

# Detección de trayectorias anómalas en flotillas de vehículo

Alexis Jair López-Gutiérrez, Perfecto Malaquias Quintero-Flores,  
Ana Karen Hernandez-Vazquez

Tecnológico Nacional de México,  
Instituto Tecnológico de Apizaco,  
Departamento de Estudios de Posgrado e Investigación,  
México

{alexistegno, anakarenhv26, parfait.phd}@gmail.com

**Resumen.** En la actualidad, las prácticas logísticas para la transportación y distribución de mercancías constituyen un elemento clave para elevar la competitividad de empresas, regiones y países. No obstante, la elección de rutas alternas ajenas a la planificación de rutas provoca que las empresas logísticas terminen proporcionando servicios de mala calidad con unidades que ponen en riesgo la entrega oportuna de las mercancías contribuyendo negativamente en el funcionamiento de la cadena de suministro. El análisis de datos espacio-temporales pueden ayudar a obtener información más precisa sobre las opciones de ruta de las flotillas de vehículos. En este sentido, el presente estudio tiene como objetivo desarrollar e implementar un módulo que permita el procesamiento, análisis y despliegue de información satelital orientada al análisis de patrones para encontrar anomalías en las trayectorias de los operadores. La implementación del algoritmo denominado TODS - Detección y segmentación de valores atípicos de trayectoria, agrupa y segmenta trayectorias en rutas normales y rutas anómalas en diferentes periodos del día, para ser de ayuda en la toma de decisiones y minimizar costos de operaciones. Para evaluar los resultados se realizó pruebas utilizando tres conjuntos de datos de trayectorias recopilados por los operadores en Tlaxcala, México. La evaluación de resultados indica que TODS encontró entre un 30 % y un 40 % de rutas anómalas. Además, es posible observar el rendimiento del algoritmo, que en promedio se gasta más del 90 % en tiempo de ejecución en la recuperación de información.

**Palabras clave:** Datos espacio-temporales, segmentación de trayectoria, agrupación de trayectoria, algoritmo basado en densidad, algoritmo basado en distancia.

## Trajectory Outlier Detection on Vehicle Fleet

**Abstract.** Nowadays, logistics for transportation and distribution of merchandise are a key element to increase the competitiveness of companies, regions, and countries. However, the election of alternative routes

outside the planned routes causes the logistic companies to provide a poor-quality service, using units that endanger the appropriate delivery of merchandise and impacting negatively the way in which the supply chain works. The analysis of the spatial-temporal data can help get more precise information about the route options for vehicle fleets. With this in mind, this paper aims to develop and implement 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 – Trajectory Outlier Detection and Segmentation, to group and segment car road trajectories in normal routes and routes abnormal during different times of the day to be able to help in the decision making, focusing on minimizing operational costs. To evaluate the results, tests were carried out using three sets of trajectory data collected by the operators in Tlaxcala, Mexico. The evaluation of results indicates that TODS found between 30 % and 40 % of failed routes. In addition, it is possible to observe the performance of the algorithm, which on average spends more than 90 % at runtime in information retrieval.

**Keywords:** Spatial-temporal data, trajectory segment, trajectory clustering, density-based algorithm, distance-based algorithm.

## 1. Introducción

La logística de transporte y distribución de mercancías a nivel global se encuentra en una fase de inflexión respecto a lo que se venía haciendo en años anteriores, saliendo a flote una serie de problemas de distribución que se han vuelto comunes en el sector. Una mala planificación de rutas logísticas de entrega, no tener los vehículos en tiempo y forma, o tener retrasos en la entrega, son parte de los problemas de distribución a los que mayormente se ven enfrentadas las empresas mexicanas. De acuerdo con Kotler et al. [5], los principales elementos de los costos de la distribución física son el transporte (37 %), el control de existencias (22 %), el almacenamiento (21 %) y otros como la recepción de órdenes, el servicio al cliente, la distribución y la administración (20 %). Por tanto, el transporte de distribución logística es la actividad económica más importante entre los componentes de las etapas de la logística empresarial.

Al entender la distribución física de productos como la actividad de trasladar los bienes desde el centro de producción hasta el consumidor final, es evidente pensar que el elemento clave de una cadena de suministro es el sistema de transporte logístico, sin embargo existen malas prácticas en las empresas causando deficiencias en la productividad para la entrega de productos con atrasos de tiempo en la gestión de la cadena de suministros, contribuyendo negativamente con más de la mitad de las pérdidas en las empresas siendo factor de merma y declive de las empresas Mexicanas de acuerdo con el estudio [1]. Por tanto, existe la necesidad de identificar y controlar todos aquellos factores del costo para garantizar la competitividad empresarial.

