



INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

TESIS

Clasificación de patrones de conducción para su
implementación en gemelos digitales

QUE PARA OBTENER EL GRADO DE
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN

PRESENTA:

Ing. Isay Gerardo Zamora de la Rosa

DIRECTORES DE TESIS:

Dr. Amadeo José Argüelles Cruz
Dra. Ilse Cervantes Camacho



Ciudad de México

Junio 2025



INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REGISTRO DE TEMA DE TESIS Y DESIGNACIÓN DE DIRECTOR DE TESIS

Ciudad de México, a de del

El Colegio de Profesores de Posgrado del en su Sesión
(Unidad Académica)

No celebrada el día del mes de , conoció la solicitud presentada por el (la) alumno (a):

Apellido Paterno:	ZAMORA	Apellido Materno:	DE LA ROSA	Nombre (s):	ISAY GERARDO
-------------------	--------	-------------------	------------	-------------	--------------

Número de registro:

del Programa Académico de Posgrado:

Referente al registro de su tema de tesis; acordando lo siguiente:

1.- Se designa al aspirante el tema de tesis titulado:

" Clasificación de patrones de conducción para su implementación en gemelos digitales "

Objetivo general del trabajo de tesis:

Realizar la clasificación eficiente y precisa de patrones de conducción, con la finalidad de identificar las conductas al volante que sirvan para la generación de gemelos digitales,

2.- Se designa como Directores de Tesis a los profesores:

Director: 2° Director:
No aplica:

3.- El Trabajo de investigación base para el desarrollo de la tesis será elaborado por el alumno en:

que cuenta con los recursos e infraestructura necesarios.

4.- El interesado deberá asistir a los seminarios desarrollados en el área de adscripción del trabajo desde la fecha en que se suscribe la presente, hasta la aprobación de la versión completa de la tesis por parte de la Comisión Revisora correspondiente.

Director(a) de Tesis

Dr. Amadeo José Argüelles Cruz

Aspirante

C. Isay Gerardo Zamora De la Rosa

2° Director de Tesis

Dra. Ilse Cervantes Camacho

Presidente del Colegio

Dr. Francisco Hiram Salvo Castro





INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

ACTA DE REVISIÓN DE TESIS

En la Ciudad de siendo las horas del día del mes de del se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Posgrado de: para examinar la tesis titulada:

Clasificación de patrones de conducción para su implementación en gemelos digitales del (la) alumno (a):

Apellido Paterno:	ZAMORA	Apellido Materno:	DE LA ROSA	Nombre (s):	ISAY GERARDO
-------------------	--------	-------------------	------------	-------------	--------------

Número de registro:

Aspirante del Programa Académico de Posgrado:

Una vez que se realizó un análisis de similitud de texto, utilizando el software antiplagio, se encontró que el trabajo de tesis tiene 05 % de similitud. **Se adjunta reporte de software utilizado.**

Después que esta Comisión revisó exhaustivamente el contenido, estructura, intención y ubicación de los textos de la tesis identificados como coincidentes con otros documentos, concluyó que en el presente trabajo SI NO SE CONSTITUYE UN POSIBLE PLAGIO.

JUSTIFICACIÓN DE LA CONCLUSIÓN: *(Por ejemplo, el % de similitud se localiza en metodologías adecuadamente referidas a fuentes original)*

El porcentaje de similitud identificado corresponde a fracciones de texto del estado del arte debidamente referenciadas.

****Es responsabilidad del alumno como autor de la tesis la verificación antiplagio, y del Director o Directores de tesis el análisis del % de similitud para establecer el riesgo o la existencia de un posible plagio.**

Finalmente y posterior a la lectura, revisión individual, así como el análisis e intercambio de opiniones, los miembros de la Comisión manifestaron **APROBAR** **SUSPENDER** **NO APROBAR** la tesis por **UNANIMIDAD** o **MAYORÍA** en virtud de los motivos siguientes:

Cumple con los requisitos normativos y con una buena presentación del trabajo.

COMISIÓN REVISORA DE TESIS

Dr. Amadeo José Argüelles Cruz
Director de Tesis

Dr. Marco Antonio Moreno Ibarra

Dra. Yenny Villuendas Reyes

Dra. Ilse Cervantes Carnacho
2º Director de Tesis

Dr. Antonio Alarcón Paredes

Dra. Ana María Magdalena Saldaña Pérez



ESTADO MEXICANO
INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN
EN COMPUTACIÓN
DIRECCIÓN
INFORMÁTICA

Dr. Juan Humberto Sossa Azuela
PRESIDENTE DEL COLEGIO DE
PROFESORES



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA DE AUTORIZACIÓN DE USO DE OBRA PARA DIFUSIÓN

En la Ciudad de México el día 27 del mes de Junio del año 2025, el que suscribe Isay Gerardo Zamora de la Rosa alumno del programa Maestría en Ciencias de la Computación con número de registro B230531, adscrito a Centro de Investigación en Computación manifiesta que es autor intelectual del presente trabajo de tesis bajo la dirección de Dr. Amadeo José Argüelles Cruz y Dra. Ilse Cervantes Camacho y cede los derechos del trabajo intitulado Clasificación de patrones de conducción para su implementación en gemelos digitales, al Instituto Politécnico Nacional, para su difusión con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expresado del autor y/o directores. Este puede ser obtenido escribiendo a las siguientes direcciones de correo isayg-zamora@hotmail.com, jamadeo@cic.ipn.mx, icervantesc@ipn.mx. Si el permiso se otorga, al usuario deberá dar agradecimiento correspondiente y citar la fuente de este.

Isay Gerardo Zamora de la Rosa

Nombre completo y firma autográfica del
estudiante

Resumen

Este trabajo desarrolla una metodología para la clasificación de patrones de conducción, que incluye el aprendizaje automático, profundo y no supervisado, orientada a su implementación en gemelos digitales vehiculares.

La metodología implementada utiliza datos vehiculares recopilados de 10 sujetos (cinco hombres y cinco mujeres) para el entorno en pista y 5 sujetos (tres hombre y dos mujeres) para el ambiente urbano, se emplearon equipos especializados de adquisición de datos instalados en un vehículo Kia Rio 2018. Se registraron variables como velocidad, revoluciones por minuto del motor (RPM), ángulo de giro del volante, entre otras.

Se implementaron y evaluaron múltiples algoritmos de clasificación, incluyendo métodos tradicionales de aprendizaje automático (K-Nearest Neighbors) y técnicas de aprendizaje profundo (*Long Short-Term Memory - LSTM* y *Gated Recurrent Unit - GRU*). La evaluación experimental se realizó mediante diferentes técnicas de validación, que incluyen la validación cruzada *K-fold* ($K=5$), *Hold-out* (80,20) y *Walk-forward cross validation* para mantener el orden de tiempo, utilizando métricas de precisión, recall y F1-score.

Los resultados obtenidos muestran la efectividad de las técnicas de aprendizaje profundo para la clasificación de maniobras de conducción. Los algoritmos LSTM y GRU mostraron un rendimiento superior en la identificación de patrones temporales en los datos vehiculares. El análisis mediante matrices de confusión reveló que estos modelos pueden distinguir efectivamente entre diferentes tipos de maniobras, proporcionando una base sólida para la implementación en sistemas de gemelos digitales.

Adicionalmente, se desarrolló un análisis no supervisado utilizando técnicas de agrupamiento (*K-means* y agrupamiento jerárquico) para identificar grupos en los datos de conducción, complementado con análisis de componentes principales (PCA) para la reducción de dimensionalidad. Estos enfoques permitieron identificar patrones latentes en el comportamiento de conducción que no son evidentes mediante análisis supervisado.

La investigación contribuye al campo de los gemelos digitales vehiculares proporcionando una metodología validada para la clasificación de patrones de conducción que puede ser integrada en sistemas de monitoreo. Los modelos desarrollados tienen aplicaciones potenciales en sistemas de asistencia al conductor, análisis de seguridad vial, optimización de rutas y desarrollo de vehículos autónomos.

Abstract

This work develops a methodology for the classification of driving patterns, which includes machine learning, deep learning, and unsupervised learning, aimed at its implementation in vehicular digital twins.

The implemented methodology uses vehicular data collected from 10 subjects (five men and five women) for the test track environment, and 5 subjects (three men and two women) for the urban environment. Specialized data acquisition equipment was installed in a 2018 Kia Rio vehicle. Variables such as speed, engine revolutions per minute (RPM), steering wheel angle, among others, were recorded.

Multiple classification algorithms were implemented and evaluated, including traditional machine learning methods (K-Nearest Neighbors) and deep learning techniques (Long Short-Term Memory - LSTM and Gated Recurrent Unit - GRU). The experimental evaluation was carried out using different validation techniques, including K-fold cross-validation (K=5), hold-out (80/20), and walk-forward cross-validation to preserve temporal order, using precision, recall, and F1-score metrics.

The results obtained demonstrate the effectiveness of deep learning techniques for driving maneuver classification. The LSTM and GRU algorithms showed superior performance in identifying temporal patterns in vehicular data. Analysis through confusion matrices revealed that these models can effectively distinguish between different types of maneuvers, providing a solid foundation for implementation in digital twin systems.

Additionally, an unsupervised analysis was developed using clustering techniques (K-means and hierarchical clustering) to identify groups in the driving data, complemented by principal component analysis (PCA) for dimensionality reduction. These approaches allowed the identification of latent patterns in driving behavior that are not evident through supervised analysis.

This research contributes to the field of vehicular digital twins by providing a validated methodology for the classification of driving patterns that can be integrated into monitoring systems. The developed models have potential applications in driver assistance systems, road safety analysis, route optimization, and the development of autonomous vehicles.

Agradecimientos

Quiero expresar mi más sincero reconocimiento y gratitud a todas las personas e instituciones que hicieron posible la realización de esta tesis.

A mis padres y mi familia, por su amor incondicional, su apoyo emocional constante y por creer en mí durante cada etapa de este proceso. Su confianza y sacrificio constituyen el pilar de este logro académico.

A mis asesores, por su guía experta, su paciencia y su retroalimentación constructiva. Su dedicación y consejo han sido fundamentales para el desarrollo de esta investigación y mi crecimiento profesional.

Al CONACYT (Consejo Nacional de Ciencia y Tecnología), por el respaldo económico durante mis estudios de posgrado. Su apoyo institucional ha sido esencial para dedicarme plenamente a mi formación académica y llevar a cabo este proyecto.

Agradezco al Laboratorio Nacional CONAHCYT en Electromovilidad Inteligente (LAN-CEI) por fomentar un entorno que fortalece la investigación en el área de movilidad inteligente.

Índice general

Resumen	IV
Abstract	V
Agradecimientos	VI
1 Introducción	1
1.1 Definición del problema	1
1.2 Próposito del estudio	2
1.3 Preguntas de investigación	2
1.4 Importancia del estudio	3
1.4.1 Marco normativo	4
1.5 Objetivos	6
1.5.1 Objetivo General	6
1.5.2 Objetivos Específicos	6
1.6 Hipótesis	6
1.7 Contribuciones	6
1.8 Limitaciones del estudio	7
1.9 Organización de la tesis	7
2 Estado del arte	9
2.1 Clasificación de estilos de conducción	12
2.1.1 Estudios y revisiones sistemáticas	13
2.1.2 Análisis del comportamiento del conductor utilizando OBD-II	13
2.1.3 Arquitecturas avanzadas de Aprendizaje Profundo	15
2.1.4 Análisis del comportamiento de conducción usando diversas técnicas de recopilación de datos.	16
2.1.5 Técnicas de agrupamiento y clasificación no supervisada	17
2.1.6 Métodos de comité para la clasificación de patrones de conducción	18
2.1.7 Sistemas integrados y evaluación comparativa de modelos	19
2.2 Gemelos digitales	20
2.2.1 Definición y aplicaciones generales	20
2.2.2 Revisiones sistemáticas en la industria automotriz	21
2.2.3 Análisis de riesgos y seguridad vial	24
2.2.4 Mantenimiento predictivo y desarrollo de sistemas	25
2.2.5 Implementaciones de gemelos digitales en vehículos motorizados	27

2.3	Conclusiones	28
3	Materiales y Metodología	30
3.1	Recolección y banco de datos	30
3.1.1	Lugar del experimento	30
3.1.2	Equipos para la Adquisición de los Datos	30
3.2	Tratamiento de la información	32
3.2.1	Lectura y tratamiento de la información aportada por el Transceptor PCAN-USB	32
3.2.2	Lectura y tratamiento de la información aportada por el VBOX3i y el IMU03	32
3.2.3	Compatibilización de Datos	32
3.2.4	Variables recopiladas	33
3.2.5	Vehículo	33
3.2.6	Sujetos	34
3.2.7	Condiciones del experimento en pista	34
3.2.8	Condiciones del experimento en Ciudad	36
3.2.8.1	Características del experimento en ciudad	37
3.2.8.2	Trayectoria en ciudad	37
3.2.8.3	Clasificación de maniobras urbanas	38
3.3	Metodología de análisis	39
3.3.1	Métodos de validación	40
3.3.1.1	<i>Hold-Out Validation</i>	40
3.3.1.2	<i>K-Fold Cross Validation</i>	40
3.3.1.3	<i>Walk-Forward Validation</i>	40
3.3.2	Análisis Supervisado	40
3.3.2.1	K-NN	41
3.3.2.1.1	Funcionamiento	41
3.3.2.1.2	Pseudocódigo	41
3.3.2.2	Bosques Aleatorios (Random Forest)	41
3.3.2.2.1	Funcionamiento	42
3.3.2.2.2	Pseudocódigo	43
3.3.2.3	Red neuronal multicapa (Multilayer Perceptron)	44
3.3.2.3.1	Funcionamiento	44
3.3.2.3.2	Pseudocódigo	45
3.3.3	Aprendizaje Profundo	46
3.3.3.1	LSTM	47
3.3.3.1.1	Funcionamiento	47
3.3.3.1.2	Pseudocódigo	47
3.3.3.2	GRU	48
3.3.3.2.1	Funcionamiento	49
3.3.3.2.2	Pseudocódigo	49
3.3.4	Análisis no supervisado	50
3.3.4.1	Análisis de Componentes Principales (PCA)	51
3.3.4.1.1	Funcionamiento	51

3.3.4.1.2	Pseudocódigo	51
3.3.4.2	Jerárquico	51
3.3.4.2.1	Enfoque Aglomerativo (<i>Bottom-Up</i>)	52
3.3.4.2.2	Enfoque Divisivo (<i>Top-Down</i>)	52
3.3.4.2.3	Pasos para realizar un agrupamiento jerárquico aglomerativo	53
3.3.4.2.4	Dendrograma: El árbol de relaciones	53
3.3.4.2.5	Pseudocódigo	54
3.3.4.3	K-means	54
3.3.4.3.1	Funcionamiento	54
3.3.4.3.2	Pseudocódigo	55
3.3.5	Evaluación	55
3.3.5.1	Índice de Calinski-Harabasz	56
3.3.5.2	Índice de la silueta (<i>Silhouette</i>)	57
3.3.5.3	Índice de Davies-Bouldin	57
4	Hallazgos en el desarrollo del de clasificador	58
4.1	Desarrollo del clasificador de maniobras	59
4.1.1	Configuración experimental para el desarrollo del clasificador	59
4.1.1.1	Estrategias de validación	60
4.1.1.2	Configuración de los hiperparámetros	60
4.1.2	Evaluación experimental y selección de algoritmos	61
4.1.2.1	K-Nearest Neighbors (k=3)	61
4.1.2.2	Long Short-Term Memory (LSTM)	63
4.1.2.3	Gated Recurrent Unit (GRU)	64
4.1.2.4	Análisis comparativo del rendimiento	65
4.2	Empleo del clasificador de maniobras para identificar los estilos de conducción en arroyo vehicular	65
4.2.1	Procedimiento con los datos de calle	65
4.2.2	Resultados	66
4.3	Análisis de agrupamiento para la identificación de estilos de conducción	69
4.3.1	Análisis de Componentes Principales (PCA)	70
4.3.2	Estrategias de Agrupamiento Implementadas	71
4.4	Determinación del número óptimo de estilos de conducción en pista	72
4.5	Estilo Agresivo: Riesgo Elevado	74
4.6	Estilo Tranquilo: Riesgo bajo	74
4.7	Estilo Equilibrado: Riesgo moderado	75
4.8	Clasificación de Estilos de Conducción	75
4.9	Distribución General de Estilos	83
4.9.1	Análisis Detallado por Variable y Maniobra	84
4.9.1.1	Variables de Aceleración (X_Accel, Y_Accel, Z_Accel)	84
4.9.1.2	Control de Dirección (Steering Angle)	84
4.9.1.3	Velocidades Angulares (YawRate, PitchRate, RollRate)	84
4.9.2	Análisis Comparativo por Maniobras	85
4.9.2.1	Maniobra de Recta	85
4.9.2.2	Curvas (Con y Sin Peralte)	85

4.9.2.3	Slalom	85
4.9.2.4	Rotonda	86
4.9.3	Clasificación Final de Estilos de Conducción por Sujeto	86
4.10	Clasificación de estilos de conducción para el arroyo vehicular	86
4.10.0.1	Resultados del Agrupamiento Jerárquico	87
4.10.0.2	Resultados del Algoritmo K-means	87
4.10.0.3	Análisis Comparativo y Decisión Final	88
4.10.1	Análisis Detallado por Variable y Maniobra	92
4.10.1.1	Variables de Aceleración (X_Accel, Y_Accel, Z_Accel)	92
4.10.1.2	Control de Dirección (Steering Angle)	93
4.10.1.3	Velocidades Angulares (YawRate, PitchRate, RollRate)	93
4.10.2	Análisis Comparativo por Maniobras	93
4.10.2.1	Maniobra de Recta	93
4.10.2.2	Vuelta Derecha	94
4.10.2.3	Vuelta Izquierda	94
4.10.3	Clasificación Final de Estilos de Conducción por Sujeto en el arroyo vehicular	94
4.10.3.1	Tabla Resumen de Clasificación por Sujeto	94
4.10.4	Validación de la Clasificación	95
4.11	Establecimiento de fundamentos para implementación en gemelos digitales	95
4.11.1	Propuesta de la implementación	95
4.11.2	Herramientas y simuladores necesarios para implementación	96
5	Conclusiones, Discusión y Recomendaciones	98
5.1	Conclusiones	98
5.1.1	Cumplimiento del objetivo general	98
5.1.2	Logros en objetivos específicos	98
5.1.3	Validación de la hipótesis de investigación	99
5.2	Discusión	99
5.2.1	Fortalezas metodológicas y experimentales	99
5.2.2	Hallazgos significativos	99
5.2.3	Limitaciones identificadas	100
5.2.4	Implicaciones teóricas y prácticas	100
5.2.5	Aplicaciones potenciales específicas	100
5.3	Sugerencias para futuras investigaciones	101
5.3.1	Procedimiento para el desarrollo de gemelos digitales basados en datos	101
5.4	Conclusión final	102

Índice de figuras

2.1	Representación del marco de investigaciones en comportamiento del conductor y gemelos digitales	10
3.1	Equipos para la adquisición de datos	31
3.2	Recorrido de pista sentido AH	35
3.3	Representación de la pista segmentada	36
3.4	Trayectoria en ciudad	38
3.5	Ejemplo de etiquetado de maniobras en entorno urbano	39
4.1	Matriz de confusión para <i>k-Nearest Neighbors</i> (k=3) para datos de pista	62
4.2	Matriz de confusión para LSTM con validación <i>walk-forward</i>	63
4.3	Matriz de confusión para GRU validación <i>walk-forward</i>	64
4.4	Ejemplo de la matriz rectangular	66
4.5	Matriz rectangular correspondiente al sujeto 1	67
4.6	Matriz rectangular correspondiente al sujeto 3	68
4.7	Matriz rectangular correspondiente al sujeto 5	68
4.8	Gráficas radar para maniobra de recta mostrando patrones de agrupamiento por sujeto	76
4.9	Gráficas radar para maniobra de curva con peralte	77
4.10	Gráficas radar para maniobra de curva sin peralte	78
4.11	Gráficas radar para maniobra de slalom	78
4.12	Gráficas radar para maniobra de rotonda	79
4.13	Diagramas radar de distribución por variable - Maniobra de recta	80
4.14	Diagramas radar de distribución por variable - Curva con peralte	80
4.15	Diagramas radar de distribución por variable - Curva sin peralte	81
4.16	Diagramas radar de distribución por variable - Slalom	82
4.17	Diagramas radar de distribución por variable - Rotonda	83
4.18	Gráficas radar para maniobra de recta mostrando patrones de clustering por sujeto (S1, S3, S4, S5, S6)	89
4.19	Gráficas radar para maniobra de vuelta derecha	90
4.20	Gráficas radar para maniobra de vuelta izquierda	90
4.21	Diagramas polares de distribución por variable - Maniobra de recta	91
4.22	Diagramas polares de distribución por variable - Vuelta derecha	91
4.23	Diagramas polares de distribución por variable - Vuelta izquierda	92

Índice de tablas

1.1	Marco normativo relevante para el desarrollo del proyecto	5
2.1	Análisis de estudios relevantes sobre comportamiento de conducción	11
2.2	Análisis de estudios relevantes sobre aplicaciones de gemelos digitales en el sector automotriz	12
3.1	Variables Físicas medidas a través de los Equipos para Adquisición de Datos	33
3.2	Especificaciones del Vehículo	33
3.3	Distribución de registros por tipo de maniobra	37
3.4	Distribución de maniobras en ciudad por conductor	39
4.1	Comparación de parámetros entre modelos LSTM y GRU	61
4.2	Resultados por clase: Precision, Recall y F1-score del KNN con método de validación <i>K-fold</i>	62
4.3	Métricas de evaluación para LSTM con método de validación <i>walk-forward</i>	63
4.4	Métricas de evaluación para GRU con validación <i>walk-forward</i>	64
4.5	Rendimiento promedio macro de los algoritmos seleccionados	65
4.6	Métricas de evaluación - Sujeto 1 con GRU	67
4.7	Métricas de evaluación - Sujeto 3 con LSTM	67
4.8	Métricas de evaluación - Sujeto 3 con KNN	69
4.9	Desempeño por conductor en modelos individuales considerando si mejor desempeño	69
4.10	Métricas de Validación - Agrupamiento Jerárquico	72
4.11	Métricas de Validación - Algoritmo K-means	72
4.12	Comparación de Métricas por Algoritmo	73
4.13	Distribución de Estilos de Conducción	84
4.14	Distribución Porcentual de Estilos de Conducción por Sujeto	86
4.15	Métricas de Validación - Agrupamiento Jerárquico	87
4.16	Métricas de Validación - Algoritmo K-means	87
4.17	Comparación de Métricas por Algoritmo	88
4.18	Distribución de Estilos de Conducción	92
4.19	Distribución Porcentual de Estilos de Conducción por Sujeto	94

1. Introducción

El avance acelerado de las tecnologías de conectividad vehicular y los sensores integrados en los automóviles modernos han generado una oportunidad sin precedentes para la recopilación y análisis de datos sobre el comportamiento de conducción. La capacidad de los vehículos para generar grandes volúmenes de información, patrones de manejo y respuestas del conductor representa un recurso valioso que puede transformar fundamentalmente nuestra comprensión de la dinámica vehicular y el comportamiento humano al volante.

