https://doi.org/10.17979/spudc.9788497498081.0575

PERFILES DE CONDUCCIÓN MEDIANTE PROCESAMIENTO INTELIGENTE Y ÁRBOLES DE DECISIÓN

Robert Perelló, Matilde Santos
Facultad de Informática, Universidad Complutense de Madrid
rperellostephens@gmail.com, msantos@ucm.es

Rafael Korbas
Universidad de Comenius, Bratislava, Eslovaquia
rafael.korbas@gmail.com

Resumen

En este trabajo se analiza la información sobre la conducción obtenida con GPS para obtener perfiles telemáticos de los conductores en base a algoritmos inteligentes. Para extraer la información significativa ha sido necesario un complejo pre-procesamiento, que incluye en primer lugar la aplicación del algoritmo Ramer-Douglas-Peucker para simplificar los datos y facilitar su manejo en etapas posteriores. Después se han aplicado operaciones para ordenar descartar viajes demasiado cortos, deshacer rotaciones y traslaciones aleatorias, etc., con el fin de poder detectar viajes repetidos o similares. Se han desarrollado dos métodos algorítmicos para detectar si dos viajes son similares. Por último, entrenando con los patrones obtenidos, se aplica para la clasificación la técnica de los árboles de decisión.

Palabras Clave: Algoritmos inteligentes, perfiles telemáticos, conducción, árboles de decisión.

1 INTRODUCCIÓN

El modelo empleado por las aseguradoras de automóviles para calcular el riesgo y valorar una póliza ha cambiado poco a lo largo de los años. En general, determinan el coste de la póliza basándose en información recogida en el momento en el que se realiza la venta: edad, sexo, kilometraje anual, tipo de vehículo e historial de accidentes.

La telemática está suponiendo un cambio importante de este modelo en el sector automovilístico. El uso de dispositivos inalámbricos para transmitir datos en tiempo real permite a los aseguradores de automóviles tomar decisiones sobre precios en base a características iniciales del vehículo o del conductor pero también incorporar información sobre el comportamiento del conductor [11]. Esto favorece el ofrecer servicios personalizados, mejorar la seguridad vial, y reducir costes de reclamaciones [9]. Por otro lado, la conducción segura permite también reducir energía [2].

Este trabajo se basa en los datos ofrecidos en la competición Kaggle "Driver Telematics Analysis" de AXA Seguros [4]. El desafío propuesto por AXA se basa en identificar conductores por medio de sus perfiles telemáticos a partir de datos GPS. Es decir, generar modelos para cada conductor según su manera de conducir. Este perfil telemático se debe extraer de una información limitada: la posición del vehículo cada segundo. Para hacer más compleja la competición los trayectos de los diferentes conductores se pueden ver sometidos a diversas rotaciones y traslaciones, desconocidas, así como lagunas de datos, trayectos falsamente asignados a un conductor, etc.

El objetivo del presente trabajo es usar los datos ofrecidos por esta competición para aplicar técnicas inteligentes para clasificar a los conductores [6]. Para ello se han analizado los datos originales, detectando carencias, fallos, ruido, distorsiones, etc. Se han pre-procesado los datos originales para extraer características telemáticas de los conductores. Se han obtenido patrones de comportamiento y se han aplicado algoritmos predictivos inteligentes para detectar trayectos que no pertenecen al conductor, que era el objetivo final del estudio. En nuestro caso se han utilizado, como primera aproximación, árboles de decisión puesto que esta técnica permite aprovechar de manera eficiente el conocimiento extraído del análisis de los datos [5]. Para abordar este problema se han propuesto otras técnicas en la literatura, como análisis de regresión lineal o redes Bayesianas [1], pero tanto en esta contribución como en [11] se trabaja con datos de sensores de pedales (freno y acelerador) para distinguir entre conductores expertos y no expertos. Un trabajo similar es el realizado por [10], quien ha tratado los mismos datos pero aplicando otras técnicas de machine learning.

La estructura del artículo es la siguiente. En la sección 2 se presentan los datos originales y los criterios de evaluación. En la sección 3 se describen algunas de las operaciones de pre-procesamiento llevadas a cabo. En la sección 4 se muestran los resultados de la clasificación con técnicas inteligentes. El trabajo termina con las conclusiones y líneas futuras.

