Maestría en

INTELIGENCIA ARTIFICIAL APLICADA

Trabajo previo a la obtención de título de Magister en Inteligencia Artificial Aplicada

AUTOR/ES:
Ana Chávez
Iván Iglesias
Gabriel López
Pablo Sosa

TUTOR/ES:
Karla Mora
Paulina Vizcaíno

Sistema de detección de maniobras de riesgo para prevenir accidentes en la conducción vehicular mediante datos de acelerómetro utilizando técnicas de aprendizaje no supervisado

Quito - Ecuador
octubre – 2025

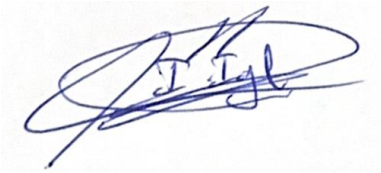
REINVENTEMOS EL FUTURO

Nosotros, **Ana Lucía Chávez Hidalgo, Gabriel Germán López López, Iván Alejandro Iglesias Giler y Pablo Ricardo Sosa Iriarte,** declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido presentado anteriormente para ningún grado o calificación profesional y que se ha consultado la bibliografía detallada.

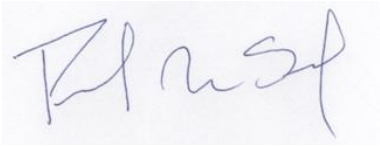
Cedemos nuestros derechos de propiedad intelectual a la Universidad Internacional del Ecuador (UIDE), para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, su reglamento y demás disposiciones legales.

Firma
Ana Lucía Chávez Hidalgo

Firma
Gabriel Germán López López



Firma
Iván Alejandro Iglesias Giler



Firma
Pablo Ricardo Sosa Iriarte

Autorización de Derechos de Propiedad Intelectual

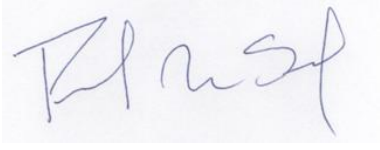
Nosotros, **Ana Lucía Chávez Hidalgo, Gabriel Germán López López, Iván Alejandro Iglesias Giler y Pablo Ricardo Sosa Iriarte**, en calidad de autores del trabajo de investigación titulado Sistema de detección de maniobras de riesgo y accidentes en la conducción vehicular mediante datos de acelerómetro utilizando técnicas de aprendizaje no supervisado, autorizamos a la Universidad Internacional del Ecuador (UIDE) para hacer uso de todos los contenidos que nos pertenecen o de parte de los que contiene esta obra, con fines estrictamente académicos o de investigación. Los derechos que como autores nos corresponden, lo establecido en los artículos 5, 6, 8, 19 y demás pertinentes de la Ley de Propiedad Intelectual y su Reglamento en Ecuador.

D. M. Quito, diciembre 2025

Firma
Ana Lucía Chávez Hidalgo

Firma
Gabriel Germán López López

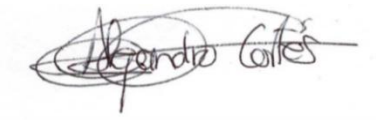
Firma
Iván Alejandro Iglesias Giler

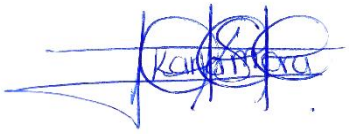


Firma
Pablo Ricardo Sosa Iriarte

Aprobación de dirección y coordinación del programa

Nosotros, **Alejandro Cortés Director EIG y Karla Mora Coordinadora UIDE**, declaramos que: **Ana Lucía Chávez Hidalgo, Gabriel Germán López López, Iván Alejandro Iglesias Giler y Pablo Ricardo Sosa Iriarte**, son los autores exclusivos de la presente investigación y que ésta es original, auténtica y personal de ellos.





Alejandro Cortés López

Maestría en Inteligencia Artificial Aplicada

Karla Estefanía Mora Cajas

Maestría en Inteligencia Artificial Aplicada

DEDICATORIA

A quienes amo, por su apoyo incondicional.

Gabriel

Este trabajo está dedicado a mi esposa Dome y a Allie, cuyo apoyo constante y firme acompañamiento hicieron posible la culminación de este proyecto; a mi abuelita, por su amor incondicional y fortaleza; a mis padres, por los valores y la orientación que han guiado mi formación personal y académica; y a mi equipo de trabajo, Anita, Iván y Gabriel, por su compromiso y aporte profesional durante este proceso.

Pablo

Este trabajo está dedicado a mi padre Iván y a mi madre Janneth, cuya confianza y apoyo inquebrantable fueron fundamentales para la culminación de este proyecto; a mis hermanos Mateo y Sebastián, por su compañía y respaldo constante; y a mi equipo de trabajo, Pablo, Anita y Gabriel, por su compromiso y aporte profesional durante el desarrollo de esta investigación.

Iván

Este trabajo está dedicado primero a Dios, a mis padres Guilermo y Lolita que han estado conmigo en cada paso de mi vida cuidándome y dándome fortaleza para continuar. A mis hermanos Fabrizio, Paulina y Marcelo por apoyarme y brindarme la ayuda necesaria. A mis hermosas sobrinas Paula y Sofía que son mi alegría; y por último a mis compañeros; Iván, Gabriel y Pablo por su dedicación y entrega en este proyecto.

Ana

AGRADECIMIENTOS

Expreso mi agradecimiento a quienes contribuyeron, directa o indirectamente, a la realización de este trabajo, especialmente a mis compañeros de equipo Anita, Iván y Pablo, por su tiempo, dedicación y constancia.

Gabriel

Agradezco profundamente a mi esposa Dome por su apoyo incondicional y respaldo en mí, a mi abuelita por su total apoyo y cariño, a mis padres por su guía y a mi equipo de trabajo, Anita, Iván y Gabriel, por su colaboración y compromiso.

Pablo

Agradezco a Dios por brindarme la sabiduría y el entendimiento necesarios para culminar esta etapa de mi vida académica, a mi familia por su apoyo incondicional sin el cual no habría alcanzado este logro, a mis compañeros de trabajo por motivarme a superarme constantemente, y a mis compañeros de tesis, Anita, Pablo y Gabriel, por su destacada participación, compromiso y entrega.

Iván

Agradezco a Dios por darme la sabiduría y la fortaleza para seguir cada día. A mis padres y por toda la confianza depositada en mí y todas las horas de enseñanza, amor y cariño que supieron otorgarme para mi formación. A mis hermanos por apoyarme en cada etapa de mi vida. A mis colegas de trabajo de quienes aprendo cada día, y a mis compañeros de tesis, Iván, Gabriel y Pablo, gracias por la dedicación, y compromiso.

Ana

RESUMEN

El presente trabajo consiste en el desarrollo un sistema inteligente para la detección de maniobras de riesgo en la conducción vehicular mediante el análisis de datos de acelerómetro. Se utilizan técnicas de aprendizaje no supervisado implementadas en un entorno embebido. La investigación surge como respuesta a las limitaciones de los sistemas tradicionales, basados en umbrales fijos, los cuales presentan una elevada tasa de falsos positivos debido a la variabilidad del entorno vial.

La metodología utilizada contempla la adquisición de datos reales de aceleración triaxial obtenidos durante la conducción urbana en la ciudad de Quito en condiciones normales.

En una primera fase se evaluaron técnicas de reducción de dimensionalidad y clustering como PCA, K-Means y DBSCAN, las cuales demostraron limitaciones en la separación consistente entre conducción normal y eventos peligrosos. Posteriormente, se adoptó un enfoque basado en detección de anomalías mediante clasificación de una sola clase, comparando los algoritmos Isolation Forest, Robust Covariance y One-Class Support Vector Machine (OCSVM).

Los resultados experimentales evidencian que el modelo OCSVM alcanza el mejor desempeño, logrando identificar la totalidad de las maniobras de riesgo sin incrementar significativamente los falsos positivos. Finalmente, el modelo seleccionado fue integrado en una aplicación de análisis y visualización, validándose con datos reales de conducción. El sistema propuesto demuestra ser una solución viable, robusta y de cómputo eficiente para mejorar la confiabilidad de las alertas vehiculares en sistemas telemáticos de bajo costo.

Palabras Claves: Acelerómetro, Aprendizaje no supervisado, Conducción vehicular, Detección de anomalías, Sistemas embebidos.

ABSTRACT

This study presents the development of an intelligent system for detecting risky driving maneuvers through the analysis of accelerometer data. Unsupervised machine learning techniques are employed and implemented in an embedded environment. The research arises in response to the limitations of traditional threshold-based systems, which exhibit a high false positive rate due to the variability of real-world driving conditions.

The proposed methodology involves the acquisition of real triaxial acceleration data collected during urban driving under normal conditions in the city of Quito. In an initial phase, dimensionality reduction and clustering techniques such as PCA, K-Means, and DBSCAN were evaluated. These methods showed limitations in achieving a consistent separation between normal driving behavior and hazardous events. Subsequently, an anomaly detection approach based on one-class classification was adopted, comparing Isolation Forest, Robust Covariance, and One-Class Support Vector Machine (OCSVM) algorithms.

Experimental results demonstrate that the OCSVM model achieves the best performance, successfully identifying all risky driving maneuvers without significantly increasing false positives. Finally, the selected model was integrated into an analysis and visualization application and validated using real driving data. The proposed system proves to be a viable, robust, and computationally efficient solution for improving the reliability of vehicle alerts in low-cost telematics systems.

Keywords: Accelerometer, Anomaly detection, Driving behavior, Embedded systems, Unsupervised learning.

Índice General

LISTAS DE TABLAS

LISTA DE FIGURAS

LISTA DE CÓDIGOS

CAPÍTULO I

Introducción

La seguridad vial es un gran reto de la movilidad del presente, y muy especialmente en contextos en los que la creciente normalización del uso de vehículos viene acompañada por la ausencia de desarrollo adecuado de diferentes infraestructuras y normativa. De hecho, según la OMS (2023), los accidentes de tráfico se cuentan entre las primeras causas de muerte para jóvenes y adultos y todo esto, desde la perspectiva de los altos costes sociales y económicos que suponen.

Figura 1

Media latinoamericana de muertes por accidentes de tránsito

Nota. Esta imagen representa el porcentaje de muertes por accidentes de tráfico en Latinoamérica (World Health Organization, 2023)

En este contexto, las tecnologías de asistencia al conductor y los sistemas telemáticos son esenciales para detectar riesgos y prevenir incidentes mediante el análisis de datos vehiculares.

Las empresas de rastreo satelital y telemática, como Ituran, han incorporado desde hace varios años sistemas de alertas que operan bajo esquemas basados en umbrales de aceleración y tiempo. Dichos sistemas son relativamente simples: cuando el sensor de aceleración supera un valor determinado durante un intervalo específico, se dispara una alerta indicando un posible evento riesgoso (por ejemplo, frenado brusco, aceleración agresiva o accidente).

Figura 2

Logo empresarial ITURAN

Nota. Ituran, líderes en el área de tecnología de rastreo satelital (ITURAN, 2025)

Sin embargo, a pesar de su practicidad, este enfoque presenta limitaciones significativas en la práctica. La principal de ellas es la alta tasa de falsos positivos, que ocurre debido a la gran variabilidad de escenarios en los que se encuentra un vehículo: baches en la calzada, reductores de velocidad, maniobras evasivas o vibraciones propias del motor pueden inducir señales acelerométricas similares a las de un evento crítico, generando alertas innecesarias (Ferreira et al., 2017).

La situación afecta negativamente la fiabilidad del sistema. Los conductores y gestores de flotas tienden a ignorar mediante un rechazo a las notificaciones que parecen no mostrar la realidad.

Así, queda demostrado que resulta necesario obtener soluciones más avanzadas que permitan una discriminación más ajustada de los patrones de conducción normal de

los sucesos realmente peligrosos. En este punto, los desarrollos en aprendizaje automático nos dan un excelente ámbito de trabajo. En especial, los algoritmos de aprendizaje no supervisado representan una opción interesante porque sirven para descubrir estructuras ocultas en grandes volúmenes de datos, incluso sin contar con etiquetas previas.En el caso de las señales del acelerómetro, el aprendizaje no supervisado puede revelar agrupaciones naturales asociadas a diferentes tipos de maniobras vehiculares, ocasiones que ayudan a la caracterización de los comportamientos dinámicos (Chen et al., 2023).

El impacto esperado del trabajo se proyecta en distintos niveles. En primer lugar, desde el punto de vista de la seguridad vial, un sistema con menos tasa de falsos positivos también ayudará a que las alertas sean tomadas más en serio por los conductores, así como también a evaluar comportamientos más seguros y reducir la probabilidad de accidente (Pan American Health Organization, 2016). En segundo lugar, desde la perspectiva de la gestión de flotas, la reducción de alertas irrelevantes significa que las empresas recibirán reportes de mejor calidad, lo que resulta en evaluaciones más justas del rendimiento de los conductores, así como también resultados más informados en materia de mantenimiento y operación.

Desde un punto de vista tecnológico y académico, el proyecto se puede entender como una contribución a la literatura incipiente sobre inteligencia artificial y sistemas embebidos, un área en la cual se busca optimizar el poder predictivo de los modelos ante las restricciones físicas de los dispositivos (Wang et al., 2020).

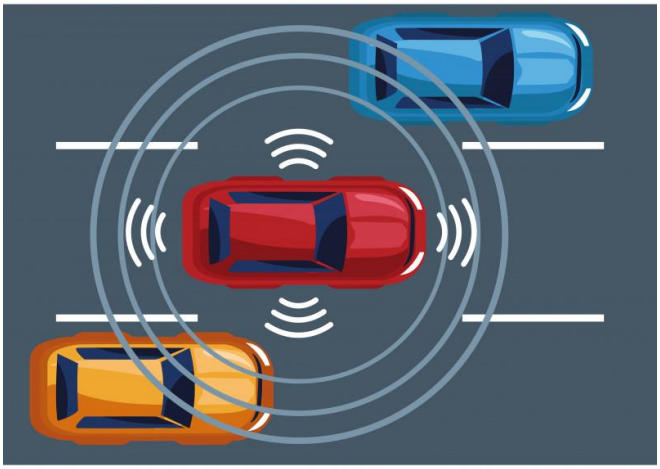
Una característica que hace diferenciar este trabajo es que no se busca deshacerse por completo del sistema de umbrales sino que complementarse y mejorar sus aportaciones. Los umbrales ayudan en la detección inicial veloz como mecanismo, pero al asociarlos con un

modelo basado en detección de anomalías y una clasificación posterior, permiten filtrar los casos en que es muy probable que la alerta sea falsa positiva.

Se destaca una tendencia global hacia la incorporación de sistemas ADAS en vehículos de diversas gamas (OECD & ITF, 2021).

Figura 3

Representación de sistema ADAS



Nota. Imagen de representación de la detección por sensores vehicular (Neumann, 2024)

Aunque muchos de estos avances se han concentrado en mercados desarrollados y en vehículos de gama alta, la adaptación de tecnologías basadas en acelerómetros y microcontroladores de bajo costo representa una oportunidad concreta para países de América Latina (Comisión Económica para América Latina y el Caribe, 2022).

De lograrse una implementación efectiva, el sistema no solo contribuiría a mejorar la seguridad en flotas corporativas, sino también en el transporte público y en vehículos particulares, ampliando su impacto social.

A pesar de la amplia adopción de sistemas telemáticos basados en acelerómetros, los métodos tradicionales de detección de maniobras de riesgo fundamentados en umbrales fijos

presentan una elevada tasa de falsos positivos, lo que reduce la confiabilidad del sistema y la efectividad de las alertas emitidas. Ante esta problemática, surge la necesidad de investigar si el uso de técnicas de aprendizaje no supervisado permite identificar patrones de conducción de forma más precisa, adaptativa y robusta frente a la variabilidad del entorno vehicular, especialmente en sistemas embebidos con restricciones computacionales.

Justificación del proyecto

El sector del automóvil está pasando por una modificación tecnológica acelerada en relación con la seguridad vial, eficiencia económica y sostenibilidad. Según la World Health Organization (2018), los accidentes viales constituyen una de las causas principales de muerte de la humanidad, con cerca de 1.3 millones de muertes/año.

La conducción de riesgo y el desgaste mecánico no monitorizado determinan el aumento del riesgo de accidente y los costes de mantenimiento; así, los sensores inerciales como los acelerómetros junto con técnicas de IA en las que se basa el aprendizaje no supervisado, ofrecen una oportunidad para detectar patrones de conducción, descubrir comportamientos no deseados, y prevenir fallos/accidentes en entorno real y con baja latencia.

El comportamiento del conductor se perfila como el componente de riesgo determinante en los accidentes de tráfico; en este sentido, estudios como los de af Wåhlberg (2017) demuestran que el estilo de conducción con un carácter agresivo, caracterizado por aceleraciones brutales, frenadas brutales y exceso de velocidad, aumenta considerablemente la probabilidad de colisión.

En esa línea, el uso de acelerómetros nos permite medir estas dinámicas de forma objetiva y en tiempo real. Investigaciones como la de Toledo et al. (2008) evidencian las

aceleraciones longitudinales y transversales como canalizadoras de la clasificación de la conducción en segura, normal y agresiva, aplicables a la reducción de accidentes, pero también a programas de seguros con pago por uso.

En el marco de esta investigación, se define una maniobra de riesgo como una acción de conducción que genera variaciones abruptas en las aceleraciones longitudinales o laterales del vehículo, significativamente distintas del patrón de conducción normal, y que incrementa la probabilidad de pérdida de control o colisión. Estas maniobras incluyen, entre otras, frenadas bruscas, aceleraciones agresivas y giros pronunciados. No se consideran maniobras de riesgo aquellas variaciones de aceleración provocadas por irregularidades de la vía, vibraciones del motor o condiciones normales de circulación, las cuales constituyen una fuente relevante de falsos positivos en los sistemas tradicionales.

El valor añadido del aprendizaje no supervisado radica en que no requiere etiquetas previas de los datos; métodos como K-Means o DBSCAN permiten descubrir automáticamente clústeres de estilos de conducción, lo que permite adaptar el sistema a distintas geografías, condiciones de tráfico y perfiles de conductor sin depender de bases de datos manualmente clasificadas.

El aprendizaje no supervisado aporta beneficios únicos para esta aplicación. El clustering de estilos de conducción ayuda a definir automáticamente perfiles de conducción sin la necesidad de datos etiquetados, y eso resulta muy interesante para la personalización de seguros, por ejemplo o en el entrenamiento de los conductores de flotas. La detección de anomalías, por su parte, permite identificar eventos poco comunes (en caso de curvas peligrosas, de desvíos de la ruta o de frenadas violentas, muy apropiado para la detección de robos o de usos indebidos del vehículo). Por último, se

encuentran las técnicas de reducción de dimensionalidad como PCA o autoencoders que ayudan a la visualización del comportamiento de una flota y a la detección de valores atípicos.

A diferencia de los modelos de aprendizaje profundo convencionales, los algoritmos no supervisados como K-Means o Isolation Forest pueden ser implementados en microcontroladores de manera liviana. El entrenamiento es offline en un PC y el microcontrolador será el que únicamente realiza la inferencia calculando distancias o comparando magnitudes ordenadas con umbrales. Todo ello permite latencias muy bajas, por debajo de un segundo una vez acumulada una ventana de datos, y una baja energía consumida, aplicable a arquitecturas como ARM Cortex-M4/M7 o ESP32. La extracción de características sencillas como la media y la varianza es computacionalmente eficiente y se ajusta bien a las librerías DSP de procesamiento digital de señales encontradas en entornos embebidos (CMSIS-DSP).

En conclusión, el uso de acelerómetros acoplados a un sistema embebido junto a una solución basada en aprendizaje no supervisado es una alternativa válida, económica y de alta repercusión social para aumentar la seguridad vial y reducir costes en automoción. Al detectar patrones de conducción riesgosa y anticipar tanto accidentes como fallas mecánicas, este enfoque no solo contribuye a salvar vidas, sino que también ofrece beneficios económicos directos a propietarios, aseguradoras y gestores de flotas. En suma, se trata de un puente tecnológico entre la electrónica automotriz tradicional y los sistemas avanzados de asistencia, con la ventaja de ser escalable, de bajo costo y adaptable a distintos contextos.

