Applied Computer Science, vol. 16, no. 2, pp. 5–17
doi:10.23743/acs-2020-09

Submitted: 2020-06-03
Revised: 2020-06-12
Accepted: 2020-06-10

spatial-temporal data,
trajectory outlier detection, trajectory clustering

Alexis J. LOPEZ [0000-0001-7755-5109]*,
Perfecto M. QUINTERO [0000-0001-7651-4364]*,
Ana K. HERNANDEZ [0000-0002-8265-1303]*

ANALYTICS AND DATA SCIENCE APPLIED TO THE TRAJECTORY OUTLIER DETECTION

Abstract

Nowadays, logistics for transportation and distribution of merchandise are a key element to increase the competitiveness of companies. However, the election of alternative routes outside the panned routes causes the logistic companies to provide a poor-quality service, with units that endanger the appropriate deliver of merchandise and impacting negatively the way in which the supply chain works. This paper aims to develop a module that allows the processing, analysis and deployment of satellite information oriented to the pattern analysis, to find anomalies in the paths of the operators by implementing the algorithm TODS, to be able to help in the decision making. The experimental results show that the algorithm detects optimally the abnormal routes using historical data as a base.

1. INTRODUCTION AND MOTIVATION

In the las few years, land transportation has been the most used by most companies for merchandise distribution. Product distribution represents an important percentage in the total amount of costs in the companies supply chain, that is why the main focus for the companies should be the cost reduction and optimization. Recent studies (Sarmento, Renneboog & Matos, 2017) underline the importance of the logistic costs, including transportation cost, for example,

* Tecnológico Nacional de México, Instituto Tecnológico de Apizaco, 90300, Carretera Apizaco-Tzompantepec, Esquina Av., Instituto Tecnologico S/N, Apizaco, Tlaxcala, México, alexistegno@gmail.com

to Latin America, it represents between 18% and 32 % of the total monetary value of the product; a really high percentage in comparison to that of the countries that belong in The Organisation for Economic Co-operation and Development (OCDE). At the same time, it focuses the logistics cost of the set of countries of the OCDE, which is in a 9% average, even a smaller percentage than in the United States of America, which is 9.5 %.

In the current context, the transportation and logistics sector are a fundamental factor to increase the competitiveness of their productive sectors, promoting the supply and distribution of goods and services. This last area generates deficiencies in the productivity of the product delivery with delays in the supply chain management. Contributing negatively to more than half of the losses in the companies, this makes it to be a shrinkage and decline factor in Mexican companies. Hence, there is a need to identify and control all those cost factors to ensure the companies competitiveness. The operators in charge to deliver supplies in many occasions define the rout planning in an empirical way by delivering thousands of supplies according to the demand of each customer. The empirical assignation consists of the way in which supplies are delivered according to the operator's experience. In many times, the election of alternate routes outside the panned ones causes the logistics companies to provide poor quality services by using vehicles that risk the merchandise and the supply chain. Small companies are the most affected party by this situation. As a consequence, they bring down medium sized and large companies in the country.

Data science techniques for the processing and analysis of the Global Positioning System (GPS) can help obtain more precise information about the routing options for the vehicle fleets, as it is shown in figure 1, and as it was probed by (Marković, Sekuła, Vander Laan, Andrienko & Andrienko, 2019), who popularized the potential of the GPS data for the research in the behavior according to the routs specially by studying the variability of time. According to what was previously stated, a good way to act in an optimum manner when facing possible facts that can interfere with the rout optimization is the detection of the paths outside the pattern. Having precise information about the operator routes for a period of time will allow to learn, calculate, and to detect abnormalities in the route planning, thus facilitating the decision-making process according to the operational costs. Taking as a base what has been said, this work tackles the problematic of finding abnormal trajectories considering the historical information of the operators. To detect spatial-temporal data abnormalities it is necessary to find atypical values or patterns that indicate atypical behavior in the driving routes of the fleet operators, in this work a hybrid algorithm is used distance-based and cluster-based, then the grouping results are used to determine abnormal routes. The algorithm used is Trajectory Outlier Detection and Segmentation – TODS (Schmitt & Baldo, 2018).

Fig. 1. Data Science and Analytics for the Detection and Control of Anomalous Routes

The rest of this paper is organized as it follows: section 2, related works; section 3, methodology; section 4, obtained results; section 5, conclusion of this work.