En este contexto, la identificación de estilos de conducción emerge como un campo de investigación en la industria automotriz. Desde la optimización del consumo energético hasta el desarrollo de sistemas de asistencia personalizados, la capacidad de caracterizar y predecir patrones de conducción individuales se posiciona como un elemento clave para el futuro de la movilidad inteligente y sostenible.

1.1. Definición del problema

Los datos vehiculares recopilados durante la conducción nos brindan una visión detallada sobre el comportamiento del conductor al volante, con el potencial de descubrir patrones distintivos de conducción que permitan identificar diferentes estilos de manejo [1], ofreciendo una fuente valiosa de información para mejorar la seguridad vial, optimizar la eficiencia energética y personalizar la experiencia de manejo. La identificación y comprensión de estos estilos permite desarrollar sistemas adaptativos que pueden ajustar automáticamente las respuestas del vehículo según las características específicas de cada conductor, reducir significativamente el consumo de combustible mediante recomendaciones personalizadas de conducción ecológica, y prevenir accidentes al anticipar comportamientos riesgosos [2, 3].

Sin embargo, el potencial transformador de esta información permanece en gran medida inexplorado debido a la ausencia de metodologías robustas que permitan identificar automáticamente estos estilos a partir de los datos vehiculares disponibles [4]. Esta limitación impide que los sistemas automotrices modernos aprovechen las ventajas inherentes del análisis de estilos de conducción, tales como la mejora en la eficiencia del combustible hasta en un 25 %, la reducción de emisiones contaminantes, y el desarrollo de sistemas de asistencia verdaderamente personalizados [5, 6].

La incapacidad de extraer automáticamente información sobre estilos de conducción limita el desarrollo de gemelos digitales efectivos y sistemas inteligentes que podrían re-

volucionar la industria automotriz mediante la creación de experiencias de manejo más seguras, eficientes y sostenibles [7, 8]. Por lo que, incorporar efectivamente el comportamiento del conductor en gemelos digitales, permite no solo simular escenarios realistas y predecir comportamientos de conducción en diversas condiciones, sino también impulsar avances significativos en los sistemas avanzados de asistencia al conductor (ADAS) y en la evolución hacia vehículos completamente autónomos [9, 10].

1.2. Propósito del estudio

El propósito de este estudio es desarrollar y validar una metodología integral para la identificación automática de estilos de conducción a partir de la extracción de datos vehiculares, mediante la implementación del aprendizaje automático.

Este trabajo tiene como objetivo crear una base metodológica sólida para identificar los estilos de conducción que sirva como base para la implementación de gemelos digitales efectivos en el análisis del comportamiento del conductor, que permitan la predicción y simulación de patrones de conducción. La metodología propuesta busca facilitar la integración de información sobre estilos de manejo en sistemas avanzados de asistencia al conductor (ADAS) y contribuir al desarrollo de vehículos autónomos que requieren una comprensión precisa del comportamiento humano para operar de manera segura y eficiente.

El propósito final es proporcionar herramientas innovadoras que contribuyan a mejorar la seguridad vial, optimizar la eficiencia energética de los vehículos y fomentar el desarrollo de una movilidad más inteligente y sostenible, estableciendo las bases para futuras investigaciones en la integración de aprendizaje automático y tecnologías de simulación digital en el sector automotriz.

1.3. Preguntas de investigación

Esta investigación busca responder las siguientes preguntas fundamentales:

¿Es posible desarrollar una metodología efectiva que combine técnicas de agrupamiento no supervisado con clasificación supervisada para identificar automáticamente estilos de conducción?

¿Qué tan precisos pueden ser los algoritmos de inteligencia artificial para predecir las maniobras de conducción?

¿Cuál es mejor: Aprendizaje profundo o los métodos tradicionales de aprendizaje automático para entender cómo manejamos?

¿Estamos listos para crear gemelos digitales que repliquen fielmente cómo maneja cada persona?

1.4. Importancia del estudio

La revolución de la inteligencia artificial (IA) ha transformado múltiples sectores de la sociedad y la industria, redefiniendo procesos y generando innovaciones disruptivas. En el ámbito automotriz, la creciente demanda de vehículos más seguros, eficientes y sostenibles ha impulsado transformaciones significativas, dando lugar a avances que redefinen el paradigma de la movilidad en el siglo XXI [11].

El ámbito del transporte y la movilidad está experimentando una transformación significativa gracias a la implementación de tecnologías avanzadas, entre las cuales destacan los gemelos digitales. Esta innovadora herramienta permite simular y optimizar el rendimiento de los vehículos, mejorar la seguridad de los sistemas autónomos y fomentar la integración entre el mundo físico y digital en la industria automotriz. [11,12]. Esto permite analizar el desempeño de los vehículos de manera simultánea, predecir su comportamiento, detectar fallas y optimizar su rendimiento [13]. Además, los gemelos digitales facilitan ajustes y mejoras continuas, reflejando con exactitud el estado, comportamiento y rendimiento de sus contrapartes físicas, lo que contribuye a la seguridad y eficiencia de los sistemas de transporte [13,14].

El potencial de los gemelos digitales va más allá de una simple simulación. Su verdadero poder reside en su capacidad para integrar y procesar vastas cantidades de datos en tiempo real, permitiendo un análisis predictivo y prescriptivo que puede anticipar fallos, optimizar el mantenimiento y mejorar continuamente el rendimiento del vehículo a lo largo de su ciclo de vida [15,16]. Esta característica los convierte en herramientas invaluable no solo para los fabricantes, sino también para los operadores y usuarios finales de los vehículos, ya que facilitan la toma de decisiones informadas y la mejora continua de los sistemas automotrices [15,17].

La implementación de un sistema de clasificación robusto y preciso de patrones de conducción es esencial para el desarrollo y perfeccionamiento de sistemas de asistencia avanzada al conductor (ADAS) y vehículos autónomos [18]. Para que estos sistemas sean verdaderamente efectivos y seguros, deben ser capaces de comprender y anticipar no solo las condiciones del entorno, sino también el comportamiento del conductor humano [19]. La variabilidad natural en los patrones de conducción, influenciada por factores como la personalidad, el estado emocional, la experiencia, las condiciones ambientales y el tráfico, representa un desafío significativo para los algoritmos de agrupamiento y clasificación supervisada, que deben adaptarse a esta complejidad para mejorar la seguridad y la eficiencia en la conducción [20,21].

La vasta cantidad de sensores presentes en los vehículos modernos, capaces de capturar una amplia gama de parámetros como velocidad, aceleración, presión de los neumáticos hasta condiciones ambientales en conjunto con sistemas adicionales que integran información adicional referente a la dinámica vehicular y la trayectoria de la ruta, resulta fundamental para un análisis completo de los estilos de conducción. La integración y el procesamiento de estos flujos de datos son esenciales para identificar patrones representativos y clasificables, que permitan una mejor comprensión y evaluación de los comportamientos al volante.

[20, 22]. La precisión en la clasificación de los modos de conducción depende directamente de un análisis exhaustivo de estos datos, que permita no solo extraer patrones relevantes, sino también realizar un etiquetado adecuado para garantizar la efectividad del proceso. Esto se logra mediante la implementación de aprendizaje automático, que permiten una mejor comprensión y anticipación del comportamiento del conductor [22].

En este contexto, el desarrollo de modelos avanzados de clasificación de estilos de conducción, integrados con tecnologías como los gemelos digitales y el aprendizaje automático, representa un paso crucial hacia una movilidad más segura, eficiente y sostenible [12]. Este trabajo busca abordar los desafíos asociados con la variabilidad en los patrones de conducción, proporcionando herramientas innovadoras para mejorar la seguridad vial, optimizar el rendimiento de los vehículos y fomentar una interacción más efectiva entre humanos y sistemas autónomos. De este modo, la investigación no solo contribuye al avance tecnológico, sino que también sienta las bases para transformar el panorama de la movilidad en el futuro [11, 12]

Por otro lado, la clasificación de patrones de conducción es un área de creciente interés en la investigación automotriz. La variabilidad en los estilos de conducción, influenciada por factores como la personalidad y las condiciones del entorno, representa un desafío para los algoritmos de aprendizaje automático [22]. La combinación de estas tecnologías, gemelos digitales y clasificación de patrones de conducción, promete abrir nuevas posibilidades para mejorar la seguridad y la eficiencia en el transporte.

1.4.1. Marco normativo

El desarrollo para realizar las pruebas sobre los comportamientos de conducción encuentra sustento en el marco normativo federal mexicano vigente, el cual no sólo permite, sino que estimula la realización de pruebas controladas en arroyo vehicular. Cumpliendo los requisitos del art. 68 del reglamento de tránsito en carreteras y puentes de jurisdicción federal (RTCPJF) [23], las normas oficiales mexicanas 068-SCT y 012-SCT (NOM) [24, 25] aplicables y los principios de la ley general de movilidad y seguridad vial (LGMSV) [26], en la Tabla 1.1 se resumen las disposiciones clave del marco legal aplicable y su relevancia directa para el desarrollo del presente proyecto.

Norma	Disposición clave	Relevancia para el proyecto
Ley General de Movilidad y Seguridad Vial (LGMSV)	Art. 1 y 2, fracc. X – Garantizar el derecho a la movilidad y tomar decisiones con base en evidencia científica y territorial [26].	Legítima la captación de datos en vía pública para mejorar la seguridad vial.
	Art. 31 – Obliga a los tres órdenes de gobierno a impulsar proyectos de movilidad que protejan la vida y promuevan tecnología para reducir riesgos [26].	Da soporte a estudios piloto con vehículos instrumentados.
Reglamento de Tránsito en Carreteras y Puentes de Jurisdicción Federal (RTCPJF)	Art. 68 – Permite eventos especiales o caravanas en carreteras previa autorización de la autoridad federal competente [23].	Vía reglamentaria para tramitar el permiso de pruebas controladas en arroyo vehicular.
Normas Oficiales Mexicanas	NOM-012-SCT-2-2017 (peso y dimensiones) [24] NOM-068-SCT-2-2014 (condiciones físico-mecánicas) [25]	Aseguran que el vehículo prototipo cumpla requisitos de seguridad e integridad estructural.

Tabla 1.1: Marco normativo relevante para el desarrollo del proyecto

Viabilidad de las pruebas en arroyo vehicular

1. Permiso federal

- Solicitud ante la Secretaría de Infraestructura, Comunicaciones y Transportes (SICT) y la Guardia Nacional, conforme al art. 68 RTCPJF.
- Plan de operación detallando tramo, horario, señalización temporal y póliza de responsabilidad civil amplia.

2. Condiciones técnicas del vehículo

- Peso y dimensiones certificadas bajo NOM-012 (garantiza que el equipo de medición no exceda límites).

3. Gestión de riesgo

- Protocolo de Seguridad Vial Operativa (mensajes variables).

4. Protección y gobernanza de datos

- Aviso de privacidad simplificado para datos de telemetría – alineado con la Ley General de Protección de Datos Personales en Posesión de Sujetos Obligados.
- Registro de variables en formato georreferenciado, compatible con el Sistema Nacional de Movilidad y Seguridad Vial (arts. 29-30 LGMSV).

1.5. Objetivos

1.5.1. Objetivo General

Desarrollar y validar un sistema de clasificación de estilos de conducción que, mediante la integración de datos en tiempo real, algoritmos de aprendizaje automático y aprendizaje profundo y escenarios de conducción realistas, permita generar gemelos digitales.

1.5.2. Objetivos Específicos

1. Recopilar un conjunto representativo de datos vehiculares, mediante equipos de instrumentación especializados y bajo condiciones de conducción diurnas —en pista y en entornos urbanos—, para entrenar y validar el sistema de clasificación de estilos de conducción.
2. Evaluar diversos algoritmos de aprendizaje automático y aprendizaje profundo para la identificación automática de maniobras de conducción, con el fin de seleccionar el modelo de mejor rendimiento para su incorporación al sistema de clasificación.
3. Identificar y caracterizar los estilos de conducción distintivos presentes entre los distintos participantes, determinando las variables clave que los diferencian y estableciendo perfiles que sirvan de referencia para el sistema de clasificación.
4. Definir los fundamentos metodológicos que permitan integrar el sistema de clasificación de estilos de conducción en arquitecturas de gemelos digitales vehiculares.

1.6. Hipótesis

Pregunta de investigación: ¿Es posible desarrollar una metodología de clasificación de patrones de conducción de tal forma que establezca las bases fundamentales para la implementación de gemelos digitales vehiculares?

Hipótesis Alternativa (H1): Se puede desarrollar un sistema de clasificación de patrones de conducción que establezca bases fundamentales para la implementación de gemelos digitales vehiculares.

Hipótesis Nula (H0): No se puede desarrollar un sistema de clasificación de patrones de conducción que establezca bases fundamentales válidas para la implementación de gemelos digitales vehiculares.

1.7. Contribuciones

El trabajo de tesis propone la integración de la inteligencia artificial y los gemelos digitales en la industria automotriz, gracias al acceso a datos y a tecnologías avanzadas de simulación y análisis [15]. Asimismo, se espera que la adopción de estas herramientas sean de gran utilidad para:

1. Mejora de la Seguridad Vial:

- Identificación temprana de patrones de conducción potencialmente peligrosos.
- Desarrollo de sistemas de asistencia al conductor más inteligentes y predictivos.
- Personalización de sistemas de seguridad basados en el estilo de conducción individual.

2. Optimización de la Eficiencia Energética:

- Ajuste de los sistemas del vehículo para maximizar la eficiencia basada en el modo de conducción detectado.
- Desarrollo de estrategias de conducción *eco-friendly* personalizadas.

3. Implicaciones para la Investigación Futura:

- Apertura de nuevas líneas de investigación en la interacción hombre-máquina en el contexto automotriz.
- Potencial para aplicaciones en otros campos, como la optimización del tráfico urbano y la planificación de infraestructuras.

1.8. Limitaciones del estudio

Este estudio presenta las siguientes limitaciones que definen su alcance:

Limitaciones poblacionales: El estudio se limita a datos de conducción de 10 participantes (cinco hombres y cinco mujeres), todos mayores de edad, sin restricciones específicas de experiencia de manejo. No se incluyen análisis de otros grupos demográficos o poblaciones especiales.

Limitaciones tecnológicas: Los datos fueron recopilados utilizando un vehículo Kia Rio modelo 2018 equipado con sistemas específicos de adquisición de datos (VBOX3i, IMU03, PCAN-USB). La investigación se limita a las variables capturadas por estos equipos y no incluye otros tipos de sensores o tecnologías de recopilación.

Limitaciones metodológicas: El estudio se enfoca exclusivamente en el desarrollo y validación de la metodología de análisis de datos para la identificación de estilos de conducción mediante agrupamiento y clasificación supervisada. La investigación no aborda la implementación real y completa del gemelo digital, limitándose al análisis de datos y desarrollo de la base metodológica para futuras implementaciones de gemelos digitales.

1.9. Organización de la tesis

Este documento se estructura en cinco capítulos que abordan de manera sistemática el desarrollo de clasificación de patrones de conducción orientada para que sea aplicable a gemelos digitales.

Capítulo 1 - Introducción: Se establece el marco de la investigación, presentando la definición del problema, el propósito del estudio, las preguntas de investigación, la importancia del trabajo, los objetivos generales y específicos, la hipótesis planteada, así como las limitaciones y supuestos que enmarcan el alcance de la investigación.

Capítulo 2 - Estado del arte: Presenta una revisión de la literatura relevante, abordando dos líneas principales de investigación: la clasificación de estilos de conducción mediante diversas técnicas de aprendizaje automático y aprendizaje profundo, y el desarrollo e implementación de gemelos digitales. Este capítulo fundamenta teóricamente la propuesta e identifica las contribuciones del trabajo.

Capítulo 3 - Materiales y metodología: Describe detalladamente la metodología empleada, incluyendo el proceso de recolección de datos, los equipos de adquisición utilizados (VBOX3i, IMU03, PCAN-USB), el procesamiento y tratamiento de los datos vehiculares, y los algoritmos de aprendizaje automático implementados tanto para clustering como para clasificación supervisada.

Capítulo 4 - Hallazgos en el desarrollo del sistema de clasificación: Presenta todo el desarrollo y los resultados obtenidos del trabajo de investigación para la identificación de patrones de conducción, esto incluye la aplicación de técnicas de aprendizaje supervisado y no supervisado. Se incluyen los hallazgos experimentales tanto para datos de pista como para datos de calle.

Capítulo 5 - Conclusiones: Presenta las conclusiones finales derivadas del trabajo de investigación, evaluando el cumplimiento de los objetivos establecidos y validando la hipótesis planteada. Además, se discuten las limitaciones encontradas durante el desarrollo de la investigación y se proponen recomendaciones para trabajos futuros que permitan avanzar hacia la implementación completa de gemelos digitales.

2. Estado del arte

El análisis del comportamiento del conductor ha emergido como una herramienta importante en el campo de la seguridad vial, la eficiencia energética y el desarrollo de sistemas de transporte inteligentes. Con el aumento constante del tráfico vehicular y la complejidad de los entornos de conducción modernos, comprender y predecir el comportamiento de los conductores se ha vuelto fundamental para mejorar la seguridad en las carreteras y optimizar los sistemas de transporte.

Esta transformación en la forma de comprender y analizar dicho comportamiento ha sido posible gracias a los avances tecnológicos recientes, los cuales abarcan desde mediciones biométricas avanzadas del conductor y el análisis en tiempo real de imágenes del entorno, hasta la interpretación de datos obtenidos a través del sistema de diagnóstico a bordo versión II (OBD-II, por sus siglas en inglés). Como una derivación natural de estos desarrollos, han surgido los sistemas avanzados de asistencia a la conducción (ADAS), los cuales integran sensores, cámaras y algoritmos que permiten monitorear continuamente el entorno y asistir al conductor en tareas críticas como el frenado automático, el mantenimiento de carril o el control de velocidad adaptativo. Al funcionar de manera integrada con otras tecnologías vehiculares, los ADAS no solo contribuyen a mejorar la seguridad y reducir la carga cognitiva del conductor, sino que también enriquecen el ecosistema de datos disponible, facilitando una comprensión más profunda y contextualizada del comportamiento humano al volante.

El estudio de estos comportamientos no solo aporta beneficios inmediatos en la prevención de accidentes y la reducción del consumo de combustible, sino que también adquiere una importancia estratégica en el contexto actual del desarrollo de vehículos autónomos. Comprender los patrones de conducción humana resulta esencial para diseñar sistemas de inteligencia artificial capaces de interactuar de manera segura y eficiente con usuarios humanos, promoviendo así una transición fluida hacia entornos de movilidad automatizada.