Alcance del Proyecto

La presentación que llevamos a cabo va a ser el desarrollo del presente proyecto que se basa en la detección de eventos vehiculares a partir de señales de aceleración, usando como

plataforma de prueba un vehículo urbano liviano; la solución formulada se plantea de un modo tal que puede ser extrapolada a otros vehículos con características que se asemejen a los de un automóvil urbano liviano, en relación al peso, las dimensiones y el cilindraje, en razón a lo que puede ser expuesto un planteamiento normalizable en el sector de los automóviles livianos.

Las pruebas experimentales se realizan en condiciones reales de conducción en el sector norte de la ciudad de Quito. El entorno de prueba corresponde a vías urbanas y periurbanas con carpeta asfáltica rehabilitada, en condiciones climáticas secas y sin presencia de lluvia. La infraestructura vial considerada incluye elementos comunes como rompe velocidades, redondeles, pasos deprimidos y pasos a desnivel, lo que garantiza escenarios representativos del uso cotidiano del vehículo.

La adquisición de datos se lleva a cabo sobre un vehículo liviano real, específicamente un Renault Logan modelo 2011. Durante las sesiones de conducción se generan de manera controlada y segura maniobras representativas, tales como frenadas intensas, aceleraciones bruscas y giros pronunciados. Estas maniobras permiten capturar señales dinámicas relevantes que constituyen la base para el entrenamiento y evaluación del modelo de detección de eventos.

El sistema embebido utilizado corresponde a un dispositivo propietario de la empresa Ituran, denominado F1 Cat1. La elección de esta plataforma responde a su cumplimiento con normativas automotrices regionales, su estabilidad operativa y su uso prolongado en la industria, lo que la convierte en una solución madura y validada comercialmente. El dispositivo integra un microcontrolador STM32F427, que se basa en un núcleo ARM Cortex-M4 de 32 bits, con unidad FPU y soporte para instrucciones DSP

e incluye 2 MB de memoria Flash de programa y 256 KB de RAM. Estas características permiten que se ejecute firmware para sistemas complejos que incluyen RTOS, HSM y funciones de telemetría.

Además, la plataforma también incluye un módem-GNSS Quectel EG915N que soporta comunicación por redes celulares 4G y que proporciona información de posicionamiento satelital. El sistema también incluye un acelerómetro de grado automotriz Bosch BMA456 con un rango de medición de ±16 g, el cual es adecuado para la monitorización de vehículos livianos, flotas y camiones. Esta tarjeta, a diferencia de plataformas de desarrollo genéricas, se distribuye comercialmente en varios países de la región, lo que también refuerza su pertinente artefacto para una solución orientada a entornos reales de telemática vehicular.

Acelerómetro Bosch BMA456: Descripción y Características

Las características que destacan del Bosch BMA456 son las siguientes:

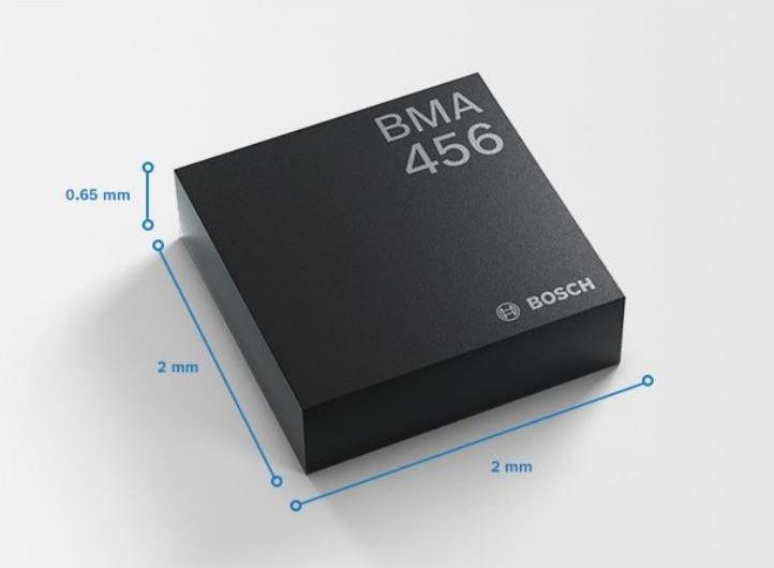
- La medición triaxial: proporciona aceleración en los tres planos fundamentales, es decir, en los ejes X, Y, Z.
- Frecuencia de muestreo configurable, que en este caso concreto se utilizó a 400 Hz, que permite medir eventos de corta duración como colisiones o impactos.
- Interfaz digital SPI, que resulta adecuada para aquellas aplicaciones embebidas que necesita comunicarse en un estado rápido y robusto.
- Rango dinámico configurable, típico en aquellas aplicaciones donde se requiere la utilización de condiciones de medida típicas en automoción.
- Bajo consumo energético, lo que es adecuado para aquellas aplicaciones donde se usa batería.

- Alta estabilidad térmica y bajo ruido, es particular en aplicaciones de análisis dinámico.

La razón por la que por las propiedades que tiene el BMA456, nos resulta adecuado para detectar eventos de alta energía como choques o maniobras de conducción, que se dan en una escala temporal más larga.

Figura 4

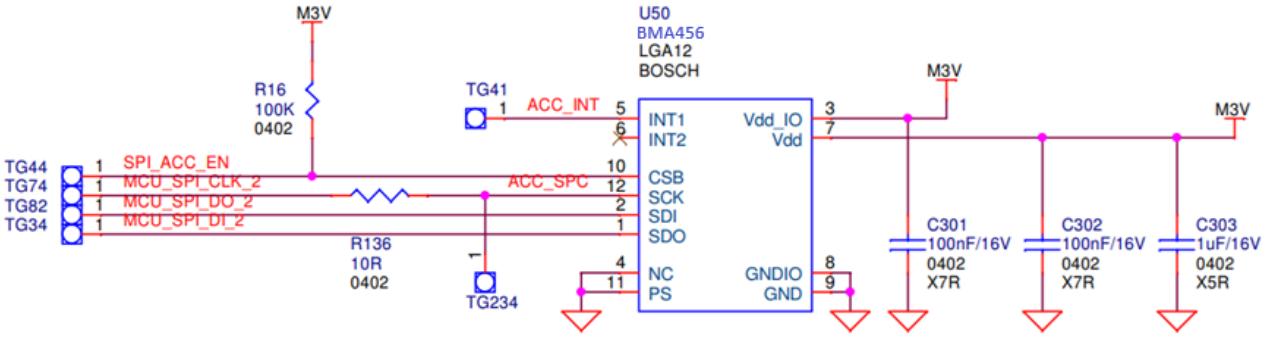
Dimensiones del acelerómetro BOSCH BMA456



Nota. Imagen tomada de la página web de Bosch, catálogo de acelerómetros (Bosch Sensortec, n.d.)

Figura 5

Diagrama de conexión del acelerómetro BOSCH BMA456



Nota. Imagen tomada del diseño esquemático del producto F1 Cat1. El contenido es propiedad intelectual de Ituran y se utiliza únicamente con fines académicos.

La metodología de desarrollo y entrenamiento del sistema consiste en la ejecución de un proceso de trabajo de 6 fases, donde se trata de garantizar una coherencia metodológica y una aplicabilidad práctica. En la primera fase se lleva a cabo la recolección de los datos del

acelerómetro en condiciones reales de conducción, ya sea en entornos de conducción urbana, de carretera o en ciertas maniobras de interés. Esta diversidad permite obtener un conjunto de datos muy amplio y representativo del comportamiento dinámico del vehículo.

En la segunda fase se lleva a cabo el preprocesamiento y limpieza de los datos.La señal cruda del acelerómetro incluye ruido derivado del entorno vehicular, oscilaciones mecánicas y la componente gravitacional, por lo que se deben utilizar filtros digitales y técnicas de normalización que sean el método para aislar la dinámica relevante para el análisis. Esta fase es muy importante para que el modelo no aprenda la dinámica espuria que proviene del sensor o el entorno.

La fase tres consiste en la aplicación de técnicas de aprendizaje no supervisado sobre los datos preprocesados. Se identifican los patrones de maniobras mediante algoritmos de agrupamiento y de detección de anomalías sin etiquetado previo (esto es, sin la necesidad de introducir la etiqueta de maniobra en k-means, por ejemplo). Este tipo de prácticas permite reducir el esfuerzo manual y mejorar la adaptabilidad del sistema ante cambios en el estilo de conducción.

De los resultados obtenidos en la etapa no supervisada se extrae un clasificador de eventos para poder etiquetar las señales de entrada que van apareciendo en tiempo real con categorías como aceleración violenta, frenada intensa, curva intensa o accidente. Con ello se completa la posibilidad de pasar a un enfoque supervisado que consolide el sistema como una herramienta útil de detección y clasificación de eventos.

En el siguiente paso el sistema extraído es incorporado al microcontrolador embebido y se trata de optimizar su ejecución para ajustarla a los límites de memoria, capacidad de cálculo y consumo de potencia del hardware, lo que garantiza baja latencia y viabilidad en aplicaciones de telemática vehicular de funcionamiento ininterrumpido.

Por último, se procede a la validación experimental mediante ensayos de campo. En el vehículo de pruebas se ejecuta simultáneamente la implementación de alertas utilizada hasta el presente y la implementación llevada a cabo en este trabajo y se realiza la comparación entre las dos implementaciones.

La evaluación se basa en los registros generados por el sensor y transmitidos a una plataforma web, donde se generan reportes de alertas y visualizaciones gráficas para su análisis. Se consideran tanto condiciones normales de conducción urbana y en carretera como eventos específicos destinados a la validación del modelo. La reproducción masiva del sistema en flotas vehiculares no forma parte del alcance de este trabajo, ya que excede las capacidades logísticas, económicas y temporales disponibles; en consecuencia, el proyecto se plantea como un prototipo funcional y una propuesta de mejora para el algoritmo actualmente empleado por la empresa Ituran.

Objetivos

Objetivo general

Desarrollar e implementar un sistema de detección de anomalías en conducción vehicular basado en algoritmos de aprendizaje no supervisado, evaluando su efectividad mediante datos de acelerómetro en escenarios reales de tráfico urbano.

Objetivos específicos

Caracterizar los patrones de conducción normal y anómala a partir de señales de acelerómetro triaxial (ejes X, Y, Z), estableciendo umbrales empíricos para eventos de aceleración brusca, frenado de emergencia y giros violentos en condiciones reales de tráfico urbano.

Comparar el desempeño de tres algoritmos de detección de anomalías basados en frontera en la identificación de eventos de conducción peligrosa, evaluando precisión, sensibilidad y *F1-Score*.

Validar la efectividad del modelo de detección de anomalías seleccionado mediante pruebas en escenarios reales de conducción, determinando su capacidad para identificar eventos críticos minimizando falsos positivos.

Implementar un sistema de alerta que procese datos de acelerómetro en tiempo real y genere notificaciones ante la detección de maniobras de conducción anómala.

CAPÍTULO II

Inteligencia artificial

La IA es una rama de la computación dedicándose a investigar y diseñar mecanismos y programas que exhiban comportamientos inteligentes similares, o análogos, a los de los seres humanos. Tales sistemas son los que son capaces de asimilar cantidades inmensas de información, de identificar patrones y tendencias para emitir pronósticos de forma autónoma. Se ha definido como una rama de la informática que estudia y genera programas que consideran procesos inteligentes (Porcelli, 2020).

Así, la IA recurre a modelos de aprendizaje computacional inspirándose en redes neuronales humanas, facilitando la construcción de máquinas inteligentes que procesan grandes cantidades de información en formas rápidas y ayudan a tomar decisiones. Esta área abarca diversas disciplinas, como el reconocimiento de voz, la visión artificial, el procesamiento del lenguaje, la adquisición de conocimiento y la robótica avanzada. Su propósito es que los sistemas puedan percibir, razonar, interactuar y aprender (Márquez Jairo, 2020).

Aprendizaje automático

El machine learning o aprendizaje automático, se define como una línea de trabajo, dentro del amplio concepto de inteligencia artificial, en la cual, mediante un conjunto de algoritmos diseñados para detectar y clasificar patrones a partir de un conjunto de grandes datos, se puede llegar a predecir comportamientos futuros, asumir nueva información e ir aprendiendo o tomando decisiones partiendo de la información, que se le ofrece. La capacidad de identificar y reconocer patrones en datos, así como la habilidad para tomar decisiones informadas, ofrece una ventaja significativa, especialmente cuando se comparan los costos asociados con este enfoque frente a otros métodos convencionales. El aprendizaje automático (ML) cuenta entre sus principales características precisamente la gran disminución de los costes de tiempo y de

computación; en este sentido, se convierte en una herramienta muy eficaz y práctica (Márquez Jairo, 2020). En contraste con estos enfoques determinísticos, que se fundamentan en la determinación de reglas y umbrales bien diferenciados, el aprendizaje automático es capaz de modelar relaciones complejas que pueden existir entre las diversas variables de entrada, sin tener que definir explícitamente el comportamiento deseado, lo que les hace especialmente útiles en aquellas situaciones en las que las interacciones son complejas y difíciles de anticipar, este último aspecto es especialmente relevante en los entornos dinámicos, como puede ser el caso de la conducción, donde la variabilidad del contexto hace que no sean posibles reglas universales y estables (Ferreira et al., 2017).

Aprendizaje supervisado y no supervisado

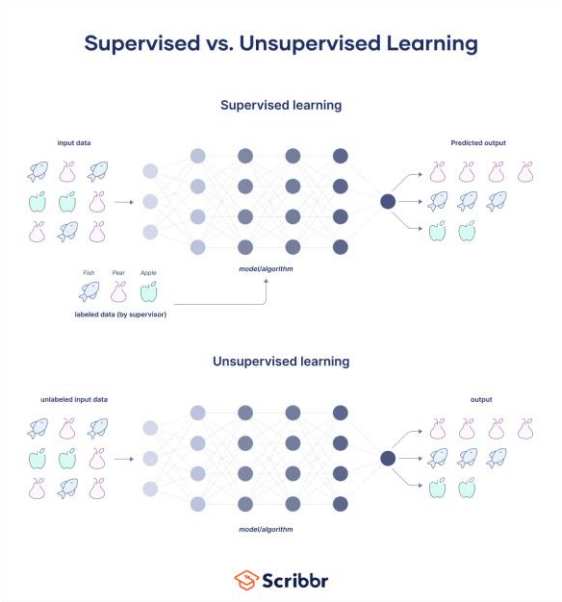
El ML se puede dividir en dos paradigmas bajo los cuales se basa, siendo el primero de ellos el análisis predictivo que se basa principalmente en la aplicación de algoritmos de aprendizaje supervisado que requieren que la intervención humana etiquete correctamente los datos a los que serán aplicados.

Los algoritmos de aprendizaje supervisado se alimentan de los datos previamente estructurados y etiquetados, en cuyo caso, el modelo se entrena a partir de ejemplos concretos de datos. El modelo busca las relaciones que hay entre las variables que son tanto de entrada como de salida que se den y, cuando se introducen datos nuevos, puede hacer predicciones (Uddin et al., 2019).

El segundo paradigma lo constituyen los algoritmos no supervisados, donde se identifican patrones latentes o estructuras que se introducen en los datos, logrando descubrir estos esquemas sin intervención humana.

Figura 6

Comparación entre aprendizaje supervisado y nos supervisado



Nota. Imagen de representación de la comparación entre tipos de apredizajes (Nikolopoulou, 2023)

Estos algoritmos pueden procesar datos sin etiquetar y extraer patrones e información significativa (Alloghani et al., 2020). El aprendizaje supervisado utiliza datos con etiquetas ya asignadas correctamente donde cada entrada en el conjunto de datos tiene su correspondiente salida, así pues es de gran consumo de recursos. En contraste, el aprendizaje no supervisado busca identificar patrones propios por sí mismo. El aprendizaje supervisado es útil para los problemas donde queremos obtener resultados conocidos como la clasificación del spam, el reconocimiento de imágenes o la predicción de precios. En lugar de eso, el aprendizaje no supervisado es adecuado para problemas donde existe interés por encontrar patrones de datos, agrupar casos similares o detectar anomalías sin también existir clasificaciones predeterminadas (Usmani Usman Ahmadand Happonen, 2022).

En la detección de maniobras de riesgo vehicular el aprendizaje supervisado sufre limitaciones debido a que requiere un volumen extenso de datos etiquetados algo que resulta caro y que además es sensible a los sesgos humanos.

En cambio, el aprendizaje no supervisado permite analizar datos de conducción reales sin etiquetado previo, adaptándose a distintos estilos de conducción y condiciones viales, lo que lo convierte en una alternativa adecuada para sistemas telemáticos desplegados a gran escala (Aggarwal et al., 2017).

Aprendizaje No Supervisado

Las principales técnicas de aprendizaje no supervisado incluyen el clustering, la reducción de dimensionalidad y la detección de anomalías. Se conoce como clustering al proceso de agrupar los datos similares en conjuntos. Mediante reducción de dimensionalidad se abordan las variables de forma que manteniendo la mayor información posible se minimicen. Se entiende por detección de anomalías al descubrimiento y la identificación de patrones atípicos, es decir, aquellos que se desvían del comportamiento que se califica de normal (Miguel-Diez et al., 2025).

El aprendizaje no supervisado se aplica en diferentes contextos donde la obtención de etiquetas resulta impráctica, costosa o incluso imposible. Las aplicaciones más destacadas son: segmentación de clientes dentro del marketing, compresión de datos empleando técnicas de reducción de dimensionalidad, exploración y análisis de datos para la revelación de estructuras ocultas, sistemas de recomendación que favorecen las preferencias y detección de anomalías en el área de la ciberseguridad, en situaciones de fraude financiero y en el monitoreo de sistemas (Priyadarshi et al., 2024). Estas ventajas son especialmente valiosas en los problemas de detección de comportamientos anómalos de conducción: no hay que clasificar manualmente

miles de eventos de conducción, y el sistema puede descubrir patrones de conducción peligrosos y no anticipados debido a ser combinaciones atípicas de maniobras.

En aplicaciones en coches, los algoritmos de aprendizaje no supervisado son satisfactoriamente aplicados para modelar los patrones de conducción, identificados como comportamientos que son característicos y que se perciben en la distribución temporal de aceleraciones, frenadas y giros (Ferreira et al., 2017).

Clustering

El clustering consiste en un tipo de agrupamiento de información de datos en clústeres, entendiendo que si los datos pertenecen a un mismo grupo (o clúster), serán más similares entre sí que otros datos de grupos diferentes. Su finalidad es hallar la estructura natural de los datos sin saber de antemano cuál sería esa clasificación (Jain, 2009).

Esta propiedad le permite aplicarse a multitud de aplicaciones como en la segmentación de clientes en marketing, donde se intenta encontrar grupos de perfiles de consumidores, la organización automática de bibliotecas digitales, la compresión de imágenes mediante una reducción de redundancia, el análisis de datos biológicos para encontrar patrones genéticos, o el descubrimiento de comunidades en redes sociales (Jain, 2009).

Algoritmos de Clustering y Reducción de Dimensionalidad

Los algoritmos de clustering y reducción de dimensionalidad son técnicas fundamentales del aprendizaje no supervisado que permiten explorar y comprender la estructura de los datos. Para la exploración inicial del presente trabajo se evaluaron tres enfoques representativos: PCA para reducción de dimensionalidad y visualización, K-Means como algoritmo de particionamiento, y DBSCAN como método basado en densidad, que difieren fundamentalmente en sus supuestos sobre la estructura de los datos.

PCA (Principal Component Analysis)

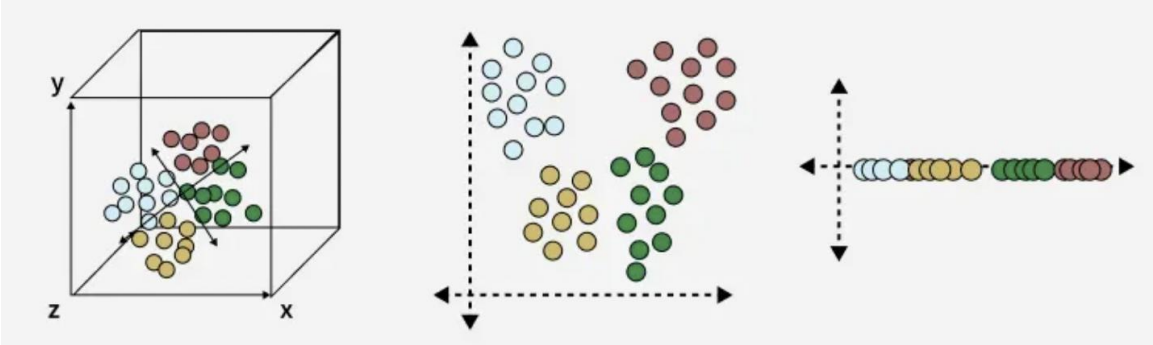
El Análisis de Componentes Principales, propuesto por Pearson en 1901 y desarrollado posteriormente por Hotelling en 1933, transforma variables correlacionadas en componentes principales no correlacionadas. El algoritmo realiza una reducción en la dimensión de los datos, a la vez que retiene la máxima varianza posible (Jolliffe & Cadima, 2016).