Los operadores encargados de repartir insumos en muchas ocasiones definen la planeación de rutas de manera empírica. La asignación empírica consiste en la repartición de los insumos de acuerdo con la experiencia del operador. En muchas ocasiones, la elección de rutas alternas ajenas a la planificación de rutas provoca que las empresas logísticas terminen proporcionando servicios de mala calidad con unidades que ponen en riesgo la entrega oportuna de las mercancías y el funcionamiento de la cadena de suministro, padeciendo esta situación principalmente las pequeñas empresas arrastrando consigo a las medianas y a las grandes empresas del país.

Con fundamento en lo anterior, la mejor manera para reaccionar ante posibles hechos anómalos que afecten a la optimización de rutas es la detección de las trayectorias fuera del patrón. Tener información precisa sobre las rutas del operador durante un período de tiempo permitirá aprender, estimar y detectar anomalías en la planeación de rutas, facilitando la toma de decisiones en los costos de operaciones. En éste trabajo se implementa un algoritmo de agrupación y segmentación de trayectorias de flotillas de los vehículos aplicando el enfoque basado en agrupación y un enfoque basado en distancia. Después, los resultados de la agrupación se utilizan para determinar rutas anómalas.

El resto de éste documento está organizado de la siguiente manera. En la sección 2 presenta los antecedentes referentes a esta investigación. La implementación del algoritmo para la detección de trayectorias anómalas se describe en la sección 3. La sección 4 presenta los resultados obtenidos y por último en la sección 5 concluye éste trabajo.

## **2. Antecedentes**

Existe varios trabajos que abordan la detección de trayectorias anómalas desde un enfoque basado en distancia como se demuestra en el trabajo [9] donde proponen un algoritmo llamado Center of Sliding Window (CSW) para detectar ubicaciones de conducción anormales causadas por condiciones de tráfico particulares como semáforos, cruces de calles o rotondas. Lee et al. [6] propusieron el algoritmo TRAjectory Outlier Detection (TRAOD) utilizando una medida de similaridad denominada distancia angular, en el cual la dirección de las sub-trayectorias anómalas difiere de las de las sub-trayectorias vecinas. Por otro lado, algunos trabajos se han centrado en la detección de trayectorias atípicas basados en densidad, estudios como [2] proponen un nuevo concepto de valor atípico local basado en densidad con datos inciertos para detectar rápidamente valores atípicos.

Los métodos de detección atípicos basados en la densidad relativa pueden procesar conjuntos de datos de densidad múltiple de manera efectiva. La complejidad temporal de éste tipo de métodos puede llegar a  $O(m^2)$ , donde  $m$  es el número de puntos en la base de datos, lo cual es comparable a los enfoques basados en la distancia. En la investigación [8] propusieron un algoritmo de detección de valores atípicos basados en la trayectoria de la densidad para compensar las desventajas del algoritmo TRAOD que no puede detectar anomalías

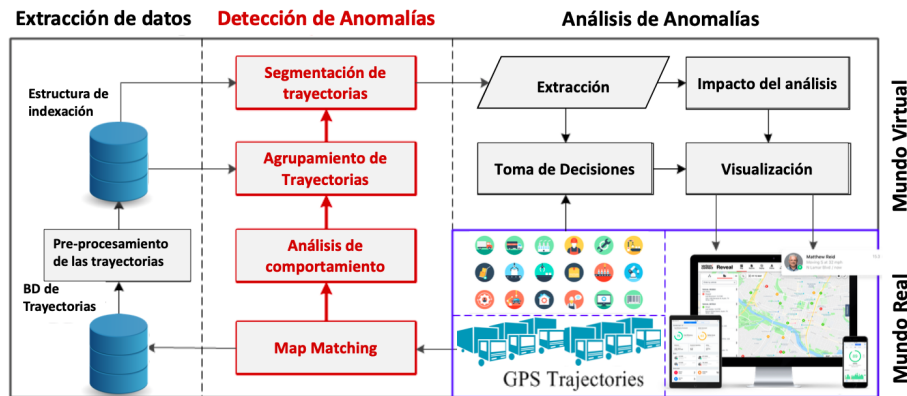
cuando existe densidad entre los puntos dentro de una única región de interés. Sin embargo, Cao et al. [2] propuso un nuevo concepto de valor atípico local basado en la densidad en datos inciertos para detectar rápidamente valores atípicos. En contraste con el enfoque basado en densidad y el enfoque basado en distancia, existen varios algoritmos de agrupación para encontrar valores atípicos en el conjunto de datos, como K-means, DBSCAN, STING y BRICH Yuan et al. [11]. Liao [7] señaló que los valores atípicos son el subproducto de la agrupación. Domínguez et al. [3] aplicaron un algoritmo de agrupación basado en la densidad, obteniendo el pulso de la ciudad permitiendo la detección de anomalías (comportamientos inesperados) entre los usuarios en Nueva York.