2. RELATED WORKS

The abnormal trajectory detection algorithms are mainly classified in four classes, based on statistics, distance, density, and the methods based on grouping. For our study, we will briefly review the different approaches.

The methods based on statistics can be quite effective and efficient when there is enough data and previous knowledge, however, this kind of methods are not ideal when high dimension data is applied. It can also use existent models or create new ones to evaluate the adjustment degree of data according to the model. The most used data models include the Gaussian distribution (Shaikh & Kitagawa, 2014) the Gaussian multivariate distribution (Hazel, 2000), among others.

Distance based methods are more widely used in statistic methods, for they are easier to design and to determine significant distance measurements in a data set that determines the statistic distribution models. The advantage of this models is based simplicity on the implementation, however, the complexity in the worst-case scenario can come up to O(m^2), this could be too expensive to handle grate volume data. Besides, this method is also sensitive to parameters. In the (Munoz-Organero, Ruiz-Blaquez & Sánchez-Fernández, 2018) work, an algorithm based in distance called Center of Sliding Window (CSW) was proposed to detect driving abnormal locations caused by particular traffic conditions like traffic lights, intersections or roundabouts. The goal was to filter the atypical driving points related to random traffic conditions, such as a congested highway. Infrastructure

elements (Lee, Han & Li, 2008) proposed the TRAjectory Outlier Detection (TRAOD) algorithm in which the similarity measurement called angular distance is applied, in which the direction of the abnormal sub- trajectories is different than the adjacent sub-trajectories.

The methods based on density are related to the atypical detection methods based on distance. This is due to density being generally defined by distance, specially the Euclidian distance as well as the angular distance (Schmitt & Baldo, 2018). From the density point of view, the atypical values are low density regions. The atypical detection methods based on relative density can process multiple density data sets in an effective way. The temporal complexity of this kind of methods can reach O(m^2), which is comparable to the approaches based on distance. (Han, Kamber & Pei, 2012) introduces several detection methods of atypical values based on density according to the calculation of the density. (Liu, Pi & Jiang, 2013) proposed a detection algorithm of atypical values based on the density trajectory to compensate for the algorithm disadvantages TRAOD Lee et al. that cannot detect abnormal flaws when there is a local factorial and dense. In the work (Cao, Shi, Wang, Han & Bai, 2014) proposed to use a new concept of atypical value based on the uncertain data density to quickly detect the atypical values.

The cluster-based methods are used to discover data groups that are strongly connected, while the atypical value detection is used to discover objects that are not strongly related to other data. In the group result, those calculi that are away from other calculi, or that are not as relevant for others can be marked as abnormalities. That's why the group can be used to detect atypical values. There are a lot of grouping algorithms to detect atypical values in a data set such as K-means, DBSCAN, STING and BRICH (Yuan, Sun, Zhao, Li & Wang, 2017). Liao (Liao, 2005) pointed out that the atypical values are a sub product of the group. Dominguez et al. (Domínguez, Redondo, Vilas & Khalifa, 2017) applied a grouping algorithm based in density, obtaining the city's pulse, thus allowing the detection of anomalies (unexpected behaviors) in the New York users. Fontes et al. (Fontes, de Alencar, Renso & Bogorny, 2013) atypical values were found between interest regions, using the grouping algorithm that takes the space and time context to find anomalies in the trajectories. The atypical semantic trajectories are calculated between interest regions where the objects have a similar movement intention. As a first step, the algorithm extracts the candidates that are the sub trajectories that have a similar move intention. In a second step, normal and abnormal trajectories are found. There are several works that present hybrid approaches using based methods on clustering and distance as it is shown in Lei and Mingchao (Lei & Mingchao, 2018) It uses the similarity measurement of the angular distance and the relative distance grouping special groups based on density on application with noise in between de data, using the DBSCAN algorithm to generate patterns from original trajectories.