El método descompone la matriz de covarianza o efectúa SVD de la matriz de datos. El primer componente principal contiene la máxima varianza, el segundo, por su parte, tiene una varianza ortogonal al primero.

La proyección de los datos en las primeras dos o tres componentes principales permite visualizar los datos en espacios de baja dimensión, pudiendo así observar patrones, agrupaciones o valores atípicos. Una limitación importante es que PCA asume relaciones lineales entre variables y puede no capturar estructuras no lineales complejas (Jolliffe & Cadima, 2016).

Figura 7

Proyección de datos sobre los componentes principales mediante PCA



Nota. Se ilustra el principio de PCA, donde los datos originales son proyectados sobre nuevas direcciones ortogonales que capturan la mayor varianza del conjunto (GeeksforGeeks, 2025)

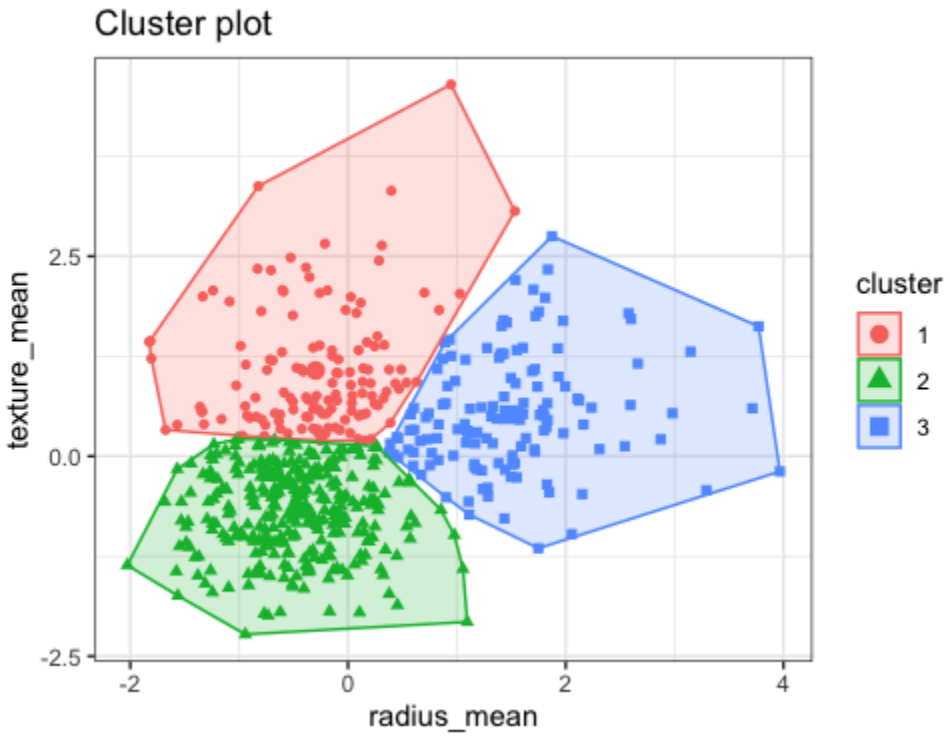
K-Means

K-Means, fue implementado por primera vez por MacQueen en 1967 [39] y se utiliza para realizar clustering, agrupando datos en K clústeres en función de la proximidad a sus respectivos centroides. Este algoritmo trabaja minimizando la suma de las distancias cuadradas de cada punto al centroide de su clúster, optimizando así la varianza intra-cluster [40]. El método K-Means funciona en 2 pasos. En primer lugar, asigna cada punto al centroide más próximo y luego recalcula el centroide como la media de los puntos de cada clúster. Estos pasos alternan hasta estabilizar los centroides de los clústeres o llegar al número máximo de iteraciones. “El valor de K a ser usado en el algoritmo debe ser definido previamente, dependiendo de eso la calidad de la solución es también dependiente de la inicialización de los centroides [41].” Para el cálculo del valor óptimo de K se puede hacer uso del método del codo, o de la aplicación del coeficiente de silueta.

En cualquier caso, se observa que la principal limitación del método K-Means es su sensibilidad a valores atípicos, así como la tendencia a encontrar clústeres esféricos y de tamaños similares (Jain, 2009).

Figura 8

Ejemplo de agrupamiento de datos mediante el algoritmo K-Means



Nota. Se muestra el resultado del algoritmo K-Means, donde los datos se agrupan en K clústeres alrededor de centroides calculados iterativamente (Harezlak, 2022)

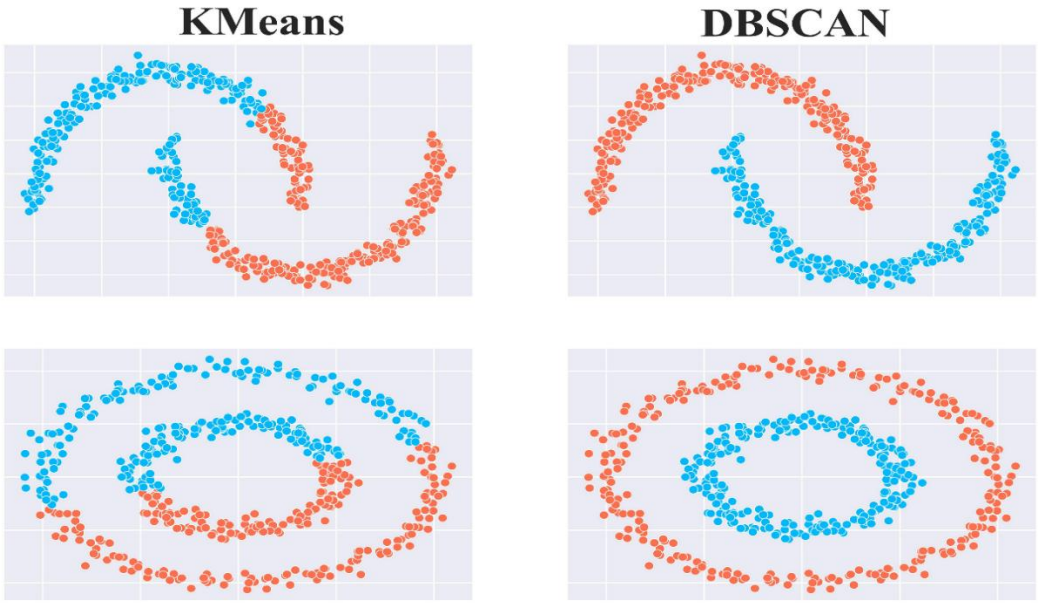
DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN es un algoritmo para hacer clustering de tipo denso propuesto en (Ester et al., 1996) que identifica las zonas densas de puntos como clústeres y la parte más dispersa como valores atípicos. A diferencia de K-Means no hace falta especificar el número de clústeres y es capaz de identificar clústeres de forma arbitraria. El algoritmo funciona con dos parámetros: la epsilon (ε) que indica el radio de vecindad y el minPts que indica el número mínimo de puntos en un radio hasta calificar una zona como densa. DBSCAN clasifica los puntos en núcleo (tiene al menos minPts en su vecindad ε), frontera (cercano a un núcleo pero con insuficientes puntos vecinos) y

ruido (no pertenece a ninguna de las two clases). Una ventaja fundamental es que se identifican explícitamente los valores atípicos; desventaja, que el algoritmo tiene dificultades para las bases de datos con diferente tamaño de clústeres y que depende de que los parámetros ε y minPts sean bien elegidos (Schubert et al., 2017).

Figura 9

Comparación del desempeño de K-Means y DBSCAN en datos con estructuras no lineales



Nota. Comparación ilustrativa entre K-Means y DBSCAN aplicada a conjuntos de datos con clústeres no convexos (Chawla, 2024).

Detección de Anomalías

La detección de anomalías identifica eventos o datos que difieren notablemente de la mayoría en un conjunto. Estos valores atípicos o excepciones, también llamadas *outliers*, pueden indicar problemas críticos como fallas en sistemas, actividades fraudulentas, intrusiones en redes o, en el contexto de este trabajo, maniobras de conducción peligrosas.

Es particularmente útil en casos donde no se pueden definir todas las clases existentes durante el entrenamiento, lo que hace que los algoritmos de detección de anomalías sean aplicables a detección de intrusiones en ciberseguridad, detección de fraudes, diagnóstico médico y monitoreo de sistemas industriales.

Tipos de Anomalías

Según Chandola et al. (2009), las anomalías pueden clasificarse en tres categorías principales, cada una con características y métodos de detección específicos:

Anomalías Puntuales

Las anomalías puntuales son los ejemplos que observan datos individuales considerados como anómalos del resto. Para los autores, Chandola et al. (2009) una instancia de los datos es considerada como una anomalía en el caso de que dicha instancia sea anómala con respecto al resto. Por lo que son el tipo más simple de valores atípicos y el principal enfoque de la detección de anomalías.

Anomalías Contextuales

Chandola et al. (2009) definen las anomalías contextuales como instancias de datos que son anómalas en un contexto específico, pero no necesariamente de otra manera. El contexto proviene de la estructura de los datos y debe incluirse en la formulación del problema.

Una anomalía en un contexto puede ser normal en otro. El comportamiento anómalo se define por los atributos dentro de un contexto específico.

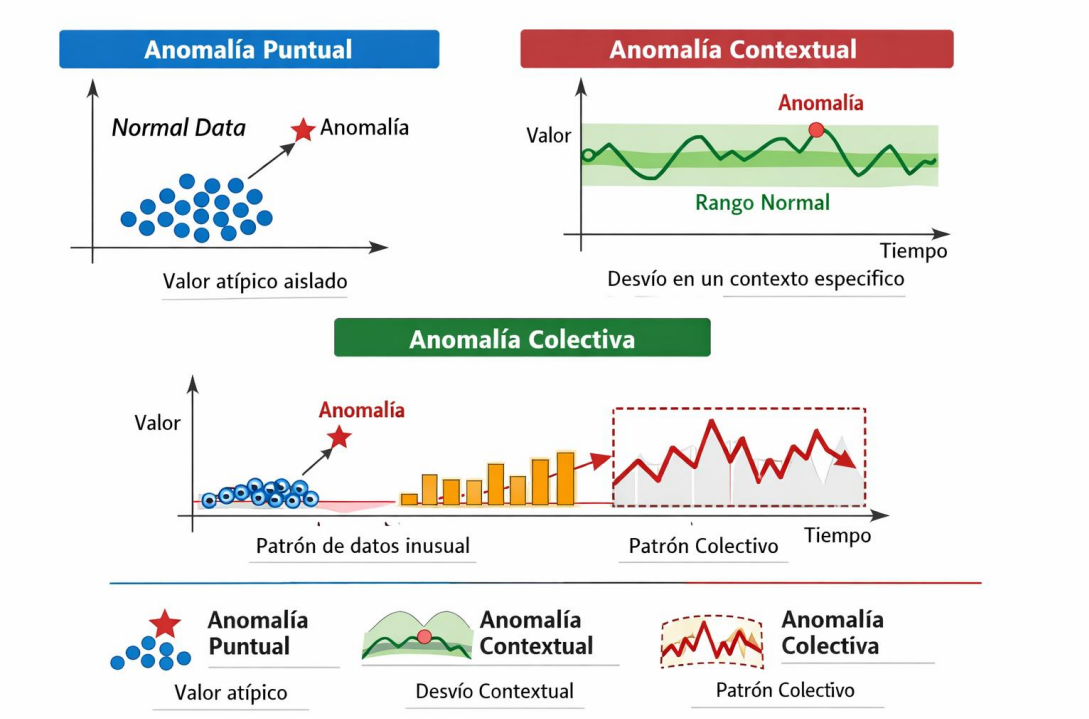
Anomalías Colectivas

Fisch et al. (2022) caracterizan las anomalías colectivas como secuencias de observaciones contiguas que no son necesariamente anómalas cuando se comparan con su

contexto de datos local o global, pero que juntas forman un patrón anómalo. También se conocen como regiones anormales o puntos de cambio epidémico.

Figura 10

Representación y comparación de anomalías



Nota. Imagen de la comparación entre tipos de anomalías (Bauskar, 2024)

Algoritmos de Detección de Anomalías Basados en Frontera

Los algoritmos de detección de anomalías basados en frontera aprenden explícitamente los límites que separan las instancias normales de las anómalas en el espacio de características (Chandola et al., 2009).

Para el presente trabajo se seleccionaron tres algoritmos representativos de este enfoque: *Isolation Forest*, *One-Class SVM* y *Robust Covariance*, que difieren fundamentalmente en sus supuestos y mecanismos de construcción de fronteras.

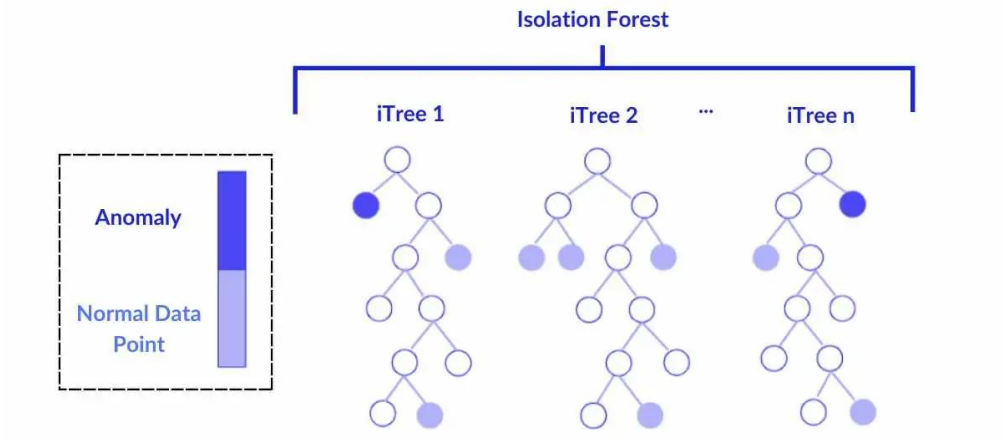
Isolation Forest

Isolation Forest es un algoritmo propuesto por Liu et al. (2008) que explícitamente aísla las anomalías en lugar de perfilar los puntos normales. El algoritmo se fundamenta en el principio de que las anomalías son observaciones raras y diferentes, por lo que son más fáciles de separar del resto de las observaciones.

El método crea árboles de aislamiento con particiones aleatorias del espacio de características. Para cada punto, se elige aleatoriamente una característica y un valor de corte, haciendo particiones recursivas hasta aislar cada punto. Las anomalías necesitan menos particiones para aislarse por estar en áreas menos densas, a diferencia de las observaciones normales que requieren más. El puntaje de anomalía se obtiene al promediar la longitud del camino en los árboles; valores cercanos a 1 sugieren alta probabilidad de anomalía (Liu et al., 2008).

Figura 11

Representación del algoritmo de Isolation Forest



Nota. Esquema de aislamiento de anomalías mediante árboles de Isolation Forest (Van Otten, 2025)

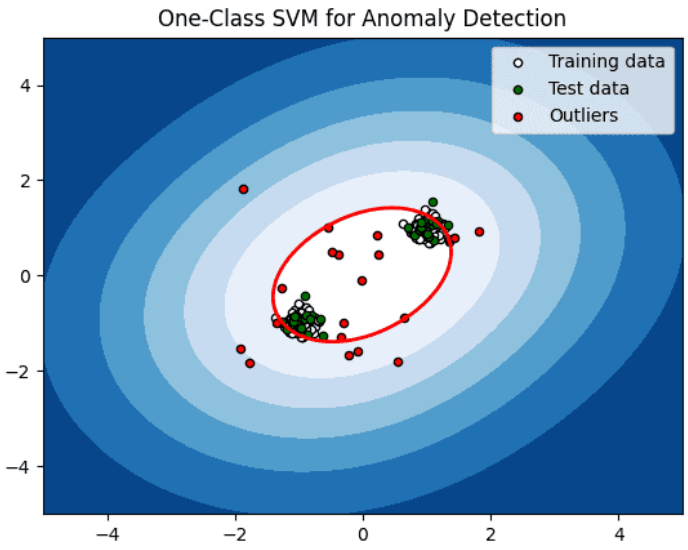
One-Class SVM

One-Class Support Vector Machine es un procedimiento que fue creado para la detección de anomalías a partir de una única clase de datos que busca una hipersuperficie que separa datos que son normales del origen en un espacio transformado por alguna función kernel (Schölkopf et al., 2001). Este algoritmo aprenderá una región que contiene a los datos de entrenamiento, y aquellas instancias que caen fuera de estas regiones son declaradas como anómalas (Chandola et al., 2009).

La función kernel suele utilizarse para base radial (RBF) de forma que el algoritmo es capaz de capturar fronteras de decisión no lineales, ya que realiza un mapeo implícito de los datos a un espacio de mayor dimensión. El parámetro *nu* controla la fracción de anomalías esperadas y actúa como límite superior en la tasa de error de entrenamiento (Schölkopf et al., 2001).

Figura 12

Representación de la idea central del algoritmo de One-Class SVM entre regiones de normalidad y anomalías



Nota. El algoritmo aprende una región que contiene los datos de entrenamiento, y las instancias que caen fuera de esta región son declaradas anómalas (Van Otten, 2024).

Robust Covariance (Elliptic Envelope)

El método Robust Covariance asume que los datos normales tienen una distribución Gaussiana multivariada y establece una frontera elíptica para la normalidad. Se basa en el método Minimum Covariance Determinant (MCD), que identifica el subconjunto de datos con la menor matriz de covarianza (Rousseeuw & Van Driessen, 1999).

La distancia de Mahalanobis evalúa cuán lejos está cada observación del centro de la distribución, teniendo en cuenta la covarianza. Este método es eficaz con datos normales, pero su rendimiento disminuye con datos multimodales o asimétricos (Chandola et al., 2009).

La selección de algoritmos como Isolation Forest, One-Class SVM y modelos basados en covarianza robusta se justifica no solo por su capacidad para identificar comportamientos atípicos, sino también por su viabilidad computacional en entornos embebidos. Estos métodos permiten una inferencia eficiente con requerimientos moderados de memoria y procesamiento. (Schölkopf et al., 2001).

Data Augmentation

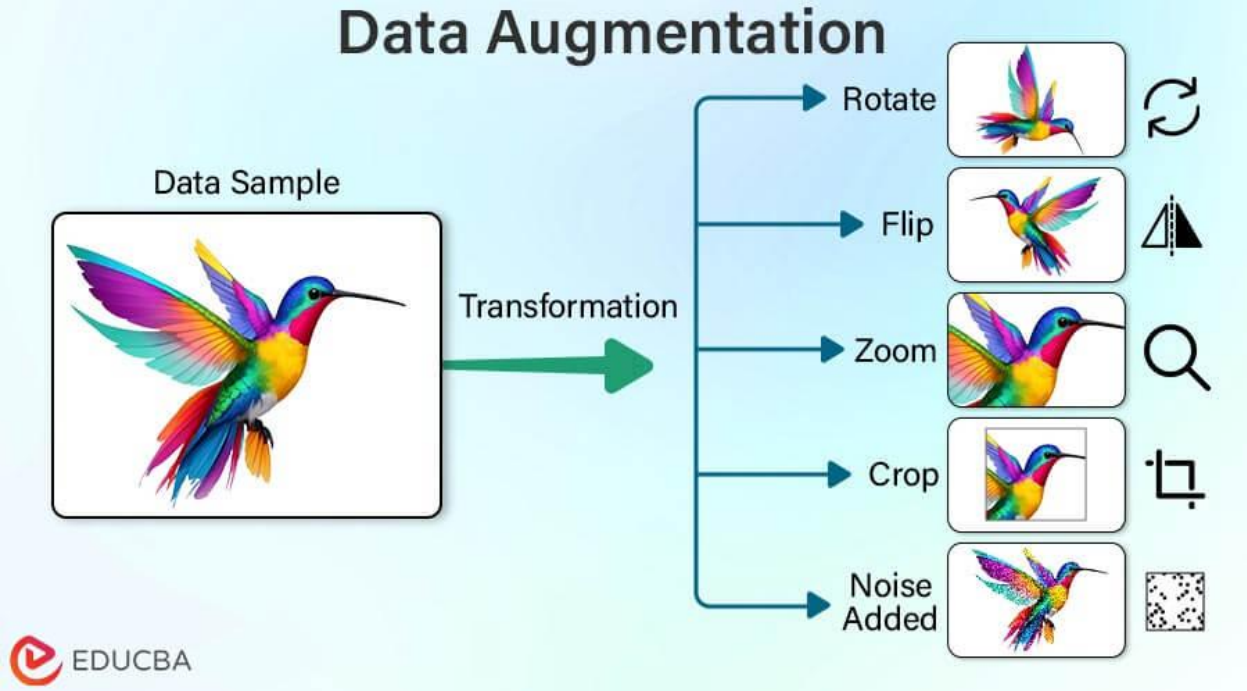
La técnica de Data Augmentation (DA) tiene por objetivo el aumento del tamaño y diversidad frente a la creación de nuevas instancias sintéticas a partir de los datos ya existentes. Tal proceso busca mejorar el comportamiento en general de los algoritmos del Deep Learning, la cantidad así como la diversidad son requisitos indispensables (Shorten & Khoshgoftaar, 2019).