En este sentido, la incorporación de técnicas avanzadas como el aprendizaje automático, las redes neuronales y el aprendizaje profundo ha permitido grandes avances en la clasificación del comportamiento del conductor. Estas herramientas no solo incrementan la precisión en la identificación de patrones, sino que también mejoran la capacidad de respuesta en tiempo real de los sistemas inteligentes.

A partir de esta base conceptual y tecnológica, en el siguiente apartado se presenta una revisión detallada de la literatura más relevante en este campo. Se comienza con estu-

dios pioneros en la recolección y análisis de datos vehiculares, para posteriormente abordar investigaciones recientes que aplican enfoques de aprendizaje máquina y aprendizaje profundo en la clasificación de patrones de conducción. Esta evolución se sintetiza en el diagrama 2.1, el cual ilustra el enfoque general de las investigaciones sobre el comportamiento del conductor y su integración con implementaciones de gemelos digitales en el sector automotriz. Mientras que, las tablas 2.1 y 2.2 muestran de forma clara y resumida las características principales de los estudios que le dan soporte a esta investigación.

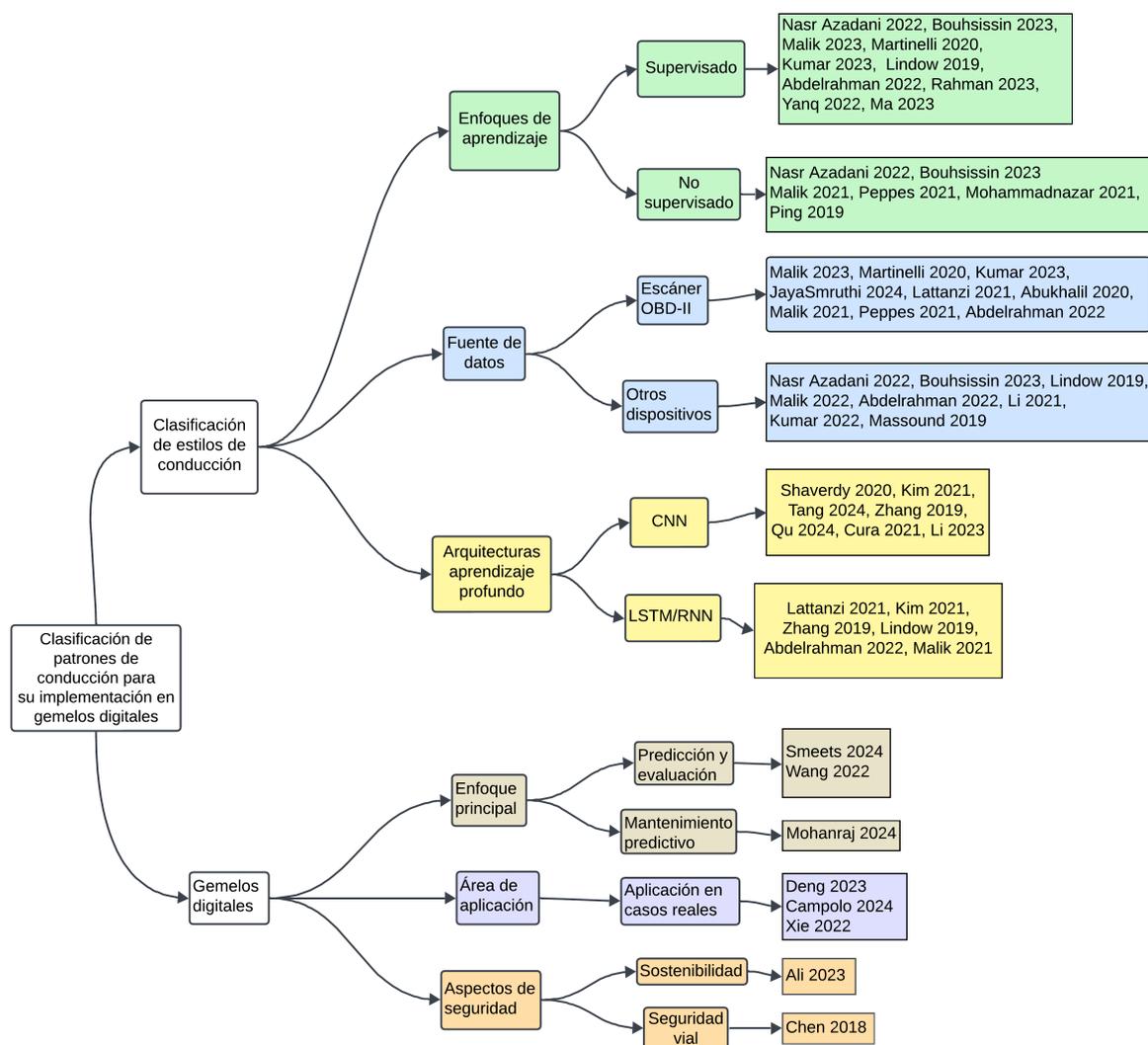


Figura 2.1: Representación del marco de investigaciones en comportamiento del conductor y gemelos digitales

Referencia	Supervisado	No Supervisado	Semi-Supervisado	KNN	SVM	Bosques aleatorios	Árboles de decisión	CNN	LSTM/RNN	CNN-LSTM	Escáner OBD-II	Sensores	Cámaras/Visión computadora	CAN-BUS	Comite de clasificación	Clasificación de estilos	Detección de anomalías
(Nasr Azadani et al., 2022 [27])	x	x			x	x	x	x	x			x	x	x	x	x	x
(Bouhsissin et al., 2023 [14])	x	x		x	x	x	x	x	x			x	x	x	x	x	x
(Malik et al., 2023 [28])	x	x		x	x	x	x	x	x			x	x		x	x	x
(Martinelli et al., 2020 [29])	x				x	x						x			x		
(Kumar & Jain, 2023 [22])	x				x	x						x					x
(Uvarov & Ponomarev, 2021 [30])	x					x						x					
(JayaSmruthi et al., 2024 [31])	x			x	x							x					x
(Lattanzi & Freschi, 2021 [32])	x				x				x			x					
(Chen et al., 2019 [33])	x					x	x								x		x
(Kim et al., 2021 [36])	x							x	x	x					x		
(Shahverdy et al., 2020 [34])	x							x									
(Tang et al., 2024 [37])	x							x	x	x							
(Zhang et al., 2019 [35])	x							x	x	x	x				x		
(Lindow & Kashevnik, 2019 [38])	x			x	x		x	x	x				x				
(Qu et al., 2024 [39])	x							x	x					x			
(Abdelrahman et al., 2022 [40])	x				x	x			x			x	x				x
(Malik et al., 2021 [41])	x	x							x			x				x	
(Peppes et al., 2021 [42])	x	x			x	x			x			x				x	
(Mohammadnazar et al., 2021 [43])		x														x	
(Bandyopadhyay et al., 2022 [44])		x														x	
(Li et al., 2021 [45])			x						x				x				
(Rahman et al., 2023 [46])	x				x	x	x		x						x		x
(Ma et al., 2023 [47])	x										x			x			
(Wang et al., 2022 [48])	x								x								
(Yang et al., 2022 [49])	x	x			x		x									x	
(Ping et al., 2019 [50])	x	x							x							x	
(Cura et al., 2021 [51])	x							x	x								
(Li et al., 2023 [52])	x							x	x	x	x						

Tabla 2.1: Análisis de estudios relevantes sobre comportamiento de conducción

Referencia	Predicción de comportamiento	Evaluación de riesgo	Mantenimiento predictivo	Aplicación en caso real	Vehículos autónomos	Sostenibilidad ambiental	Impacto en la seguridad
(Deng et al., 2023 [55])				X			
(Ali et al., 2023 [58])					X	X	
(Wang et al., 2022 [59])		X		X			X
(Chen et al., 2018 [6])							X
(Mohanraj et al., 2024 [61])			X	X			
(Campolo et al., 2024 [62])				X	X		
(Smeets et al., 2024 [63])	X			X			
(Xie et al., 2022 [65])				X			

Tabla 2.2: Análisis de estudios relevantes sobre aplicaciones de gemelos digitales en el sector automotriz

2.1. Clasificación de estilos de conducción

En esta sección se comienza con revisiones de la literatura sistemática que proporcionan un panorama general del campo. Posteriormente, se describen estudios que establecen metodologías para la recolección y análisis de datos vehiculares mediante sistemas OBD-II, sensores y técnicas de visión por computadora. Continuando, se examinan investigaciones que emplean técnicas de agrupamiento, así como algoritmos de clasificación supervisada para identificar y categorizar patrones específicos de conducción, los cuales se describen como agresivos, defensivos, moderados, entre otros. También se revisan estudios de clasificación de estilos de conducción enfocados en su impacto en la eficiencia energética y sostenibilidad ambiental.

2.1.1. Estudios y revisiones sistemáticas

Comenzando con las revisiones sobre el análisis de los comportamientos de conducción, el artículo "*Driving Behavior Analysis Guidelines for Intelligent Transportation Systems*" [27], realiza un análisis exhaustivo de las metodologías actuales para el análisis del comportamiento de conducción (DBA, por sus siglas en inglés) en el contexto de los sistemas de transporte inteligentes. El análisis de comportamiento de conducción ha captado un gran interés tanto en la academia como en la industria debido a su potencial para mejorar la seguridad vial, optimizar la gestión de combustible y contribuir a la creación de sistemas de asistencia avanzada para la conducción (ADAS).

El artículo proporciona una revisión de los modelos y algoritmos utilizados en DBA, clasificados en tres categorías principales: algoritmos basados en umbrales y lógica difusa, algoritmos de aprendizaje automático y algoritmos de aprendizaje profundo [27].

Un estudio similar es el artículo "*Driver Behavior Classification: A Systematic Literature Review*" [14], que realiza una revisión exhaustiva de la investigación sobre la clasificación del comportamiento de los conductores, centrándose en estudios publicados entre 2015 y 2022. El objetivo es analizar y organizar de manera sistemática las diferentes metodologías y enfoques utilizados para detectar y clasificar los comportamientos de los conductores, destacando el uso de algoritmos de inteligencia artificial como el aprendizaje automático (ML) y el aprendizaje profundo (DL). Los algoritmos más empleados incluyen Redes Neuronales Convolucionales (CNN), Redes Neuronales Recurrentes (RNN) como las LSTM, Máquinas de Soporte Vectorial (SVM), Bosques Aleatorios (*Random Forests*), *K-Nearest Neighbors* (KNN), y Árboles de Decisión [14].

Un estudio adicional es "*A framework on driving behavior and pattern using On-Board diagnostics (OBD-II) tool*" [28], en el cual se presenta un análisis detallado de las técnicas y enfoques para analizar el comportamiento del conductor utilizando datos de sistemas OBD-II y algoritmos de aprendizaje automático. Los autores exploran diferentes aspectos, tales como técnicas de aprendizaje automático (Redes Neuronales, Máquinas de Soporte Vectorial y Árboles de Decisión), técnicas de agrupamiento como K-means, y sistemas basados en Lógica Difusa y Neuro-Difusos para clasificar y predecir comportamientos de conducción.

2.1.2. Análisis del comportamiento del conductor utilizando OBD-II

Una de las fuentes de datos más importantes y ampliamente utilizadas en el análisis del comportamiento de conducción es el sistema de diagnóstico a bordo versión II (OBD-II). Este protocolo estándar, implementado en la mayoría de los vehículos modernos, proporciona acceso directo a una gran cantidad de parámetros del vehículo, incluyendo velocidad, aceleración, revoluciones por minuto del motor, consumo de combustible y posición del acelerador, entre otros. La importancia del OBD-II radica en su capacidad para ofrecer datos precisos y continuos sobre el comportamiento del vehículo y, por extensión, del conductor, sin requerir la instalación de sensores adicionales.

El uso de datos OBD-II para el análisis del comportamiento de conducción ha experimentado un crecimiento significativo en los últimos años, impulsado por la necesidad de desarrollar sistemas más inteligentes y adaptativos para la gestión de flotas, la evaluación de riesgos de seguros, y el desarrollo de tecnologías de asistencia al conductor. La extracción y procesamiento de estos datos permite identificar patrones específicos de conducción, clasificar diferentes estilos de manejo y detectar comportamientos potencialmente peligrosos. A continuación, se presenta una revisión de los estudios más relevantes que han utilizado datos OBD-II como base para el análisis y clasificación del comportamiento de conducción.

Un estudio relevante en la recolección de datos es "*Human behavior characterization for driving style recognition in vehicle system*", realizado por Martinelli et al. [29], donde se creó un método capaz de verificar de manera continua la identidad del propietario del vehículo con alto grado de precisión. Se emplearon técnicas de aprendizaje automático para identificar y diferenciar entre propietarios de vehículos e impostores según su comportamiento al volante. Se recopilieron 51 características de conducción de 10 personas distintas utilizando un escáner OBD-II, aplicando cinco algoritmos de clasificación diferentes. Los resultados mostraron una precisión y exhaustividad promedio de 0.99 % [29].

El trabajo de Kumar y Jain [22] desarrolló un modelo de clasificación que identifica 10 clases distintas de comportamientos de conducción, calculando 11 métricas adicionales a partir de atributos del escáner OBD-II como consumo de combustible, velocidad del vehículo y revoluciones por minuto. Se emplearon algoritmos como SVM, *AdaBoost* y *Random Forest*, logrando tasas de precisión del 99 %, 99 % y 100 %, respectivamente.

El artículo realizado por Kirill Uvarov y Andrew Ponomarev [30] examina el uso de datos OBD-II para identificar el comportamiento y las características de los conductores, empleando el conjunto de datos del escáner OBD-II del artículo anterior. Se destaca que, inicialmente, no se conocía el significado de muchas de las variables extraídas por los escáneres OBD-II. Los autores realizaron una identificación de esta información para llevar a cabo un análisis más profundo de la clasificación, evaluando su precisión mediante cinco algoritmos diferentes. El algoritmo de Random Forest resultó ser el más efectivo, superando a los demás.

Estudios como "*Driving Behavior Classification by Vehicle OBD Data Using Machine Learning*" [31] se centran en desarrollar un modelo para clasificar el comportamiento al volante utilizando datos recopilados del escáner OBD-II. El conjunto de datos incluye parámetros como consumo de combustible, velocidad del motor, velocidad del vehículo, entre otros, con la implementación de múltiples conductores. Se evaluaron varios algoritmos de aprendizaje automático, incluyendo SVM, KNN, Gradient Boosting, Perceptron, para desarrollar el modelo. El algoritmo Gradient Boosting obtuvo los mejores resultados, con tasas de 99.4 %, demostrando capacidad para generalizar a patrones de conducción no vistos.

El artículo "*Machine Learning Techniques to Identify Unsafe Driving Behavior by Means of In-Vehicle Sensor Data*" [32], propone una metodología para clasificar un modo de conducción seguro e inseguro, observando la relación entre la velocidad y la aceleración

lateral y longitudinal del vehículo. Se entrenaron y probaron dos herramientas de clasificación diferentes: máquinas de soporte vectorial y redes neuronales. Del escáner OBD-II se extraen características utilizando descriptores estadísticos, parámetros de Hjorth, parámetros de Curtosis y Asimetría. La red neuronal supera a la SVM, logrando una precisión media superior al 90%, con mejoras notables en precisión y recall para ambas clases de conducción.

La investigación *Comparisons of Machine Learning Algorithms for Driving Behavior Recognition Using In-Vehicle CAN Bus Data* [33], realizó una matriz de correlación con las características más significativas para un proceso de clasificación de comportamientos de conducción. Entre los algoritmos empleados, los basados en comités de clasificadores. *Random Forest* logró la mayor precisión en la prueba del 97.5%, mientras que Decision Tree y Gradient Boosting obtuvieron precisiones superiores al 95%.

2.1.3. Arquitecturas avanzadas de Aprendizaje Profundo

En la última década, el campo del análisis del comportamiento de conducción ha experimentado una revolución significativa con la introducción de arquitecturas avanzadas de aprendizaje profundo. Estas tecnologías han permitido superar las limitaciones de los métodos tradicionales de aprendizaje automático, ofreciendo capacidades superiores para el procesamiento de secuencias temporales complejas y la extracción automática de características relevantes a partir de datos en bruto [34]. Las arquitecturas híbridas, que combinan redes neuronales convolucionales (CNN) con redes neuronales recurrentes (RNN), han demostrado ser particularmente efectivas para el análisis de patrones de conducción, ya que pueden capturar tanto las características espaciales como las dependencias temporales presentes en los datos vehiculares. [35]

La evolución hacia el uso de modelos de aprendizaje profundo en el análisis del comportamiento de conducción ha sido impulsada por la disponibilidad de grandes volúmenes de datos y el incremento en el poder computacional. Estas técnicas han permitido el desarrollo de sistemas más robustos y precisos para la clasificación de estilos de conducción, la detección de anomalías comportamentales y la predicción de patrones futuros de manejo. En esta subsección se revisan las investigaciones más destacadas que han implementado arquitecturas avanzadas de aprendizaje profundo para la identificación y clasificación de comportamientos de conducción.

En años recientes, ha surgido una nueva generación de investigaciones que implementan arquitecturas híbridas de aprendizaje profundo para la clasificación de estilos de conducción. Shahverdy et al. [34] establecieron un paradigma innovador al convertir señales temporales en representaciones espaciales mediante gráficos de recurrencia, aplicando posteriormente CNN para clasificar cinco estilos de conducción: normal, agresivo, distraído, somnoliento y conducción ebria. Su metodología alcanzó una precisión del 99.99%, representando el estado del arte en transformación de datos temporales a espaciales.

Otro enfoque destacado es el desarrollado por Kim et al. [36], quienes implementaron un sistema CNN-LSTM de cuatro capas con dos LSTM apilados para clasificar compor-

tamiento de conducción en tres categorías: defensivo, normal y agresivo. Su arquitectura híbrida combina extracción de características espaciales mediante CNN con modelado temporal mediante LSTM, logrando precisiones superiores al 95 % en clasificación de series temporales de aceleración.

Zhang et al. [35] presentaron un enfoque end-to-end que fusiona CNNs y RNNs con mecanismos de atención para identificación automática de características comportamentales. Su arquitectura demuestra capacidad de aprendizaje automático de patrones sin ingeniería manual de características, utilizando datos temporales para clasificación de estilos de conducción. El modelo propuesto demuestra un rendimiento superior en la identificación de comportamientos de conducción en comparación con métodos tradicionales.

Tang et al. [37] propusieron un modelo CNN-LSTM para identificación de heterogeneidad en conductores, clasificando comportamiento mediante análisis de patrones de car-following. Su sistema utiliza *Coupled Hidden Markov Models* (CHMM) para segmentación semántica de comportamiento, seguido de CNN-LSTM para predicción, superando consistentemente a LSTM tradicional en precisión predictiva.

2.1.4. Análisis del comportamiento de conducción usando diversas técnicas de recopilación de datos.

Si bien el uso de datos OBD-II representa una de las aproximaciones más directas y fácilmente adoptadas para el análisis del comportamiento de conducción, existen diversas técnicas alternativas y complementarias que han demostrado ser valiosas para la comprensión integral de los patrones de manejo. Estas metodologías abarcan desde el uso de sensores integrados en dispositivos móviles hasta sistemas de visión por computadora y análisis multimodal que combinan múltiples fuentes de información. La diversificación en las técnicas de recopilación de datos ha sido impulsada por la necesidad de obtener una perspectiva más holística del comportamiento del conductor, considerando no solo los parámetros operativos del vehículo, sino también las condiciones del entorno, el estado fisiológico del conductor y las características específicas del contexto de conducción.

La importancia de explorar técnicas diversas de recopilación de datos radica en que diferentes enfoques pueden revelar aspectos únicos del comportamiento de conducción que podrían no ser detectables mediante una sola metodología. Por ejemplo, mientras que los datos OBD-II proporcionan información precisa sobre el comportamiento del vehículo, los sensores de smartphones pueden capturar movimientos más sutiles y patrones de aceleración que reflejan el estilo personal de conducción. De manera similar, el análisis de imágenes puede detectar distracciones visuales o fatiga del conductor que no se reflejan directamente en los parámetros operativos del vehículo. A continuación, se presenta una revisión de las investigaciones que han implementado técnicas alternativas para el análisis del comportamiento de conducción.

Existen diversas técnicas para llevar a cabo un análisis del comportamiento de conducción. Una de ellas, previamente explorada, consiste en el uso de escáneres OBD-II. En esta sección, nos enfocaremos en otras técnicas igualmente relevantes.