2 MATERIALES Y MÉTODOS

Los datos sobre los que se ha trabajado fueron los cedidos por AXA para la competición Kaggle. Se trata de 2.736 carpetas, cada una numerada y representando un conductor distinto. En cada carpeta hay 200 archivos CSV (comma seperated values) que corresponden a 200 viajes presuntamente realizados por ese conductor. Pero un número reducido y aleatorio no pertenecen a dicho conductor (ni a ninguno de los otros 2.735 conductores). Por lo tanto, el reto final es clasificar los trayectos de cada conductor en los que realmente ha realizado él, porque son coherentes con su comportamiento en la conducción, y los que han sido falsamente asignados a ese conductor.

Estos archivos contienen los datos GPS, indicando la posición del vehículo, en metros, cada segundo.

```
x,y
0.0,0.0
-1.9,5.8
-5.1,9.3
…
```

Los datos presentan lapsus y pueden estar incompletos. Se supone que cada fila corresponde con una muestra tomada en sucesivos segundos, pero no siempre es así. De esta manera, podemos encontrar que de una fila a otra la posición varía en varios cientos de metros, algo imposible en el plazo de un segundo.

Para el desarrollo de este trabajo se ha empleado MATLAB© 2014a. En concreto, se ha recurrido a la librería Statistics and Machine Learning Toolbox de MATLAB para generar los clasificadores.

Aunque existen numerosas técnicas de clasificación dentro del campo de aprendizaje máquina, dado que nos encontramos ante un problema de clasificación binaria (pertenece o no a una clase) y gran cantidad de datos iniciales, la técnica elegida es los árboles de decisión.

2.1 CRITERIOS DE EVALUACIÓN

Para evaluar la tasa de éxito de un modelo para clasificar trayectos (o discernir si un trayecto pertenece a un conductor o no), la plataforma Kaggle emplea una curva ROC (Receiver Operating Characteristic, o Característica Operativa del Receptor). Se trata de una representación gráfica del rendimiento de un sistema clasificador binario según se varía el umbral de discriminación. La curva se obtiene representando el ratio de verdaderos positivos frente al ratio de falsos positivos. En concreto, se usa el área bajo esta curva para evaluar los resultados.

El área bajo la curva (AUC) es igual a la probabilidad de que el clasificador puntúe una instancia positiva aleatoria por encima de una instancia negativa aleatoria.

Los resultados se suben a Kaggle por medio de un archivo CSV con el formato siguiente:

```
driver_trip,prob
1_1,1   -> trayecto 1 pertenece al conductor 1 (probabilidad 1)

1_2,1   -> trayecto 2 pertenece al conductor 1 (probabilidad 1)

1_3,0   -> trayecto 2 pertenece al conductor 1 (probabilidad 0)

1_4,1   -> trayecto 2 pertenece al conductor 1 (probabilidad 1)
```

El primer número indica el conductor, el segundo número indica el trayecto, y el tercer número representa la probabilidad de que ese trayecto pertenezca (1) o no (0) al conductor concreto.

3 PRE-PROCESAMIENTO

Los pasos seguidos en el pre-procesamiento de los datos originales son los siguientes: encontrar y descartar trayectos cortos; simplificar trayectos para reducir el número de puntos con los que se trabaja, manteniendo su contorno; deshacer cualquier rotación o traslación a la que se haya podido someter el trayecto; detectar trayectos iguales; y por último, suavizar los recorridos, compensando tramos sin datos o descartando puntos sin sentido.

3.1 DATOS ORIGINALES

Los 200 trayectos reales de cada uno de los 2.736 conductores, para proteger la privacidad de la ubicación de los conductores, se centraron en el origen (0, 0) y fueron sometidos a rotaciones e inversiones aleatorias. Además, unos tramos cortos fueron eliminados al inicio y final de cada viaje.

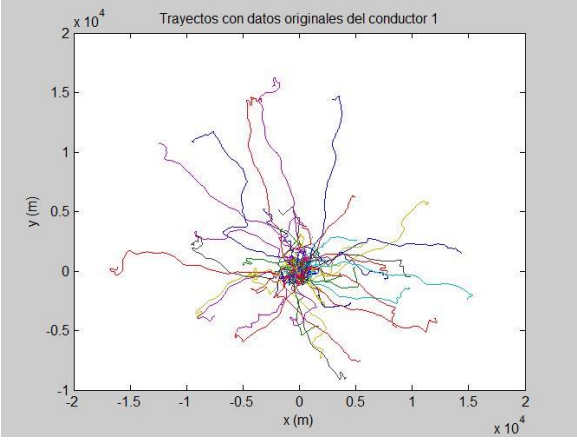

Figura 1: Trayectos del conductor 1

En la figura 1 se observan los 200 trayectos del conductor número 1.