3. PROPOSED METHOD

To tackle this problem, a hybrid approach based in grouping and in distance was applied, using the similarity measurement called Euclidian distance and angular distance with the purpose of segmenting normal routes with a high number of drivers on that path and abnormal trajectories with a minor number of drivers in determined time intervals, with the implementation of TODS algorithm (Schmitt & Baldo, 2018) to provide a solution for the detection of anomalies in trajectories using historical data. As described in figure 2, the trajectory data of the devices is gathered and stored in the data base. After they are processed using a series of filters to finally group them and segment them in two sets abnormal and normal accordingly. The main algorithm is divided in 3 phases; 1 extraction and data processing; 2 grouping and 3 segmentation. It can be seen on algorithm 1 and their entrances are defined as follows:

- Start region (SR) and end region (ER): They represent the interest regions where the selected trajectories should intersect.
- Start hour (SH) and end hour (EH): They represent the time interval in which the trajectories SR and ER should intersect. This should respect the inequality SH < EH.
- Interpolant size (I), standard deviation (SD) and sigma (σ): these are parameters used for processing and cleaning trajectories.
- The similarity measurements applied for this work are the Euclidian distance (D) and the angular distance (θ): They are used to identify the couple of segments; abnormal and normal.
- The number of groups that make a referral to same pattern groups (KS): It is the minimum number of normal groups to find.

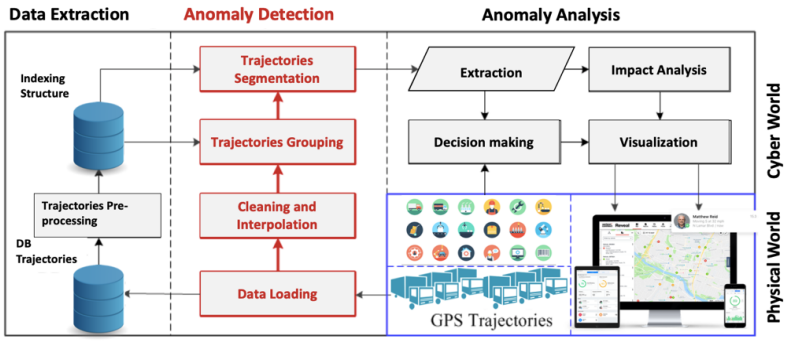

Fig. 2. Data extraction and analysis process for the detection of abnormal trajectories

The algorithm output are normal and abnormal arranged trajectories sets with their respective segments classified in sequences.

Algorithm 1 Main algorithm

Require: $SH, EH, SR, ER, I, SD, \sigma, D, \theta, KS$

```
1: t ← FindTrajectories(SR, ER)
2: for i ← 0 to Length(t) do
3:     FilterNoisePoints(t[i], SD,σ)
4:     InterpolatePoints(t[i], I)
5: end for
6: C ← GetCandidates(t, SH, EH, SR, ER)
7: idx ← CreateClusteringGrid(C, D)
8: GT ← GetTrajectoriesGroups(C, idx)
9: SGT ← GetStandardTrajectories(GT, KS)
10: R ← GetTrajectoriesRoutes(GT, SGT, D, θ)
11: return SGT,R
```

Fig. 3. Main Algorithm

3.1. Data extraction and preprocessing

There is data being obtained from the trajectories without processing from the data base (Find Trajectories line 1 on figure 3). The data base system applied was PostgreSQL, due to its capability to execute geometrical consultation to recuperate all the stored trajectories that intersect two different rectangular trajectories (SR y ER). Once the data has been extracted, the FilterNoisePoints (line 3 on figure 3) method follows with the preprocessing of the data, invalidating two invalid points in two steps. The first steps calculate a normal distribution using the distance in between two adjacent points. The points that extrapolate the confidence interval defined by σ are eliminated with the InterpolatePoints method (line 4 on figure 3). The parameter I defines the superior limit distance between two adjacent points. For that reason, if the distance between two adjacent points is bigger than the I value, an interpolation is executed to insert as many points as needed to adjust the distance if the point is shorter than I. The interpolation increases the points of density in the trajectories to improve the grouping set by the grouping algorithm. The second step calculates the standard deviation of the trajectory points to completely eliminate the trajectories that surpass the SD defined parameter. The next step is to eliminate the trajectories that fail to match the time interval given by SH y EH, filtering segments that respect the direction of the trajectory with the GetCandidates (line 6) method. For the recovery and optimization of the space time consultations, in this work, a system of index structure based in data grid was used to facilitate the consult's resolution by the grouping algorithm. In a way that it is efficient mapping, he points out the trajectories as it can be seen in the algorithm. The grouping algorithm used in this work is called Density-based spatial clustering of applications with noise (DBSCAN) (Gan & Tao, 2015).