El artículo realizado por Friedrich Lindow y Alexey Kashevnik [38] abordó el uso de datos de sensores de smartphones y métodos de aprendizaje automático para monitorear el comportamiento del conductor. Se identifican varios enfoques de aprendizaje automático, como Máquinas de Vectores de Soporte, *K-nearest neighbor* más cercanos, redes neuronales y árboles de decisión, que pueden ser utilizados para clasificar y agrupar el comportamiento del conductor basado en datos de sensores del celular.

La investigación de Qu et al. [39] ofrece una revisión exhaustiva sobre los sistemas de monitoreo del comportamiento del conductor, con un enfoque en la integración de visión por computadora para el análisis detallado de su conducta. El estudio destaca avances como los modelos de aprendizaje profundo y arquitecturas de segmentación, específicamente HBPS (*Human Body Parts Segmentation*) y CDCL (*Cross-Domain Complementary Learning*), que permiten aislar con precisión partes clave del cuerpo para detectar distracciones de manera eficaz. Se subraya la relevancia de los indicadores fisiológicos en la evaluación del estado y desempeño del conductor, sugiriendo que un enfoque multimodal puede mejorar la robustez de los sistemas de monitoreo.

El artículo "*Robust Data-Driven Framework for Driver Behavior Profiling Using Supervised Machine Learning*" [40] propone un método para la creación de perfiles de comportamiento de conductores utilizando aprendizaje automático supervisado. Se basa en datos obtenidos del estudio *Strategic Highway Research Program 2* (SHRP2), que incluye más de 9,000 eventos de choques y casi choques, así como 20,000 eventos de conducción normal. Se emplearon varias técnicas tanto de aprendizaje automático como de aprendizaje profundo, tales como *Random Forest* (RF), máquinas de soporte vectorial (SVM), y redes neuronales profundas (DNN). El modelo de *Random Forest* mostró el mejor rendimiento, con predicciones precisas incluso con datos limitados.

2.1.5. Técnicas de agrupamiento y clasificación no supervisada

Una de las principales dificultades en el análisis del comportamiento de conducción es la ausencia de datos etiquetados que permitan la aplicación directa de técnicas de aprendizaje supervisado. En muchos escenarios del mundo real, los datos de conducción se recopilan de manera continua sin una categorización previa de los diferentes estilos o patrones de manejo, lo que hace necesario el uso de técnicas de aprendizaje no supervisado para identificar estructuras inherentes en los datos. Las técnicas de agrupamiento han demostrado ser particularmente valiosas en este contexto, ya que permiten organizar automáticamente grandes volúmenes de datos de conducción en grupos que reflejan los diferentes estilos de manejo.

El desarrollo de metodologías robustas de agrupamiento para datos de conducción ha sido impulsado por la necesidad de procesar eficientemente los grandes volúmenes de información generados por los sistemas modernos de monitoreo vehicular. Estas técnicas no solo permiten la identificación automática de patrones de conducción, sino que también facilitan el etiquetado posterior de los datos para su uso en modelos de aprendizaje supervisado. Además, los enfoques no supervisados son particularmente útiles para detectar comportamientos anómalos o estilos de conducción poco comunes que podrían no

estar representados en conjuntos de datos etiquetados manualmente. En esta subsección se revisan las investigaciones que han aplicado técnicas de agrupamiento y clasificación no supervisada para el análisis del comportamiento de conducción.

El estudio de Malik et al. [41] propone un modelo para agrupar datos recopilados mediante escáner OBD-II. Los datos fueron preprocesados y se aplicaron técnicas de reconocimiento de patrones para identificar tendencias de comportamiento de conducción. Se realizó análisis de correlación para identificar características comunes, continuando con análisis exhaustivo de cada grupo generado para asignarles etiquetas significativas. Mediante técnicas de aprendizaje profundo, se logró una precisión del 99 % en la clasificación.

Un artículo adicional donde se propone un agrupamiento de datos es "*Driving Behaviour Analysis Using Machine and Deep Learning Methods for Continuous Streams of Vehicular Data*" [42], que presenta un análisis del comportamiento de conducción utilizando métodos de aprendizaje automático y aprendizaje profundo aplicados a flujos de datos vehiculares obtenidos mediante el escáner OBD-II. Se aplicó el algoritmo de K-means para agrupar los datos basados en características como las RPM y la velocidad del vehículo, teniendo como resultado dos grupos: conducción ecológica y contaminante.

Mohammadnazar et al. [43] aplicaron ML no supervisado, específicamente K-means, para clasificación de estilos de conducción en tres categorías: agresivo, normal y calmado. Su metodología procesó más de 27 millones de observaciones de mensajes de seguridad básicos de más de 1300 vehículos. Los resultados mostraron que K-means ofreció mayor precisión comparado con K-medoids en la clasificación de patrones comportamentales.

Una investigación adicional que implementa técnicas avanzadas de aprendizaje no supervisado es la desarrollada por Bandyopadhyay et al. [44], quienes utilizaron autocodificadores seq-2-seq multicapa con agrupamiento jerárquico para análisis no supervisado de patrones de conducción. Su aproximación demostró efectividad en el aprendizaje de representaciones para identificación de estilos sin supervisión directa. Esta metodología es particularmente relevante para escenarios donde los datos de conducción carecen de etiquetas predefinidas.

El enfoque desarrollado por Li et al. [45] implementó modelos LSTM semi-supervisados que combinan datos etiquetados y no etiquetados para identificación de maniobras de conducción. Su aproximación supera métodos supervisados tradicionales en precisión, recall, F1-score y *area under the ROC curve* (AUC), demostrando efectividad del aprendizaje semi-supervisado en contextos con datos parcialmente etiquetados. Esta metodología representa una solución práctica para casos donde solo una fracción de los datos de conducción está etiquetada.

2.1.6. Métodos de comité para la clasificación de patrones de conducción

La complejidad en el análisis del comportamiento de conducción, caracterizada por la alta variabilidad entre individuos, la influencia de múltiples factores contextuales y la naturaleza multidimensional de los datos vehiculares, ha hecho que el desarrollo de enfoques

sea más sofisticado, yendo más allá de los algoritmos de clasificación tradicionales. Los comités de clasificadores, que combinan las predicciones de múltiples modelos para obtener resultados más robustos y precisos, han emergido como una solución particularmente efectiva para abordar estos desafíos. Estas técnicas aprovechan las fortalezas complementarias de diferentes algoritmos de aprendizaje automático, reduciendo la varianza y el sesgo inherentes en modelos individuales.

El desarrollo de técnicas avanzadas de clasificación para el análisis del comportamiento de conducción ha sido impulsado por la necesidad de lograr altos niveles de precisión y confiabilidad en aplicaciones críticas de seguridad. Los comités de clasificadores, incluyendo Random Forest, Gradient Boosting, y técnicas de votación, han servido como una herramienta valiosa para la clasificación de patrones de conducción. Además, la integración de arquitecturas de aprendizaje profundo con mecanismos de atención y técnicas de fusión de datos ha abierto nuevas posibilidades para el procesamiento de información multimodal y la captura de dependencias temporales de largo plazo. A continuación, se presenta una revisión de las investigaciones que han implementado un comité de clasificadores para el análisis del comportamiento de conducción.

Rahman et al. [46] compararon múltiples modelos de aprendizaje automático (Random Forest, SVM, Decision Trees, Neural Networks) para clasificación de comportamiento normal vs. anómalo. Su estudio demostró que *Random Forest* alcanza una precisión de 99.9% en la clasificación binaria de patrones comportamentales, estableciendo un límite inferior de comité de clasificadores en la detección de anomalías comportamentales.

La investigación de Ma et al. [47] desarrolló una arquitectura de doble módulo para detección temporal de distracciones en conducción. Su sistema alcanza rendimiento superior sobre métodos previamente realizados, estableciendo un nuevo paradigma para procesamiento de secuencias temporales largas.

Wang et al. [48] introdujeron el algoritmo LSTM-Residual (LSTM-R) de dos pasos para detección en tiempo real de comportamiento anormal de conducción. Su metodología utiliza predicción LSTM seguida de análisis estadístico de residuales temporales, logrando F1-score de 0.866% en detección de anomalías comportamentales sin requerir etiquetado extensivo de datos anómalos.

2.1.7. Sistemas integrados y evaluación comparativa de modelos

La evolución en el campo del análisis del comportamiento de conducción ha dado lugar al desarrollo de sistemas integrados que combinan múltiples técnicas de procesamiento de datos, algoritmos de clasificación, y plataformas de evaluación en soluciones exhaustivas para aplicaciones del mundo real. Los avances teóricos y prácticos en el análisis de datos vehiculares sirven como base fundamental para satisfacer las necesidades de la industria automotriz, los sistemas de transporte inteligente, y las aplicaciones de seguridad vial. La importancia de estos enfoques integrados radica en su capacidad para abordar los desafíos complejos.

El desarrollo de plataformas integradas para el análisis del comportamiento de conduc-

ción ha sido motivado por la necesidad de traducir los avances de investigación en aplicaciones prácticas que puedan beneficiar directamente a conductores, operadores de flotas y unidades de control de tránsito. Estos sistemas incorporan capacidades de procesamiento en tiempo real, interfaces de usuario intuitivas y metodologías de evaluación rigurosas que permiten validar la efectividad de diferentes enfoques de los algoritmos. En esta subsección se revisan las investigaciones que han desarrollado sistemas integrados y metodologías de evaluación comparativa para el análisis del comportamiento de conducción.

Yang et al. [49] desarrollaron un marco para clasificar y evaluar en tiempo real los niveles de seguridad en el comportamiento de conducción. Se utilizó un simulador de conducción en entorno controlado con 260 individuos, registrando 36 variables relacionadas con el comportamiento del conductor. Los datos se agruparon utilizando los algoritmos k-means, agrupamiento jerárquico y *Gaussian Mixture Models* (GMM), siendo k-means el más efectivo. La combinación de árboles de decisión con k-means alcanzó la mayor precisión, con un 99.6 %.

Ping et al. [50] analizaron la relación entre el comportamiento del conductor y el consumo de combustible mediante técnicas avanzadas de aprendizaje automático. Los datos se analizaron mediante un algoritmo de agrupamiento espectral, identificando tres estilos de conducción principales. Posteriormente, se aplicó un modelo de redes neuronales LSTM para predecir el consumo de combustible a corto plazo, integrando información del comportamiento del conductor y del entorno dinámico.

La investigación de Cura et al. [51] implementó arquitecturas CNN y LSTM para la clasificación de comportamiento en conductores de autobús, evaluando cuatro categorías específicas: desaceleración, aceleración del motor, giros y cambios de carril. Sus resultados mostraron que CNN supera significativamente a LSTM en la identificación de conducción agresiva, con mejores índices de rendimiento en el modelado comportamental. Este estudio estableció puntos importantes de referencia para la comparación directa entre arquitecturas CNN y LSTM en la clasificación de patrones de conducción.

El trabajo desarrollado por Li et al. [52] propuso DSDCLA, un framework híbrido CNN-LSTM con mecanismos de atención multinivel que combina self-attention para características espaciales con multi-head attention para relaciones temporales. Esta arquitectura demuestra mejoras significativas en la identificación de patrones espacio-temporales complejos comparado con modelos sin atención, estableciendo un nuevo paradigma para el procesamiento de secuencias de conducción en tiempo real.

2.2. Gemelos digitales

2.2.1. Definición y aplicaciones generales

Como se ha revisado hasta el momento en la última década, el sector automotriz ha analizado al conductor desde múltiples perspectivas, explorando su comportamiento al volante, los estilos de manejo, los factores de seguridad y su contribución hacia una movilidad más sostenible.

Con la implementación de la inteligencia artificial y la tecnología de los gemelos digitales, la industria automotriz ha experimentado una transformación significativa que también viene impulsada por el Internet de las Cosas (IoT) y la conectividad. En este contexto, el concepto de gemelo digital se ofrece como herramienta que permite realizar simulación y análisis en tiempo real de vehículos y sistemas automotrices [53]. Un gemelo digital es una representación virtual de un objeto físico, que integra datos en tiempo real para reflejar su estado, comportamiento y rendimiento [54]. Esta tecnología no solo optimiza el diseño y la fabricación de vehículos, sino que también mejora la gestión del ciclo de vida del producto, la seguridad y la sostenibilidad [55].

La aplicación de gemelos digitales en la industria automotriz abarca diversas áreas, desde el mantenimiento predictivo hasta la simulación de comportamientos de conducción y la optimización de sistemas de energía [55]. A medida que los vehículos se vuelven más complejos y conectados, la necesidad de herramientas que faciliten la toma de decisiones informadas y la mejora continua se vuelve cada vez más crítica [56]. En este sentido, los gemelos digitales ofrecen un enfoque sistemático para abordar los desafíos contemporáneos que enfrenta la industria, como la reducción de costos, la mejora de la eficiencia operativa y el cumplimiento de regulaciones ambientales cada vez más estrictas.

Esta sección de los antecedentes se centra en la revisión y análisis de la literatura existente sobre la tecnología de gemelos digitales aplicada al estudio de la conducción. Primero se realizará una revisión sobre los desarrollos recientes y las aplicaciones prácticas. Se presentarán estudios que destacan la importancia de los gemelos digitales en la mejora de la seguridad vial, la optimización del mantenimiento de vehículos y la personalización de la experiencia del conductor.

2.2.2. Revisiones sistemáticas en la industria automotriz

Comenzando con la investigación de Deng et al., [57] en la cual se realiza una revisión sistemática sobre el concepto y las aplicaciones del *Digital Twin* (DT) en la industria automotriz, se clasifican las investigaciones en tres áreas principales: la industria automotriz tradicional, el transporte inteligente y los sistemas de energía.

Principales aplicaciones del gemelo digital [57]

- **Diseño y producción:** Mejora la eficiencia mediante simulaciones para análisis NVH (ruido, vibración y aspereza) y validación virtual de sistemas eléctricos.
- **Fabricación:** Monitoreo en tiempo real para optimizar líneas de ensamblaje y reducir costos.
- **Mantenimiento predictivo:** Anticipa fallos en componentes como frenos y motores utilizando simulaciones y aprendizaje automático, incrementando la confiabilidad y prolongando la vida útil de los sistemas [57].

Sistemas de transporte y vehículos autónomos [57]

- **Gestión de tráfico:** Modelos virtuales que permiten predecir intenciones del conductor, optimizar la movilidad y reducir congestiones.
- **Conducción autónoma:** Validación de tecnologías mediante simulaciones complejas, garantizando seguridad en su implementación.
- **Sistemas ADAS:** Mejora de la experiencia del usuario con ajustes en tiempo real para maximizar seguridad y comodidad [57].

Sistemas de energía [57]

- **Baterías:** Gestión optimizada del estado de carga (SOC) y salud (SOH) en baterías de litio mediante modelos digitales con aprendizaje profundo (LSTM) y datos en tiempo real.
- **Celdas de combustible (PEMFC):** Modelización tridimensional para abordar desafíos operativos y de durabilidad, mejorando el rendimiento general [57].

Beneficios del gemelo digital [57]

- **Monitoreo en tiempo real:** Facilita el seguimiento continuo del ciclo de vida de los sistemas, permitiendo la implementación de soluciones proactivas.
- **Simulación y verificación:** Aplicaciones como los sistemas de carga automática para vehículos eléctricos mejoran la seguridad y calidad antes de la producción.
- **Integración tecnológica:** En combinación con *Internet of Things* y *PLM que significa Product Lifecycle Management* (IoT y PLM), impulsa la manufactura inteligente, mejorando la eficiencia y reduciendo costos [57].

La siguiente revisión literaria realizada por Ali *et al.* [58] se enfoca en el uso de gemelos digitales (Digital Twins, DT) en sistemas de transporte inteligente (ITS), destacando aplicaciones en vehículos eléctricos (EV) y autónomos (AV). Explora su integración con tecnologías avanzadas como IoT, Big Data, IA, aprendizaje automático (ML) y redes 5G, analizando aplicaciones, beneficios y desafíos [58].

Principales aplicaciones de los gemelos digitales en transporte y electromovilidad [58].

- **Optimización del tráfico:** Predicción de flujos y gestión de semáforos para reducir tiempos de espera.
- **Gestión de baterías (BMS):** Monitoreo en tiempo real del estado de carga (SoC) y salud de baterías, optimizando rendimiento y previniendo fallos.
- **Sensores virtuales (VS):** Simulación de sensores físicos para estimar consumo energético y rutas óptimas.

- **Sistemas autónomos y ADAS:** Mejora de seguridad y experiencia del usuario mediante algoritmos predictivos.
- **Redes de transporte inteligente (DTN):** Gestión de flotas y recursos energéticos en ciudades inteligentes [58].

Tecnologías clave integradas con gemelos digitales [58].

- **IoT:** Conexión en tiempo real para mejorar eficiencia en la carga de vehículos.
- **Big Data:** Análisis masivo para optimizar logística y planificación de rutas.
- **Redes 5G:** Comunicación rápida y confiable entre vehículos e infraestructura.
- **IA y ML:** Predicción de fallos, optimización de operaciones y mejora de ciberseguridad [58].

Desafíos en la electromovilidad [58]

- **Rastreo y monitoreo:** Ansiedad por la autonomía y limitada infraestructura de carga, resuelto con algoritmos predictivos.
- **Gestión de baterías:** Necesidad de mayor durabilidad y eficiencia mediante *Building Management System* (BMS) basado en la nube.
- **Conectividad:** Problemas de interoperabilidad y ciberseguridad [58].
- **Seguridad y privacidad:** Validación de redes para detectar vulnerabilidades.
- **Análisis de datos:** Uso de ML y Big Data para mantenimiento predictivo y reducción de costos [58].

Estudios de caso destacados

- Simulación de tráfico en Zhenjiang [58].
- Detección de ciberataques en vehículos autónomos [58].
- Evaluación de infraestructura de carga rápida y su impacto en redes eléctricas [58].

El artículo concluye que los DT son esenciales para resolver los desafíos de la movilidad inteligente, ofreciendo un marco sólido para el desarrollo de soluciones sostenibles y eficientes [58].

2.2.3. Análisis de riesgos y seguridad vial

Pasando a investigaciones directamente relacionadas con un análisis de los riesgos viales en autopistas, particularmente en accidentes por pérdida de control como derrapes o vuelcos, tenemos el artículo "*Digital Twin Analysis for Driving Risks Based on Virtual Physical Simulation Technology*" [59], el cual implementa gemelos digitales para modelar vehículos y carreteras, evaluando las dinámicas y los riesgos asociados.

Recopilación de datos

- Uso de drones para capturar videos aéreos de vehículos en movimiento.
- Extracción de trayectorias y velocidades mediante visión por computadora (algoritmo YOLOv5)

Construcción del gemelo digital.

- Creación de un espacio virtual que integra dinámicas de conducción, modelos de vehículos y carreteras basados en datos reales.
- **Software empleado:** Carsim y Trucksim para simular dinámica vehicular y construir modelos precisos.

Análisis de riesgos de conducción

- Evaluación de factores como derrapes (ángulo de derrape del centroide), vuelcos (tasa de transferencia de carga lateral - LTR) y colisiones (distancia lateral - SD).
- Identificación de áreas de alto riesgo debido a la geometría y condiciones del pavimento.

Resultados y aplicación. En una prueba realizada en una rampa de la autopista LK en China con 13 vehículos:

- Se identificaron riesgos críticos, como derrapes y vuelcos, en vehículos con velocidades excesivas.
- El análisis resaltó que ciertas áreas de la carretera presentaban mayor riesgo debido a factores estructurales y del pavimento.
- Los resultados muestran que la mayoría de los vehículos tenían un riesgo bajo, pero algunas secciones específicas aumentaron significativamente los peligros.

Características clave

- **Análisis en tiempo real:** Transmisión de datos desde drones mediante redes 4G.
- **Enfoque integral:** Considera tanto el diseño del camino como el comportamiento del vehículo, en lugar de centrarse solo en el conductor.
- **Métricas clave:** Ángulo de derrape, LTR y distancia lateral.

Un estudio similar es el realizado por Chen *et al.* [60], que propone un marco conceptual para mejorar la seguridad en vehículos conectados mediante el uso de gemelos digitales de comportamiento. Este enfoque permite modelar y predecir el comportamiento de los conductores al compartir datos entre vehículos conectados, ayudando a anticipar comportamientos futuros y evitar colisiones. Para representar contextos de conducción y acciones de los conductores, se utiliza un modelo basado en Procesos de Decisión de Markov (MDP), donde el comportamiento del conductor se abstrae en cinco posibles acciones: girar a la izquierda, girar a la derecha, acelerar, frenar o mantener la velocidad. Algoritmos de clasificación como árboles de decisión y *K-nearest neighbors* se emplean para aprender y predecir estos comportamientos [60].

El marco define el riesgo como la probabilidad de que un conductor entre en un estado inseguro de conducción, dependiendo de un contexto y una acción específicos. Se proponen dos métodos para calcular esta probabilidad: uno centralizado, que utiliza información compartida entre vehículos, y otro local, que no depende del intercambio de datos. Para evaluar el sistema, se desarrolló una plataforma virtual integrada con Unity y MATLAB, que permite simular escenarios de conducción, recopilar datos y probar el desempeño del modelo [60].