### 3. Desarrollo de la propuesta de solución

Para abordar éste problema, se utilizo un enfoque híbrido basado en agrupación y basado en distancia utilizando la medida de similaridad denominada distancia euclidiana y distancia angular con el fin de segmentar rutas normales con un alto número de conducción por esa vía y trayectorias anómalas con menor número de conducción en ciertos intervalos de tiempo, con la implementación del algoritmo TODS [10] para dar solución a la detección de anomalías en trayectorias utilizando datos históricos. La selección del algoritmo se debió a una exhaustiva revisión de trabajos referentes al área, además porque satisface la solución para encontrar trayectorias que salgan del patrón entre los operadores logísticos.

Como se describe en la Figura 1, los datos de las trayectorias de los dispositivos son recopilados y almacenados en una base de datos. Después se procesan aplicando una serie de filtros para finalmente agruparlos y segmentarlos en dos conjuntos, anómalos y normales respectivamente. El algoritmo principal esta dividido en 3 fases; Extracción y Preprocesamiento de Datos, Agrupación y Segmentación. Se observa en el algoritmo 1 y sus entradas se definen de la siguiente manera:

- Región de inicio ( $SR$ ) y región de finalización ( $ER$ ): representan las regiones rectangulares donde se deben cruzar las trayectorias seleccionadas;
- Hora de inicio ( $SH$ ) y hora de finalización ( $EH$ ): representan el intervalo de tiempo. en el cual las trayectorias deben cruzar  $SR$  y  $ER$ , estos deben respetar la desigualdad  $SH < EH$ ;
- Tamaño de interpolación ( $I$ ), desviación estándar ( $SD$ ) y sigma ( $\sigma$ ) : son parámetros que se utilizan para preprocesamiento y limpieza de las trayectorias;
- Las medidas de similaridad utilizadas para éste trabajo son la distancia euclidiana ( $D$ ) y la distancia angular ( $\theta$ ): se utilizan para identificar el par de segmentos; anómalos y normales.
- El número de grupos que hacen referencias a grupos con un mismo patrón ( $KS$ ): Es el numero mínimo de agrupaciones normales a encontrar.



**Fig. 1.** Proceso de Extracción y análisis de datos para la detección de trayectorias anómalas.

La salida del algoritmo son dos conjuntos ordenados de trayectorias normales y anómalas, además con sus respectivos segmentos clasificados en secuencias.

---

**Algoritmo 1** Algoritmo principal.

---

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

- 1:  $t \leftarrow FindTrajectories(SR, ER)$
  - 2: **Para**  $i \leftarrow 0$  **to**  $Length(t)$  **Hacer**
  - 3:      $FilterNoisePoints(t[i], SD, \sigma)$
  - 4:      $InterpolatePoints(t[i], I)$
  - 5: **Fin Para**
  - 6:  $C \leftarrow GetCandidates(t, SH, EH, SR, ER)$
  - 7:  $idx \leftarrow CreateClusteringGrid(C, D)$
  - 8:  $GT \leftarrow GetTrajectoriesGroups(C, idx)$
  - 9:  $SGT \leftarrow GetStandardTrajectories(GT, KS)$
  - 10:  $R \leftarrow GetTrajectoriesRoutes(GT, SGT, D, \theta)$
  - 11: **return**  $SGT, R$
- 

**3.1. Fase 1 - Extracción y preprocesamiento de trayectorias**

Se comienza a obtener los datos de las trayectorias sin procesar desde la base de datos con el método *FindTrajectories* (línea 1). El sistema de base de datos utilizado fue PostgreSQL, debido a su capacidad para ejecutar consultas geométricas para recuperar todas las trayectorias almacenadas que se cruzan con dos geometrías rectangulares distintas (SR y ER). Una vez recuperado los datos, se procede al procesamiento de los datos en el método *FilterNoisePoints* (línea 3) eliminando puntos inválidos en dos pasos. El primer paso calcula una distribución normal usando la distancia entre todos los puntos adyacentes. Se eliminan los puntos que extrapolan el intervalo de confianza definido por  $\sigma$  con

el método InterpolatePoints (línea 4). El parámetro  $I$  define la distancia límite superior entre dos puntos adyacentes.

Por lo tanto, si la distancia entre dos puntos adyacentes es mayor que el valor de  $I$ , se ejecuta una interpolación para insertar tantos puntos como sea necesario para ajustar la distancia del punto más corto que  $I$ . La interpolación aumenta la densidad de puntos de las trayectorias para mejorar el agrupamiento realizada por el algoritmo de agrupación. El segundo paso calcula la desviación estándar de los puntos de trayectoria para eliminar por completo las trayectorias que superan el parámetro  $SD$  definido. El siguiente paso es eliminar las trayectorias que no cumplen con el intervalo de tiempo dado por  $SH$  y  $EH$ , filtrando segmentos que respetan la dirección de la trayectoria con el método GetCandidates (línea 6). Para la recuperación y optimización de consultas espacio - temporales, en éste trabajo se utilizó un sistema de estructura de índice basada en cuadrícula para facilitar la resolución de consultas por el algoritmo de agrupación en una forma eficiente mapeando los puntos de las trayectorias como se puede observar en el algoritmo 2. El algoritmo de agrupación utilizado en éste trabajo se denomina Density-based spatial clustering of applications with noise (DBSCAN).