3.2. Grouping

The grouping phase uses the similarity measurements based on distance; the Euclidian distance and the angular distance. Algorithm 2 details the function GetTrajectoriesGroups (line 8) from algorithm 1. The entries of the algorithm are the candidate's C set and the indexation system idx. Both variables are produced by the extraction phase and preprocessing, the output of the algorithm is a group team (GT), where each group contains trajectories with a similar route between SR y ER, as it can be seen on Figure 4.

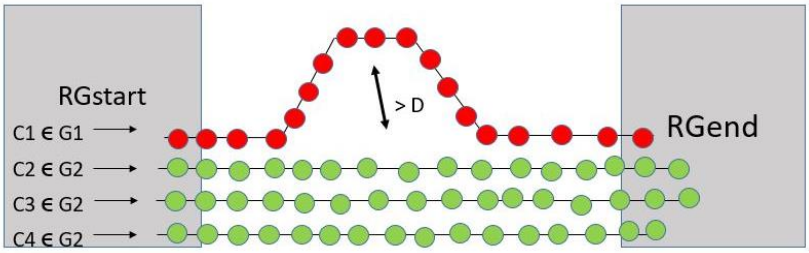

Fig. 4. Candidates grouping

Algorithm 2 starts to arrange the list of candidates in ascending order by the total of points (line 1). For each candidate, the algorithm starts to clean the variable currentGroup (line 4 on figure 5). After that, it makes a comparison between the current C candidate and the elements in the group GT (line 6). If the candidate matches an existing group, the variable currentGroup stores this respective group and finalizes the search (lines 7–8 on figure 5). The FitInGroup function uses the structure of the index to verify the distance of all the points that belong to c in the current trajectory grouping G. of the G group is not found (line 11 on figure 5), a new group is created to assign the candidate (line 12 on figure 5) and it is added to the GT group. Finally, the candidate c is added to the current group (line 15 on figure 5) and the next candidate is taken to comparison (line 3 on figure 5).

Algorithm 2 Trajectory grouping algorithm

Require: C, idx
1: $SortTrajectories(C)$
2: GT $\leftarrow CreateSetStructure()$
3: **for all** c $\in$ C **do**
4: $currentGroup \leftarrow empty$
5: **for all** c $\in$ C **do**
6: **if** $FitInGroup(c, G, idx)$ **then**
7: $currentGroup \leftarrow$ G
8: $break$
9: **end if**
10: **end for**
11: **if** $currentGroup = empty$ **then**
12: $currentGroup \leftarrow CreateGroup()$
13: $AddGroup(GT, currentGroup)$
14: **end if**
15: $AddCandidate(currentGroup, c)$
16: **end for**
17: **return** GT

Fig. 5. Trajectory grouping algorithm

3.3. Segmentation

Before proceeding with the segmentation, the standard group set is defined (STG) and the non-standard group is set (NSTG) in the GetStandardTrajectories method on the line 9 on figure 3, arranging the trajectory groups in descending order and selecting the superior KS groups with the most number of trajectories as the standard data set (STG) and the rest are marked as non-standard data sets (NSTG). The input parameters on figure 6 are: group set (GT), standard group set (STG), non-standard group set (NSTG), grouping distance (D), and the threshold of the angular distance (θ).

The segmentation process consists of two steps. The first step (found on figure 7) makes the segmentation by grouping the trajectories using the Euclidian distance. It identifies all the points in the trajectories as normal or abnormal using the grouping function HasNear. It classifies a specific point as normal due to its proximity to other points in the trajectories on figure 7. The second step makes the correction of segmentation, assessing the angular difference between trajectory segments, reclassifying the points from the beginning to the end of the abnormal segments calculating the angular difference between them and the normal segments using the BackExtension and FrontExtension, functions respectively. In figure 8, the segmentation can be seen by angular difference. At last, a composed route is created by normal and abnormal segments labeled as DS y NDS, respectively.