En las simulaciones realizadas, los modelos lograron predecir con precisión el comportamiento de los conductores y el riesgo de colisión en escenarios de autopista. Además, se implementó una estrategia de prevención basada en estas predicciones, que resultó en una reducción significativa de las colisiones en las pruebas experimentales. El estudio demuestra el potencial de los gemelos digitales de comportamiento para aumentar la seguridad en vehículos conectados mediante la integración de modelos predictivos y estrategias preventivas [60].

2.2.4. Mantenimiento predictivo y desarrollo de sistemas

Una aplicación eminente de los gemelos digitales es el mantenimiento predictivo, como lo elaborado en el artículo "*Digital Twins for Automotive Predictive Maintenance*" [61]. Este trabajo presenta un marco metodológico para implementar sistemas de mantenimiento predictivo habilitados por gemelos digitales, destacando su capacidad para anticipar fallas mediante el análisis de datos históricos y en tiempo real. Este enfoque contrasta con el mantenimiento reactivo tradicional, proporcionando un método más eficiente y proactivo [61].

El marco incluye varias etapas clave: la recolección y preprocesamiento de datos, que utiliza información de sensores vehiculares, registros de mantenimiento y parámetros ambientales, garantizando calidad mediante limpieza y normalización de datos; la ingeniería de características, donde se seleccionan atributos relevantes como fluctuaciones de presión de aceite o patrones de temperatura del motor, optimizando los datos mediante escalado y reducción de dimensionalidad; y el desarrollo de gemelos digitales, creando modelos virtuales detallados con herramientas como Blender, integrando datos de sensores y simulaciones para replicar componentes automotrices [61].

El artículo también detalla el uso de algoritmos avanzados como *Random Forests*, LSTM y *Gradient Boosting* para detectar anomalías y realizar predicciones, optimizando su desempeño mediante técnicas como la búsqueda en cuadrícula. Los modelos se entrenan, validan y prueban utilizando conjuntos de datos divididos y métricas como *precision* y *recall* para garantizar su efectividad. Posteriormente, los sistemas se implementan y monitorean continuamente mediante arquitecturas basadas en la nube o *edge computing*, generando retroalimentación para la mejora continua [61].

Finalmente, se evalúan métricas operativas como el tiempo medio entre fallas (*MTBF*) y el tiempo medio de reparación (*MTTR*), y se aplican algoritmos de aprendizaje adaptativo para refinar las estrategias de mantenimiento. El artículo concluye que este marco puede transformar las prácticas de mantenimiento automotriz, ofreciendo ventajas competitivas en un sector en constante evolución [61].

Una investigación relacionada con el diseño de un gemelo digital es la realizada por Campolo *et al.*, en el artículo [62]. Este trabajo describe el diseño, implementación y evaluación de un marco de gemelos digitales (DT) para vehículos conectados y autónomos (VCA), utilizando *edge computing* para optimizar la gestión de datos, la predicción y el rendimiento vehicular.

El marco propuesto responde a las limitaciones actuales en la aplicación de DT en VCA, como la falta de implementaciones específicas, evaluaciones detalladas de la huella de comunicación y modelos de datos interoperables. Incluye un diseño arquitectónico con componentes físicos y digitales, protocolos de comunicación eficientes y una prueba de concepto (*PoC*) evaluada experimentalmente. Los componentes del sistema abarcan:

- **Vehículo físico:** Módulos de sensorización como CAN bus, GNSS y cámaras.
- **Unidad a bordo (OBU):** Preprocesa y transmite datos.
- **Gemelo Digital Vehicular (VDT):** Un microservicio containerizado que almacena datos en InfluxDB2 y ofrece análisis y predicciones en tiempo real [62].

Las interfaces sur y norte gestionan la comunicación con el vehículo y las aplicaciones externas, respectivamente.

La implementación se realizó con hardware como un módulo ELM327 para datos *Controller Area Network* (CAN) y un microcontrolador ESP32, y software como contenedores Docker y protocolos MQTT y HTTP. Los datos se transmiten y procesan con alta eficiencia, utilizando JSON y estándares como OMA-LwM2M para describir objetos y recursos vehiculares [62].

La evaluación experimental midió métricas como huella de comunicación, tiempo de procesamiento y consumo de recursos en configuraciones de alta frecuencia (10 Hz). Los resultados muestran que la operación *Read-Composite* reduce la huella de comunicación hasta un 24%, mientras que la carga de procesamiento y el consumo de recursos se mantienen manejables, demostrando la escalabilidad del sistema [62].

El artículo concluye que este marco no solo resuelve las limitaciones actuales en DT para CAVs, sino que también establece una base sólida para aplicaciones futuras, promoviendo una movilidad más inteligente mediante un diseño integral que conecta todos los componentes e interfaces del sistema [62].

2.2.5. Implementaciones de gemelos digitales en vehículos motorizados

La siguiente investigación es de suma importancia para la elaboración del proyecto, realizada por Smeets *et al.*, en el artículo "*Digital Twin for Motorcycle Riding Profile Prediction*" [63]. Este trabajo presenta el desarrollo de un gemelo digital diseñado para predecir perfiles de conducción de motocicletas, integrando modelos del vehículo, su entorno y el comportamiento del conductor.

El modelo de comportamiento del conductor se basa en datos recolectados de más de 100,000 viajes en motocicletas BMW en Alemania durante 2022, abarcando 2.5 millones de kilómetros. Este modelo incluye:

- Un análisis del comportamiento lateral, que evalúa la relación entre la curvatura de la carretera y el ángulo de inclinación de la motocicleta.
- Un análisis del comportamiento longitudinal, que utiliza diagramas g-g (aceleración longitudinal y lateral).

Los perfiles de conducción se dividen en cuatro categorías: no dinámico, novato, experimentado y dinámico, y se modelan matemáticamente mediante funciones optimizadas como arcotangentes y elipses [63].

El modelo del entorno utiliza datos geoespaciales de la plataforma HERE [64], para representar rutas con características como curvatura, pendiente y límites de velocidad. Además, simula una línea de conducción optimizada que minimiza la curvatura y maximiza la visibilidad. Por su parte, el modelo de la motocicleta predice el perfil de velocidad y el ángulo de inclinación, considerando límites físicos y capacidades del vehículo [63].

La metodología incluye la recopilación de datos de más de 2.5 millones de curvas, optimización matemática para los modelos y simulaciones realizadas en un paso de montaña alemán de 5 km. Los resultados muestran una alta correlación entre las simulaciones y los datos reales, especialmente para perfiles experimentados (percentil 75°), con errores promedio bajos en velocidad e inclinación. Aunque el perfil dinámico presentó mayor error debido a sus características extremas, sigue siendo valioso en escenarios específicos [63].

El artículo concluye que este gemelo digital es una herramienta efectiva para predecir perfiles de conducción dinámicos y mejorar la seguridad y el diseño de motocicletas, validando su precisión con datos reales y destacando su potencial para aplicaciones futuras en entornos reales [63].

Concluyendo, los gemelos digitales en el artículo "*Digital Twinning Based Adaptive Development Environment for Automotive Cyber-Physical Systems*" [65] proponen un entorno de desarrollo adaptativo basado en gemelos digitales para sistemas ciberfísicos automotrices. Se destacan dos tipos de gemelos digitales:

- **ECU *Twin*:** Replica unidades de control electrónico (ECU) para pruebas y simulaciones.
- ***Component Twin*:** Representa componentes físicos, como luces y puertas, en formato digital.

Se identifican dos desafíos principales:

- La necesidad de clonar cada entidad física (ECUs, componentes, fuentes de prueba) en un gemelo digital para garantizar una representación precisa y en tiempo real.
- La interacción efectiva entre gemelos digitales y entidades físicas para asegurar la relevancia de los datos y simulaciones.

El entorno incluye innovaciones como un flujo de clonación para ECUs y componentes, utilizando simuladores como SkyEye y software como Java o Flash, y un tablero inteligente que actúa como puente entre el mundo físico y digital, manteniendo consistencia en señales y registros automáticos [65]

El artículo concluye que los gemelos digitales son una solución prometedora para los retos actuales en el desarrollo de sistemas ciberfísicos automotrices. Al integrarlos en el proceso, se mejora la eficiencia, adaptabilidad y efectividad, estableciendo un modelo para futuras investigaciones e innovaciones en la industria [65].

2.3. Conclusiones

La forma en que estudiamos y entendemos la conducción ha cambiado por completo gracias a mejores tecnologías y herramientas de análisis más inteligentes. Los fabricantes de automóviles y los investigadores ahora tienen acceso a una gran cantidad de datos útiles provenientes de diversas fuentes, como escáneres OBD-II, sensores adicionales en el vehículo, cámaras, entre otros. Estos datos ayudan a identificar diferentes estilos de conducción y a predecir cómo podrían manejar las personas en ciertas situaciones. Mientras que, para los gemelos digitales, siendo una representación virtual de un entorno real, han sido un gran avance, permitiendo probar nuevas funciones de forma segura y predecir cuándo los vehículos podrían necesitar reparaciones.

Gracias al avance del aprendizaje automático y la inteligencia artificial en el procesamiento de toda esta información, estamos viendo mejoras reales en la seguridad vial y la eficiencia del combustible. Los autos se están volviendo más inteligentes al adaptarse al estilo de cada conductor, lo que ayuda a prevenir accidentes o inclusive impacto en el ahorro energético. Las investigaciones muestran que la combinación de inteligencia artificial con gemelos digitales está ayudando a crear vehículos que no solo comprenden mejor a sus conductores,

sino que también pueden predecir y responder a posibles problemas antes de que ocurran.

Mirando hacia el futuro, estos avances apuntan a vehículos que son más que simples medios de transporte: están convirtiéndose en máquinas inteligentes. Desde identificar patrones de conducción de riesgo de forma temprana hasta sugerir hábitos de manejo más eficientes, la tecnología está haciendo que conducir sea más seguro y sostenible. Este cambio hacia vehículos más inteligentes y conectados no solo está transformando cómo conducimos, sino que también está revolucionando toda la industria automotriz y los sistemas de transporte.

3. Materiales y Metodología

Este capítulo presenta la metodología integral empleada para el desarrollo del sistema de clasificación de patrones de conducción y su implementación en gemelos digitales. El capítulo se estructura en tres componentes fundamentales: primero, se detalla el proceso experimental de recolección de datos, incluyendo la descripción de los equipos especializados utilizados, las condiciones controladas del experimento y las características del conjunto de datos obtenido; segundo, se describe el tratamiento y procesamiento de la información vehicular recolectada mediante múltiples dispositivos de adquisición; y tercero, se presenta el marco teórico y metodológico de los algoritmos de aprendizaje automático aplicados, abarcando tanto técnicas supervisadas (*k-nearest neighbors*, *Random Forest*, MLP) como no supervisadas (agrupamiento jerárquico, K-means) y de aprendizaje profundo (LSTM, GRU).

La metodología aquí descrita constituye la base experimental y teórica que permite la identificación, clasificación y modelado de comportamientos de conducción, estableciendo los fundamentos para la posterior implementación de gemelos digitales capaces de simular y predecir patrones de manejo.

3.1. Recolección y banco de datos

3.1.1. Lugar del experimento

Los experimentos en campo se realizaron en la pista de pruebas del Instituto Mexicano del Transporte (IMT), ubicada en Carretera El Colorado - Galindo Km. 12, Pedro Escobedo, Qro. La pista de pruebas tiene una longitud total de 2.5 Km. De manera más específica, las dimensiones de la pista se pueden consultar en [66,67].

3.1.2. Equipos para la Adquisición de los Datos

Para la adquisición de la información se emplearon tres equipos: un registrador de datos, una unidad de medición inercial (o IMU) y un tranceptor PCAN-USB. En la Figura 3.1 se muestra la conjunción de dichos dispositivos.

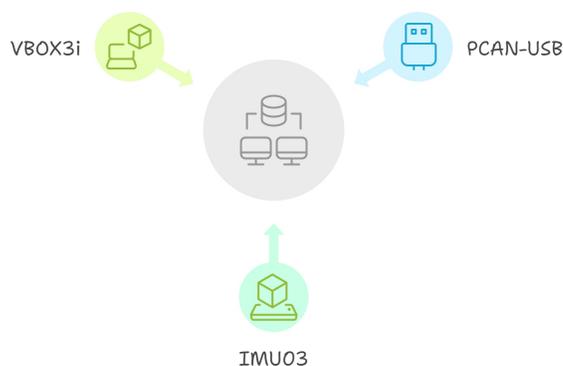


Figura 3.1: Equipos para la adquisición de datos

Registrador de Datos

El primero de los equipos de adquisición es el *VBOX3i Dual Antenna GNSS*, este es un dispositivo registrador de datos vehiculares para pruebas exhaustivas en el campo de la automoción de la marca RACELOGIC. El dispositivo tiene las siguientes características [68]:

- Medición simultánea del ángulo de deslizamiento, ángulo de cabeceo/balaneo, velocidad de guiñada, rumbo verdadero y velocidad lateral/longitudinal.
- Latencia muy baja: 8.5 ± 1 ms.
- Cuatro entradas analógicas / Dos salidas digitales.
- Un puerto CAN Bus predeterminado para ampliación del módulo de entrada.
- Un CAN Bus configurable para interfaz CAN del vehículo o salida de datos VBOX.
- Adquisición de datos inalámbrico a 100 Hz (frecuencia de muestreo) basado en el GPS.
- Interfaz serie: Bluetooth, USB y RS-232.
- Registro de datos en tarjeta Compact Flash.
- Dos salidas analógicas de 16 bits configurables por el usuario.

Unidad de Medición Inercial (o IMU)

El segundo equipo implementado fue una IMU (*Inertial Measurement Unit, por sus siglas en inglés*), de la misma marca que el registrador de datos, la cual lleva el nombre de IMU03. La IMU, al igual que el VBOX, tiene una frecuencia de muestreo de 100 Hz.

Transceptor PCAN-USB

El tercer equipo que se utilizó es un Transceptor PCAN-USB de la marca *Peak System*, el cual, fue el encargado de tomar información del vehículo (con una frecuencia de muestreo base de 50 Hz. (o ajustable a selección del usuario) mediante el protocolo *Controller Area Network (CAN)*, para posteriormente enviarlo a un ordenador.

3.2. Tratamiento de la información

La siguiente etapa corresponde al tratamiento de la información. En este proceso, los datos generados por los dispositivos de recolección deben ser leídos, compatibles entre sí y organizados adecuadamente para su posterior utilización. A continuación, se describen los pasos que conforman esta etapa.

3.2.1. Lectura y tratamiento de la información aportada por el Transceptor PCAN-USB

El equipo arroja datos en bruto (conocidos también como datos raw) que hace referencia a que son datos sin procesar, que finalmente son almacenados en formato .csv (*comma separated values, por sus siglas en inglés*), ello por cada prueba (para todos los sujetos y para ambos sentidos), tanto en pista como en ciudad. El archivo contiene información recolectada a partir de módulos a bordo del vehículo, los cuales están integrados a través de un sistema de adquisición de datos que incluye un módulo GPS y un transceptor PCAN-USB conectado al puerto de diagnóstico del automóvil mediante el protocolo CAN. Esta red permite la transmisión de datos entre distintos sistemas del vehículo, cada uno identificado mediante un Identificador CAN (ID).

3.2.2. Lectura y tratamiento de la información aportada por el VBOX3i y el IMU03

El VBOX3i y el IMU03 son dispositivos que van acoplados, por lo que las variables de ambos sistemas se guardan en un mismo archivo de extensión .vbo. Para poder leer dichos archivos mediante ordenador, la empresa fabricante de los equipos ofrece algunas opciones, entre ellas la lectura mediante el software MATrix LABORatory (MATLAB).

3.2.3. Compatibilización de Datos

Los tres dispositivos para la recopilación de información se sincronizaron para que empezaran a tomar datos al mismo tiempo. El VBOX3i y el IMU03 ya están configurados de base para una frecuencia de muestreo de 100 Hz., es por ello que la única frecuencia de muestreo que se ajustó fue la del PCAN-USB a 100 Hz.

Para este momento la información provenientes de las tres fuentes son iguales en tamaño, y es por esta razón que se pueden leer y concatenar para que queden guardados en un mismo archivo.

3.2.4. Variables recopiladas

Las variables recopiladas por los tres dispositivos incluyen tanto datos físicos como del vehículo. La Tabla 3.1 detalla estas variables, indicando su categoría y resolución, tal como fueron medidas o calculadas por los sistemas de adquisición de información.

Tabla 3.1: Variables Físicas medidas a través de los Equipos para Adquisición de Datos

Equipo	Tipo de Variable	Variable Física	Unidad	Resolución	Categoría
VBOX	Traslacionales	Latitud	[min]	1.8 mm	Vehículo
	Traslacionales	Altitud	[m]	0.01	Vehículo
	Traslacionales	Vel. Lateral	[Km/h]	0.01	Vehículo
	Traslacionales	Vel. Vertical	[m/s]	0.01	Vehículo
	Rotacionales	Ángulo de Deslizamiento Lateral (Slip Angle)	[°]	0.01	Vehículo
	Rotacionales	Ángulo de Balanceo (Roll)	[°]	0.01	Vehículo
	Aceleraciones	Acel. Lateral	[g]	0.01	Vehículo
	Aceleraciones	Aceleración Total	[g]	0.01	Vehículo
	Volante	Ángulo Volante	[°]	0.1	Vehículo
IMU	Pedales	Posición Acelerador	[%]	2 mV	Vehículo
	Aceleraciones	Acel X	[g]	0.001	Vehículo
	Aceleraciones	Acel Z	[g]	0.001	Vehículo
PCAN-USB	Rotacionales	Yaw Rate	[°/s]	0.01	Vehículo
	Pedales	Posición del acelerador	[%]	1	Conductor
	Pedales	Posición del freno	[%]	1	Conductor
	Motor	Revoluciones por minuto (RPM)	[rev/min]	1	Conductor
	Transmisión	Velocidad engranada	[#]	1	Conductor
	Volante	Ángulo de giro del volante	[°]	1	Conductor

3.2.5. Vehículo

Para la ejecución de las pruebas se empleó un vehículo Kia Rio, modelo 2018. En la Tabla 3.2 se presentan algunas de las características del vehículo usado [69]

Tabla 3.2: Especificaciones del Vehículo

Característica	Valor
Transmisión	Caja automática de 6 velocidades
Tracción	Delantera
Cilindros	4 en línea
Número de asientos	5
Tamaño del motor	1.4 L
Caballos de fuerza	99 hp @ 6,000 rpm
Torque	133 lb-ft @ 4,000 rpm
Número de válvulas	16
Altura	1.475 m
Largo	4.485 m

3.2.6. Sujetos

Para las pruebas en pista se consideraron un total de 10 sujetos, conformados por cinco hombres y cinco mujeres. No se establecieron restricciones en cuanto a la experiencia de conducción; el único requisito fue que los participantes fueran mayores de edad.

Los sujetos fueron ordenados alfabéticamente por su nombre, y con base en este orden se les asignó un identificador numérico, el cual es utilizado a lo largo de este trabajo, con la nomenclatura Sujeto 1 (S1), Sujeto 2 (S2), ..., Sujeto 10 (S10).

En el caso de las pruebas realizadas en entorno urbano (ciudad), se contó con la participación de cinco sujetos (tres hombres y dos mujeres), los cuales corresponden a un subconjunto de los mismos participantes que fueron evaluados en pista. Aunque los participantes urbanos también fueron ordenados alfabéticamente, la asignación de identificadores respetó el orden ya establecido en pista. En este contexto el participante Sujeto 2 (S2) en pista, no estuvo en las pruebas realizadas en ciudad. Sin embargo para mantener un orden y una comparación directa los demás Sujetos, mantuvieron el mismo orden designado en pista. Por lo tanto, en ciudad contamos con: Sujeto 1 (S1), Sujeto 3 (S3), Sujeto 4 (S4), Sujeto 5 (S5), Sujeto 6 (S6), así son referidos los sujetos para el entorno de ciudad, en esta investigación

3.2.7. Condiciones del experimento en pista

A continuación se presentará las condiciones experimentales para cada prueba, primero se describe para los datos recolectados en pista y posteriormente se realizará la descripción para los datos recaudado en la ciudad.

Cada prueba consistió en cinco vueltas al circuito en un sentido y cinco vueltas al sentido contrario por cada conductor. A los sujetos se les pidió que manejaran de manera natural durante los ensayos. Se tienen dos sentidos: sentido antihorario (AH) y sentido horario (H). En la Figura 3.2 se ilustra el sentido AH del recorrido en pista (la estrella indica el inicio/fin de la trayectoria).