### 3.2. Fase 2 - Agrupación

La fase de agrupación usa las medidas de similitud basada en distancia; la distancia euclidiana y la distancia angular. El Algoritmo 2 detalla la función GetTrajectoriesGroups (línea 8) del Algoritmo 1. Las entradas del algoritmo son: el conjunto de candidatos  $C$  y el sistema de indexación  $idx$ . Ambas variables son producidas por la fase de Extracción y Preprocesamiento de Trayectorias. La salida del algoritmo es un conjunto de grupos ( $GT$ ), donde cada grupo contiene trayectorias con una ruta similar entre  $SR$  y  $ER$ , como se puede ilustrar en la figura 2.

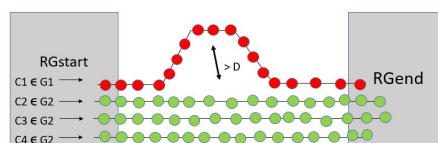


Fig. 2. Agrupación de candidatos.

El algoritmo 2 comienza a ordenar la lista de candidatos en orden ascendente por el total de puntos (línea 1). Para cada candidato, el algoritmo comienza a limpiar la variable currentGroup (línea 4). Después de eso, realiza una comparación con el candidato actual  $c$  con los elementos del grupo en  $GT$  (línea 6). Si el candidato coincide en un grupo existente, la variable currentGroup almacena ese grupo respectivo y da por finalizado la búsqueda (líneas 7-8). La función

FitInGroup utiliza la estructura de índice para verificar la distancia de todos los puntos pertenecientes a  $c$  en la agrupación de trayectorias actuales  $G$ .

Si no se ha encontrado el grupo  $G$  (línea 11), se crea un nuevo grupo para asignar el candidato (línea 12) y éste se agrega al grupo  $GT$ . Finalmente, el candidato  $c$  se asigna al grupo actual (línea 15) y el siguiente candidato repite el proceso (línea 3).

---

**Algoritmo 2** Algoritmo de agrupación de trayectorias.

---

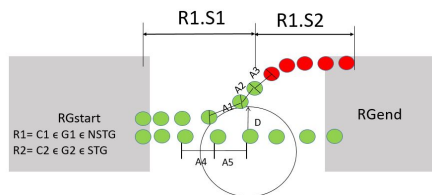
**Entrada:**  $C, idx$   
1:  $SortTrajectories(C)$   
2:  $GT \leftarrow CreateSetStructure()$   
3: **Para todo**  $c \in C$  **Hacer**  
4:      $currentGroup \leftarrow empty$   
5:     **Para todo**  $c \in C$  **Hacer**  
6:         **Si**  $FitInGroup(c, G, idx)$  **entonces**  
7:              $currentGroup \leftarrow G$   
8:              $break$   
9:         **Fin Si**  
10:     **Fin Para**  
11:     **Si**  $currentGroup = empty$  **entonces**  
12:          $currentGroup \leftarrow CreateGroup()$   
13:          $AddGroup(GT, currentGroup)$   
14:     **Fin Si**  
15:      $AddCandidate(currentGroup, c)$   
16: **Fin Para**  
17: **return**  $GT$

---

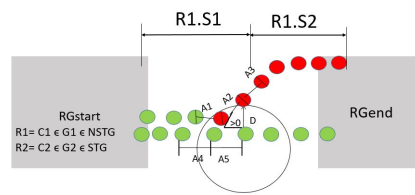
### 3.3. Fase 3 - Segmentación

Los parámetros de entrada del algoritmo 3 son: conjunto de grupos ( $GT$ ), conjunto de grupos normales ( $STG$ ), conjunto de grupos anómalos ( $NSTG$ ), distancia de agrupamiento ( $D$ ), y el umbral de la diferencia angular ( $\theta$ ). El proceso de segmentación consta de dos pasos, en el primer paso en la figura 3 realiza la segmentación agrupando las trayectorias usando la distancia euclidiana. Identifica todos los puntos de trayectoria como normales o anómalas utilizando la función de agrupación HasNear, clasificando un punto dado como normal debido a su proximidad a puntos de trayectorias cercanos.

En el segundo paso realiza la corrección de segmentación evaluando la diferencia angular entre segmentos de trayectorias, reclasificando los puntos del principio al final de los segmentos anómalos calculando la diferencia angular entre ellos y los segmentos normales utilizando las funciones BackExtension y FrontExtension, respectivamente. En la figura 4 se observa la segmentación por diferencia angular. Finalmente se crea una ruta compuesta por segmentos normales y anómalos etiquetados por  $DS$  y  $NDS$ , respectivamente. Finalmente, el algoritmo regresa los segmentos de rutas anómalos separados en conjuntos



**Fig. 3.** Segmentación de candidatos por distancia.



**Fig. 4.** Segmentación de candidatos por diferencia angular.

normales y anómalos, para la previa visualización utilizando el Api de Google MAPS.