Observando los trayectos cuidadosamente se pueden detectar trayectos repetidos, es decir, recorridos que el conductor repite varias veces. Esto supone una importante fuente de información pues podemos suponer que los viajes repetidos sí han sido realizados por él, y usaremos estos viajes para entrenar los algoritmos de clasificación.

Uno de los mayores obstáculos presentes en los datos originales se debe a la pérdida de señal GPS, por lo que aparecen tramos sin puntos. En la figura 2 se puede ver un ejemplo.

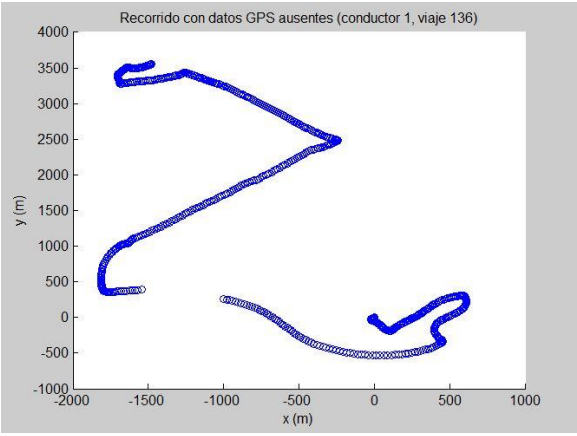

Figura 2: Recorrido con tramos con pérdida de señal GPS

Existe un número reducido de trayectos que no supera siquiera un kilómetro de recorrido. Estos viajes se desestiman ya que no ofrecen suficiente información (tanto cuantitativa como cualitativa) que pueda ser de utilidad. Para su detección simplemente se calcula la distancia recorrida cada segundo. Si la distancia recorrida en todo el viaje es inferior a un kilómetro, el viaje se marca como corto y se elimina.

Para acelerar la búsqueda de viajes iguales el resto de los trayectos son simplificados ya que para la detección de recorridos similares basta con conocer la forma o contorno del viaje. Se aplica el algoritmo de Ramer-Douglas-Peucker (RDP) [3, 8]. Este algoritmo divide recursivamente el segmento a simplificar hasta conseguir una figura con la misma forma pero muchos menos puntos que la original. Se ha utilizado un valor de $\varepsilon = 2$ como tolerancia.

De esta manera, y como ejemplo, el trayecto 1 del conductor 1 se redujo de 863 puntos a 124, manteniendo la forma de recorrido original. Esto reduce enormemente las tareas computacionales de los siguientes pasos de pre-procesado.

3.2 DESHACER ROTACIONES Y TRASLACIONES

El siguiente paso es deshacer las rotaciones y traslaciones aleatorias a las que han sido sometidos los viajes, con el fin de eliminar los efectos de estas transformaciones y poder detectar viajes iguales. En la figura 3 se muestra un ejemplo de dos trayectos iguales, aunque visualmente no sea fácil de detectar.

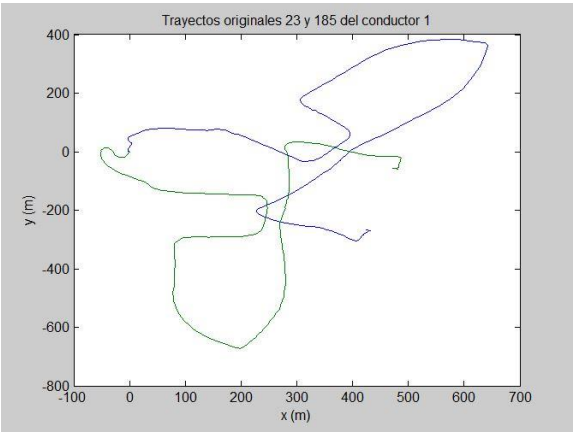

Figura 3: Trayectos iguales sin procesar

Es necesario automatizar el proceso de esta detección debido al elevado número de viajes con los que se trabaja. El primer paso es calcular si la mayoría de los puntos de la curva se encuentran por debajo del eje x. Si es así, se invierte la curva para que pase a estar por encima del eje. El siguiente paso consiste en calcular una matriz rotacional para cada curva, rotando la misma hasta que su punto final esté sobre el eje x. Así, tanto el punto inicial en el origen, como el punto final se encuentran sobre el eje x.