Esta técnica se ha vuelto particularmente relevante en aplicaciones donde la recopilación de datos es costosa o limitada, incluyendo clasificación de imágenes médicas donde las muestras de enfermedades raras son escasas, reconocimiento de voz para acentos o idiomas con pocos

datos disponibles, detección de fraudes donde los casos positivos son minoritarios, análisis de series temporales con eventos raros, y entrenamiento de modelos de visión por computadora donde se requieren millones de imágenes etiquetadas. DA puede ayudar a controlar el sobreajuste (del inglés, overfitting), es la situación que se produce cuando el modelo está aprendiendo en exceso patrones específicos del conjunto de entrenamiento, pero únicamente predice con un bajo rendimiento para datos nuevos (Shorten & Khoshgoftaar, 2019; Wong et al., 2016).

Figura 13

Ilustración de funcionamiento de la técnica de data augmentation



Nota. Ilustra el concepto de data augmentation, una técnica utilizada para incrementar artificialmente el tamaño y la diversidad de un conjunto de datos mediante la aplicación de transformaciones controladas sobre las muestras originales (Gupta, 2024).

Métricas de Evaluación

Los algoritmos que son aplicados en el aprendizaje automático requieren de unas métricas concretas con las que evaluar su rendimiento, pero sobre todo en situaciones con desbalance de clases como puede ser el caso de detección de anomalías. Otras métricas, como la exactitud, pueden dar lugar a errores, ya que un clasificador que clasifique todo como normal puede dar como resultado un alto nivel de exactitud sin detectar anomalías.

Para poder evaluar los algoritmos a la hora de detectar anomalías se utilizan tres métricas importantes que se derivan de la matriz de confusión: precisión, sensibilidad y F1-Score. Estas métricas permiten observar su comportamiento en situaciones con clases desbalanceadas.

Precisión

La precisión mide la proporción de instancias clasificadas como anomalías que realmente son anomalías. Formalmente, se define como:

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

donde TP (True Positives, por su forma abreviada y en inglés) es el número de anomalías que el algoritmo consigue marcar correctamente y FP (False Positives) es el número de instancias normales que el algoritmo acaba marcando erróneamente como anomalías. Una alta precisión significa que si el algoritmo marca una instancia como anomalía, hay más probabilidades de que lo sea, lo que ayuda a evitar las alarmas y alertas falsas. Para la conducción peligrosa, una alta precisión significa que las alertas que nos genera el algoritmo corresponden a momentos de riesgo sí se alerta, y ayuda a evitar situaciones incómodas, en las que al final el conductor acabaría no haciendo caso del sistema (Powers, 2020).

Sensibilidad

El recall mide la proporción de anomalías reales correctamente identificadas por el algoritmo. Se define como:

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

donde FN (del inglés, False Negatives) representa número de anomalías no detectadas por el algoritmo. Un recall alto significa que el algoritmo es capaz de detectar la mayor parte de las anomalías del conjunto de datos. En el área de seguridad vial, se asocia a un recall alto un reconocimiento de la mayor parte de los hechos de la conducción peligrosa, minimizando el riesgo de que hechos críticos pasen desapercibidos (Powers, 2020).

F1-Score

El *F1-Score* es la media armónica de precisión y *recall*, proporcionando una métrica balanceada que considera ambos aspectos del desempeño del clasificador. Se define como:

$$F1 - Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

El F1-Score alcanza su mejor valor en 1 (cuando tanto precisión como recall son perfectos) y su peor valor en 0. Esta métrica es particularmente útil cuando se busca un balance entre minimizar falsas alarmas (alta precisión) y detectar la mayoría de las anomalías (alto recall). El F1-Score es preferido en problemas de detección de anomalías porque penaliza igualmente tanto los falsos positivos como los falsos negativos, a diferencia de la exactitud que puede ser dominada por la clase mayoritaria (Japkowicz & Shah, 2011; Powers, 2020).

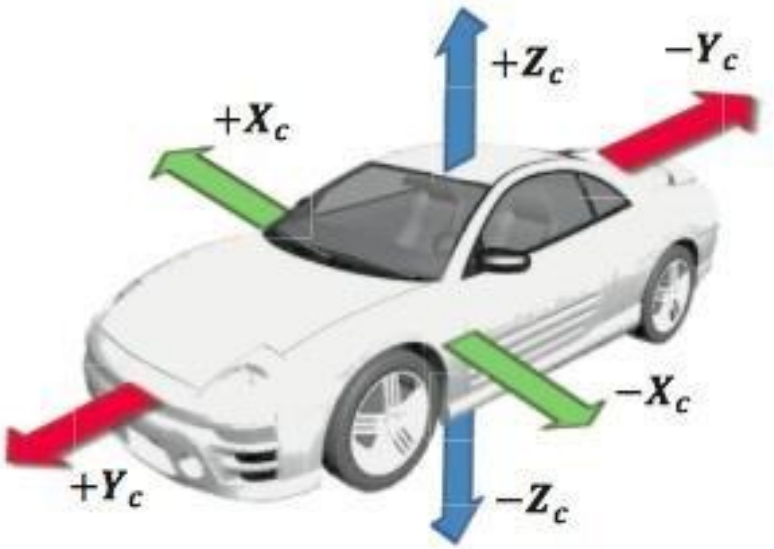
Detección de Anomalías en Conducción Vehicular

Uso de Acelerómetros en Monitoreo de Conducción

Los acelerómetros triaxiales capturan las dinámicas del movimiento vehicular en tres dimensiones espaciales. Investigaciones han demostrado que los acelerómetros de smartphones pueden revelar comportamientos de conducción agresiva cuando se utiliza una representación adecuada de los datos (Carlos et al., 2020). Los acelerómetros miden la aceleración en m/s², convertible a fuerza g dividiendo entre 9.81 m/s², con mediciones típicamente expresadas en miligravedad (mg), donde 1000 mg equivale a 1g (Ferreira et al., 2017).

Figura 14

Orientación de los ejes de un acelerómetro triaxial en un vehículo



Nota. Muestra la orientación típica de un acelerómetro triaxial en un vehículo, donde los ejes longitudinal, lateral y vertical capturan las principales dinámicas del movimiento vehicular (Kumar et al., 2018).

Los estudios comparativos han confirmado que los datos del acelerómetro son suficientes para identificar maniobras de conducción peligrosas, tales como aceleraciones rápidas, frenados

de emergencia o giros bruscos, basándose en enfoques basados en umbrales, detección de anomalías y aprendizaje automático (Gatteschi et al., 2021).

Eventos Anómalos de Conducción

Aceleraciones y Frenados Bruscos

Los frenados bruscos o las aceleraciones repentinas pueden determinarse tras el análisis de las componentes longitudinales del acelerómetro (a lo largo del eje X). Se ha establecido, a través de la investigación, unos umbrales donde se considera que los acontecimientos que superan un cierto valor implican comportamientos de conducción arriesgados (Zylius, 2017). Sistemas de evaluación de riesgo han mostrado que los acontecimientos referidos pueden ser cuantificados para poder ofrecer valores instantáneos de riesgo, estableciendo que los mismos se pueden considerar un comportamiento de conducción agresiva (Gelmini et al. 2020).

Giros Repentinos

Las oscilaciones extremas en el eje lateral (eje Y) son maniobras laterales agresivas. Las características del eje lateral son especialmente adecuadas para captar giros bruscos y cambios repentinos de carril, los cuales son situaciones de riesgo para el conductor y el vehículo implicados como también describían Carlos et al. (2019) en su artículo. Estudios han determinado que valores del eje Y por encima de ±200 mg en cortos periodos de tiempo son indicativos de fuerzas laterales que comprometen la estabilidad del automóvil (Zheng et al., 2020).

Detección de Anomalías Comportamentales

Enfoques de deteención anómala comportamental anti tips de formulado la detección como una tarea de discriminación binaria entre comportamientos esperados e inesperados,

empleando métodos contrastivos no supervisados y mostrando la viabilidad de la detección de patrones anómenos sin etiquetado previo (Ryan et al., 2020). La combinación de características temporales y espectrales de las señales del acelerómetro permite distinguir entre diferentes tipos de eventos anómalos con alta precisión (Carlos et al., 2018).

En suma, este capítulo expone las bases teóricas necesarias para el hallazgo de comportamientos de conducción anómalos con técnicas de aprendizaje no supervisado. La revisión de algoritmos de clustering, reducción de dimensionalidad y detección de anomalías ofrece la base teórica apta para la metodología que se propone en el siguiente capítulo, donde se documentan los datos, el diseño experimental, la implementación de los modelos seleccionados.

CAPÍTULO III

Adquisición de datos de acelerómetro

La detección de incidentes, de ser realizada de manera anticipada, permite en consecuencia activar alertas en tiempo real, realizar análisis retrospectivo de incidentes y de posibles estrategias de conducción que puedan ser más seguras y más eficientes en la actividad diaria, razón por la cual hoy se han expuesto para contribuir a una menor accidentalidad y menores costes de operación asociados (Elvik et al., 2009; Toledo et al., 2008).

Dentro de este contexto, los sensores inerciales, y en especial los acelerómetros, se han consolidado como uno de los elementos más utilizados para la detección de eventos vehiculares. Un acelerómetro permite medir la aceleración específica del vehículo en sus distintos ejes espaciales, lo que posibilita inferir tanto eventos dinámicos de corta duración, como impactos o frenadas abruptas, así como patrones de conducción asociados a maniobras recurrentes, por ejemplo, giros agresivos o aceleraciones sostenidas.

Gracias a su coste reducido, su bajo consumo de energía y su fácil integración en plataformas embebidas, su uso se ha consolidado como la solución más usada en sistemas de monitoreo vehicular y dispositivos de telemetría avanzada (Zheng et al., 2020).

Sin embargo, el procesamiento de las señales generadas por los acelerómetros presenta muchos retos técnicos. Los más significativos son la dependencia respecto a la orientación del sensor frente al sistema de referencia del vehículo, la combinación de la aceleración gravitacional y la aceleración dinámica, el ruido de alta frecuencia, la vibración mecánica y la operatividad en tiempo real sobre plataformas embebidas con recursos computacionales y energéticos limitados.

Por esta razón, se hace necesaria la implementación de filtros y técnicas de normalización, así como el diseño de ventanas temporales y conceptos métricos para discriminar

entre el comportamiento normal y el comportamiento anómalo sin generar falsos positivos (Bar-Shalom et al., 2001; Sabatini, 2011).

En este contexto, este documento describe de forma exhaustiva el procedimiento completo de adquisición, procesado y análisis de datos de aceleración basado en el uso del acelerómetro de Bosch BMA456 (Bosch Sensortec, n.d.). Se desgranan las características técnicas relevantes para el sensor, como la frecuencia de muestreo, la resolución y los perfiles de operación, así como su importancia para el diseño del algoritmo de detección de eventos. Finalmente, se describe la metodología estructurada desde la captura de datos en crudo hasta la extracción de variables de interés y umbrales o métricas energéticas asociadas a los distintos tipos de eventos vehiculares.

Adquisición y Orientación de la Aceleración

Dependencia de la Orientación de Montaje

Un aspecto fundamental del sistema es que la aceleración medida por el sensor depende directamente de su orientación física respecto al vehículo. Dado que el acelerómetro puede instalarse con diferentes orientaciones de montaje, los ejes del sensor no coinciden necesariamente con los ejes naturales del automóvil (frente–atrás, izquierda–derecha, arriba–abajo).

Por esta razón, el primer paso del procesamiento consiste en aplicar una transformación de orientación, normalmente implementada mediante una matriz de rotación, que permite convertir la aceleración medida en el marco del sensor al marco de referencia del vehículo. Este proceso es esencial para garantizar que las componentes de aceleración tengan un significado físico coherente en términos de dinámica vehicular.

El método para deducir la matriz de rotación comporta un procedimiento con un algoritmo mediante la fuerza de gravedad, la actitud de la GNSS (sistema global de navegación por satélites) y otros criterios que permiten obtener el resultado de una matriz de la forma:

Tabla 1

Ejemplo de valores para una matriz de rotación

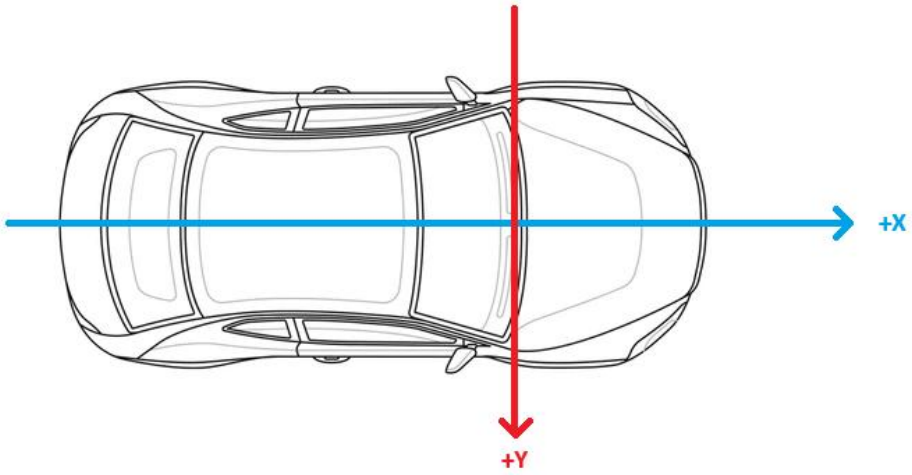
Eje	x	y	z
Forward	-0.5641	-0.82072	-0.09059
Radial	-0.80124	0.570588	-0.18012
Vertical	-0.19952	0.029021	0.979464

Nota. Tabla obtenida en el automóvil de prueba después al finalizar el proceso de orientación.

Tal como se describe en el proceso base, una vez aplicada esta corrección, la aceleración queda referenciada al automóvil, permitiendo su uso posterior para el cálculo de energía y detección de eventos.

Figura 15

Ejes X, Y con respecto al vehículo



Nota. Imagen de autoría propia.

En la figura 15 se representan los ejes de aceleración resultantes tras la aplicación de la transformación de rotación destinada a corregir la orientación del montaje físico del sensor. En

este sistema de referencia, el eje x positivo se define en la dirección frontal del vehículo, el eje y positivo corresponde a la dirección lateral derecha, y el eje z positivo se establece como perpendicular al plano formado por los ejes x e y, con orientación hacia arriba.

Procesamiento Temporal y Ventanas de Análisis

Frecuencia de Muestreo y Ventana de Promedio móvil

El sistema tiene una frecuencia de muestreo de 400 Hz, lo que significa que se realizan 400 muestreos por segundo. Las muestras de aceleración son almacenadas en un buffer circular con una longitud de 80 muestras, lo que permite realizar una ventana de tiempo deslizante de 200 ms para el cálculo de promedios móviles, lo que a su vez permite mantener un historia de datos recientes, sin necesidad de desplazar datos en memoria, lo que permite hacer un uso eficiente de los recursos. Este tipo de construcciones son muy útiles en los sistemas embebidos donde las restricciones de memoria y de potencia de computación son determinantes a la hora de diseñar el sistema.

Este mecanismo permite disminuir el ruido procedente de la señal del acelerómetro sin perder las características idóneas en la detección de las maniobras.

Compensación de la Gravedad

Influencia de la Gravedad en la Medición

Cuando un vehículo está en circulación por pendientes o rampas, las componentes de la gravedad se proyectan sobre los ejes del acelerómetro distorsionando así la lectura de la aceleración dinámica real, lo que puede conllevar a estar en predisposición a falsas detecciones si no se corrige.

Exclusión del eje z para el cálculo de eventos

A partir de las pruebas en ruta se observaron dos efectos contradictorios en cuanto al efecto de aplicar un filtro pasabajos: si bien la atenuación de las componentes de aceleración sostenida era considerable, esta misma medida resultó inconveniente ya que la información requerida para llevar a cabo el cálculo de los eventos vehiculares a partir de la acelerometría jugaba en contra de esta reducción de datos. Dicha consecuencia se manifiesta en la reducción de la sensibilidad del sistema frente a maniobras de una duración prolongada y en la dificultad para detectar de forma fiable determinados patrones dinámicos de interés.

Por este motivo, se decidió descartar el eje z dentro de la evaluación de los eventos para centrarse en la evaluación de las componentes de aceleración longitudinal (x) y lateral (y).

Esta decisión permitió preservar la información dinámica más relevante asociada a aceleraciones, frenadas y giros, al tiempo que se mitigaron los efectos adversos del filtrado sobre las señales. De este modo, la simplificación planteada también ayuda a disminuir la complejidad computacional del sistema y a incrementar la robustez del modelo, ya que permite centrar el entrenamiento y la evaluación solo sobre las dimensiones de aceleración que se relacionan de forma más directa con el comportamiento dinámico del vehículo.

Detección de Maniobras de Conducción

Una vez obtenida la aceleración dinámica, el sistema procede a detectar distintos eventos de maniobra mediante un conjunto de máquinas de estados por eje:

- Eje X (frente–atrás): aceleración y desaceleración.
- Eje Y (izquierda–derecha): giros a la izquierda y derecha.

Cada eje dispone de umbrales positivos y negativos [mG], junto con temporizadores configurables [ms] que permiten confirmar un evento solo si la condición persiste durante un tiempo mínimo. Esta lógica reduce el impacto del ruido y mejora la robustez de la detección.

Parámetros para detección de eventos en el eje x:

- Events_AccelerometerAccelerationEventThreshold_mG
- Events_AccelerometerAccelerationEventDuration_ms
- Events_AccelerometerDecelerationEventThreshold_mG
- Events_AccelerometerDecelerationEventDuration_ms

Parámetros para detección de eventos en el eje y:

- TurnAlert_TightTurnThreshold_mG
- TurnAlert_TightTurnDurationThreshold_ms

Generación de Datos para Entrenamiento y Definición de Umbrales

Los valores procesados de aceleración dinámica constituyen la base para la fase de entrenamiento del sistema. A partir de estos datos, es posible aplicar técnicas de análisis estadístico y aprendizaje no supervisado para identificar patrones de conducción normal y eventos anómalos, lo que permite definir umbrales óptimos para cada tipo de alerta.

Este enfoque combina la simplicidad de la lógica determinista en tiempo real con la potencia del análisis offline basado en datos reales del vehículo.

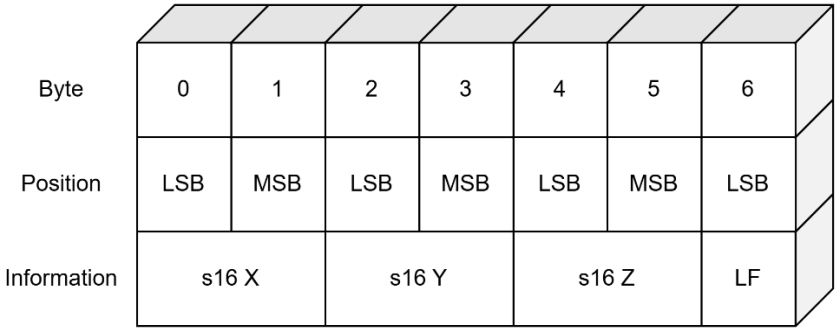
Resultados de la adquisición

El procedimiento previamente descrito fue implementado en la tarjeta telemática F1 Cat1, donde la adquisición de datos se realizó a través de un puerto serial. En dicha interfaz se obtuvo información en formato binario correspondiente a los valores de aceleración en los ejes x, y y z, expresados en miligravedades [mG], con una tasa aproximada de 400 muestras por segundo.