Algorithm 3 Trajectory segmentation algorithm

Require: GT, STG, NSTG, D, θ

$res \leftarrow CreateList()$
for all C $\in$ NSTG **do**
  for all p $\in$ C.points **do**
    $p.std \leftarrow HasNear(STG, p, D)$
  end for
  for all p $\in$ C.points **do**
    if p.std = FALSE **then**
      $BackExtension(p, C, STG, D, \theta)$
      $FrontExtension(p, C, STG, D, \theta)$
    end if
  end for
  $R \leftarrow CreateRoute()$
  $R \leftarrow CreateRoute()$
  $Split(R.DS, R.NDS, C)$
  $AddRoute(res, R)$
end for
return res

Fig. 6. Trajectory segmentation algorithm

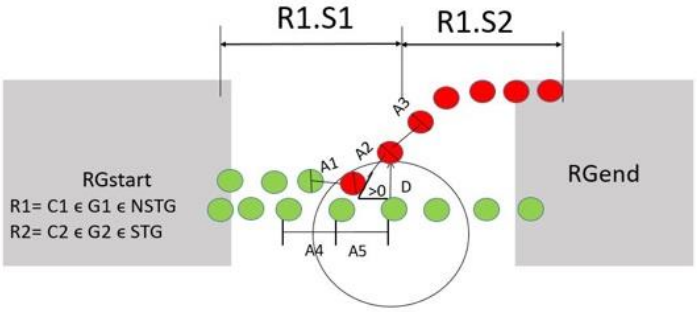

Fig. 7. Candidate segmentation by angular difference

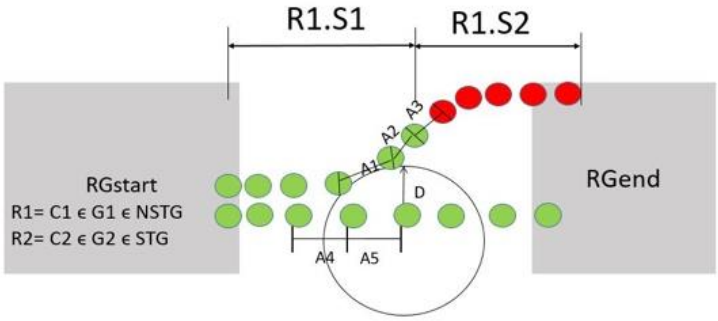

Fig. 8. Candidate segmentation by distance

4. EXPERIMENTAL RESULTS

For the development of the represented module on figure 9, Java was used (JDK version 1.8.0_121) as a programing language due to its execution and performance. For the management of consultations and information recovery, the data base manager applied was PostgreSQL, for it counts with an extension to manage geo spatial data and has as main characteristics the spatial kind of data, spatial indexes and functions that operate over them. According to the hardware specifications, a macBook Air was used with an Intel Core I5 processer and 8GB RAM to execute the module.

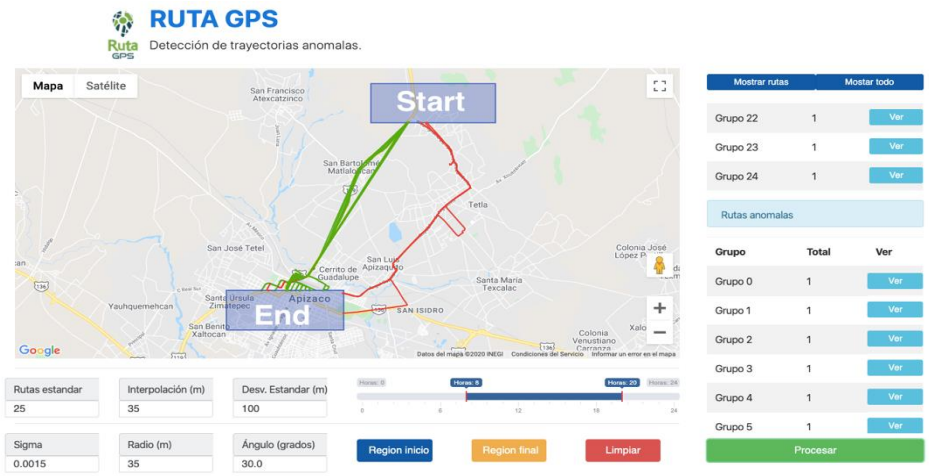

Fig. 9. Module interface

Tab. 1. RutasGPS format route

RutaGPS				
Device	**Latitude**	**Longitude**	**Time**	**Attribute**
15	19.5607	-98.130251	2019-04-05 11:48:01	{distance:14.43,totalDistance:41148}
15	19.5606	-98.130277	2019-04-05 11:50:53	{distance:14.43,totalDistance:41148}
15	19.5609	-98.130509	2019-04-05 11:51:15	{distance:14.44,totalDistance:42111}
15	19.5610	-98.130599	2019-04-05 11:58:04	{distance:14.44,totalDistance:42500}
15	19.5613	-98.130476	2019-04-05 12:00:04	{distance:14.44,totalDistance:45123}

For this work the data set of the project RutaGPS was applied, thus allowing space, time information of more than 50 users in a 6-month time period, which means from March 2019 through September 2019. It contains 17,621 trajectories and 6,798,282 points, including the users. On chart 1, the data format can be seen. On Figure 10, the visual representation of the spatial representation is shown.