La matriz rotacional viene definida por la siguiente expresión:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \qquad (1)$$

Siendo θ el ángulo de giro anti-horario. Para calcular este parámetro se emplea la tangente inversa de la posición final del trayecto.

3.3 DETECCIÓN DE TRAYECTOS IGUALES

Para detectar si dos recorridos son iguales se han aplicado dos métodos consecutivamente. Si se cumplen las condiciones especificadas en cada uno de ellos, se considera que los trayectos son iguales. Dado el coste computacional del segundo método, si no se cumple el primero, éste no se aplica.

El primer método consiste en calcular ciertos parámetros que pueden definir el contorno de un recorrido: la longitud del recorrido, el área bajo la curva, el número de puntos que describen la curva, los puntos máximo y mínimo sobre el eje y, número de puntos positivos y negativos, y distancia del punto final al inicial. Para todos estos parámetros se obtienen unos ratios al comparar dos recorridos, y estos ratios se combinan linealmente para obtener un ratio final. Si este ratio total se encuentra entre unas tolerancias superior e inferior, se procede al segundo método. Se han usado 0.6 y 1.4 como umbrales inferior y superior en este primer algoritmo.

El siguiente método genera una lista con las longitudes de cada segmento del trayecto (distancias entre puntos RDP) y los ángulos entre segmentos. Una vez obtenidos, se buscan pares longitud-ángulo iguales entre los dos viajes. Si se encuentra un número por encima de un umbral, se considera que ambos trayectos son iguales. En nuestro caso se ha establecido que para que dos distancias o dos ángulos se considerasen iguales sus valores no podían variar más de un 10%.

3.4 SUAVIZADO Y CORRECCIÓN DE TRAYECTOS

El último paso del pre-procesamiento consiste en compensar las carencias y datos erróneos presentes en los datos originales. Esto se consigue observando la coherencia entre puntos contiguos, dado que representan la posición del vehículo cada segundo.

Para ello se recorren los puntos de cada trayecto, midiendo la distancia entre puntos contiguos (o, de manera equivalente, la velocidad del coche cada segundo) para encontrar anomalías. Cuando se detecta una, se observan los siguientes puntos para ver si se trata de un único punto anómalo o de un tramo en el que faltan datos. Si es un único punto, se descarta. Si se trata de un tramo sin información GPS, se calcula la velocidad justo antes y después de este tramo, y la distancia del mismo. Seguidamente se introducen puntos equidistantes con la velocidad media en el tramo.

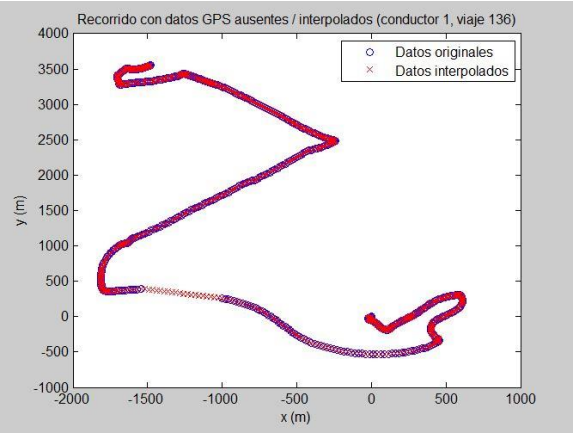

Figura 4: Trayecto reconstruido

En la figura 4 se puede ver en azul un trayecto original con un tramo sin información, y en rojo el trayecto reconstruido.

4 CLASIFICACIÓN

Una vez que todos los trayectos han sido corregidos, se procede a extraer las características de los mismos para generar las matrices de entrenamiento y clasificación.

4.1 EXTRACCIÓN DE CARACTERÍSTICAS

Para entrenar los algoritmos de predicción y posteriormente clasificar trayectos es necesario obtener características telemáticas de los conductores a partir de los datos pre-procesados. Dentro de todas las posibles, se deben seleccionar las más relevantes dado que el modelo que se desea obtener debe ser representativo de la manera de conducir de cada conductor.

Se realizaron pruebas con un primer conjunto de características y después con un segundo conjunto que contenía las características anteriores y otras adicionales para intentar mejorar los resultados. Algunas características fueron sencillas de obtener, como la velocidad, la aceleración o el frenado, y parámetros similares o derivados del vehículo. Además se implementaron otras más complejas, como agresividad de giro, media de la velocidad por el ángulo de giro o la media de la aceleración por el ángulo de giro, percentiles, etc.