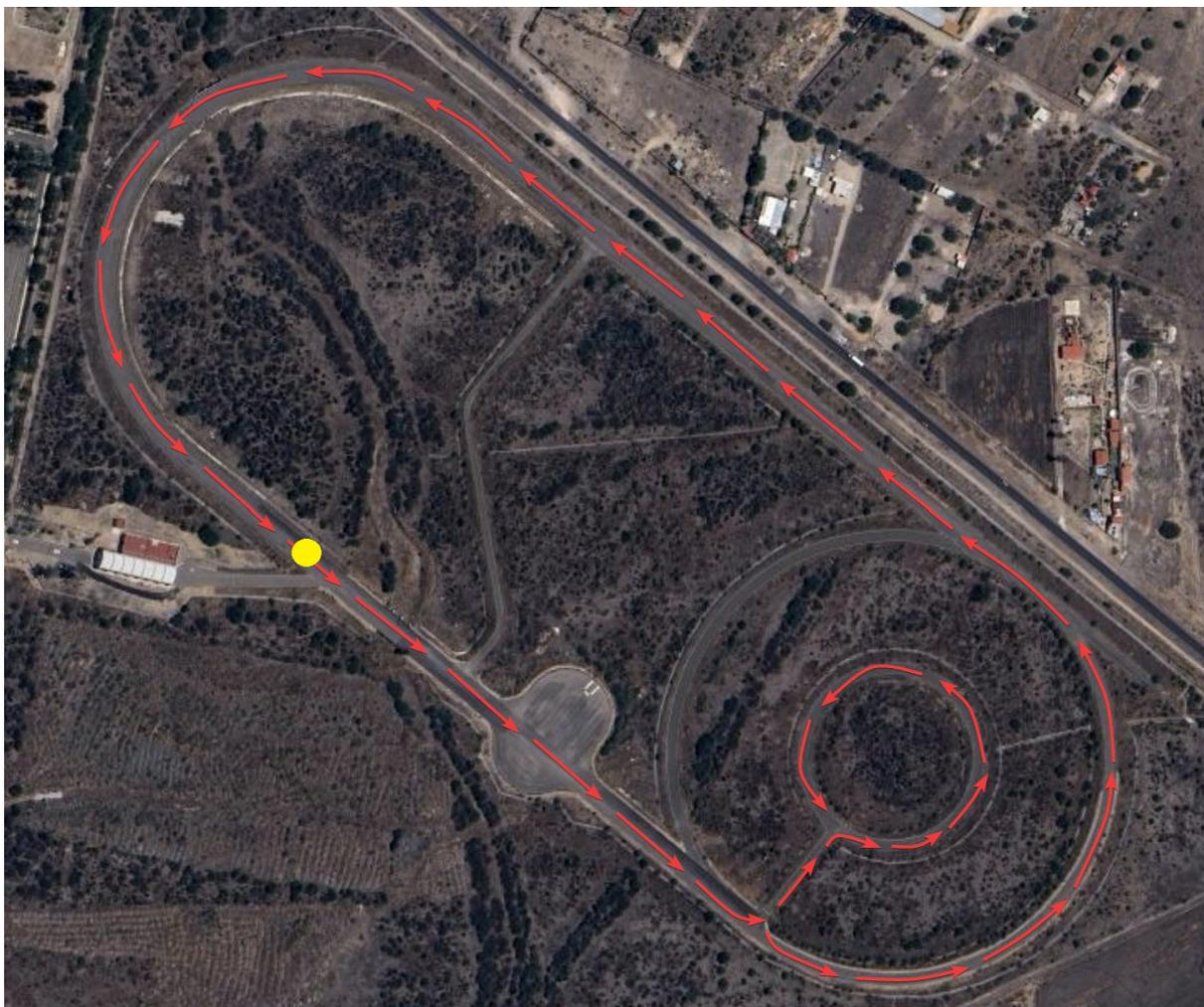


Figura 3.2: Recorrido de pista sentido AH

La conducción de cada participante en una prueba, ya sea en sentido H o AH, debía de cumplir con algunas condiciones de maniobras. Siendo seis las maniobras asignadas a realizar por cada sujeto: Curva con peralte; Curva sin peralte; Codo; Rotonda; Recta; y Maniobra de Evasión de cinco Obstáculos o Prueba de Slalom (ZIG-ZAG). En la Figura 3.3 se presentan la segmentación de la maniobras en una prueba AH.

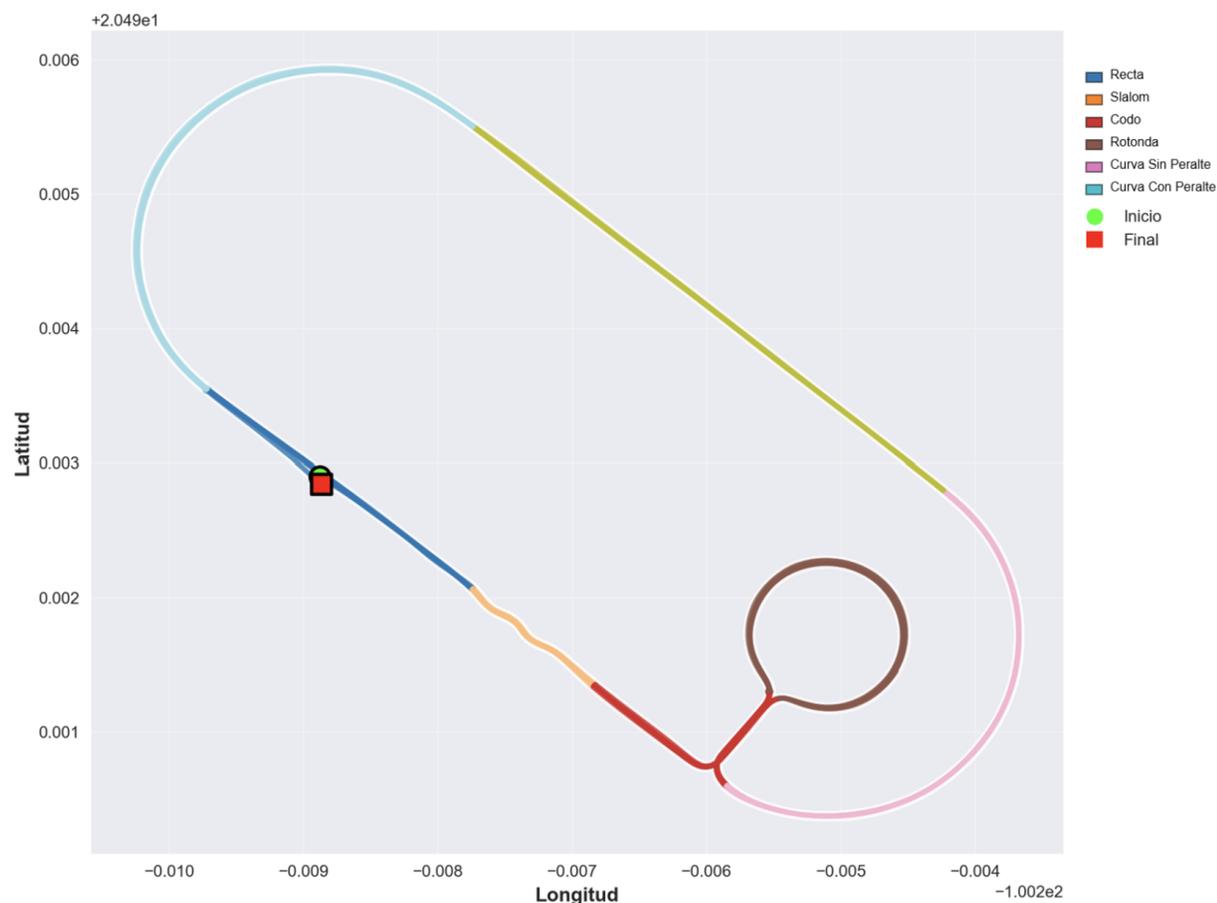


Figura 3.3: Representación de la pista segmentada

Distribución de datos recolectados

Una vez que cada sujeto completó la prueba, se recopilaron un total de 1,669,158 registros. Estos datos se distribuyen entre las diversas maniobras realizadas, cuya distribución detallada se presenta en la Tabla 3.3.

- NOTA:** Los datos identificados por la zona en color rojo, llamado "Rotonda", vease en la Figura 3.3 no se incluyeron en el análisis por no ser representativos del estudio, además de ser una sección con pocos datos. Incluir estos segmentos podría introducir ruido en el modelo de clasificación. Por lo que el total de datos a considerar para su uso posterior, es de 1,366,788, como se muestra en la Tabla 3.3

3.2.8. Condiciones del experimento en Ciudad

Complementando los experimentos de pista, se ejecutó una fase de experimentación en entorno urbano en la ciudad de Querétaro, México. Esta fase tuvo como objetivo evaluar los estilos de conducción en condiciones de entorno urbano.

Tabla 3.3: Distribución de registros por tipo de maniobra

Maniobra	Registros	Porcentaje
Curva sin Peralte	264,218	19.3 %
Curva con Peralte	246,918	18.0 %
Rectas	395,643	28.9 %
Slalom	139,360	10.2 %
Rotonda	320,649	23.4 %
Total	1,366,788	100.0 %

3.2.8.1. Características del experimento en ciudad

Para la fase urbana se seleccionaron cinco conductores (tres hombres, dos mujeres) del grupo original de 10 participantes que habían completado las pruebas en pista. Esta selección permitió realizar comparaciones directas entre el comportamiento del mismo conductor en ambos entornos de conducción.

Los experimentos se realizaron utilizando la misma configuración de equipos empleada en pista: VBOX3i con frecuencia de muestreo de 100 Hz, IMU03 para mediciones inerciales, y PCAN-USB para datos del vehículo a 50 Hz. Se mantuvo el mismo vehículo Kia Rio 2018 para garantizar consistencia en las mediciones.

3.2.8.2. Trayectoria en ciudad

Se diseñó una ruta específica de aproximadamente 35 km, teniendo cada prueba una duración entre 35 y 50 minutos. La ruta recorrida por cada uno de los sujetos se muestra en la Figura 3.4:

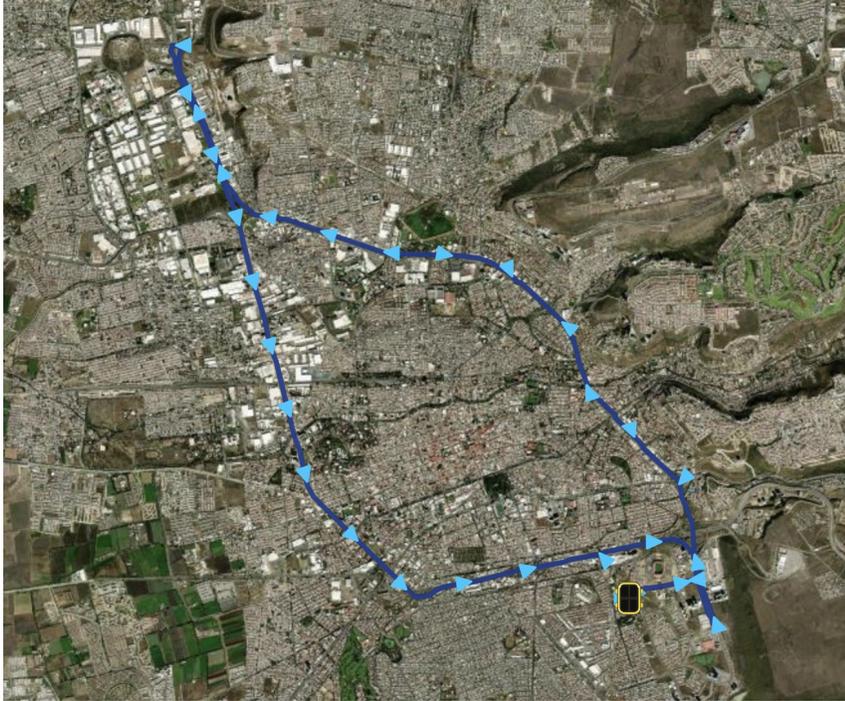


Figura 3.4: Trayectoria en ciudad

3.2.8.3. Clasificación de maniobras urbanas

El entorno urbano requirió simplificar el esquema, realizando una clasificación semi-supervisada que consta de tres categorías basadas en el ángulo del volante:

- **Recta:** Se considera un ángulo del volante entre -10° y $+10^\circ$
- **Vuelta Derecha:** Ángulo del volante mayor a $+10^\circ$
- **Vuelta Izquierda:** Ángulo del volante menor a -10°

Un ejemplo del resultado del proceso de etiquetado se muestra en la Figura 3.5, en donde:

- **Color verde:** Recta
- **Color morado:** Vuelta derecha
- **Color amarillo:** Vuelta izquierda

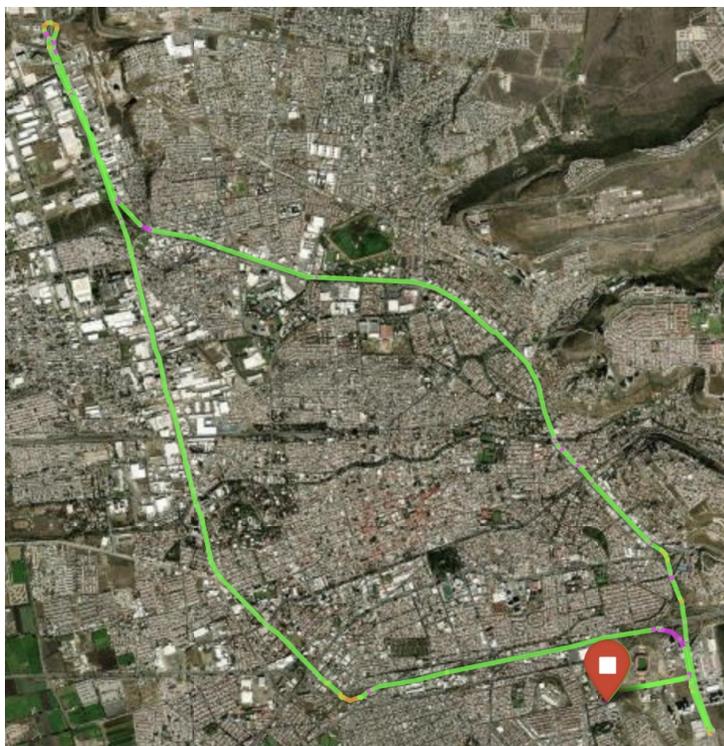


Figura 3.5: Ejemplo de etiquetado de maniobras en entorno urbano

La distribución de maniobras obtenida por conductor se presenta en la Tabla 3.4:

Tabla 3.4: Distribución de maniobras en ciudad por conductor

Conductor	Rectas	Giro Der.	Giro Izq.	Total
Sujeto 1	179,230 (86.6 %)	19,125 (9.2 %)	8,632 (4.2 %)	206,987
Sujeto 2	256,562 (87.0 %)	16,498 (5.6 %)	22,600 (7.4 %)	295,660
Sujeto 3	175,118 (82.8 %)	17,141 (8.1 %)	18,885 (8.9 %)	211,144
Sujeto 4	176,929 (91.3 %)	8,936 (4.6 %)	7,909 (4.1 %)	193,774
Sujeto 5	250,993 (84.8 %)	22,008 (7.4 %)	23,587 (8.0 %)	296,588
Total	1,038,832 (86.1 %)	83,708 (6.9 %)	81,613 (6.8 %)	1,204,153

Con esto se da por finalizado el proceso de tratamiento de la información, para dar el siguiente paso; establecer la metodología de análisis que permita extraer patrones significativos de los datos recolectados.

3.3. Metodología de análisis

Esta etapa consiste en establecer los métodos de validación, los algoritmos y las métricas utilizadas durante el proceso de desarrollo de esta investigación.

3.3.1. Métodos de validación

Empezando por los métodos de validación, se emplearon tres enfoques diferentes. El uso de múltiples métodos de validación permite evaluar los modelos desde diferentes perspectivas, el método de validación *Hold-out* proporciona una evaluación directa, *K-Fold* elimina el sesgo de partición y *Walk-Forward* es especialmente apropiado para datos temporales como los registros vehiculares. Esta combinación metodológica garantiza una evaluación más completa y confiable de cada algoritmo implementado.

3.3.1.1. *Hold-Out Validation*

El método Hold-Out consiste en dividir el conjunto de datos en dos partes independientes: un conjunto de entrenamiento y un conjunto de prueba. Tradicionalmente se utiliza una división 80-20 o 70-30, donde el porcentaje mayor corresponde al entrenamiento.

3.3.1.2. *K-Fold Cross Validation*

La validación cruzada K-Fold divide el conjunto de datos en K subconjuntos (folds) de tamaño aproximadamente igual. El proceso se repite K veces, utilizando cada fold como conjunto de prueba una vez.

3.3.1.3. *Walk-Forward Validation*

La validación *Walk-Forward* es especialmente apropiada para datos temporales o secuenciales. Simula condiciones reales donde se entrena con datos históricos y se predice el futuro.

Proceso del método *Walk-Forward Validation* [70]

1. Se establece una ventana temporal inicial para entrenamiento
2. Se entrena el modelo con los datos de esa ventana
3. Se evalúa en el siguiente período temporal
4. Se expande la ventana de entrenamiento incluyendo el período recién evaluado
5. Se repite el proceso avanzando temporalmente

3.3.2. Análisis Supervisado

Para el desarrollo de esta investigación se hace uso del paradigma supervisado del aprendizaje automático, para el proceso de clasificación, correspondiente a los datos en pista. Por lo que, se hace la descripción de cada algoritmo empleado.

3.3.2.1. K-NN

El algoritmo *K-Nearest Neighbors* (KNN) apareció inicialmente en 1951 en "*Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties*" [71], un trabajo de Fix y Hodges. En 1967, su análisis fue ampliado en "*The Nearest Neighbor Decision Rule*", un artículo de T. M. Cover y P. E. Hart [72].

El algoritmo determina la clasificación de un punto no clasificado basándose en las clases de sus k vecinos más cercanos dentro de un espacio de características. El algoritmo opera bajo la suposición de que los puntos ubicados cerca unos de otros tienen una mayor probabilidad de pertenecer a la misma clase. Para medir la proximidad, KNN emplea funciones de distancia, una de las más empleadas es la distancia euclidiana pero cabe recalcar que no es la única métrica que existe. [72].

La idea central del algoritmo es que los puntos de datos similares tienden a estar cerca unos de otros. Asigna una clase a un punto no clasificado mediante una votación mayoritaria de las clases de sus vecinos más cercanos [71, 72].

3.3.2.1.1 Funcionamiento

Definición del valor de k : Se elige un valor de k , que representa el número de vecinos más cercanos que se tendrán en cuenta para la clasificación [72].

Cálculo de distancias: Para clasificar un nuevo punto, es necesario calcular su distancia con respecto a todos los puntos del conjunto de entrenamiento, utilizando cualquier distancia [72].

Selección de vecinos: Una vez calculadas las distancias, se identifican los k puntos más cercanos al punto que se desea clasificar. Estos vecinos serán los que influyan directamente en la asignación de la clase [72].

Asignación de clase: El algoritmo asigna al punto no clasificado la clase más frecuente entre sus k vecinos más cercanos [73].

En caso de empate entre varias clases, se pueden aplicar distintos criterios de desempate, como la selección aleatoria de una de las clases empatadas o el uso de la distancia promedio a los vecinos de cada clase para tomar una decisión más precisa.

3.3.2.1.2 Pseudocódigo El algoritmo *K-Nearest Neighbors* se implementa siguiendo los pasos a continuación descritos [72]:

3.3.2.2. Bosques Aleatorios (Random Forest)

El algoritmo *Random Forest* (Bosques Aleatorios) fue desarrollado por Leo Breiman en 2001 en su artículo "*seminal Random Forests*" publicado en *Machine Learning* [74]. Este trabajo se basó en desarrollos previos de métodos de *ensemble*, incluyendo el concepto de *bagging* introducido por el mismo Breiman en 1996 [75] y el método de subespacios aleatorios propuesto por Ho en 1998 [76].

Entrada: conjunto de entrenamiento $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, punto de prueba x_q , número de vecinos K

Salida: clase predicha para x_q

Proceso:

1. Para cada punto x_i en D :
 - Calcular la distancia $d(x_q, x_i)$
 2. Ordenar las distancias de menor a mayor
 3. Seleccionar los K vecinos más cercanos a x_q
 4. Asignar la clase más frecuente entre esos K vecinos
 5. Devolver la clase asignada
-

Random Forest es un algoritmo de aprendizaje automático supervisado que utiliza comité de ensamble para tareas de clasificación y regresión. El algoritmo construye múltiples árboles de decisión durante el entrenamiento y produce la predicción final combinando las predicciones individuales de cada árbol mediante votación mayoritaria (para clasificación). La idea fundamental es que un conjunto de modelos débiles puede formar un modelo fuerte cuando se combinan adecuadamente [77].