---

**Algoritmo 3** Algoritmo de segmentación de trayectorias.

---

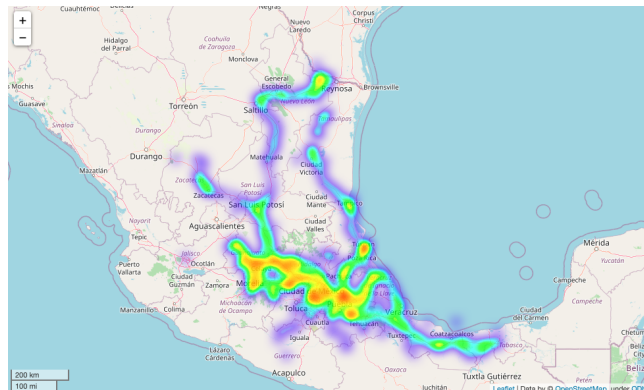
**Entrada:** GT, STG, NSTG, D,  $\theta$

- 1:  $res \leftarrow CreateList()$
  - 2: **Para todo**  $C \in NSTG$  **Hacer**
  - 3:   **Para todo**  $p \in C.points$  **Hacer**
  - 4:      $p.std \leftarrow HasNear(STG, p, D)$
  - 5:   **Fin Para**
  - 6:   **Para todo**  $p \in C.points$  **Hacer**
  - 7:     **Si**  $p.std = FALSE$  **entonces**
  - 8:        $BackExtension(p, C, STG, D, \theta)$
  - 9:        $FrontExtension(p, C, STG, D, \theta)$
  - 10:    **Fin Si**
  - 11:   **Fin Para**
  - 12:    $R \leftarrow CreateRoute()$
  - 13:    $R \leftarrow CreateRoute()$
  - 14:    $Split(R.DS, R.NDS, C)$
  - 15:    $AddRoute(res, R)$
  - 16: **Fin Para**
  - 17: **return**  $res$
- 

## 4. Resultados

Para el desarrollo del módulo, se utilizó Java (JDK versión 1.8.0\_121) como lenguaje de programación, por sus bondades a la hora de la ejecución y rendimiento. Para el manejo de consultas y recuperación de la información, se utilizó el gestor de base de datos PostgreSQL, porque cuenta con una extensión para manejar datos geoespaciales, y tiene como principales características, los tipos de datos espaciales, índices espaciales y funciones que operan sobre ellos. Con respecto a la especificación de hardware, se utilizó una MacBook Air con procesador Intel Core I5 y 8GB de RAM para ejecutar el módulo.





**Fig. 5.** Representación visual de la densidad espacial.

Para éste trabajo se utilizo el conjunto de datos del proyecto RutaGPS, teniendo información espacio - temporal de más de 50 usuarios en un período de 6 meses, es decir, marzo del 2019 hasta septiembre del 2019. Contiene 17,621 trayectorias y 6,798,282 puntos, incluidos todos los usuarios, en la tabla 1 se observa un extracto de los datos.

En la figura 5 se muestra la representación visual de la densidad espacial. Todo el análisis se realizó utilizando tres conjuntos de datos heterogéneos.

El primer conjunto de datos contiene aproximadamente 5024 trayectorias recopiladas durante 2019 por 24 semanas, por un operador que hace su repartición de productos entre la ciudad de Tlaxco, Tlaxcala a Apizaco Tlaxcala. El segundo conjunto tiene aproximadamente 14074 trayectorias recopiladas durante 2019 por 50 semanas, por un operador que se desplaza desde la zona conurbada de San Pablo Apetatitlán, Santa Ana Chiahutempan y Santa Cruz Tetela, en el estado de Tlaxcala. Y el tercer conjunto cuenta con 5880 trayectorias durante 2019 por 22 semanas, por un operador logístico que se desplaza de la ciudad de Apizaco, Tlaxcala a la ciudad de Santa Ana Chiahutempan. La Tabla 2 presenta el resumen de los 3 conjuntos de datos.

Los valores para los parámetros definidos para el primer conjunto son;  $SH = 8h$ ,  $EH = 20h$ ,  $I = 35m$ ,  $SD = 100m$ ,  $\sigma = 99,7\%$ ,  $D = 50m$ ,  $\theta = 30$  y  $KS = 20$ . Para el segundo conjunto los valores seleccionados son:  $SH = 10h$ ,  $EH = 16h$ ,  $I = 30m$ ,  $SD = 120m$ ,  $\sigma = 99,7\%$ ,  $D = 50m$ ,  $\theta = 30$  y  $KS = 3$ . Y para el tercer conjunto los parámetros seleccionados son:  $SH = 9h$ ,  $EH = 20h$ ,  $I = 35m$ ,  $SD = 100m$ ,  $\sigma = 99,7\%$ ,  $D = 50m$ ,  $\theta = 30$  y  $KS = 2$ . Algunos parámetros como  $SD$ ,  $\sigma$  y  $I$  pueden definirse automáticamente en futuros estudios. Estos valores de parámetros se definieron empíricamente en base a la densidad de puntos que cuenta por naturaleza cada conjunto de datos seleccionados, así como por las sugerencias relacionadas por la literatura.