El segundo conjunto incluye además histogramas de intervalos de velocidad y aceleración, tiempo durante un recorrido a cierta velocidad, etc.

En total se han extraído 94 características distintas de los trayectos. Estas características se han almacenado

en una de dos matrices: si el trayecto del que se han extraído se ha detectado como un recorrido repetido, se almacenan en una matriz de entrenamiento. Por el contrario, si no provienen de un viaje repetido, se almacenan en una matriz a clasificar.

4.2 CLASIFICACIÓN CON ÁRBOLES DE DECISIÓN

Para realizar la clasificación se ha optado por la técnica de los árboles de decisión [7]. Para ello se ha empleado la función *TreeBagger* de MATLAB. Esta función genera conjuntos de árboles de decisión a partir de una matriz MxN de entrenamiento y un vector de clases. La matriz tiene M filas correspondientes a M observaciones. En concreto, se trata de los M viajes repetidos detectados para el conductor en cuestión. Cada una de las N columnas de la matriz representa una de las características extraídas, por lo que N=94. El vector de clases contiene una fila por cada observación, por lo que es de longitud M. En este vector se etiqueta cada observación como que pertenece (1) o no (0) al conductor.

A la matriz de entrenamiento de cada conductor se le deberán añadir las matrices de entrenamiento de otros conductores para obtener un conjunto de observaciones, unas que pertenecen al conductor y otras que no. Se ha procurado buscar un balance para tener suficientes observaciones en la matriz como para generar modelos fiables de predicción, pero no tantos como para ralentizar excesivamente la computación de los modelos.

La función *TreeBagger* además permite especificar el número de árboles con el que construir el *random forest*. Una vez más se debe procurar buscar un equilibrio entre el número de árboles y el tiempo que supone computarlos.

Por último, antes de generar los modelos de clasificación, se han normalizado entre 0 y 1 los valores de las características dentro de la matriz de entrenamiento, evitando así que el algoritmo se centre excesivamente en valores grandes y descarte valores menores.

Tras obtener un modelo telemático para cada conductor a partir de su matriz de entrenamiento, se usa ese modelo para predecir si las observaciones almacenadas en su matriz de clasificación pertenecen o no al conductor. Finalmente, estos resultados se insertan en un archivo CSV para suministrar al evaluador automático de Kaggle y obtener una evaluación de la metodología. Este evaluador devuelve una puntuación AUC (área bajo la curva ROC).

4.3 DISCUSIÓN DE RESULTADOS

A continuación se recogen los resultados AUC de las pruebas de clasificación que se han realizado con el algoritmo de detección de los árboles de detección.

Tabla 1: Resultados de la clasificación.

Test	AUC	Nº árboles	Set de características
1	0,50363	20	49 características
2	0,52626	250	49 características
3	0,53432	400	49 características
4	0,53921	1.000	49 características
5	0,51845	20	94 características
6	0,53178	250	94 características
7	0,53744	400	94 características
8	0,54502	1.000	94 características

Estos valores no representan tasa de predicciones acertadas o fallidas, sino la probabilidad de que se puntúe una instancia positiva aleatoria por encima de una instancia negativa aleatoria. Lógicamente al incrementar el número de árboles con el que se generan los modelos, la respuesta mejora, pero también aumenta el coste computacional. La mejor puntuación se obtiene con el uso del segundo conjunto de características, el más completo.

Los organizadores de la competición no han publicado los resultados reales (qué trayectos pertenecen realmente a cada conductor), por lo que no es posible calcular otras métricas como tasa de fallos para evaluar los clasificadores.

5 CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se han analizado y obtenido perfiles telemáticos de conductores en base a algoritmos inteligentes a partir de datos GPS de los mismos.

Para lograr este objetivo ha sido necesario un análisis detallado de los datos originales. Con este análisis se ha podido comprobar el estado de los datos y la información relevante que se podía extraer de ellos para caracterizar la manera de conducir de los conductores. Se han detectado las carencias y posibles errores en estos los datos, los cuales se han tenido que solventar durante las etapas de pre-procesado.

El pre-procesamiento ha requerido el desarrollo y aplicación de diversos algoritmos, algunos de diseño propio. En concreto, se han estudiado todos los viajes, descartando los demasiado cortos, simplificando los datos por medio del algoritmo Ramer-Douglas-Peucker, se han ordenado, deshaciendo traslaciones y rotaciones aleatorias.