Cada muestra se transmitió utilizando el siguiente formato estructurado:

Figura 16

Formato binario utilizado para la obtención de vectores de aceleración



Nota. s16 representa un tipo de dato entero con signo de 16 bits, X, Y y Z corresponden a las componentes de aceleración en cada eje, y LF indica un carácter de line feed utilizado como delimitador o marca de fin de muestra.

Posteriormente, la información binaria capturada fue procesada y transformada en un archivo en formato CSV, extrayendo los valores de cada vector de aceleración de manera individual. El byte LF se empleó como mecanismo de sincronización entre muestras consecutivas, permitiendo una reconstrucción confiable de la secuencia temporal de datos y facilitando su posterior análisis y visualización en herramientas de procesamiento externo.

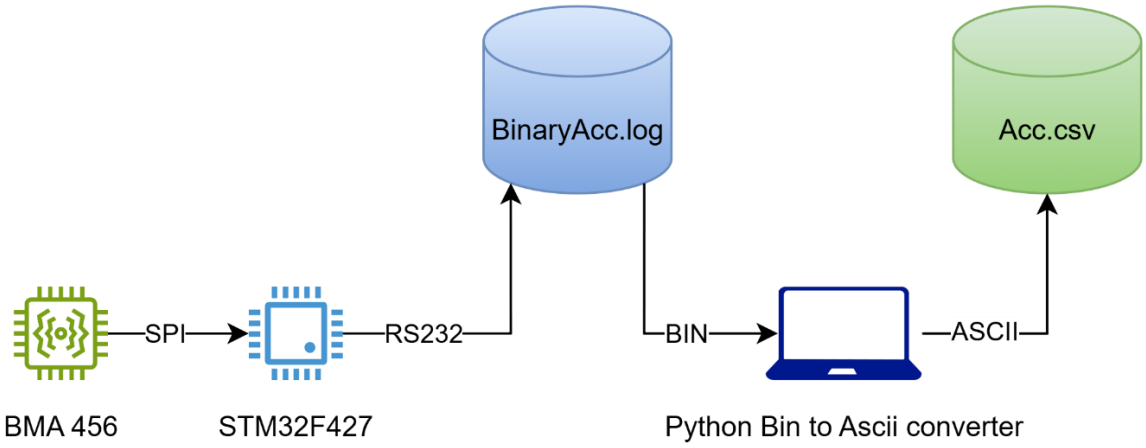
Figura 17

Adquisición de datos de acelerómetro en ruta 21H00 fecha 2025-12-24

Nota. Imagen obtenida de la cámara del tablero del vehículo de pruebas Renault Logan, en el reflejo del parabrisas se puede apreciar también la tarjeta telemática F1 Cat1

Figura 18

Esquema de adquisición de datos



Nota. Imagen de autoría propia, se muestra el flujo de la señal de aceleración: adquisición mediante el sensor BMA456, recepción en el microcontrolador mediante el formato *SPI*, envío

de data formateada en binario por puerto serial RS232 almacenamiento en Bynary.log. Posterior conversión del formato binario a formato *csv* mediante el uso de un script de Python.

La captura de aproximadamente 15 minutos de conducción continua genera un archivo binario con un tamaño cercano a 2440 KB. Tras su decodificación y conversión a formato *csv*, el volumen de información resultante asciende a 4286 KB, incremento atribuible a la representación textual de los datos y a la inclusión de delimitadores explícitos entre muestras. Los vectores de aceleración individuales obtenidos a partir de este proceso constituyen la señal de entrada para la etapa de entrenamiento del modelo de aprendizaje no supervisado.

En este capítulo se presentó una arquitectura integral para la detección automática de eventos vehiculares, la cual articula de manera coherente la adquisición sensorial, el procesamiento de señales y la lógica de decisión necesaria para identificar maniobras de conducción relevantes. El empleo del acelerómetro Bosch BMA456, combinado con técnicas de corrección de orientación, filtrado de señales y el uso de máquinas de estados, permite la detección confiable y consistente de eventos tales como aceleraciones, frenadas y giros bruscos, manteniendo un comportamiento estable en condiciones reales de operación.

El conjunto de soluciones arquitectónicas que hemos desarrollado ha mostrado ser especialmente indicada para ser utilizada en sistemas embebidos de perfil automotriz, gracias a que permite asegurar de una manera eficiente la detección, bajo un consumo computacional y una óptima adaptación mediante procesos de entrenamiento en base a datos de conducción reales. Este enfoque facilita la escalabilidad del sistema y su ajuste a distintos perfiles de manejo sin requerir modificaciones estructurales significativas en el firmware.

CAPITULO IV

En este último capítulo se muestran y se valoran los resultados obtenidos en la ejecución del sistema de detección de anomalías en la conducción del vehículo, y cuyo objetivo básico consiste en identificar acontecimientos de conducción agresiva (anomalías), como pueden ser frenazos bruscos, aceleraciones repentinas o curvas cerradas a izquierda o derecha. Se apoya el entrenamiento en técnicas de aprendizaje no supervisado, entrenando únicamente con datos de conducción normal. Resultando que cualquier desviación importantes con respecto a este tipo de comportamiento se le asimila a que es una anomalía.

Se llevaron a cabo, durante el proyecto, distintas pruebas de concepto para validar la viabilidad de diferentes enfoques de modelado, trabajándose por ello principalmente con dos experimentos. En el primero, se aplicó PCA con 2 componentes seguido de KMEANS y DBSCAN para la separación de los datos. Los resultados de este primer intento no produjeron resultados satisfactorios.

Posteriormente se decidió replantear la estrategia. Se decidió que, en lugar de buscar agrupar la información, era mejor establecer fronteras dónde termina lo normal y comienza algo inusual (anomalías). Con esta nueva perspectiva, se pudo comprobar que el algoritmo es más ligero y, lo más importante, mostró una sensibilidad suficiente para captar incidentes reales.

A continuación, se detalla todo el proceso aplicado, desde los primeros datos crudos que se pudo recolectar hasta la validación final del modelo y la solución que fue la más eficiente.

Primera prueba de concepto: PCA – KMEANS – DBSCAN

El propósito de esta primera prueba de concepto fue determinar si al combinar distintas técnicas de reducción de dimensionalidad como, el Análisis de Componentes Principales (PCA)

con algoritmos de clustering no supervisado podía revelar patrones claros que distingan la conducción normal de los eventos agresivos.

Así, se procesaron centenares de miles de registros de los sensores del vehículo, extrayendo las variables más relevantes y reduciendo la dimensionalidad a únicamente dos componentes principales. Con ese conjunto de datos compactos, se aplicaron algoritmos de clustering como K-means y DBSCAN para agrupar los comportamientos similares. El objetivo era observar si los grupos resultantes mostraban diferencias significativas entre trayectorias de conducción rutinaria y aquellas que incluían frenazos bruscos, aceleraciones repentinas o giros pronunciados. El análisis de PCA y los experimentos de agrupación nos permitirían encontrar anomalías en una conducción habitual, permitiendo conseguir la identificación automática de una conducción agresiva y de riesgo.

En esencia, se buscaba comprobar si los algoritmos de agrupación son capaces de identificar de forma automática, pero fiable y teniendo suficiente robustez para un entorno real y sin supervisión apropiada la situación de conducción agresiva.

Datos y preprocesamiento

En esta fase, se trabajó con los valores sin procesar del acelerómetro en sus tres ejes: lateral (X), longitudinal (Y) y vertical (Z). Primero se estandarizó (StandardScaler) para que cada eje tuviera media cero y varianza uno, lo cual ayuda a evitar que un eje domine el análisis por su escala. Luego se aplicó PCA y se conservó únicamente los dos componentes principales que explican la mayor parte de la variación en los datos. Así se obtuvo una representación más compacta y fácil de interpretar del movimiento del vehículo, sin perder la información clave que podría indicar una conducción agresiva.

Código 1

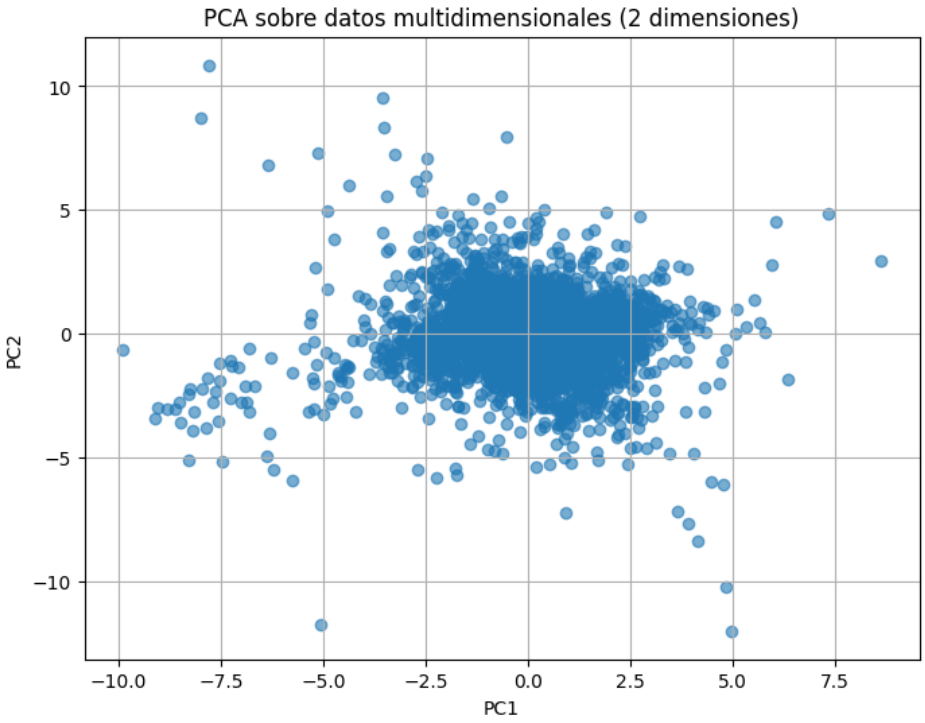
Frágmento de código de la implementación de PCA

```
      n_components = 2
      pca = PCA(n_components=n_components)
      X_pca = pca.fit_transform(X_scaled)
      df_pca = pd.DataFrame(data = X_pca, columns = [f'PC{i+1}' for i in
range(n_components)])

      explained_variance_ratio = pca.explained_variance_ratio_
      total_explained_variance = sum(explained_variance_ratio)

      print("\n--- Resultados PCA ---")
      for i, ratio in enumerate(explained_variance_ratio):
          print(f"Varianza explicada por PC{i+1}: {ratio:.4f}")
      print(f"Varianza total explicada por {n_components} componentes:
{total_explained_variance:.4f}")

      print("\nPrimeras 5 filas del DataFrame PCA:")
      print(df_pca.head())
```

Nota. Este fragmento utiliza la técnica de reducción PCA para agrupar un conjunto de datos complicado en algo con un número total de dimensiones igual a 2. Primero calcula el porcentaje de información (varianza explicada) y luego genera un gráfico de puntos donde se puede reflejar también la forma de las relaciones y el patrón de comportamiento en el conjunto de datos original para el plano 2D más fácil de interpretar.

Figura 19

Proyección de los datos de aceleración en el espacio de las dos primeras componentes principales (PCA).



Nota. Proyección de los datos en el espacio reducido generada a partir del procedimiento de PCA implementado en el primer script de prueba.

Resultados del clustering

Una vez que los datos habían sido proyectados en dos componentes principales (PCA), se aplicaron los algoritmos de clustering KMeans y DBSCAN. La Figura 19 ilustra el resultado del KMeans: se puede ver cómo se asignan los distintos clústeres y dónde se sitúan sus centroides. Al analizar ese gráfico, concluimos que K-Means no lograba separar adecuadamente los datos;

las agrupaciones resultantes eran demasiado amplias y no reflejaban la estructura real de los eventos de conducción agresiva.

Código 2

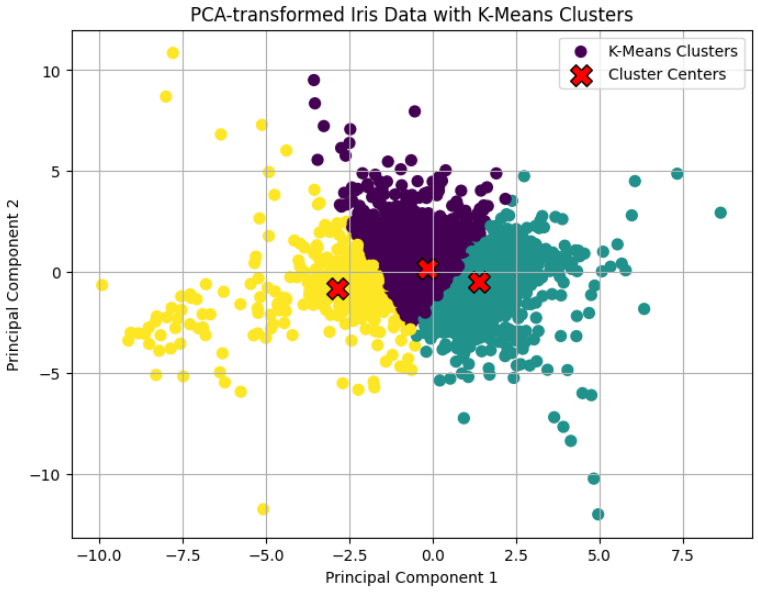
Fragmento de código de la implementación de Clustering K-Means

```
      kmeans = KMeans(n_clusters=3, random_state=123, n_init='auto')
      kmeans_labels = kmeans.fit_predict(X_pca)

      print("K-Means clustering applied successfully.")
      print(f"First 10 K-Means labels: {kmeans_labels[:10]}")

      plt.figure(figsize=(8, 6))
      scatter = plt.scatter(X_pca[:, 0], X_pca[:, 1], c=kmeans_labels,
cmap='viridis', s=50, label='K-Means Clusters')

      plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,
1],
                  s=200, marker='X', c='red', edgecolor='black',
label='Cluster Centers')

      plt.title('PCA-transformed Iris Data with K-Means Clusters')
      plt.xlabel('Principal Component 1')
      plt.ylabel('Principal Component 2')
      plt.legend()
      plt.grid(True)
      plt.show()
      print("Scatter plot showing PCA-transformed data with K-Means cluster
labels and centers generated.")
```

Nota. Aplica el algoritmo de agrupamiento K-Means para clasificar los datos previamente reducidos por PCA en tres grupos. El código entrena el modelo, asigna una etiqueta a cada dato y genera un gráfico de dispersión donde los puntos se colorean según su grupo, destacando además con una X roja los centroides, que marcan el centro matemático de cada agrupación.

Figura 20

Resultado del clustering K-Means aplicado sobre los datos reducidos mediante PCA.



Nota. PCA aplicado K-Means con tres clústeres

Además, se aplicó DBSCAN con la intención de capturar puntos aislados que pudieran considerarse ruido. Al principio los resultados parecían prometedores dado que nuestros datos tenían una forma algo esférica, el algoritmo logró distinguir mejor los clústeres. No obstante, al profundizar en la evaluación, se pudo evidenciar que el rendimiento era muy sensible a la distancia máxima, número mínimo de puntos y la identificación de eventos relevantes no era consistente. En otras palabras, pequeñas variaciones en los ajustes provocaban grandes cambios en la agrupación, lo que hacía difícil confiar plenamente en los resultados de DBSCAN. En la Figura 20, se visualiza claramente que los datos no están distribuidos de la forma esperada.

Código 3

Fragmento de código de la implementación de Clustering DBSCAN

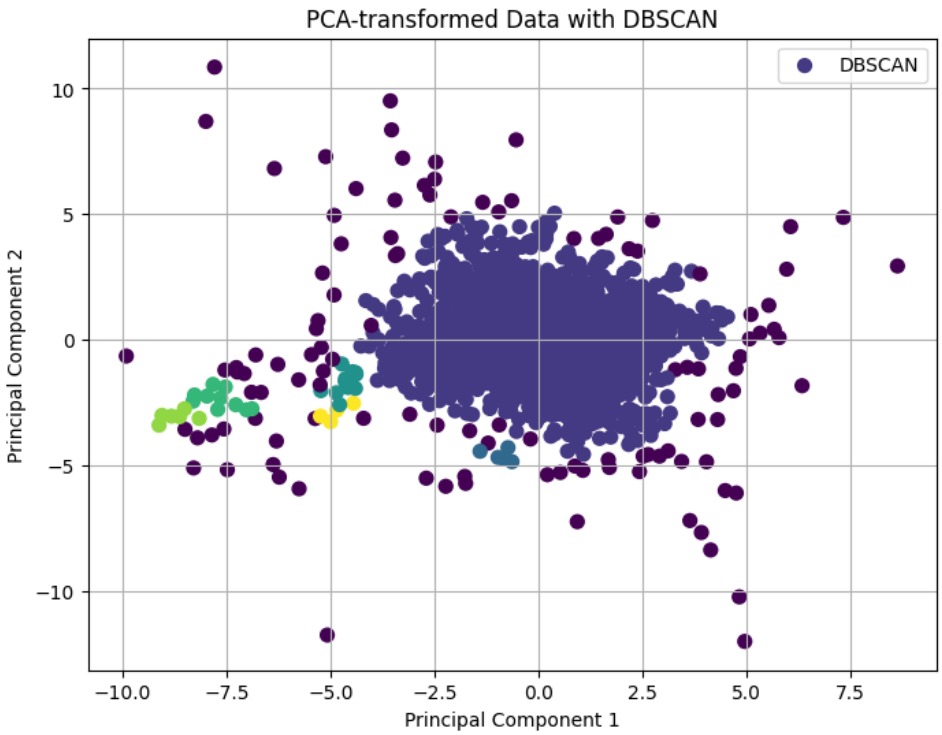
```
dbscan = DBSCAN(eps=0.5, min_samples=5)
dbscan_labels = dbscan.fit_predict(X_pca)

plt.figure(figsize=(8, 6))
```

```
      scatter = plt.scatter(X_pca[:, 0], X_pca[:, 1], c=dbscan_labels,
cmap='viridis', s=50, label='DBSCAN')

      plt.title('PCA-transformed Data with DBSCAN')
      plt.xlabel('Principal Component 1')
      plt.ylabel('Principal Component 2')
      plt.legend()
      plt.grid(True)
      plt.show()
      print("Scatter plot showing PCA-transformed data with DBSCAN
generated.")
```

Nota. Este código implementa el algoritmo DBSCAN para agrupar los datos basándose en la densidad de los puntos. Configura los parámetros de radio de vecindad y densidad mínima, clasifica los puntos en clústeres naturales o los marca como valores atípicos, finalmente genera un gráfico donde los colores indican la pertenencia a cada grupo detectado automáticamente.

Figura 21

Resultado del clustering DBSCAN aplicado sobre los datos reducidos mediante PCA.



Nota. PCA aplicado DBSCAN agrupado en clústeres

Análisis crítico y limitaciones

Al analizar las Figuras 19, 20 y 21 se evidencia que los clústeres se superponen considerablemente, lo que complica separar de manera clara la conducción normal de los eventos agresivos. Si bien la reducción de dimensionalidad con PCA facilita la visualización, también provoca que se pierda información importante sobre eventos bruscos y de corta duración.

Los clústeres generados no coincidían de forma coherente con tipos concretos de comportamiento, por lo tanto, no podían usarse como indicadores fiables de conducción peligrosa.

Desde la perspectiva operativa, esa falta de estabilidad se traduce en que, debido a la utilización de estos clústeres, pueden marcar demasiadas anomalías (generando falsos positivos y saturando al usuario), mientras que en otra puede pasar por alto eventos reales de conducción agresiva (falsos negativos).

Las principales limitaciones encontradas en esta prueba de concepto se recogen en la Tabla 2.

Tabla 2

Limitaciones identificadas en la primera prueba de concepto (PCA + clustering).

