., geo-based queries and spatial index. The API makes the endpoints available on JSON format, which allows clients to consume the provided trajectory analyses.

Currently, the API presents four endpoints. The first one is the Trajectory Metadata (*journeys-by-date*) that returns trajectories' ids given a start and end date. This endpoint is essential to filter proper trajectories when users select a range of data on the Exploratory Data Analysis screen, as shown in Figure 2a. The second, the GPS endpoint, returns trajectories for a given route[7], date interval, and selected features on the side-bar (e.g., period of the day and speed). The returned information of this endpoint is used on the EDA screen, where users can observe basic statistics from trajectories behavior. Specifically, the request returns a list of trajectories and their attributes, respectively, such as timestamp, speed, acceleration, delta distance, delta time (time difference between two consecutive points), and cumulative distance.

The third endpoint, the anomaly detection endpoint, returns the top-k anomaly trajectories of a given route along with their respective scores and trajectories ids. It

[5]https://fastapi.tiangolo.com/

[6]https://www.postgresql.org/

[7]A *route* can be defined as a set of road segments [Liu et al. 2017], whereby journeys in this route generate trajectories on different periods of time.

allows users to filter anomalous trajectories in descending order of scores. This endpoint is used by the WebApp, in the Anomaly Trajectory Filter. Lastly, the model evaluation endpoint requests a route and the name of the method to return recall, precision, and threshold from all trajectories.

3. Use Cases

Conference attendees will be able to see how MobApp can be used from three use cases, one for each analysis the system provides.

Exploratory Analysis. Visualizing general statistics of the data is the first step to understanding its characteristics and distribution. Using MobApp, practitioners/scientists can explore trajectory data to discover, for instance, which period of the day the buses are slowest, the impact of known events (e.g., heavy rain) in the traffic flow, etc. To illustrate a real scenario, a user accesses the EDA screen and selects the data period between 2013-01-01 and 2013-04-01 on the sider-bar, as shown in Figure 2a. Next, he selects the morning option as a period of the day and chooses route 007D1001 along with the velocity filter, all at the same sider-bar. As a result, the MobApp returns trajectories behavior on the maps, as shown in Figure 2b. In addition, the application presents a dispersion graph where the user can observe probably traffic jams effects between 10k and 16k of cumulative distance, as shown in Figure 2c. Overall, the EDA functionalities can help users understand traffic behavior and provide information for better decision-making.

Bus Anomaly Trajectories. Identifying anomaly trajectories for a public transportation agency is essential for implementing better policies in a city. For example, consider an analyst from a department of public transportation who decides to analyze thousands of daily trajectories (trips) to identify anomalies trajectories to support future decisions(releasing or retaining buses regarding detected anomalies). Unfortunately, it is not easy to manually inspect a massive volume of trajectories since they are multivariate temporal series with hundreds of points per trip. Therefore, considering that context, the analyst can use MobApp to rank anomalies trajectories by the anomaly level. To do that, the user can browse to the Anomaly Detection screen and select which route he wants (by the select input component) to filter the top-5 anomaly trajectories and analyze each one by the map on the right side. Figure 3a shows an example in which a user/analyst selects route 1, and the MobApp returns the most anomaly trajectory on the map along with the region where the anomaly occurs (red points).

Model Comparison. Comparison between anomaly detection models in ongoing work is expected in most quantitative research. In this direction, a researcher looking to compare trajectory anomaly detection models can use the MobApp application. For that, users can access the Model Evaluation screen and evaluate all models based on selected routes through the Precision-Recall curve. Also, on the same screen at the bottom, users can analyze a particular trajectory from the selected route using the models individually. For example, an analyst can follow two steps to compare its model. First, he browses to the Evaluation screen and selects route 35. MobApp application, in turn, returns the Precision-Recall curve summarizing the approaches in a sample of 100 trajectories, as shown in Figure 3c. Second, on the bottom of the same screen, the analyst selects trajectory 68 and the Transformer method to analyze results individually, Figure 4a and 4b show the results of this interaction.

4. Conclusions and Future Work

In this demo, we have worked with Dublin dataset and developed three screens: Exploratory Data Analysis, Anomaly Trajectory Filter, and Model Evaluation. As the application has only one dataset, we are developing an end-point to store other datasets (GPS trajectories) following a default pattern to be generic enough to feed the anomaly detection approaches. Moreover, we intend to link the selected trajectories across screens as an end-to-end analysis. For example, one can select a specific route on the EDA screen and use the trajectories on the other screens to get a complete analysis. Lastly, we are developing another end-point to receive other approach results (anomaly detection results) and evolve the user interface (UI).

Acknowledgment

The authors would like to thank the INES 2.0, APQ 0388-1.03/14 e CNPq 465614/2014-0 for his Ph.D. supporting and funding.

References

Belhadi, A., Djenouri, Y., Lin, J. C.-W., and Cano, A. (2020). Trajectory outlier detection: Algorithms, taxonomies, evaluation, and open challenges. *ACM Transactions on Management Information Systems (TMIS)*, 11(3):1–29.

Bessa, A., Silva, F. d. M., Nogueira, R. F., Bertini, E., and Freire, J. (2016). Riobusdata: Outlier detection in bus routes of rio de janeiro. *arXiv preprint arXiv:1601.06128*.

Bose, R. (2009). Advanced analytics: opportunities and challenges. *Industrial Management & Data Systems*.

Buja, A., Cook, D., Hofmann, H., Lawrence, M., Lee, E.-K., Swayne, D. F., and Wickham, H. (2009). Statistical inference for exploratory data analysis and model diagnostics. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906):4361–4383.

Chen, C., Zhang, D., Castro, P. S., Li, N., Sun, L., Li, S., and Wang, Z. (2013). iboat: Isolation-based online anomalous trajectory detection. *IEEE Transactions on Intelligent Transportation Systems*, 14(2):806–818.

Cruz, M. and Barbosa, L. (2020). Learning gps point representations to detect anomalous bus trajectories. *IEEE Access*, 8:229006–229017.

Liu, K., Gao, S., Qiu, P., Liu, X., Yan, B., and Lu, F. (2017). Road2vec: Measuring traffic interactions in urban road system from massive travel routes. *ISPRS International Journal of Geo-Information*, 6(11):321.

Liu, Y., Zhao, K., Cong, G., and Bao, Z. (2020). Online anomalous trajectory detection with deep generative sequence modeling. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 949–960. IEEE.

Pappalardo, L., Simini, F., Barlacchi, G., and Pellungrini, R. (2019). scikit-mobility: A python library for the analysis, generation and risk assessment of mobility data. *arXiv preprint arXiv:1907.07062*.