También se han corregidos fallos en los datos originales, como puntos lejanos o segmentos del recorrido sin datos del GPS sobre la posición del vehículo, descartando o interpolando en los tramos carentes de información. Todo ello con el fin de detectar viajes repetidos o similares.

Se han desarrollado dos métodos para detectar si dos viajes son similares. Este ha sido uno de los pasos más costosos computacionalmente ya que se ha tenido que iterar sobre cada conjunto de 200 viajes de los 2.736 conductores, buscando características similares entre trayectos que describan la forma de los recorridos, y pares de distancias de segmentos y ángulos iguales. Los viajes similares se han marcado como tal para poder ser usados posteriormente para entrenar los algoritmos de clasificación.

Por último, se han empleado árboles de decisión para calcular la probabilidad de que una observación pertenezca a una clase u otra o el coste de una mala predicción. Aunque no predicen correctamente todos los viajes, en todos los casos han superado los resultados que obtendría una predicción aleatoria. Los resultados son mejorables pero a costa de un gran tiempo computacional, mientras que este trabajo se ha centrado en las etapas de pre-procesamiento.

Dadas las restricciones de los datos y la poca información de la que se parte, los resultados son satisfactorios y suponen una prometedora señal de que se pueden generar predictores de comportamiento de conducción a partir de información telemática del conductor. Esta contribución abre las puertas a la implementación de analizadores de perfiles telemáticos dentro del sector de seguros de automóviles.

Como trabajo futuro se propone a corto plazo comparar con otras técnicas de clasificación que permitan mejorar las clasificaciones, como SVM, así como dar un tratamiento borroso a algunas de las características de la conducción (agresividad de una curva, o de la aceleración, etc.)

Otra línea pasa por contar con más y mejor información sobre los conductores, vehículos y recorridos, lo que mejoraría los modelos (sensores incorporados a los pedales del vehículo para conocer cómo acelera, frena y cambia de marcha el conductor, conocer las distancias que el conductor deja entre su vehículo y el inmediatamente anterior, cómo conduce en distintos tipos de carretera o ante adversidad climatológicas, etc.).

Referencias

[1] Amata, H., Miyajima, C., Nishino, T., Kitaoka, N., & Takeda, K. (2009, October). “Prediction model of driving behavior based on traffic conditions and driver types”. In: *Intelligent Transportation Systems*, 2009. ITSC'09. 12th International IEEE Conference on (pp. 1-6). IEEE.

[2] Dijksterhuis, C., Lewis-Evans, B., Jelijs, B., Tucha, O., de Waard, D., & Brookhuis, K. (2016). “In-car usage based insurance feedback strategies. A comparative driving simulator study”. *Ergonomic*s, (just-accepted), 1-25.

[3] Douglas, D. H., Peucker, T. K., (1973) “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature.Cartographica”, *The International Journal for Geographic Information and Geovisualization*, 10(2), 112-122.

[4] Kaggle (2016). “Drive Telematics Analysis”. https://www.kaggle.com/c/axa-driver-telematics-analysis

[5] Kotsiantis, S. B. (2013). “Decision trees: a recent overview”. *Artificial Intelligence Review*, 39(4), 261-283.

[6] Perelló, R. (2015). “Obtención y análisis de perfiles telemáticos de conducción mediante algoritmos inteligentes”. Máster en Ingeniería de Sistemas y de Control, Universidad Complutense de Madrid, Madrid.

[7] Quinlan, J. R. (1986). “Induction of decision trees”. *Machine learning*, 1(1), 81-106.

[8] Ramer, U. (1972) “An iterative procedure for the polygonal approximation of plane curves”, *Computer Graphics and Image Processing*, 1(3), 244-256.

[9] Reifel J., Hales M., Xu G., Lala, S. (2010). “Telematics: The Game Changer. Reinventing auto insurance.” A.T. Kearney White Paper. http://www.atkearney.co.uk/documents/10192/19079b53-8042-43ea-b870-ef42b1f033a6

[10] Vakati, K. (2015), “Driver Telematics Analysis”. Master's Projects, SJSU. Paper 394. http://scholarworks.sjsu.edu/etd_projects/394

[11] Van Ly, M., Martin, S., & Trivedi, M. M. (2013, June). “Driver classification and driving style recognition using inertial sensors”. In: *Intelligent Vehicles Symposium* (IV), IEEE (pp. 1040-1045).