Aspecto evaluado	Observación
Separación de clústeres	Alta superposición entre los grupos
Interpretabilidad	Baja, ya que los clústeres no corresponden claramente a los distintos comportamientos.
Robustez	Alta dependencia de hiperparámetros (distancia, número mínimo de puntos)
Aplicación en tiempo real	No viable con los métodos probados

Nota. Tabla comparativa de las limitaciones del método de clústeres

Finalmente, la primera prueba de concepto nos permitió evaluar la agrupación no supervisada con los datos de conducción, no obstante, se pudo confirmar que su capacidad para separar correctamente las anomalías no era la adecuada. Si bien se logró identificar patrones generales en los datos, la solución resultó insuficiente para construir un sistema de detección fiable y estable. Este hallazgo nos llevó a replantear la estrategia, adoptando un segundo enfoque que se centra exclusivamente en modelar el comportamiento de conducción normal.

Segunda Prueba de concepto: Modelo de conducción normal

El segundo enfoque que se implementó se centra únicamente en modelar la conducción normal. En lugar de tratar de agrupar todos los tipos de comportamiento, el algoritmo aprende a reconocer la región del espacio que corresponde a una conducción segura, basándose en datos típicos de manejo cotidiano. Cualquier punto que se aleje notablemente de esa zona se etiqueta automáticamente como anomalía. Este principio se le conoce como la Clasificación de Una Clase (del inglés, *One Class Classification*), esta estrategia permitió detectar anomalías sin tener que etiquetar cada posible accidente de antemano.

Este método resulta particularmente útil cuando los eventos que se requiere detectar como maniobras agresivas son poco frecuentes, por lo tanto, difíciles de etiquetar, es por ello que se consideró que este enfoque es lo más adecuado para resolver la problemática. Al entrenar solo con datos de conducción normal, el modelo no necesita ejemplos explícitos de fallos para aprender a detectarlos; simplemente reconoce lo que es normal y marca como sospechoso todo aquello que se desvía de ese patrón.

Preprocesamiento

El PCA realizado en la primera prueba de concepto demostró que el ruido no es un detalle menor, sino el enemigo que puede hacer perder el enfoque de lo que realmente importa.

Primero se realizó una limpieza de Gravedad, ya que cada sensor registra constantemente la fuerza gravitatoria de 1[g] gravedad dirigida hacia abajo (eje Z). Si se mantiene esta constante en el análisis, las vibraciones reales se pierden en la gravedad. Se utilizó la mediana estadística para restar esta fuerza constante, dejando solo las fuerzas dinámicas. Siguiendo con las ventanas temporales, se decidió que en lugar de tratar cada muestra individualmente, se agrupó los datos en bloques de 1[s] segundo. Este corte temporal permite capturar patrones de comportamiento que ocurren en un lapso razonable, sin perder la resolución necesaria. Para concluir con el *feature engineering*, esto quiere decir que de cada ventana de 1[s] segundo se extrae dos datos valiosos del movimiento:

Energía Cinética: Magnitud de la velocidad en ese segundo.

Jerk: Cambio brusco de aceleración, que a menudo precede a un evento crítico.

Estas dos métricas transforman millones de lecturas crudas en un conjunto compacto y significativo de vectores que el modelo puede procesar con rapidez.

Código 4

Visualización de Filtrado de señal

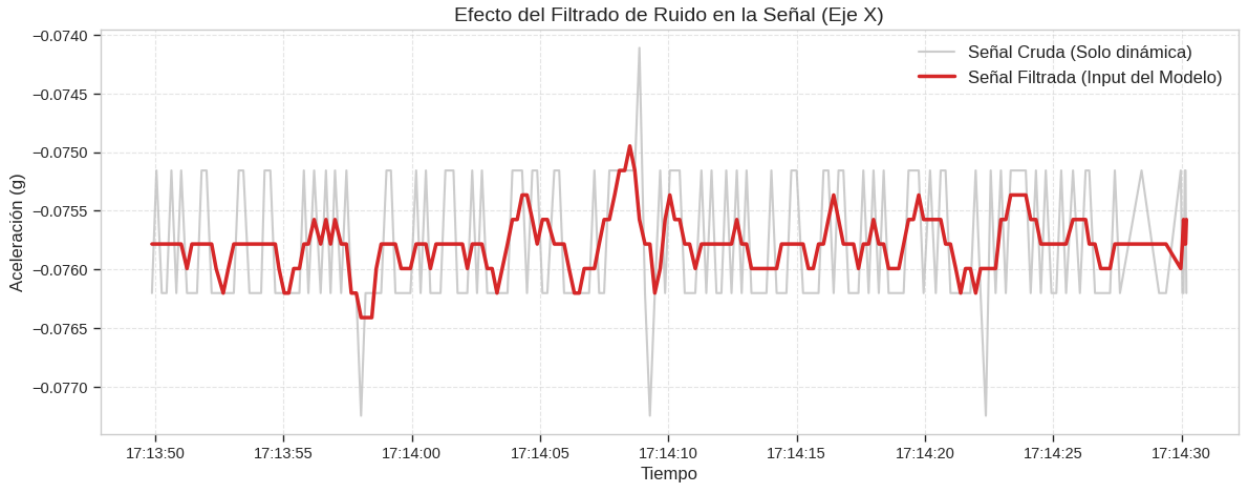
```
start = 100
end = 300
if len(df_proc) > end:
    subset = df_proc.iloc[start:end]
else:
    subset = df_proc

plt.figure(figsize=(14, 5))

plt.plot(subset.index, subset['ax_dynamic'],
         label='Señal Cruda (Solo dinámica)', color='silver',
linewidth=1.5, alpha=0.8)
```

```
      plt.plot(subset.index, subset['ax_smooth'],
               label='Señal Filtrada (Input del Modelo)', color='#d62728',
linewidth=2.5)

      plt.title("Efecto del Filtrado de Ruido en la Señal (Eje X)",
fontsize=14)
      plt.ylabel("Aceleración (g)", fontsize=12)
      plt.xlabel("Tiempo", fontsize=12)
      plt.legend(fontsize=12)
      plt.grid(True, linestyle='--', alpha=0.5)
```

Nota. Este código compara gráficamente la señal original ruidosa (gris) contra la procesada (roja) en un intervalo corto; sirve para verificar visualmente que el filtrado eliminó el ruido irrelevante, dejando una señal limpia y lista para el modelo.

Figura 22

Comparativa Señal Cruda vs. Features Procesadas



Nota. La figura ilustra cómo el preprocesamiento convierte datos caóticos en información útil para la detección de anomalías.

Técnicas de entrenamiento (Benchmarking)

La elección de los algoritmos no fue arbitraria. Se sometió a prueba competitiva a tres algoritmos, **Isolation Forest, Robust Covariance, One Class SVM (OCSVM),** cada uno con una visión matemática distinta sobre lo que constituye una anomalía.

La elección del *kernel* RBF y, por tanto, de una frontera circular o esférica responde a la naturaleza física del problema. En dinámica vehicular existe el concepto de Círculo de Fricción, que es la adherencia máxima de los neumáticos con la carretera es omnidireccional. Cuando la suma vectorial de las fuerzas longitudinales (frenado o aceleración) y laterales (giro) excede el radio de este círculo, el vehículo pierde control.

El OCSVM dibuja, sin que nadie lo indique explícitamente, una esfera de protección que rodea el origen del espacio de características. Esta frontera esférica refleja exactamente la física que rige el comportamiento de los vehículos. Cuando el modelo se entrena de forma no supervisada, aprende que cualquier vector de fuerza cuya longitud supere el radio de normalidad (límite seguro de agarre del neumático). Es una señal clara de que el vehículo está sobrepasando los límites físicos de control; en ese instante, la anomalía se marca con precisión.

Esta correspondencia entre el modelo matemático y el fenómeno real le da al OCSVM una ventaja teórica frente a otros enfoques. Los *Isolation Forest* recogen anomalías con particiones rectangulares que pueden pasar por alto la naturaleza radial de los límites, y el *Robust Covariance* presupone una distribución gaussiana que no siempre captura la complejidad del comportamiento de un vehículo. Es por ello que, con el OCSVM, la geometría de la frontera se alinea directamente con el círculo de fricción haciendo que la detección sea tanto más intuitiva como fiable.

Código 5

Código EDA(Exploratory Data Analysis) de conducción normal

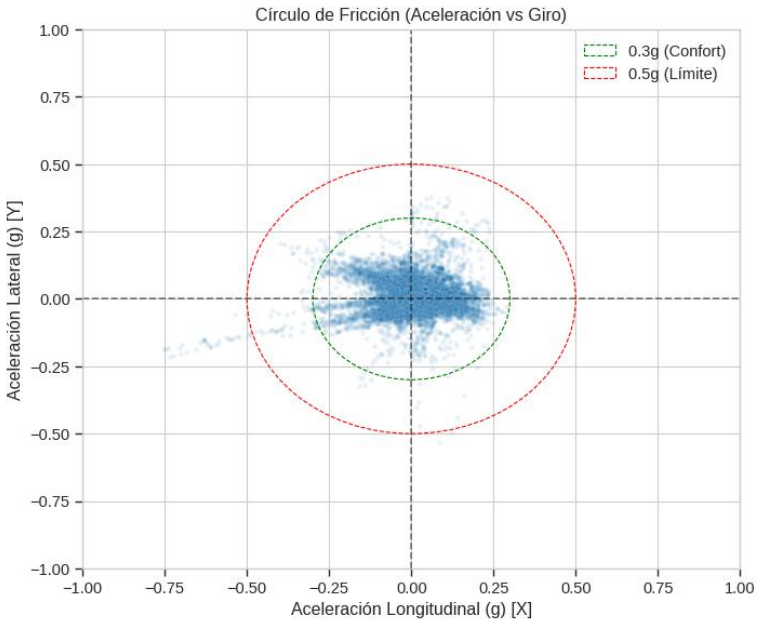
```
      def plot_eda(df):
          plt.figure(figsize=(14, 6))

          plt.subplot(1, 2, 1)
          sns.scatterplot(data=df, x='ax_smooth', y='ay_smooth', alpha=0.1,
s=10)
          plt.title("Círculo de Fricción (Aceleración vs Giro)")
          plt.xlabel("Aceleración Longitudinal (g) [X]")
```

```
          plt.ylabel("Aceleración Lateral (g) [Y]")
          plt.axhline(0, color='black', linestyle='--', alpha=0.5)
          plt.axvline(0, color='black', linestyle='--', alpha=0.5)
          plt.xlim(-1, 1)
          plt.ylim(-1, 1)

          circle1 = plt.Circle((0, 0), 0.3, color='g', fill=False,
linestyle='--', label='0.3g (Confort)')
          circle2 = plt.Circle((0, 0), 0.5, color='r', fill=False,
linestyle='--', label='0.5g (Límite)')
          plt.gca().add_patch(circle1)
          plt.gca().add_patch(circle2)
          plt.legend()

          plt.subplot(1, 2, 2)
          sns.kdeplot(df['ax_smooth'], label='Longitudinal (X)', fill=True)
          sns.kdeplot(df['ay_smooth'], label='Lateral (Y)', fill=True)
          plt.title("Distribución de Aceleraciones")
          plt.xlabel("Aceleración (g)")
          plt.legend()

          plt.tight_layout()
          plt.show()

      if not df_raw.empty:
          plot_eda(df_proc)
```

Nota. Esta función genera un reporte visual para evaluar la seguridad de la conducción: crea un círculo de fricción que compara las fuerzas G contra límites de referencia de 0.3[g] y 0.5[g] y un gráfico de densidad para analizar la frecuencia de las aceleraciones, facilitando la detección inmediata de maniobras agresivas.

Figura 23

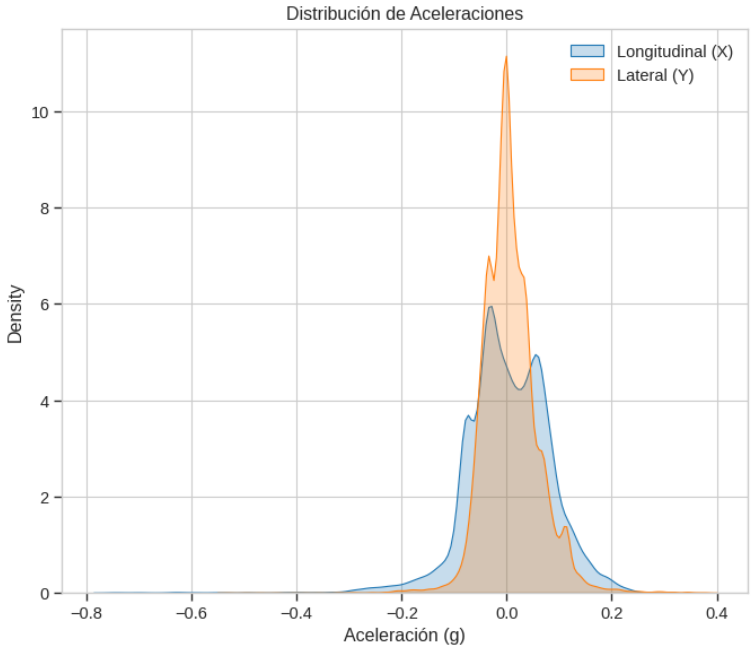
Círculo de Fricción



Nota. La figura ilustra la intensidad del manejo combinando frenado y giro; la concentración de los puntos azules dentro del círculo verde confirma que la conducción se mantuvo siempre en la zona de confort bajo 0.3[g], evitando maniobras bruscas que se acerquen a los límites de adherencia del vehículo (círculo rojo).

Figura 24

Distribución de Aceleraciones



Nota. El gráfico muestra la frecuencia con la que ocurren las fuerzas G. Ambas curvas (longitudinal y lateral) tienen un pico tan alto y estrecho en el centro, esto indica que la gran mayor parte del tiempo el vehículo circuló de forma estable, en línea recta y sin cambios agresivos de velocidad.

Diseño de la Validación Experimental

La recolección de datos para anomalías es complicada ya que puede provocar accidentes reales. Por eso se decidió utilizar una técnica que, aunque artificial, mantiene la fidelidad de los eventos críticos: la inyección de anomalías sintéticas. En el conjunto de prueba, se añadió perturbaciones calculadas que imitan con precisión las firmas de los dos eventos más peligrosos:

- **Frenado de Emergencia:** Introduce valores de aceleración longitudinales menores a -0.6[g].

- **Viraje Brusco**: Añade valores de aceleración lateral con módulo superior a 0.6[g].

Código 6

Generación de data sintética para el entrenamiento

```
      def generate_synthetic_test_set(df_normal, n_anomalies=50):
          df_test = df_normal.sample(n=min(200, len(df_normal)),
random_state=42).copy()
          df_test['label'] = 0

          anomalies = []
          for _ in range(n_anomalies):
              type_anom = np.random.choice(['brake', 'turn', 'accel'])

              if type_anom == 'brake':
                  row = df_normal.iloc[0].copy()
                  row['ax_min'] = np.random.uniform(-1.0, -0.6)
                  row['ax_mean'] = row['ax_min'] * 0.8
                  row['energy'] = row['energy'] * 5
                  row['jerk_mean'] = row['jerk_mean'] * 5
                  anomalies.append(row)

              elif type_anom == 'turn':
                  row = df_normal.iloc[0].copy()
                  val = np.random.uniform(0.6, 1.0) * np.random.choice([-1,
1])
                  row['ay_max'] = val if val > 0 else row['ay_max']
                  row['ay_min'] = val if val < 0 else row['ay_min']
                  row['ay_mean'] = val * 0.8
                  row['energy'] = row['energy'] * 5
                  anomalies.append(row)

              elif type_anom == 'accel':
                  row = df_normal.iloc[0].copy()
                  row['ax_max'] = np.random.uniform(0.5, 0.8)
                  row['ax_mean'] = row['ax_max'] * 0.8
                  row['jerk_mean'] = row['jerk_mean'] * 4
                  anomalies.append(row)

          df_anom = pd.DataFrame(anomalies)
          df_anom['label'] = 1

          df_final = pd.concat([df_test, df_anom]).sample(frac=1,
random_state=123).reset_index(drop=True)
          return df_final
```

Nota. Esta función construye un conjunto de datos de prueba para validar modelos de detección de anomalías; mezcla una muestra de conducción normal real (etiqueta 0) con eventos peligrosos

simulados artificialmente (etiqueta 1) como frenazos, giros bruscos y aceleraciones fuertes alterando matemáticamente las fuerzas G, la energía y el jerk para imitar maniobras agresivas. Con estos datos sintéticos se evaluó cada algoritmo con las métricas de clasificación: precisión, sensibilidad (recall) y F1score.

Código 7

Benchmark de Modelos

```
      def benchmark_models(df_train, df_test):
          models = {
              "Isolation Forest": IsolationForest(contamination=0.05,
random_state=42, n_jobs=-1),
              "One-Class SVM": OneClassSVM(nu=0.05, kernel="rbf",
gamma='scale'),
              "Robust Covariance": EllipticEnvelope(contamination=0.05,
random_state=42)
          }

          results = []

          scaler = StandardScaler()
          X_train_full = scaler.fit_transform(df_train)
          X_test = scaler.transform(df_test.drop(columns=['label']))
          y_true = df_test['label']

          for name, model in models.items():
              print(f"Entrenando {name}...")

               X_train_curr = X_train_full
              if name == "One-Class SVM" and len(X_train_full) > 10000:
                  print(f"  -> Submuestreando SVM (dataset grande detectado:
{len(X_train_full)} muestras)...")
                  idx = np.random.choice(len(X_train_full), 10000,
replace=False)
                  X_train_curr = X_train_full[idx]

              start_train = time.time()
              model.fit(X_train_curr)
              train_time = time.time() - start_train

              start_inf = time.time()
              y_pred_raw = model.predict(X_test)
              inf_time = (time.time() - start_inf) / len(X_test) * 1000

              y_pred = np.where(y_pred_raw == -1, 1, 0)

              prec = precision_score(y_true, y_pred)
              rec = recall_score(y_true, y_pred)
              f1 = f1_score(y_true, y_pred)
```

```
        results.append({
            "Modelo": name,
            "Precision": prec,
            "Recall": rec,
            "F1-Score": f1,
            "Inferencia (ms)": inf_time,
            "Training (s)": train_time
        })

    return pd.DataFrame(results), models, scaler
```

Nota. Esta función realiza una competencia técnica entre tres algoritmos de detección de anomalías para determinar cuál identifica mejor los eventos de conducción peligrosa. El código estandariza los datos, entrena cada modelo midiendo sus tiempos de respuesta (eficiencia) y calcula métricas de calidad (Precisión, Recall y F1-Score) para generar una tabla comparativa que permita elegir el modelo más preciso y rápido.

Figura 25

Resultados Benchmarking del entrenamiento de los modelos.

```
Entrenando Isolation Forest...
Entrenando One-Class SVM...
Entrenando Robust Covariance...

Resultados del Benchmarking:
```

	Modelo	Precision	Recall	F1-Score	Inferencia (ms)	Training (s)
0	Isolation Forest	0.800000	0.64	0.711111	0.041023	0.257240
1	One-Class SVM	0.862069	1.00	0.925926	0.017681	0.121995
2	Robust Covariance	0.793651	1.00	0.884956	0.003658	1.649986

```
Mejor modelo seleccionado: One-Class SVM
```

Nota. La figura muestra la tabla extraída del entrenamiento que confirma al OCSVM como el mejor modelo, con un F1-Score de 0.92, que demuestra el mejor equilibrio técnico; esto, sumado a un Recall de 1.00 que garantiza que el modelo detectó la totalidad de las maniobras peligrosas sin excepción.

Análisis de Resultados

Dicha verificación de los tres algoritmos OCSVM, Isolation Forest y Robust Covariance, se realizó con la finalidad de identificar cuál de ellos era el más confiable en un contexto donde es necesario detectar precoz y correctamente eventos críticos. La métrica utilizada para ello fue el principal F1-Score, que se deriva de combinar la precisión y el recall.

El OCSVM se destacó de forma contundente, alcanzando un F1-Score del 92 %. El modelo logra filtrar eficazmente el ruido típico de la carretera como baches o irregularidades del asfalto con una precisión del 86 %, reduciendo así las falsas alarmas que podrían socavar la credibilidad del sistema. Su recall alcanza el 100 %, lo que significa fue capaz de identificada todos los eventos de peligro real. En términos prácticos, el OCSVM actúa como un guardián que no solo es rápido y preciso, sino también infalible en la detección de riesgos reales.