Para el primer escenario, el operador tiene un patrón de movilidad mediante la carretera federal Tlaxco-Apizaco, observada en color verde en la figura 6. Por otra parte, se observa una anomalía con una trayectoria atípica pintada en

**Tabla 1.** Formato de datos del conjunto RutaGPS.

ID	Deviceid	Fixtime	Latitude	Longitude	Speed (kn)
1	1	2019-01-31 15:13:13	9.5607245	-98.130251	0.12
2	1	2019-01-31 15:13:13	19.560675	-98.130277	0.13
3	1	2019-01-31 15:13:14	19.560788	-98.130509	0.02
.	.	.	.	.	.
5601	2	2019-02-02 08:04:04	19.47644	-98.12148	2.94
5602	2	2019-02-02 08:05:25	19.47924	-98.12568	7.86
.	.	.	.	.	.
11304	3	2019-02-04 14:04:04	19.452497	-98.1027815	6.76
11305	3	2019-02-04 14:06:06	19.45248	-98.1027885	6.80

color rojo en la figura 6 tomando una ruta distinta a la frecuente desviando su trayectoria por segmentos que conducen a rutas fuera del patrón de movilidad del operador, llegando a la ciudad de Apizaco por segmentos distintos a la carretera federal Tlaxco-Apizaco.

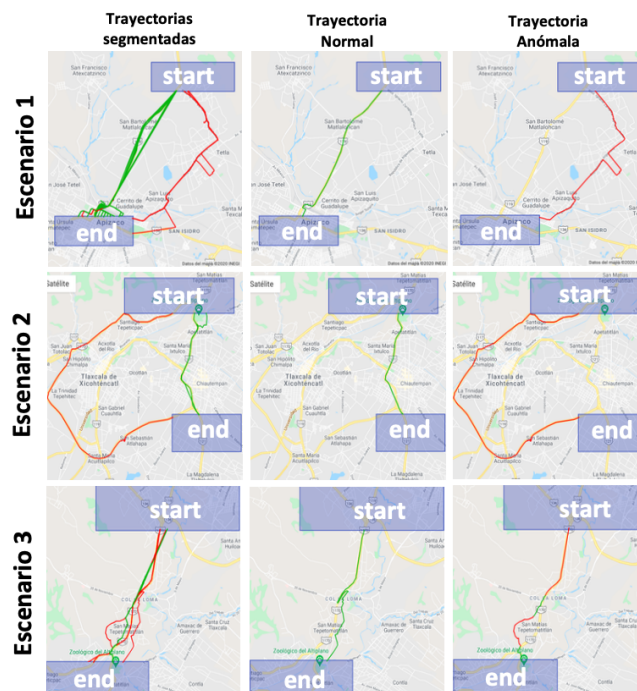
Para el segundo escenario, el operador tiene una trayectoria cotidiana mediante la carretera federal Vía corta Sta. Ana-Puebla/México, observada en color verde en la figura 6 tomando una ruta distinta a la frecuente, desviando su trayectoria por segmentos que conducen a rutas fuera del patrón de movilidad del operador, llegando a la población de Magdalena Tlaxtepec por segmentos distintos a la carretera federal Vía corta. La ruta anómala se desvía antes de tomar la carretera federal Apizaco - Chiahutempan, tomando una trayectoria por la carretera federal Apizaco - Tlaxcala, con destino al entronque hacia el libramiento de Tlaxcala. Esta anomalía se puede explicar, ya que las trayectorias tomadas durante éste proceso son trayectorias que aluden al periodo de mantenimiento y trabajos de ampliación en la carretera Apizaco - Chiahutempan. Razón probable que en éste periodo existía un embotellamiento de autos durante la hora pico. Por ultimo, para el escenario tres, el operador logístico tiene una ruta desde la ciudad de Apizaco, Tlaxcala a la ciudad de Santa Ana Chiahutempan, Tlaxcala, tal y como se observa en la Figura 6. Para el escenario 3 se observan distintos patrones en el mapa de carretera, estos patrones de igual manera se vieron afectados por los trabajos de mantenimiento y de ampliación en la carretera federal Apizaco-Chiahutempan. Existe una trayectoria atípica resaltada en color rojo observada en la Figura 6, tomando una ruta distinta a la frecuente desviando su trayectoria por segmentos que rodean a la carretera federal. Cabe mencionar que la ruta seleccionada como anómala, es una ruta común para el transporte publico durante los trabajos de mantenimiento.