La característica distintiva de *Random Forest* es la introducción de aleatoriedad en dos niveles: primero, cada árbol se entrena con una muestra *bootstrap* del conjunto de datos original, y segundo, en cada nodo del árbol solo se considera un subconjunto aleatorio de características para determinar la mejor división [78]. Esta doble aleatoriedad reduce la correlación entre los árboles individuales y mejora la capacidad de generalización del modelo.

3.3.2.2.1 Funcionamiento

Generación de muestras: Para cada árbol en el bosque, se genera una muestra *bootstrap* del conjunto de datos original. Esto significa que se selecciona aleatoriamente el mismo número de muestras que el conjunto original, pero con reemplazo, lo que resulta en que algunas muestras aparezcan múltiples veces y otras no aparezcan [74].

Construcción de árboles con selección aleatoria de características: Cada árbol se construye utilizando solo un subconjunto aleatorio de características en cada nodo para determinar la mejor división. Típicamente, se utiliza \sqrt{p} características para clasificación y $p/3$ para regresión, donde p es el número total de características [74].

Crecimiento completo de los árboles: Los árboles individuales se hacen crecer profundamente sin poda, utilizando solo las características seleccionadas aleatoriamente en cada nodo. Esto permite que cada árbol capture patrones complejos en su subespacio de características [79].

Agregación de predicciones: Para realizar una predicción sobre una nueva muestra, cada árbol en el bosque genera su propia predicción. En clasificación, la clase final se determina por votación mayoritaria, mientras que en regresión se calcula el promedio de todas las predicciones [80].

Estimación *out-of-bag*: Las muestras que no fueron seleccionadas durante el *bootstrap* de cada árbol (aproximadamente 36.8% de las muestras) se utilizan para estimar el error de generalización sin necesidad de un conjunto de validación separado. Esta estimación *out-of-bag* proporciona una medida confiable del rendimiento del modelo [81].

El número de árboles en el bosque ($n_estimators$) es un hiperparámetro importante que generalmente se establece en un valor suficientemente grande para asegurar la convergencia del error, típicamente entre 100 y 1000 árboles, dependiendo del conjunto de datos y los recursos computacionales disponibles.

3.3.2.2 Pseudocódigo El algoritmo Random Forest se implementa siguiendo los pasos a continuación descritos, el primer pseudocódigo es para el proceso de entrenamiento y el segundo pseudocódigo es para el proceso de clasificación [74]:

Entrada: conjunto de entrenamiento $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, número de árboles N , profundidad máxima d_{max} , número de características m

Salida: conjunto de árboles entrenados $\{T_1, T_2, \dots, T_N\}$

Proceso:

1. **Para** $i = 1$ hasta N :
 - Crear muestra bootstrap D_i desde D
 - Seleccionar aleatoriamente m características de las p disponibles
 - Entrenar árbol de decisión T_i con D_i , limitado a profundidad d_{max}
2. **Fin Para**
3. Retornar conjunto de árboles $\{T_1, T_2, \dots, T_N\}$

Algorithm 1: Entrenamiento del algoritmo Random Forest

Entrada: punto de prueba x_q , conjunto de árboles entrenados $\{T_1, T_2, \dots, T_N\}$

Salida: clase predicha \hat{y}

Proceso:

1. **Para** cada árbol T_i :
 - Obtener predicción $y_i = T_i(x_q)$
2. **Fin Para**
3. Calcular la clase más frecuente: $\hat{y} = \text{moda}(y_1, y_2, \dots, y_N)$
4. Retornar clase predicha \hat{y}

Algorithm 2: Clasificación de un nuevo punto con Random Forest

3.3.2.3. Red neuronal multicapa (Multilayer Perceptron)

La red neuronal multicapa tiene sus orígenes en los trabajos pioneros sobre redes neuronales artificiales. Los fundamentos conceptuales fueron establecidos por McCulloch y Pitts en 1943 con su modelo de neurona artificial [82], y posteriormente desarrollados por Rosenblatt en 1958 con el *Perceptron* [83]. Sin embargo, el avance decisivo llegó en 1986 cuando Rumelhart, Hinton y Williams popularizaron el algoritmo de retropropagación (*backpropagation*) en su influyente artículo "*Learning representations by back-propagating errors*" publicado en *Nature* [84].

El Multilayer Perceptron (MLP) es una red neuronal artificial de alimentación hacia adelante (*feedforward*) que consiste en múltiples capas de neuronas completamente conectadas. A diferencia del perceptrón simple, que solo puede resolver problemas linealmente separables, el MLP puede manejar problemas complejos no lineales mediante el uso de capas ocultas y funciones de activación no lineales. El algoritmo utiliza el método de retropropagación para ajustar los pesos de las conexiones, permitiendo que la red aprenda representaciones internas útiles de los datos [85].

La arquitectura básica del MLP consta de una capa de entrada que recibe los datos, una o más capas ocultas que procesan la información, y una capa de salida que produce las predicciones finales. Cada neurona realiza una combinación lineal de sus entradas seguida de una función de activación no lineal, típicamente la función sigmoide, tangente hiperbólica o ReLU (*Rectified Linear Unit*) [86].

3.3.2.3.1 Funcionamiento

Propagación hacia adelante (*Forward propagation*): Durante la fase de propagación hacia adelante, los datos de entrada se procesan capa por capa desde la entrada hasta la salida. Cada neurona calcula una suma ponderada de sus entradas, añade un término de sesgo, y aplica una función de activación para producir su salida. Matemáticamente, para una neurona j en la capa l : $a_j^{(l)} = f\left(\sum_i w_{ji}^{(l)} a_i^{(l-1)} + b_j^{(l)}\right)$ [84].

Cálculo de la función de pérdida: Una vez que la red produce una salida, se calcula una función de pérdida que mide la discrepancia entre la salida predicha y la salida deseada. Para problemas de clasificación, comúnmente se utiliza la entropía cruzada, mientras que para regresión se emplea típicamente el error cuadrático medio [87].

Retropropagación del error: El algoritmo de retropropagación calcula los gradientes de la función de pérdida con respecto a todos los pesos y sesgos de la red. Utilizando la regla de la cadena del cálculo diferencial, el error se propaga hacia atrás desde la capa de salida hasta la capa de entrada, calculando las derivadas parciales necesarias para la actualización de parámetros [84].

Actualización de pesos mediante descenso de gradiente: Los pesos y sesgos se actualizan utilizando algoritmos de optimización basados en gradientes, típicamente el descenso de gradiente estocástico (SGD) o variantes más sofisticadas como lo puede ser Adam [88]. La actualización sigue la regla: $w_{ji}^{(l)} \leftarrow w_{ji}^{(l)} - \eta \frac{\partial L}{\partial w_{ji}^{(l)}}$, donde η es la tasa de aprendizaje [86].

Iteración del proceso de entrenamiento: El proceso de propagación hacia adelante, cálculo de pérdida, retropropagación y actualización de pesos se repite iterativamente para múltiples épocas hasta que la red converge a una solución satisfactoria. Durante este proceso, la red ajusta gradualmente sus parámetros para minimizar la función de pérdida y mejorar su capacidad de predicción [85].

3.3.2.3.2 Pseudocódigo Red Neuronal Multicapa (MLP) El algoritmo de Red Neuronal Multicapa con retropropagación se implementa siguiendo los pasos a continuación descritos, se tienen dos pseudocódigos una para el proceso de entrenamiento y otra para la fase

Entrada: conjunto de entrenamiento $D = \{(x_1, y_1), \dots, (x_n, y_n)\}$, tasa de aprendizaje α , número de épocas E , arquitectura de la red (capas, neuronas, función de activación)

Salida: pesos $W^{(l)}$ y sesgos $b^{(l)}$ entrenados para cada capa l

Proceso:

1. Inicializar pesos $W^{(l)}$ y sesgos $b^{(l)}$ aleatoriamente para cada capa l

2. **Para** $e = 1$ hasta E (épocas):

▪ **Para** cada muestra (x_i, y_i) en D :

• **Propagación hacia adelante:** calcular activaciones $a^{(l)} = \sigma(W^{(l)}a^{(l-1)} + b^{(l)})$

• **Retropropagación:**

◦ Calcular error de salida: $\delta^{(L)} = (a^{(L)} - y_i) \odot \sigma'(z^{(L)})$

◦ Para cada capa l hacia atrás: $\delta^{(l)} = (W^{(l+1)T}\delta^{(l+1)} + b^{(l+1)}) \odot \sigma'(z^{(l)})$

• **Actualizar parámetros:**

◦ $W^{(l)} := W^{(l)} - \alpha \delta^{(l)} a^{(l-1)T}$

◦ $b^{(l)} := b^{(l)} - \alpha \delta^{(l)}$

▪ **Fin Para**

3. **Fin Para**

de clasificación [84].

Algorithm 3: Entrenamiento de una red neuronal multicapa (MLP)

Entrada: punto de prueba x_q , pesos $W^{(l)}$ y sesgos $b^{(l)}$ entrenados para cada capa

Salida: clase predicha \hat{y}

Proceso:

1. Establecer activación inicial $a^{(0)} = x_q$

2. **Para** cada capa l :

▪ Calcular $z^{(l)} = W^{(l)}a^{(l-1)} + b^{(l)}$

▪ Calcular $a^{(l)} = \sigma(z^{(l)})$

3. **Fin Para**

4. La salida final es $a^{(L)}$, de donde se obtiene $\hat{y} = \arg \max(a^{(L)})$

5. Retornar clase predicha \hat{y}

Algorithm 4: Clasificación de un dato con una red neuronal entrenada

El entrenamiento de un MLP requiere una selección de hiperparámetros como la arquitectura de la red, la tasa de aprendizaje, el tamaño del lote (*batch size*), y estrategias de regularización para prevenir el sobreajuste y asegurar una buena generalización.

3.3.3. Aprendizaje Profundo

Para el desarrollo de esta investigación se emplean técnicas de aprendizaje profundo como complemento al paradigma supervisado del aprendizaje automático. Este análisis es

de utilidad para el proceso de clasificación de patrones temporales, consiguiendo de esta manera no romper la secuencia de los datos, característica esencial para la clasificación de maniobras de conducción que presentan patrones complejos en el tiempo.

3.3.3.1. LSTM

Las redes *Long Short-Term Memory* (LSTM) fueron introducidas por primera vez por Sepp Hochreiter y Jürgen Schmidhuber en 1997 en su trabajo “*Long Short-Term Memory*” [89].

LSTM es una arquitectura especializada de redes neuronales recurrentes (RNN) diseñada para resolver el problema del desvanecimiento del gradiente que afecta a las RNN tradicionales. Las LSTM son capaces de aprender dependencias a largo plazo en secuencias de datos mediante el uso de una estructura de celdas de memoria y tres compuertas especializadas que controlan el flujo de información [90,91]. Esta arquitectura permite que la red mantenga información relevante durante largos períodos de tiempo mientras descarta información irrelevante, lo que las hace especialmente eficaces para tareas de procesamiento de secuencias temporales [92].

3.3.3.1.1 Funcionamiento

- La LSTM utiliza una celda de memoria (C_t) que actúa como un estado interno capaz de mantener información a través del tiempo [89].
- La **compuerta de olvido** (f_t) decide qué información debe ser eliminada del estado de la celda anterior. Esta compuerta utiliza una función sigmoide para determinar qué valores entre 0 y 1 asignar a cada componente del estado anterior [93].
- La **compuerta de entrada** (i_t) determina qué nueva información será almacenada en el estado de la celda. Trabaja en conjunto con una capa tanh que crea nuevos valores candidatos (\tilde{C}_t) que podrían ser añadidos al estado [91].
- El **estado de la celda** se actualiza combinando la información que se mantiene del estado anterior (controlada por la compuerta de olvido) con la nueva información seleccionada (controlada por la compuerta de entrada): $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$ [89].
- La **compuerta de salida** (o_t) controla qué partes del estado de la celda serán utilizadas para generar la salida. La salida final se obtiene aplicando una función tanh al estado de la celda y multiplicándola por la compuerta de salida: $h_t = o_t * \tanh(C_t)$ [93].
- El proceso se repite de forma iterativa para cada paso temporal en la secuencia, permitiendo que la red procese secuencias de longitud variable y capture patrones temporales complejos [91].

3.3.3.1.2 Pseudocódigo El algoritmo LSTM se implementa siguiendo los pasos a continuación descritos, primero se describe el proceso de entrenamiento y después esta descrito el proceso de clasificación [90]:

Entrada: secuencias de entrenamiento $X = \{x_1, x_2, \dots, x_T\}$ y sus etiquetas Y , número de épocas E , tasa de aprendizaje α

Salida: parámetros entrenados de la red: pesos W y sesgos b

Proceso:

1. Inicializar pesos W y sesgos b aleatoriamente
2. **Para** $e = 1$ hasta E (épocas):
 - Inicializar estado oculto h_0 y estado de celda C_0 en cero
 - **Para** cada paso temporal $t = 1$ hasta T :
 - Calcular compuerta de olvido: $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
 - Calcular compuerta de entrada: $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$
 - Calcular candidatos de celda: $\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$
 - Actualizar celda: $C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$
 - Calcular compuerta de salida: $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$
 - Calcular estado oculto: $h_t = o_t * \tanh(C_t)$
 - **Fin Para**
 - Calcular pérdida y aplicar retropropagación a través del tiempo (BPTT) para actualizar W y b
3. **Fin Para**

Algorithm 5: Entrenamiento de una red LSTM

Entrada: secuencia de prueba x_1, x_2, \dots, x_T , parámetros entrenados W, b

Salida: clase predicha \hat{y}

Proceso:

1. Inicializar estado oculto h_0 y estado de celda C_0 en cero
2. **Para** cada paso temporal $t = 1$ hasta T :
 - Calcular $f_t, i_t, \tilde{C}_t, C_t, o_t$, y h_t como en el entrenamiento
3. **Fin Para**
4. Usar h_T (último estado oculto) para calcular la salida final
5. Obtener predicción: $\hat{y} = \arg \max(\text{softmax}(W_y \cdot h_T + b_y))$
6. Retornar clase \hat{y}

Algorithm 6: Clasificación de una secuencia con LSTM

3.3.3.2. GRU

Las redes *Gated Recurrent Unit (GRU)* fueron introducidas por Kyunghyun Cho y sus colaboradores en 2014 en su trabajo “*Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*” [94].

GRU es una arquitectura simplificada de redes neuronales recurrentes que constituye una variante más eficiente de las LSTM. Las GRU utilizan únicamente dos compuertas (*reset* y *update*) en lugar de las tres compuertas de las LSTM, lo que reduce la complejidad computacional mientras mantiene la capacidad de modelar dependencias a largo plazo [90, 95]. Esta arquitectura combina el estado oculto y el estado de celda en una sola representación, simplificando el flujo de información y reduciendo el número de parámetros necesarios para el entrenamiento [96].

3.3.3.2.1 Funcionamiento

- La GRU utiliza un estado oculto único (h_t) que sirve tanto como memoria a largo plazo como salida de la unidad [94].
- La **compuerta de reset** (r_t) determina cuánta información del estado anterior debe ser olvidada. Se calcula como: $r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$ [95].
- La **compuerta de update** (z_t) controla cuánta información del estado anterior se mantiene y cuánta información nueva se añade. Se calcula como: $z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$ [94].
- El **estado candidato** (\tilde{h}_t) se calcula utilizando la compuerta de reset para modular la influencia del estado anterior: $\tilde{h}_t = \tanh(W_h \cdot [r_t * h_{t-1}, x_t] + b_h)$ [95].
- El **estado oculto final** se obtiene como una interpolación lineal entre el estado anterior y el estado candidato, controlada por la compuerta de update: $h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$ [94].
- El proceso se repite iterativamente para cada paso temporal, permitiendo que la red capture patrones secuenciales complejos con menor costo computacional que LSTM [95].

3.3.3.2.2 Pseudocódigo El algoritmo GRU se implementa siguiendo los pasos a continuación descritos, se muestra el pseudocódigo correspondiente a la fase de entrenamiento y después se muestra el pseudocódigo de la fase de clasificación [90]:

Entrada: secuencias de entrenamiento $X = \{x_1, x_2, \dots, x_T\}$ con sus etiquetas Y , número de épocas E , tasa de aprendizaje α

Salida: pesos W y sesgos b entrenados

Proceso:

1. Inicializar pesos W y sesgos b aleatoriamente
2. **Para** $e = 1$ hasta E :
 - Inicializar estado oculto $h_0 = 0$
 - **Para** $t = 1$ hasta T :
 - Calcular compuerta de reseteo: $r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$
 - Calcular compuerta de actualización: $z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$
 - Calcular estado candidato: $\tilde{h}_t = \tanh(W_h \cdot [r_t * h_{t-1}, x_t] + b_h)$
 - Actualizar estado oculto: $h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$
 - **Fin Para**
 - Calcular la pérdida y aplicar retropropagación a través del tiempo (BPTT) para actualizar W y b
3. **Fin Para**

Algorithm 7: Entrenamiento de una red GRU

Entrada: secuencia de prueba x_1, x_2, \dots, x_T , pesos W , sesgos b entrenados

Salida: clase predicha \hat{y}

Proceso:

1. Inicializar estado oculto $h_0 = 0$
2. **Para** $t = 1$ hasta T :
 - Calcular compuerta de reseteo: $r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r)$
 - Calcular compuerta de actualización: $z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z)$
 - Calcular estado candidato: $\tilde{h}_t = \tanh(W_h \cdot [r_t * h_{t-1}, x_t] + b_h)$
 - Actualizar estado oculto: $h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$
3. **Fin Para**
4. Calcular salida final a partir de h_T
5. Obtener clase predicha: $\hat{y} = \arg \max(\text{softmax}(W_y \cdot h_T + b_y))$
6. Retornar \hat{y}

Algorithm 8: Clasificación de una secuencia con GRU

3.3.4. Análisis no supervisado

Pasando ahora al enfoque no supervisado del aprendizaje automático, este paradigma es esencial para el conseguir el objetivo de la investigación, debido a que los datos recopilados

no cuentan con una supervisión, este arquetipo presenta una solución para formar grupos en datos que carecen de una etiqueta, ayudando así a poder identificar los diferentes estilos de conducción. Es por esto mismo que para el desarrollo de este trabajo se emplearan dos algoritmos de agrupamiento con dos enfoques diferentes, también se hace el uso de herramientas estadísticas como PCA pertenecientes al aprendizaje no supervisado

3.3.4.1. Análisis de Componentes Principales (PCA)

El Análisis de Componentes Principales (PCA) fue desarrollado por Karl Pearson en 1901 en su trabajo "*On Lines and Planes of Closest Fit to Systems of Points in Space*" [97]. Posteriormente, Harold Hotelling formalizó y extendió el método en 1933 en el artículo "*Analysis of a Complex of Statistical Variables into Principal Components*" [98]. PCA es una técnica de reducción de dimensionalidad que transforma un conjunto de variables correlacionadas en un conjunto menor de variables no correlacionadas llamadas componentes principales, preservando la mayor cantidad de varianza posible de los datos originales.

3.3.4.1.1 Funcionamiento

Estandarización de datos: Se normaliza el conjunto de datos para que todas las variables tengan media cero y varianza unitaria, evitando que variables con diferentes escalas dominen el análisis [98].

Cálculo de la matriz de covarianza: Se computa la matriz de covarianza de los datos estandarizados para identificar las relaciones lineales entre las variables [97].

Descomposición de valores propios: Se calculan los valores propios y vectores propios de la matriz de covarianza. Los vectores propios definen las direcciones de máxima varianza (componentes principales) [98].

Selección de componentes: Se ordenan los componentes por sus valores propios correspondientes y se seleccionan los k primeros que explican el porcentaje deseado de varianza total [97].

Transformación de datos: Se proyectan los datos originales al espacio de menor dimensión definido por los componentes principales seleccionados [98].

3.3.4.1.2 Pseudocódigo El algoritmo PCA se implementa siguiendo los pasos a continuación descritos [97, 98]:

3.3.4.2. Jerárquico

El algoritmo de agrupamiento jerárquico se formalizó en el artículo de S. C. Johnson en 1967, titulado *Hierarchical Clustering Schemes* [73]. En este artículo, Johnson presentó uno de los primeros enfoques formales para la creación de jerarquías en el agrupamiento de datos, introduciendo los conceptos de agrupamiento jerárquico aglomerativo y divisivo.