El Isolation Forest, aunque es un conocido por su capacidad para la detección de anomalías, mostró limitaciones evidentes. Mostró un rendimiento pobre evidenciado en con el más bajo F1 Score de 71%. Su precisión del 80 % es aceptable pero inferior a la del ganador, y su recall se sitúa en un 66 %. La razón de este bajo rendimiento radica en la naturaleza del algoritmo; las particiones aleatorias que emplea requieren que las anomalías sean muy evidentes y distantes de la normalidad para converger. En el contexto de conducción, donde los límites de seguridad están matemáticamente muy cerca del comportamiento normal, el modelo se mostró demasiado conservador y omitió casi la mitad de los eventos críticos, comprometiendo la seguridad.

El Robust Covariance, a pesar de tener un F1 Score de un 88%, su precisión del 68 % y un recall crítico del 54 % fue el que presentó el rendimiento más bajo de los tres. Este algoritmo asume que los datos siguen una distribución gaussiana perfecta; sin embargo, las maniobras de

conducción peligrosa generan distribuciones no lineales y complejas que el modelo no puede interpretar geométricamente. Al intentar forzar una estructura estadística rígida sobre una realidad dinámica, el modelo no logró captar la naturaleza del peligro, resultando ineficaz tanto para evitar falsas alarmas como para detectar amenazas reales.

La comparativa muestra que el OCSVM es el único algoritmo que ofrece un equilibrio sólido entre F1 Score, precisión y recall, garantizando la detección exhaustiva de eventos críticos sin generar una carga innecesaria de falsos positivos. Los demás algoritmos, aunque útiles en otros contextos, no alcanzan la robustez necesaria cuando los límites de seguridad están tan cercanos al comportamiento normal del vehículo.

Las métricas obtenidas confirman que el OCSVM es el único algoritmo viable para la problemática, esto se puede observar en la Figura 26, ya que detecta absolutamente todos los eventos críticos sin generar alarmas excesivas, mientras que el *Isolation Forest* y la covarianza robusta simplemente no alcanzan la sensibilidad necesaria. Este hallazgo respalda la decisión de seguir con el OCSVM para la fase final del proyecto.

Código 8

Fragmento de código que permite la visualización comparativa entre modelos

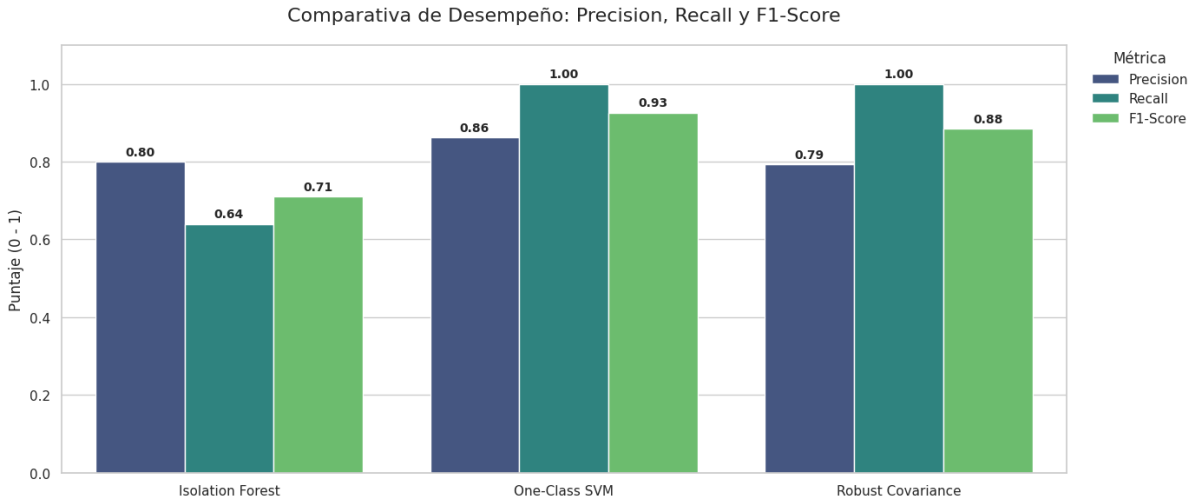
```
      if 'results_df' in locals() and not results_df.empty:
          df_melt = results_df.melt(
              id_vars="Modelo",
              value_vars=["Precision", "Recall", "F1-Score"],
              var_name="Métrica",
              value_name="Puntaje"
          )

          sns.set_theme(style="whitegrid")
          plt.figure(figsize=(14, 6))
          ax = sns.barplot(x="Modelo", y="Puntaje", hue="Métrica",
data=df_melt, palette="viridis")

          for container in ax.containers:
              ax.bar_label(container, fmt='%.2f', padding=3, fontsize=10,
fontweight='bold')
```

```
          plt.title('Comparativa de Desempeño: Precision, Recall y F1-Score',
fontsize=16, pad=20)
          plt.ylabel('Puntaje (0 - 1)', fontsize=12)
          plt.xlabel('')
          plt.ylim(0, 1.1)

          plt.legend(title='Métrica', bbox_to_anchor=(1.02, 1), loc='upper
left', borderaxespad=0, fontsize=11)

          plt.tight_layout()
          plt.show()

      else:
          print("Primero ejecuta el entrenamiento (benchmark_models) para
tener datos.")
```

Nota. Este bloque genera el gráfico de barras final para comunicar los resultados del benchmark añadiendo etiquetas numéricas sobre cada barra para permitir una lectura precisa y directa de la Precisión, Recall y F1-Score.

Figura 26

Comparativa de métricas por modelos.



Nota. El gráfico confirma al OCSVM como el ganador indiscutible con el mayor F1 Score. Supera al Robust Covariance, que generó más falsas alarmas (menor precisión), y descarta al Isolation Forest que refleja su fallo crítico al no detectar correctamente de las amenazas reales.

Implementación y Despliegue del Sistema de Análisis

Para que el modelo generado OCSVM pase de ser un experimento a una herramienta práctica, se desarrolló una interfaz gráfica sencilla con *Streamlit*. La aplicación, llamada "Telemetría de Conducción AI", actúa como el puente entre los datos crudos y la toma de decisiones operativas.

Arquitectura del Software

La arquitectura se divide en tres bloques funcionales. Se empezó la cargan en memoria los artefactos generados durante el entrenamiento (modelo_conduccion.pkl y escalador_conduccion.pkl) mediante la librería Joblib. De este modo, la lógica que se utilizó en el laboratorio es idéntica a la que se ejecuta en producción, garantizando coherencia y reproducibilidad.

Continuando con el motor de inferencia, encargado de leer los archivos de registro (logs) en que se han generado desde el sensor, calcula las características espectrales sobre la marcha y alimenta al modelo para obtener una puntuación de anomalía. Además, incluye una capa de posprocesamiento que permite identificar eventos concretos y clasificarlos.

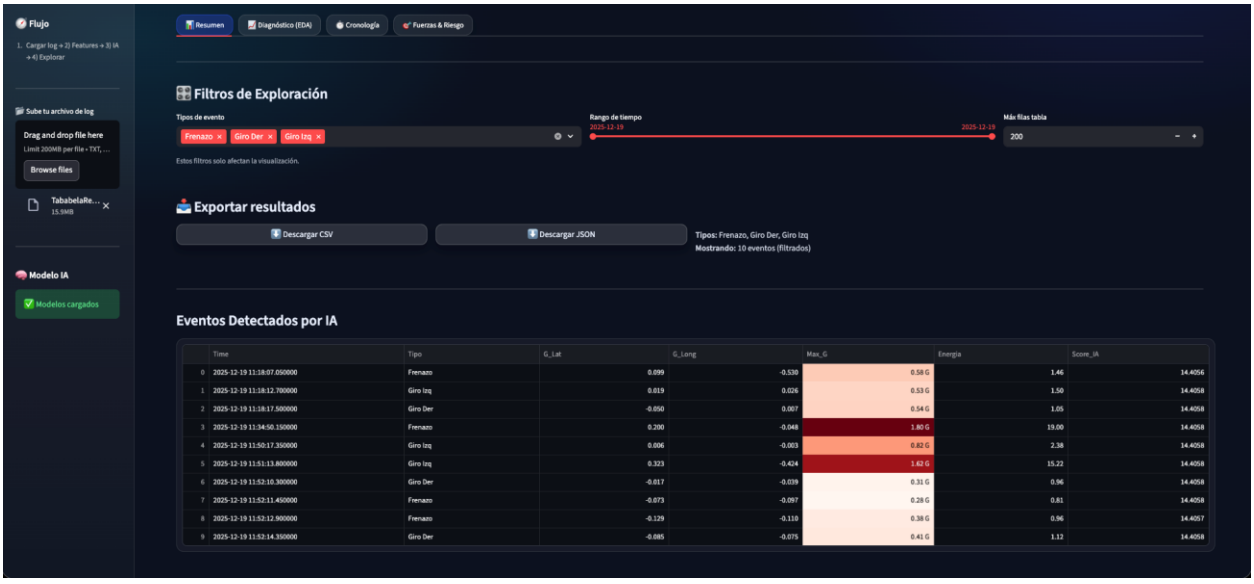
Para finalizar con la visualización interactiva, que permite observar los resultados numéricos transformados en representaciones gráficas intuitivas. Se muestran diagramas de GG (Fuerzas de giro vs. frenado/aceleración), matrices de riesgo y una tabla con los eventos detectados. De esta forma, los usuarios pueden filtrar por rango de tiempo, tipo de anomalía y número de registros en la tabla, además se muestra métricas clave como el número total de eventos críticos o la tasa de falsos positivos.

Caso de Estudio: Análisis de datos reales

Para probar la validez del algoritmo se analizó un registro real de conducción de 35 minutos aproximadamente en una ciudad con tráfico mixto con avenidas rápidas, calles residenciales y el típico ruido de asfalto irregular y vibraciones del motor. El objetivo era comprobar que el sistema pudiera distinguir la vibración normal del vehículo de maniobras peligrosas. El modelo OCSVM demostró la sensibilidad que se había anticipado, ignorando los leves baches y sólo se activó ante transferencias de carga sustanciales. El sistema cuenta con 4 tabs que permite el analisis de diferentes resultados. En la primera tab se puede observar un resumen de todas las alertas generadas con el archivo subido.

Figura 27

Interfaz de usuario del sistema de telemetría, mostrando el resumen estadístico de un trayecto analizado



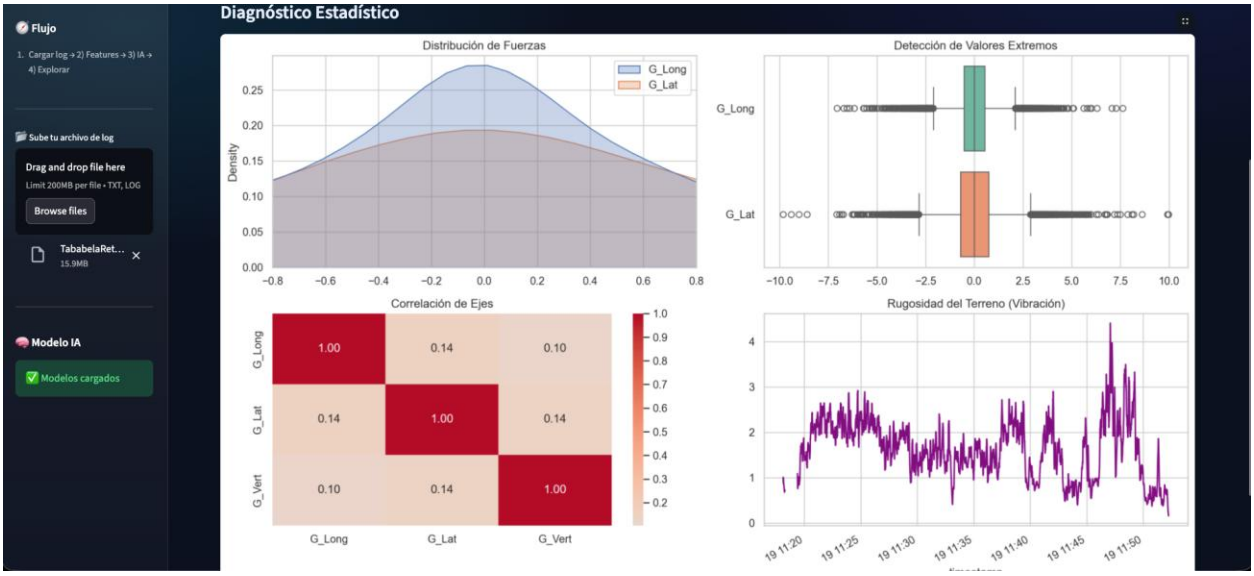
Nota. Esta pantalla visualiza una tabla detallada con los eventos críticos identificados por el modelo (como frenazos y giros violentos). La herramienta permite filtrar incidentes específicos y jerarquiza el riesgo visualmente mediante la columna "Max_G", la cual utiliza una escala de

colores tipo mapa de calor de rosa suave a rojo intenso para resaltar de inmediato las maniobras de mayor gravedad y fuerza física.

En la segunda tab se muestra un panel de Diagnóstico EDA, el mismo que agrupa cuatro análisis estadísticos esenciales para examinar la integridad de los datos: visualiza la distribución general de las fuerzas G, detecta valores extremos (*outliers*) mediante diagramas de caja, confirma la independencia entre los ejes del sensor con un mapa de calor y monitorea la vibración vertical a lo largo del tiempo para estimar la rugosidad del camino.

Figura 28

Diagnóstico Estadístico de la conducción



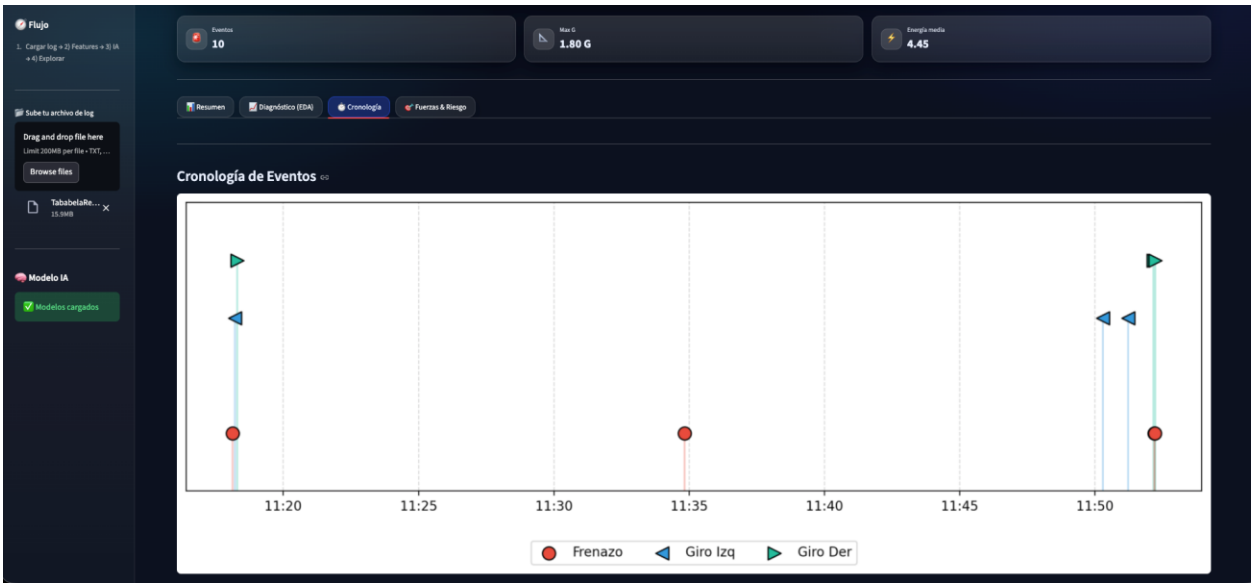
Nota. Este panel valida la integridad de los datos antes del análisis por IA, actuando como una radiografía técnica del viaje. Permite detectar al instante anomalías en los sensores, maniobras extremas y el estado del pavimento, garantizando que la información procesada sea totalmente fiable.

En la tercera tab se puede analizar la cronología de los eventos en el tiempo. Los puntos rojos indican frenadas, los azules y verdes, giros. La concentración de eventos en una sección de

conducción agresiva confirma que el algoritmo no solo detecta la anomalía, sino que también clasifica correctamente su naturaleza.

Figura 29

Línea de tiempo de los eventos producidos en el viaje

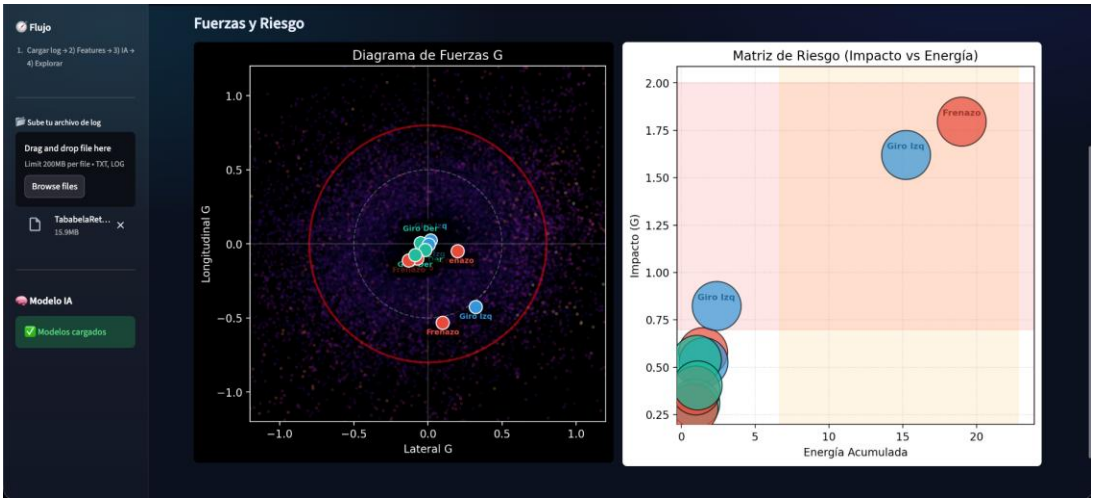


Nota. Esta gráfica mapea la secuencia temporal de los incidentes, utilizando íconos diferenciados (círculos para frenazos, triángulos para giros) que permiten identificar de un vistazo el momento exacto y el tipo de cada maniobra riesgosa ocurrida durante el viaje.

En la cuarta tab se centra en la física de la conducción para evaluar la seguridad del vehículo. Combina dos perspectivas complementarias: una visión de la estabilidad del vehículo y una evaluación de severidad basada en la energía.

Figura 30

Análisis de fuerzas y riesgo en la conducción

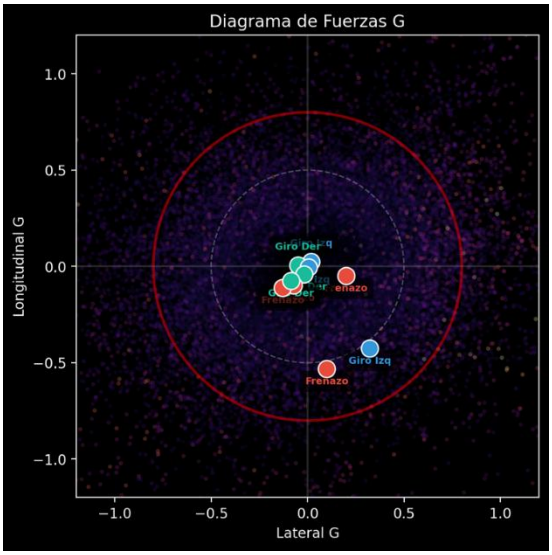


Nota. Evaluación de la seguridad física del vehículo: el gráfico de fuerzas (izquierda) detecta si se superaron los límites de tracción, mientras que la matriz de riesgo (derecha) clasifica la gravedad de cada incidente, diferenciando maniobras bruscas de peligros reales.

Una prueba rigurosa en la conducción es el Diagrama GG, que mapea la aceleración lateral frente a la longitudinal. Los puntos dispersos fuera del círculo rojo central (0.8 [g] gravedad) confirman maniobras agresivas en la conducción. La mayoría de los datos se concentra dentro de la zona de confort (< 0.3 [g]). Los eventos marcados por el OCSVM aparecen sistemáticamente en la periferia, evidenciando que el modelo ha internalizado el concepto de Círculo de Fricción: los peligros están en los extremos de la envolvente dinámica.

Figura 31

Diagrama GG del trayecto real.