El consumo de combustible es mayor en las constantes paradas entre cruces que en segmentos donde la velocidad promedio puede alcanzar hasta los 100 km/h. Si bien la trayectoria observada en color rojo representa un evento anómalo, se puede concluir que el consumo de combustible es mayor en las constantes paradas entre cruces de camino cada 200 metros que en segmentos donde la velocidad promedio puede alcanzar hasta los 100 km/h. Además, el tiempo que se demora es significativamente menor en segmentos de carretera

**Tabla 2.** Resumen estadístico de conjuntos de datos sobre trayectorias y puntos recopilados (coordenadas).

Estadísticas	Escenario 1	Escenario 2	Escenario 3
Cantidad de trayectorias	5024	14074	5880
Cantidad de puntos de trayectoria (Media)	252.9	45.1	270.3
Cantidad de puntos de trayectoria (SD)	494.9	32.9	501.8
Cantidad de puntos de trayectoria (Max)	4810	1127	5120
Cantidad de puntos de trayectoria (Min)	1	1	1
Cantidad de puntos	20354	90450	15223
Distancia de puntos adyacentes (Media)	6.4(m)	48.9(m)	6.8(m)
Distancia de puntos adyacentes (SD)	7.9(m)	908.1(m)	8.1(m)
Distancia de puntos adyacentes (Max)	234.9(m)	669323.1(m)	238.9(m)
Distancia de puntos adyacentes (Min)	0.0(m)	0.0(m)	0.0(m)

que en segmentos donde existen relativamente cruces por cada intersección de calles.



**Fig. 6.** Resultados de cada escenario.

Analizando los resultados de las trayectorias en la tabla 3 es posible observar que el algoritmo TODS encontró más rutas normales que anómalas.

**Tabla 3.** Estadísticas en la detección de Trayectorias anómalas por escenario.

VARIABLES	Conjunto 1	Conjunto 2	Conjunto 3
Tiempo (ms) Consulta BD	2697.41	402.20	1214.29
Tiempo (ms) Obtención de candidatos	492.01	25.40	31.14
Tiempo (ms) Agrupación de candidatos	9.84	0.06	0.24
Tiempo (ms) Obtención de trayectorias normales	0.03	0	0.02
Tiempo (ms) Segmentación de trayectorias	21.13	0.23	0.61
Trayectorias analizadas	37	5	6
Grupos trayectorias normales	25	3	4
Grupos trayectorias anómalas	12	2	2
Trayectorias analizadas %	100 %	100 %	10 %
Trayectorias normales %	67 %	60 %	66.6 %
Trayectorias anómalas %	33 %	40 %	33.3 %

Se explica debido al hecho de que el algoritmo agrupa las trayectorias similares en dos fases; 1 en la primera fase agrupa las trayectorias usando la distancia euclidiana. En la segunda fase agrupa las trayectorias con la diferencia angular. Además, se observa el tiempo por cada fase medido en milisegundos, esto por la cantidad mínima de trayectorias analizadas. La selección de las dos regiones denominadas "startz .end", delimita el espacio de búsqueda en el conjunto de datos, al evaluar un área de interés. El resultado de analizar áreas de interés, las observamos en la Tabla 3, con 37 trayectorias analizadas, 25 grupos de trayectorias normales y 12 grupos de trayectorias anómalas para el escenario 1. En el escenario 2 se analizaron 5 trayectorias en el espacio de búsqueda, se encontraron 3 grupos de trayectorias normales y 2 grupos de trayectorias anómalas. Y para el escenario 3, se analizaron 6 trayectorias, con 4 trayectorias normales y 2 anómalas.

#### 4.1. Evaluación

Para evaluar el algoritmo TODS, se realizó un análisis cualitativo y cuantitativo. El análisis cualitativo evalúa el proceso del algoritmo para identificar si es adecuado para detectar anomalías.

Además, el análisis cuantitativo evalúa la eficiencia en tiempo de ejecución del algoritmo con respecto a uno de los algoritmos referentes para la detección de anomalías en trayectorias de acuerdo en la literatura denominado TRA-SOD [4].

Para los valores definidos para el algoritmo TRA-SOD son los sugeridos en [4], que son:  $maxDist = 50$  y  $minSup = 4$ . Los algoritmos TODS y TRA-SOD se compararon bajo los mismos escenarios. Los resultados obtenidos se pueden ver en la Figura 7, observando la clasificación completa de las trayectorias con respecto a las rutas normales y las trayectorias pertenecientes a rutas anómalas. Analizando la clasificación de trayectorias es posible observar que TRA-SOD encontró las mismas rutas anómalas como lo hizo el algoritmo TODS para el escenario uno y tres.



Fig. 7. Comparativa entre los algoritmos TODS y TRA-SOD.