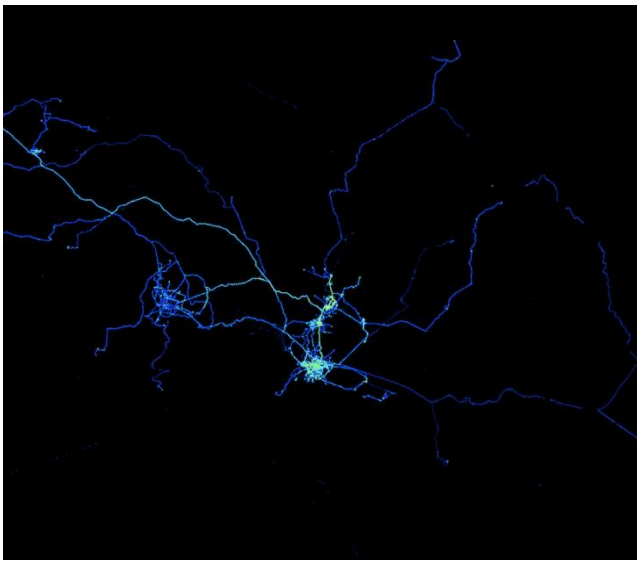

Fig. 10. Visual representation of spatial density

4.1. Case studies

For testing effects and to show an algorithm behavior, a module for a user who travels from Tlaxco, Tlaxcala to Apizaco, Tlaxcala was executed. The operator generally has a mobility pattern following a trajectory using the "carretera federal" Tlaxco-Apizaco for the logistics process can be seen in color green on Figure 11a. On the other side, an abnormal trajectory with the color red can be seen on Figure 11b, taking a different route, deviating from his trajectory in some segments that lead to routes outside the mobility pattern of the operator, arriving to Apizaco, Tlaxcala city using different segments from the "carretera federal Tlaxco-Apizaco".

Although the observed trajectory in red represents an abnormal event, it can be determined that the gas usage is larger due to the frequent stops in each intersection, every 200 meters than in segments where the average speed can reach 100 km/h. Besides, the time spent in road segments is a lot less than the time spent in segments where there are cross streets.

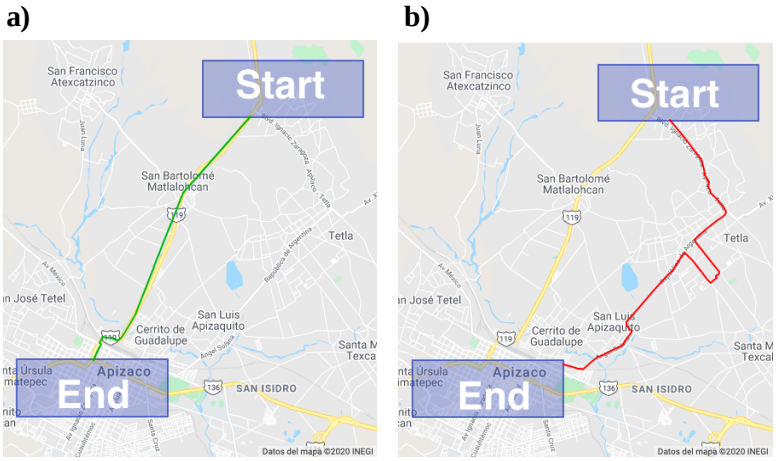

Fig. 11. Trajectory: a) normal; b) abnormal

5. CONCLUSIONS AND FUTURE WORKS

The detection of atypical values in the trajectories is greatly important to reduce and to optimize the operational costs, and to discover the particular events on transportation and distribution in the supply chain. In this paper, TODS algorithm was implemented to detect atypical values in the trajectory, to assign a segmentation and to group trajectories of the operator's technique with a normal and abnormal segments trough the analysis of the historical data. The experimental results showed the effectiveness of the algorithm in the detection of abnormal events according to the performance and viability that was analyzed in environments with real data resources for the detection of anomalies in merchandise logistics transportation and distribution processes. Regarding future researches, the detection in real time can be tested, as well as the auto-parametrization to reduce the configuration of empirical data.