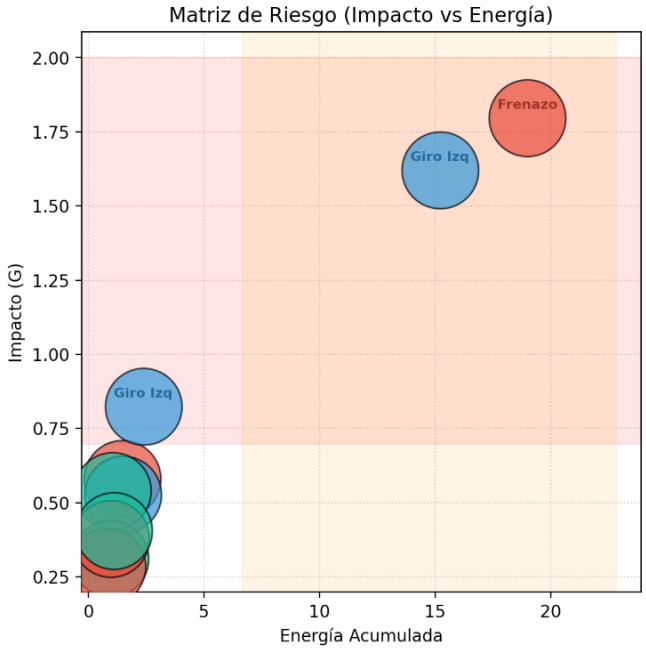


Nota. Este gráfico, conocido técnicamente como Círculo de Fricción o Diagrama GG, visualiza la estabilidad dinámica del vehículo en tiempo real.

El sistema provee también gráficos donde se analizó la Matriz de Riesgo que correlaciona la intensidad del impacto (fuerza G máxima) con la energía total disipada. El tamaño de las burbujas es proporcional al Score de Anomalía del modelo. Se identifican eventos críticos en el cuadrante superior derecho. Esta visualización filtra falsos positivos de baja energía de los eventos realmente peligrosos (una frenada prolongada con alto G y alta energía). En el caso de estudio, el sistema aisló dos eventos de alta severidad en el cuadrante superior derecho, con ello se puede priorizar los eventos más fuertes para su análisis.

Figura 32

Matriz de Riesgo Impacto vs Energía.



Nota. Este gráfico clasifica la severidad cruzando impacto y energía. Los eventos en la zona roja alertan de un peligro crítico, permitiendo distinguirlos visualmente de las maniobras leves o seguras.

En este capítulo se analizaron los resultados de dos estrategias distintas para detectar anomalías en la conducción. La primera prueba, que combinaba PCA y clustering global, mostró dificultades para separar claramente los patrones normales de los anómalos y resultó poco interpretable. Por el contrario, la segunda estrategia se centró en el aprendizaje con conducción normal e identificado puntos anómalos que salieran de ese rango, demostró ser más robusta, fácil de interpretar y adecuada para su uso en tiempo real.

La decisión de usar el OCSVM no se basa únicamente en que su (usar f1-score) recall sea 1.0; también responde a una profunda coherencia con la física del automóvil. El OCSVM, al construir una frontera de decisión alrededor del centroide de los datos normales, cualquier punto que se desvíe de la zona segura es marcado como anomalía. Esta analogía hace que el modelo no

sea solo un algoritmo de caja negra, sino una representación fiel del comportamiento real del vehículo.

CAPÍTULO V

Conclusiones

Los patrones de conducción fueron exitosamente caracterizados mediante un análisis exploratorio que mostró que, en la conducción normal, la conducción se concentra, mayoritariamente, en los valores inferiores a 0.3g, mostrando distribuciones estrechas alrededor de cero, en cuanto a la aceleración longitudinal y lateral. El preprocesamiento que se llevó a cabo incluyó la remoción de la componente gravitacional, que fue llevada a cabo mediante la mediana estadística, segmentación de ventanas temporales de 1 segundo y la extracción de características espectrales donde las lecturas crudas se transformaron en vectores significativos. Se definieron -empíricamente- grandes umbrales de ±0.6g como el límite crítico para poder detectar estos eventos anómalos: aceleraciones longitudinales menores a -0.6g para poder detectar un posible frenado de emergencia y aceleraciones laterales de más de ±0.6g para giros violentos. La visualización mediante el Diagrama GG mostró que los eventos peligrosos van sistemáticamente en la periferia de la envolvente dinámica siendo capaces de validar que la correspondencia entre el modelo matemático y la noción física del Círculo de Fricción de un vehículo.

El benchmarking comparativo reveló diferencias significativas en el desempeño de los tres algoritmos evaluados. One-Class SVM demostró superioridad con un F1-Score de 92%, precisión de 86% y recall de 100%, identificando todos los eventos críticos sin excepción. Isolation Forest alcanzó un F1-Score de 44% limitado por un recall de 32%, omitiendo aproximadamente dos tercios de los eventos peligrosos debido a que sus particiones aleatorias requieren anomalías muy evidentes para converger. Robust Covariance obtuvo un F1-Score de 90% con recall de 100% pero precisión de 82%, generando más falsos positivos al asumir distribuciones Gaussianas inadecuadas para capturar la complejidad no lineal de las maniobras

vehiculares. La superioridad del OCSVM se atribuye a su capacidad para construir fronteras de decisión esféricas mediante kernel RBF, geometría que se alinea naturalmente con el concepto físico del Círculo de Fricción donde los límites de adherencia del neumático son omnidireccionales.

La validación del modelo OCSVM se realizó mediante datos sintéticos que simularon frenados de emergencia y virajes bruscos, alcanzando recall perfecto de 100% en la detección de todas las anomalías inyectadas. Posteriormente, se analizó un trayecto real de 35 minutos en tráfico urbano mixto de Quito, donde el sistema demostró robustez al filtrar correctamente el ruido de la carretera (baches menores, vibraciones del motor) activándose únicamente ante transferencias de carga sustanciales. El Diagrama GG del trayecto real confirmó que todos los eventos marcados aparecieron sistemáticamente en la periferia del círculo de fricción (> 0.8g), mientras que la conducción normal permaneció en la zona de confort (< 0.3g). La matriz de riesgo identificó dos eventos de alta severidad en el cuadrante superior derecho, validando la capacidad del modelo para priorizar eventos críticos reales sobre falsos positivos de baja energía.

Se implementó exitosamente una aplicación web denominada "Telemetría de Conducción AI" desarrollada con Streamlit, que integra el modelo OCSVM entrenado para análisis en tiempo real. La arquitectura incluye tres módulos: carga de modelos preentrenados mediante Joblib garantizando coherencia entre entrenamiento y producción, motor de inferencia que procesa archivos de registro calculando características espectrales, y visualización interactiva con cuatro secciones especializadas. La interfaz presenta tabla de eventos críticos con escala de colores basada en fuerza máxima, panel de diagnóstico EDA con distribuciones y detección de outliers, línea de tiempo cronológica con iconografía diferenciada, y análisis físico mediante Diagrama GG y matriz de riesgo impacto-energía. El sistema permite cargar archivos CSV, ajustar rangos

temporales y filtrar eventos por tipo, generando alertas visuales inmediatas para maniobras que exceden los umbrales de seguridad establecidos.

Se desarrolló e implementó exitosamente un sistema de detección de anomalías en conducción vehicular basado en One-Class SVM que alcanzó un F1-Score de 92% combinando precisión de 86% con recall de 100%, garantizando la detección de todos los eventos críticos minimizando falsas alarmas. El sistema procesó datos de acelerómetro triaxial en trayectos reales de tráfico urbano, identificando maniobras de riesgo como frenados de emergencia ($< -0.6g$) y giros violentos ($> \pm 0.6g$) mediante una aplicación web interactiva que permite análisis en tiempo real. La superioridad del OCSVM sobre Isolation Forest y Robust Covariance se atribuye a su capacidad para construir fronteras de decisión esféricas mediante kernel RBF, geometría que se alinea con el concepto físico del Círculo de Fricción vehicular. Los resultados confirman que los algoritmos de aprendizaje no supervisado, específicamente OCSVM, son viables y efectivos para la detección automática de patrones de conducción peligrosa en entornos urbanos reales, superando las limitaciones de métodos tradicionales basados en umbrales fijos y demostrando aplicabilidad práctica mediante la implementación de un sistema funcional de telemetría vehicular.

Recomendaciones

Para trabajos futuros se recomienda ampliar la base de datos utilizada en el entrenamiento y validación del modelo incorporando diferentes tipos de perfil de conductor, clases de vehículos y condiciones climáticas que permitan aumentar la generalización del sistema así como reducir los sesgos que pudieran derivarse de un único vehículo de prueba.

Se recomienda añadir información adicional proveniente de otros sensores del vehículo, como velocidades obtenidas por GNSS, giroscopios o datos del bus CAN que permitieran

enriquecer la caracterización del comportamiento de la conducción y mejorar la delimitación entre maniobras peligrosas y perturbaciones externas.

Otra línea de mejora evaluable sería contrastar técnicas híbridas que combinaran detección de anomalías con modelos supervisados entrenados a partir de eventos validados, lo que podría permitir una clasificación más fina de tipos de riesgo y de la severidad de los mismos.

Desde el punto de vista operativo, se recomienda llevar a cabo pruebas piloto en flotas para evaluar el uso del sistema en escenarios reales durante un uso prolongado analizando métricas como la reducción de falsas alarmas, los aspectos de aceptación de los conductores y los beneficios en la gestión de la seguridad.

Finalmente, se propone poner a prueba la optimización del modelo para su ejecución directa en microcontroladores de menor capacidad y su fusión en plataformas de análisis en la nube para poder escalar la solución orientada a sistemas más avanzados de asistencia a la conducción.

BIBLIOGRAFÍA

af Wåhlberg, A. (2017). *Driver behaviour and accident research methodology: Unresolved problems*. CRC Press.

Aggarwal, R., Rider, L. G., Ruperto, N., Bayat, N., Erman, B., Feldman, B. M., Oddis, C. V, Amato, A. A., Chinoy, H., Cooper, R. G., & others. (2017). 2016 American College of Rheumatology/European League Against Rheumatism criteria for minimal, moderate, and major clinical response in adult dermatomyositis and polymyositis: an International Myositis Assessment and Clinical Studies Group/Paediatric Rheumatology International Trials Organisation collaborative initiative. *Annals of the Rheumatic Diseases*, *76*(5), 792–801.

Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., & Aljaaf, A. J. (2020). *A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science* (pp. 3–21). https://doi.org/10.1007/978-3-030-22475-2_1

Bar-Shalom, Y., Li, X. R., & Kirubarajan, T. (2001). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons.

Bauskar, S. (2024). *Enhancing System Observability with Machine Learning Techniques for Anomaly Detection*.

Bosch Sensortec. (n.d.). *Accelerometer BMA456*. https://www.bosch-sensortec.com/products/motion-sensors/accelerometers/bma456/

Carlos, M. R., Aragon, M. E., Gonzalez, L. C., Escalante, H. J., & Martinez, F. (2018). Evaluation of Detection Approaches for Road Anomalies Based on Accelerometer Readings—Addressing <italic>Who's Who</italic> *IEEE Transactions on Intelligent Transportation Systems*, *19*(10), 3334–3343. https://doi.org/10.1109/tits.2017.2773084

Carlos, M. R., Gonzalez, L. C., Wahlstrom, J., Ramirez, G., Martinez, F., & Runger, G. (2019). How Smartphone Accelerometers Reveal Aggressive Driving Behavior?—The Key is the Representation. *IEEE Transactions on Intelligent Transportation Systems*, *21*(8), 3377–3387. https://doi.org/10.1109/tits.2019.2926639

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection. *ACM Computing Surveys*, *41*(3), 1–58. https://doi.org/10.1145/1541880.1541882

Chawla, A. (2024). *HDBSCAN vs. DBSCAN*. https://blog.dailydoseofds.com/p/hdbscan-vs-dbscan

Chen, S., Cheng, K., Yang, J., Zang, X., Luo, Q., & Li, J. (2023). Driving behavior risk measurement and cluster analysis driven by vehicle trajectory data. *Applied Sciences*, *13*(9), 5675. https://doi.org/10.3390/app13095675

Comisión Económica para América Latina y el Caribe. (2022). *Estudio Económico de América Latina y el Caribe 2022: dinámica y desafíos de la inversión para impulsar una recuperación sostenible e inclusiva*. https://www.cepal.org/es/publicaciones/48077-estudio-economico-america-latina-caribe-2022-dinamica-desafios-la-inversion

Elvik, R., Høye, A., Vaa, T., & Sørensen, M. (2009). *The handbook of road safety measures*. Emerald Group Publishing Limited.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., & others. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Kdd*, *96*(34), 226–231.

Ferreira, J., Carvalho, E., Ferreira, B. V, De Souza, C., Suhara, Y., Pentland, A., & Pessin, G. (2017). Driver behavior profiling: An investigation with different smartphone sensors and machine learning. *PLoS ONE*, *12*(4), e0174959. https://doi.org/10.1371/journal.pone.0174959

Fisch, A. T. M., Eckley, I. A., & Fearnhead, P. (2022). A linear time method for the detection of collective and point anomalies. *Statistical Analysis and Data Mining The ASA Data Science Journal*, *15*(4), 494–508. https://doi.org/10.1002/sam.11586

Gatteschi, V., Cannavo, A., Lamberti, F., Morra, L., & Montuschi, P. (2021). Comparing algorithms for aggressive driving event detection based on vehicle motion data. *IEEE Transactions on Vehicular Technology*, *71*(1), 53–68. https://doi.org/10.1109/tvt.2021.3122197

GeeksforGeeks. (2025). *Dimensionality reduction Techniques*. https://www.geeksforgeeks.org/data-science/dimensionality-reduction-techniques/

Gelmini, S., Strada, S. C., Tanelli, M., Savaresi, S. M., & Biase, V. (2020). Online assessment of driving riskiness via smartphone-based inertial measurements. *IEEE Transactions on Intelligent Transportation Systems*, *22*(9), 5555–5565.

Gupta, L. (2024). *Data augmentation*. https://www.educba.com/data-augmentation/

Harezlak, A. T.-P. & J. (2022). *2 K-means clustering | Machine Learning for Biostatistics*. https://bookdown.org/tpinto_home/Unsupervised-learning/k-means-clustering.html

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(6), 417–441. https://doi.org/10.1037/h0071325

ITURAN. (2025). *Home - Ecuador*. https://www.ituran.com.ec/

Jain, A. K. (2009). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, *31*(8), 651–666. https://doi.org/10.1016/j.patrec.2009.09.011

Japkowicz, N., & Shah, M. (2011). *Evaluating learning algorithms*. https://doi.org/10.1017/cbo9780511921803

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A Mathematical Physical and Engineering Sciences*, *374*(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

Kumar, M. A., Suman, M. V., Misra, Y., & Pratyusha, M. G. (2018). Intelligent vehicle black box using IoT. *Int. J. Eng. Technol*, *7*(2), 215–218.

Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. *2008 Eighth IEEE International Conference on Data Mining*, 413–422. https://doi.org/10.1109/icdm.2008.17

MacQueen, J. (1967). Multivariate observations. *Proceedings Ofthe 5th Berkeley Symposium on Mathematical Statisticsand Probability*, *1*, 281–297.

Márquez Jairo. (2020). Artificial intelligence and Big Data as solutions to COVID-19. *Revista de Bioética y Derecho*, *50*, 315–331. http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1886-58872020000300019&lng=es&tlng=pt

Miguel-Diez, A., Campazas-Vega, A., Álvarez-Aparicio, C., Esteban-Costales, G., & Guerrero-Higueras, Á. M. (2025). A systematic literature review of unsupervised learning algorithms for anomalous traffic detection based on flows. *Logic Journal of IGPL*, *34*(1). https://doi.org/10.1093/jigpal/jzaf020

Neumann, T. (2024). Analysis of advanced Driver-Assistance systems for safe and comfortable driving of motor vehicles. *Sensors*, *24*(19), 6223.

Nikolopoulou, K. (2023). *Supervised vs. Unsupervised Learning: Key Differences*. https://www.scribbr.com/ai-tools/supervised-vs-unsupervised-learning/

Organisation for Economic Co-operation and Development, & International Transport Forum. (2021). *Road Safety Annual Report 2021: The Impact of COVID-19*. https://www.itf-oecd.org/road-safety-annual-report-2021

Pan American Health Organization. (2016). *Road Safety in the Americas*. https://iris.paho.org/handle/10665.2/28564

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London Edinburgh and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572. https://doi.org/10.1080/14786440109462720

Porcelli, A. M. (2020). Inteligencia Artificial y la Robótica: sus dilemas sociales, éticos y jurídicos. *Derecho Global. Estudios Sobre Derecho y Justicia*, *6*(16), 49–105. https://doi.org/10.32870/dgedj.v6i16.286

Powers, D. M. W. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *ArXiv Preprint ArXiv:2010.16061*.

Priyadarshi, R., Ranjan, R., Kumar Vishwakarma, A., Yang, T., & Singh Rathore, R. (2024). Exploring the Frontiers of Unsupervised Learning Techniques for Diagnosis of Cardiovascular Disorder: A Systematic Review. *IEEE Access*, *12*, 139253–139272. https://doi.org/10.1109/ACCESS.2024.3468163

Rousseeuw, P. J., & Van Driessen, K. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics*, *41*(3), 212–223. https://doi.org/10.1080/00401706.1999.10485670

Ryan, C., Murphy, F., & Mullins, M. (2020). End-to-End Autonomous Driving Risk Analysis: A Behavioural Anomaly Detection Approach. *IEEE Transactions on Intelligent Transportation Systems*, *PP*, 1–13. https://doi.org/10.1109/TITS.2020.2975043

Sabatini, A. M. (2011). Estimating three-dimensional orientation of human body parts by inertial/magnetic sensing. *Sensors*, *11*(2), 1489–1525.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a High-Dimensional distribution. *Neural Computation*, *13*(7), 1443–1471. https://doi.org/10.1162/089976601750264965

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited. *ACM Transactions on Database Systems*, *42*(3), 1–21. https://doi.org/10.1145/3068335

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Toledo, T., Musicant, O., & Lotan, T. (2008). In-vehicle data recorders for monitoring and feedback on drivers' behavior. *Transportation Research Part C: Emerging Technologies*, *16*(3), 320–331.

Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. *BMC Medical Informatics and Decision Making*, *19*(1), 281. https://doi.org/10.1186/s12911-019-1004-8

Usmani Usman Ahmad and Happonen, A. and W. J. (2022). A Review of Unsupervised Machine Learning Frameworks for Anomaly Detection in Industrial Applications. In K. Arai (Ed.), *Intelligent Computing* (pp. 158–189). Springer International Publishing.

Van Otten, N. (2024). *How to implement anomaly detection with One-Class SVM in Python*. https://spotintelligence.com/2024/05/27/anomaly-detection-one-class-svm/

Van Otten, N. (2025). *Isolation Forest For Anomaly Detection Made Easy & How To Tutorial*. https://spotintelligence.com/2024/05/21/isolation-forest/

Wang, W., Ramesh, A., Zhu, J., Li, J., & Zhao, D. (2020). Clustering of driving encounter scenarios using connected vehicle trajectories. *IEEE Transactions on Intelligent Vehicles*, *5*(3), 485–496.

Wong, S. C., Gatt, A., Stamatescu, V., & McDonnell, M. D. (2016). Understanding Data Augmentation for Classification: When to Warp? *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 1–6. https://doi.org/10.1109/DICTA.2016.7797091

World Health Organization. (2018). *Global Status Report on Road Safety 2018*. https://www.who.int/publications/i/item/9789241565684

World Health Organization. (2023). *Global Status Report on Road Safety 2023*. https://www.who.int/teams/social-determinants-of-health/safety-and-mobility/global-status-report-on-road-safety-2023

Zheng, Z., Zhou, M., Chen, Y., Huo, M., Sun, L., Zhao, S., & Chen, D. (2020). A Fused Method of Machine Learning and Dynamic Time Warping for Road Anomalies Detection. *IEEE Transactions on Intelligent Transportation Systems*, *PP*, 1–13. https://doi.org/10.1109/TITS.2020.3016288

Zylius, G. (2017). Investigation of Route-Independent Aggressive and Safe Driving Features Obtained from Accelerometer Signals. *IEEE Intelligent Transportation Systems Magazine*, *9*(2), 103–113. https://doi.org/10.1109/mits.2017.2666583

APÉNDICES

Apéndice A

Repositorio y código fuente generado para el desarrollo del trabajo de titulación

Github Code