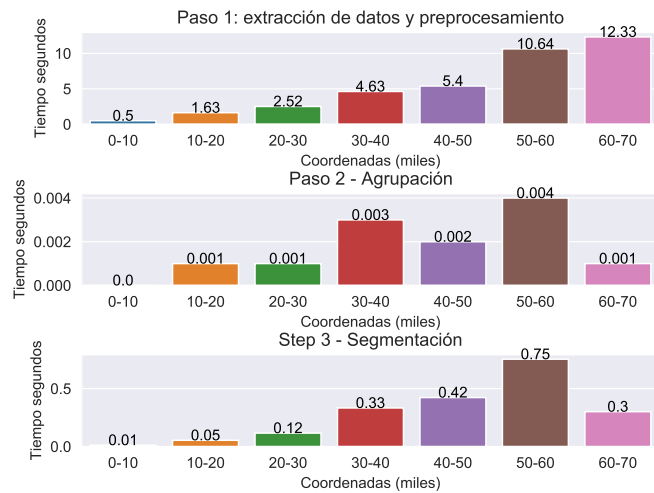


Fig. 8. Tiempo de ejecución por fases.

Sin embargo, para el escenario dos el algoritmo TRA-SOD detecta todas las trayectorias como anómalas. Esto sucede debido a la dificultad para definir el valor apropiado para el parámetro de vecindad  $k$  más cercano.

Para evaluar el rendimiento del tiempo de ejecución del algoritmo TODS, se seleccionaron 7000 muestras de pares de regiones generadas aleatoriamente que tienen, al menos, una ruta anómala. Al analizar el rendimiento en tiempo de ejecución del algoritmo TODS considerando cada una de sus fases, en la Figura 8 es posible ver que en promedio se gasta más del 90% del tiempo de ejecución

en el paso 1 (Recuperación). Esto se explica porque el proceso de generar los grupos y crear la estructura de indexación lleva mucho tiempo. Por el contrario, las fases restantes han reducido el tiempo de ejecución debido a los beneficios aportados por la estructura de índice que aceleran la búsqueda de trayectorias vecinas.

## 5. Conclusión

La detección de valores atípicos de trayectoria es de gran importancia para reducir y optimizar los costos de operaciones y descubrir eventos particulares en la transportación y distribución en la cadena de suministros. En éste estudio, implementamos el algoritmo TODS para detectar valores atípicos de trayectoria, utilizando una técnica de agrupamiento para agrupar y segmentar trayectorias de los operadores como segmentos normales y anómalos mediante el análisis de los datos históricos. Los resultados experimentales mostraron la efectividad del algoritmo en la detección eventos anómalos. Además, se realizó una comparación con el algoritmo TRA-SOD, mostrando que el algoritmo TODS es mas efectivo en la detección de rutas anómalas. En cuanto a investigaciones futuras, se puede probar la detección en tiempo real. Además, la auto-parametrización para reducir la configuración de los parámetros empíricos.

## Referencias

1. 6 problemas de distribución logística de productos [última milla], <https://www.beetrack.com/es/blog/logistica-de-distribucion/>, library Catalog: [www.beetrack.com](http://www.beetrack.com)
2. Cao, K., Shi, L., Wang, G., Han, D., Bai, M.: Density-Based Local Outlier Detection on Uncertain Data. In: Li, F., Li, G., Hwang, S.w., Yao, B., Zhang, Z. (eds.) *Web-Age Information Management*. pp. 67–71. Springer International Publishing, Cham (2014)
3. Domínguez, D.R., Redondo, R.P.D., Vilas, A.F., Khalifa, M.B.: Sensing the city with Instagram: Clustering geolocated data for outlier detection. *Expert Systems with Applications* 78, 319–333 (2017)
4. Fontes, V.C., de Alencar, L.A., Renso, C., Bogorny, V.: Discovering trajectory outliers between regions of interest. In: *Proceedings of XIV GEOINFO*, November 24-27, 2013, Campos do Jordão, Brazil. vol. 1, pp. 49–60 (2013)
5. Kotler, P., Roberto, E.L.: *Marketing social: estrategias para cambiar la conducta pública*. Ediciones Díaz de Santos (Dec 1991), google-Books-ID: vMRDpBfowEMC
6. Lee, J.G., Han, J., Li, X.: Trajectory Outlier Detection: A Partition-and-Detect Framework. In: *2008 IEEE 24th International Conference on Data Engineering*. pp. 140–149 (Apr 2008), iISSN: 2375-026X
7. Liao, T.W.: Clustering of time series data—a survey. *Pattern Recognition* 38(11), 1857–1874 (2005)
8. Liu, Z., Pi, D., Jiang, J.: Density-based trajectory outlier detection algorithm. *Journal of Systems Engineering and Electronics* 24(2), 335–340 (Apr 2013)

9. Munoz-Organero, M., Ruiz-Blaquez, R., Sánchez-Fernández, L.: 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 (2018)
10. Schmitt, J.P., Baldo, F.: A Method to Suggest Alternative Routes Based on Analysis of Automobiles' Trajectories. In: 2018 XLIV Latin American Computer Conference (CLEI). pp. 436–444 (2018)
11. Yuan, G., Sun, P., Zhao, J., Li, D., Wang, C.: A review of moving object trajectory clustering algorithms. *Artificial Intelligence Review* 47(1), 123–144 (2017)