REFERENCES

Cao, K., Shi, L., Wang, G., Han, D., & Bai, M. (2014). Density-Based Local Outlier Detection on Uncertain Data. In: F. Li, G. Li, S.W. Hwang, B. Yao & Z. Zhang, (Eds.), *Web-Age Information Management* (pp. 67–71). Springer International Publishing, Cham.

Domínguez, D.R., Redondo, R.P.D., Vilas, A.F., & Khalifa, M.B. (2017). Sensing the city with Instagram: Clustering geolocated data for outlier detection. *Expert Systems with Applications, 78*, 319–333.

Fontes, V.C., de Alencar, L.A., Renso, C., & Bogorny, V. (2013). Discovering Trajectory Outliers between Regions of Interest. In *Proceedings of XIV GEOINFO* (p. 12). Campos do Jordao, Brazil.

Gan, J., & Tao, Y. (2015). DBSCAN Revisited: Mis-Claim, Un-Fixability, and Approximation. In: Proceedings of the 2015 ACM SIGMOD *International Conference on Management of Data – SIGMOD* '15 (pp. 519–530). ACM Press, Melbourne, Victoria, Australia.

Han, J., Kamber, M., & Pei, J. (2012). *Data mining concepts and techniques.* Third edition. Elsevier.

Hazel, G.G. (2008). Multivariate Gaussian MRF for multispectral scene segmentation and anomaly detection. In *IEEE Transactions on Geoscience and Remote Sensing*, *38*(3), 1199–1211.

Lee, J.G., Han, J., & Li, X. (2008). Trajectory Outlier Detection: A Partition-and-Detect Framework. In: *2008 IEEE 24th International Conference on Data Engineering* (pp. 140–149). https://doi.org/10.1109/ICDE.2008.4497422

Lei, B., & Mingchao, D. (2018). A distance-based trajectory outlier detection method on maritime traffic data. In *2018 4th International Conference on Control, Automation and Robotics (ICCAR)* (pp. 340–343). https://doi.org/10.1109/ICCAR.2018.8384697

Liao, T.W. (2005). Clustering of time series data—a survey. *Pattern Recognition*, *38*(11), 1857–1874.

Liu, Z., Pi, D., & Jiang, J. (2013). Density-based trajectory outlier detection algorithm. *Journal of Systems Engineering and Electronics*, *24*(2), 335–340.

Markovic, N., Sekula, P., Vander Laan, Z., Andrienko, G., & Andrienko, N. (2019). Applications of Trajectory Data From the Perspective of a Road Transportation Agency: Literature Review and Maryland Case Study. *IEEE Transactions on Intelligent Transportation Systems*, *20*(5), 1858–1869. https://doi.org/10.1109/TITS.2018.2843298

Munoz-Organero, M., Ruiz-Blaquez, R., & Sánchez-Fernández, L. (2018). Automatic detection of traffic lights, street crossings and urban roundabouts combining outlier detection and deep learning classification techniques based on GPS traces while driving. *Computers, Environment and Urban Systems*, *68*, 1–8. https://doi.org/10.1016/j.compenvurbsys.2017.09.005

Sarmento, J., Renneboog, L., & Matos, P.V. (2017). Measuring highway efficiency by a DEA approach and the Malmquist index. *European Journal of Transport and Infrastructure Research*, *17*(4), 530–551.

Schmitt, J.P., & Baldo, F. (2018). A Method to Suggest Alternative Routes Based on Analysis of Automobiles' Trajectories. In: *2018 XLIV Latin American Computer Conference (CLEI)* (pp. 436–444). http://doi.org/10.1109/CLEI.2018.00059.

Shaikh, S.A., & Kitagawa, H. (2014). Efficient distance-based outlier detection on uncertain datasets of Gaussian distribution. *World Wide Web*, *17*(4), 511–538.

Yuan, G., Sun, P., Zhao, J., Li, D., & Wang, C. (2017). A review of moving object trajectory clustering algorithms. *Artificial Intelligence Review*, *47*(1), 123–144.