Los métodos jerárquicos de agrupamiento son técnicas de agrupación de datos que organizan los puntos en grupos basándose en lo cercanos que están entre sí, pero lo hacen de manera progresiva y en niveles, como si se estuviera construyendo una especie de "árbol familiar" pero con los datos. Esto permite organizar los datos de manera estructurada,

Entrada: matriz de datos $X \in R^{n \times p}$, porcentaje de varianza explicada deseado v_{target}

Salida: nueva representación reducida Y en k componentes principales, con $k \leq p$

Proceso:

1. Estandarizar la matriz de datos: $X_{std} = \frac{X - \mu}{\sigma}$
2. Calcular matriz de covarianza: $C = \frac{1}{n-1} X_{std}^T X_{std}$
3. Calcular valores propios λ_i y vectores propios v_i de C
4. Ordenar los pares (λ_i, v_i) en orden descendente de λ_i
5. Seleccionar el número mínimo de componentes k tal que: $\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^p \lambda_i} \geq v_{target}$
6. Construir la matriz V_k con los k vectores propios seleccionados
7. Transformar los datos: $Y = X_{std} \cdot V_k$
8. Retornar Y

Algorithm 9: Reducción de dimensionalidad usando PCA

formando una jerarquía que puede representarse con un dendrograma. Los métodos jerárquicos se dividen en dos enfoques [99] [100].

3.3.4.2.1 Enfoque Aglomerativo (*Bottom-Up*)

- *Bottom-up* significa que comienza desde abajo, es decir, desde los objetos individuales [100].
- Al inicio, cada objeto forma su propio grupo o clúster.
- A medida que avanzamos, el método fusiona los grupos o los objetos que están más cercanos entre sí. Esta fusión continúa hasta que:
 - Todos los objetos pertenecen a un solo clúster, o
 - Se cumple alguna condición de terminación (por ejemplo, un número específico de clústeres o una distancia mínima entre ellos).
- Es por eso por lo que se le llama "jerárquico aglomerativo": va aglomerando o fusionando pequeños grupos en otros más grandes [100].

3.3.4.2.2 Enfoque Divisivo (*Top-Down*)

- Este enfoque es el opuesto al aglomerativo.
- Aquí, se comienza desde arriba, es decir, todos los objetos están inicialmente en un solo clúster [100].
- En cada paso, se divide un clúster en clústeres más pequeños. Esta división continúa hasta que:

- Cada objeto pertenece a un clúster propio, o
 - Se cumple una condición de terminación.
- Este método se llama "jerárquico divisivo" porque va dividiendo grandes grupos en otros más pequeños [100].

3.3.4.2.3 Pasos para realizar un agrupamiento jerárquico aglomerativo

1. **Cada dato es un grupo independiente:** Al principio, cada punto de los datos es un clúster individual. Si se tienen 100 puntos, entonces habrá 100 grupos, uno para cada punto [100].
2. **Encuentra los dos más cercanos:** El algoritmo mide la distancia entre todos los puntos o grupos, y encuentra los dos que están más cerca según la métrica seleccionada. Entre las más utilizadas se encuentran la distancia euclidiana, Manhattan y coseno [99, 100].
3. **Unir los pequeños grupos en uno más grande:** Una vez que encuentra los dos puntos o grupos más cercanos, los une y los convierte en un solo grupo. Ahora se tiene un grupo que contiene esos dos puntos y otros grupos con los puntos restantes [100].
4. **Definir cómo medir la distancia entre grupos:** A medida que se agrupan puntos, es necesario definir cómo calcular la distancia entre grupos. Esto se determina mediante un criterio de enlace (*linkage*), que puede ser [99]:
 - **Single linkage:** utiliza la distancia mínima entre los elementos de ambos grupos.
 - **Complete linkage:** utiliza la distancia máxima entre los elementos de ambos grupos.
 - **Average linkage:** calcula el promedio de todas las distancias posibles entre los elementos de los grupos.
 - **Ward:** minimiza la varianza dentro de los grupos al fusionarlos.
5. **Repetir el proceso:** El algoritmo sigue buscando los grupos más cercanos (que ahora pueden ser grupos de más de un punto) y los va uniendo poco a poco. Al principio, agrupa pares de puntos, luego agrupa pequeños grupos en grupos más grandes, y así sucesivamente [100].
6. **Hasta que todos estén en un solo grupo:** Este proceso continúa hasta que todos los puntos están unidos en un único grupo que contiene a todos los datos [100].

3.3.4.2.4 Dendrograma: El árbol de relaciones

El resultado de este proceso de agrupación se representa visualmente en un dendrograma, que es un árbol que muestra cómo se unieron los datos en cada paso. Los niveles en el dendrograma indican en qué momento del proceso de agrupación se unieron los puntos o grupos.

- **NOTA:** Si se corta el dendrograma en un cierto nivel, se obtendrán los clústeres finales. Por ejemplo, si se corta el dendrograma en un nivel medio, se obtienen varios grupos de datos en lugar de un solo grupo gigante.

3.3.4.2.5 Pseudocódigo

El algoritmo de clustering aglomerativo se implementa siguiendo los pasos descritos a conti-

Entrada: conjunto de datos $X = \{x_1, x_2, \dots, x_n\}$, número de clústers deseado K (opcional)

Salida: estructura jerárquica de agrupamientos (dendrograma) o partición final con K clústers

Proceso:

1. Inicializar cada punto como un clúster individual
2. Calcular la matriz de disimilitud D entre todos los puntos (o clústers)
3. **Mientras** el número de clústers $> K$ (o hasta que quede uno si K no está definido):

nuación [101]:

- Encontrar los dos clústers C_a y C_b más cercanos según el criterio de enlace (e.g., single, complete, average, Ward)
- Fusionar clústers: $C_{a \cup b} = C_a \cup C_b$
- Calcular cardinalidad del nuevo clúster: $N_{a \cup b} = N_a + N_b$
- Actualizar la matriz de distancias para reflejar las distancias entre $C_{a \cup b}$ y los demás clústers, conservando la información histórica

4. **Fin Mientras**

5. Retornar la jerarquía completa (dendrograma) o la partición final en K clústers

Algorithm 10: Agrupamiento jerárquico aglomerativo

3.3.4.3. K-means

El algoritmo k-means fue introducido por primera vez por Stuart Lloyd en 1957 en el trabajo “*Least squares quantization in PCM*” [102]. K-Means es un algoritmo del aprendizaje no supervisado, en el cual se busca agrupar un conjunto de datos en K grupos de manera que los datos dentro de un grupo sean similares entre sí y distintos de los datos de otros grupos. Se basa en centroides, donde cada grupo está representado por un centroide, que es el punto medio de los datos asignados a ese grupo [87, 100].

3.3.4.3.1 Funcionamiento

- Se elige un número de k -clusters [100].
- Aleatoriamente se asignan centroides iniciales entre todos los datos [100].

- Después de establecer los centroides, se asigna cada patrón del conjunto de datos al centroide más cercano. Normalmente se utiliza la distancia euclidiana para medir la proximidad [99, 100]. Este proceso agrupa los datos en función de qué centroide está más cerca a ellos.
- Luego de que todos los datos se han asignado a un grupo, el centroide de cada grupo se recalcula. El nuevo centroide se obtiene calculando el promedio de todos los puntos de datos en ese grupo, el cual representa la posición media de todos los puntos en ese grupo [100].
- Se repiten los pasos de asignación de puntos y recálculo de los centroides. Continuando el proceso de forma iterativa hasta que los centroides ya no cambien significativamente o hasta el número máximo de iteraciones que se haya estipulado [99, 100].

3.3.4.3.2 Pseudocódigo El algoritmo K-means se implementa siguiendo los pasos a continuación descritos [101]:

Entrada: conjunto de datos $X = \{x_1, x_2, \dots, x_n\}$, número de clusters K

Salida: conjunto de centroides finales $\{\mu_1, \dots, \mu_K\}$, asignaciones de cada punto a un cluster

Proceso:

1. Inicializar K centroides μ_1, \dots, μ_K (aleatoriamente o con algún método como K-Means++)
2. **Repetir:**
 - Asignar cada punto x_i al cluster con el centroide más cercano:

$$\text{cluster}(x_i) = \arg \min_j \|x_i - \mu_j\|^2$$
 - Recalcular los centroides como la media de los puntos asignados a cada cluster:

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$
3. **Hasta que** los centroides no cambien significativamente o se alcance el número máximo de iteraciones
4. Retornar centroides finales y asignaciones de clusters

Algorithm 11: Algoritmo de agrupamiento K-Means

3.3.5. Evaluación

Para realizar una evaluación sobre qué tan bien se ha realizado una partición existen diferentes métricas de validación interna de grupos, las cuales son técnicas para evaluar la calidad de un agrupamiento generado previamente por un algoritmo de agrupación [103] [104].

Estas métricas de validación interna son esenciales para determinar la cantidad óptima de grupos como, por ejemplo, el valor ideal de k en K-means y para seleccionar entre diferentes algoritmos o configuraciones de agrupamiento. Al no depender de etiquetas predefinidas, son ideales en contextos no supervisados donde no se conoce la estructura verdadera de los datos [103].

Entre las principales métricas de validación interna se encuentran:

3.3.5.1. Índice de Calinski-Harabasz

Fue propuesto por T. Calinski y J. Harabasz en 1974 en un artículo titulado “*A dendrite method for cluster analysis*” [105]. Este índice surgió como parte de sus investigaciones en análisis de grupos y se diseñó como una métrica de evaluación interna que no requiere etiquetas conocidas.

Relaciona la dispersión entre los grupos con la dispersión dentro de ellos. Un valor alto indica que los grupos están bien separados y compactos [105].

Fórmula [104]:

$$\text{CH} = \frac{\sum_i n_i d^2(c_i, c) / (NC - 1)}{\sum_i \sum_{x \in C_i} d^2(x, c_i) / (n - NC)}$$

Donde:

- D : Conjunto de datos.
- $d(x, y)$: Distancia entre x e y .
- n : Número de objetos en D .
- c : Centroide de D .
- P : Número de atributos en D .
- NC : Número de grupos.
- C_i : El i -ésimo grupo.
- n_i : Número de patrones en C_i .
- c_i : Centroide de C_i .
- $\sigma(C_i)$: Varianza del vector C_i .

Valores más altos del índice de Calinski-Harabasz indican que los grupos están bien separados y los puntos dentro de cada grupo están cerca entre sí. Esto implica que el agrupamiento es de mejor calidad [105].

Valores más bajos indican que los grupos no están bien definidos, es decir, están más dispersos o solapados [105].

3.3.5.2. Índice de la silueta (Silhouette)

Mide la calidad del agrupamiento evaluando cuán cerca están los puntos de datos dentro de un mismo cluster y cuán lejos están de los puntos en otros clusters. Un valor cercano a 1 indica una buena separación entre clusters [106].

Fórmula [104]:

$$\frac{1}{N_C} \sum_i \left\{ \frac{1}{n_i} \sum_{x \in C_i} \frac{b(x) - a(x)}{\max[b(x), a(x)]} \right\}$$

Donde:

- N_C : Número de grupos.
- n_i : Número de puntos en el grupo C_i .
- $a(x)$: Distancia promedio de un punto x a los otros puntos en su mismo grupo, quiere decir qué tan cerca está de los puntos de su propio grupo.
- $b(x)$: Distancia promedio de un punto x al grupo más cercano que no es el suyo, quiere decir qué tan lejos está un grupo de otros.

3.3.5.3. Índice de Davies-Bouldin

Evalúa la compacidad de los clusters (intra-cluster) y su separación (inter-cluster). Valores más bajos indican grupos mejor formados [106].

Fórmula [104]:

$$\frac{1}{N_C} \sum_i \max_{j, j \neq i} \left\{ \frac{\frac{1}{n_i} \sum_{x \in C_i} d(x, c_i) + \frac{1}{n_j} \sum_{x \in C_j} d(x, c_j)}{d(c_i, c_j)} \right\}$$

Donde:

- N_C : Número de grupos.
- n_i : Número de puntos en el grupo i .
- $d(x, c_i)$: Distancia de un punto x al centro del grupo c_i .
- $d(c_i, c_j)$: Distancia entre los centros de los grupos i y j .

4. Hallazgos en el desarrollo del de clasificador

Este capítulo presenta la evaluación para el desarrollo del sistema de clasificación de patrones de conducción, el cual, abarca desde la implementación de algoritmos de aprendizaje automático hasta su validación en entornos reales. La investigación se estructura en múltiples fases experimentales con la finalidad de generar una robustez en la identificación de los estilos de conducción, sirviendo y aportando una base fuerte y sólida para la creación de un gemelo digital.

El desarrollo experimental se abordó de acuerdo a los hallazgos que hubo durante la fase de experimentación para determinar los estilos de conducción de las diferentes sujetos.

Abordando diferentes metodologías, partiendo del estado del arte en donde, se hace mención de que para identificar los estilos de conducción se puede realizar partiendo directo desde el agrupamiento, con el desarrollo de esta investigación, resultó no ser así, para identificar los estilos es necesario tener una referencia con datos controlados, es este caso pista, sin esa información no es posible identificar los estilos de conducción por lo que en este capítulo se aborda el proceso de completo para identificar los estilos de conducción de las sujetos, tanto en un entorno controlado, así con el arroyo vehicular siguiendo estos pasos se logra realizar una caracterización exhaustiva de los patrones de conducción:

Por lo que para describir detalladamente el proceso de la caracterización de los estilos de conducción, se dividió en cuatro etapas este capítulo, desarrollando lo que se realizó para llevar al cumplimiento del objetivo.

1. **Desarrollo del clasificador de maniobras:** Se implementaron y evaluaron algoritmos de aprendizaje automático y aprendizaje profundo para la identificación automática de las maniobras de conducción establecidas en el capítulo 3, las cuales incluyen técnicas tradicionales del aprendizaje automático (K-NN, Random Forest) hasta arquitecturas avanzadas del aprendizaje profundo (LSTM, GRU).
2. **Empleo del clasificador de maniobras para identificar los estilos de conducción en arroyo vehicular:** Los algoritmos entrenados con datos de pista se utilizaron como base para aplicarlos en el arroyo vehicular. Este proceso fue fundamental para entender, como afecta el entorno controlado en pista y como el análisis empleado en pista puede ser aplicado en los datos de ciudad.

3. **Análisis de agrupamiento para identificación de estilos de conducción:** Se aplicaron técnicas de agrupamiento del paradigma no supervisado para identificar patrones en los estilos de conducción distintivos entre conductores.
4. **Establecimiento de fundamentos para implementación en gemelos digitales:** Se definieron los criterios técnicos y metodológicos necesarios para la integración del sistema de clasificación en arquitecturas de gemelos digitales vehiculares.

Los resultados experimentales presentados en este capítulo constituyen la base empírica para validar las hipótesis de investigación planteadas y demostrar la viabilidad técnica de la metodología propuesta para su implementación en aplicaciones del mundo real. La evaluación multidimensional permite no solo identificar el algoritmo de mejor rendimiento, sino también comprender las limitaciones, fortalezas y oportunidades de mejora que orientarán el desarrollo futuro de sistemas inteligentes de análisis vehicular.

4.1. Desarrollo del clasificador de maniobras

Para desarrollar un clasificador de maniobras de conducción, se establecen los siguientes puntos, con el fin de asegurar un proceso adecuado en la construcción de un clasificador inteligente de patrones a partir de los datos obtenidos en pista:

- **Contar un conjunto de datos:** Como se menciona en la sección anterior, se cuenta con dos conjuntos de datos, uno recopilado con datos de un entorno en pista y otro de un entorno urbano o ciudadano, el cual tuvo un proceso de limpieza previo.
- **Seleccionar un método de validación:** En el desarrollo de esta investigación fueron seleccionados tres enfoques diferentes para el método de validación: *Hold-out validation*, *K-Fold Cross validation* y *Walk Forward Validation*
- **Selección de los algoritmos:** Fueron seleccionados diferentes algoritmos con dos enfoques diferentes, aprendizaje automático y aprendizaje profundo.
- **Medida de desempeño:** Con la finalidad de comparar el rendimiento de cada uno de los métodos y algoritmos seleccionados y destacar aquellos que mejores resultados tuvieron, se seleccionaron las siguientes medidas de desempeño: *Recall*, *Precision* y *F1-Score*

4.1.1. Configuración experimental para el desarrollo del clasificador

Normalización: Primero se aplicó normalización ($\mu = 0$, $\sigma = 1$) a todas las variables para garantizar comparabilidad entre algoritmos.

4.1.1.1. Estrategias de validación

Los valores seleccionados para los diferentes métodos de validación son los siguientes:

- *Hold-out*: 80 % entrenamiento y 20 % prueba.
- *K-Fold Cross Validation*: $K=5$.
- *Walk-forward cross validation*: Ventana de entrenamiento 80 %, ventana de prueba 20 %, paso de avance del 10 % y un número de iteraciones de cinco ventanas deslizantes.

4.1.1.2. Configuración de los hiperparámetros

Se establecieron configuraciones específicas de hiperparámetros para cada algoritmo seleccionado. Las configuraciones implementadas se presentan a continuación:

K-Nearest Neighbors (KNN):

- Valores de K evaluados: 3, 5, 7
- Función de distancia: Euclidiana

Random Forest:

- Número de árboles: 100 y 150
- Profundidad máxima: 10
- Criterio de división: Gini

Multilayer Perceptron (MLP):

- Configuración 1: 1 capa oculta con 50 neuronas
- Configuración 2: 2 capas ocultas (100 y 50 neuronas)
- Función de activación: ReLU
- Optimizador: Adam

LSTM y GRU: Las configuraciones establecidas para el entrenamiento de LSTM y GRU se presentan en la Tabla 4.1.

Tabla 4.1: Comparación de parámetros entre modelos LSTM y GRU

Hiperparámetros	LSTM	GRU
Capas ocultas	128 por capa	128 por capa
Número de capas	2	2
<i>Timestep</i> de entrada	50	50
<i>Dropout</i>	0.3	0.3
Bidireccional	Sí	Sí
Optimizador	AdamW (lr=0.001)	AdamW (lr=0.001)
<i>Batch size</i>	256	256

4.1.2. Evaluación experimental y selección de algoritmos

Una vez establecidas las configuraciones iniciales de los métodos de validación como de los hiperparámetros, se obtuvo un vasta cantidad de resultados, siendo este el motivo por cual, se decidió presentar únicamente los resultados de aquellos algoritmos que mostraron el mejor rendimiento en las métricas de evaluación.

Fueron identificados tres modelos con rendimiento superior que además representan diferentes paradigmas del aprendizaje automático se obtuvo; *k-Nearest Neighbors* (k=3) con el método de validación cruzada el que obtuvo mejor desempeño y por otra parte las redes neuronales recurrentes LSTM y GRU como representantes del aprendizaje profundo son los otros algoritmos, con el métodos de validación cruzada de avance (*Walk Forward Validation*).

Una vez estableciendo lo anterior. A continuación se presentan los resultados mediante matrices de confusión y sus respectivas medidas de desempeño por el clasificador de maniobras de conducción en pista.

4.1.2.1. K-Nearest Neighbors (k=3)

El algoritmo KNN con k=3 mostró un rendimiento de 99.0% en F1-score *macro average* (promedio macro) seleccionado para proporcionar igual ponderación a cada clase de maniobra independientemente de su frecuencia en el conjunto de datos, demostrando una excelente capacidad de clasificación. La Figura 4.1 presenta la matriz de confusión expresada en porcentaje, mientras que la Tabla 4.2 detalla las métricas por clase.

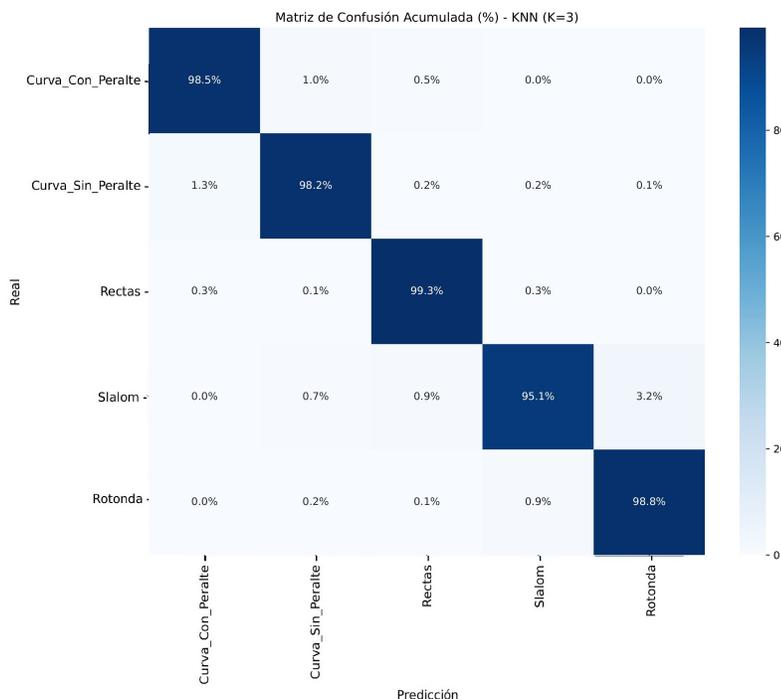


Figura 4.1: Matriz de confusión para k -Nearest Neighbors ($k=3$) para datos de pista

Tabla 4.2: Resultados por clase: Precision, Recall y F1-score del KNN con método de validación K -fold

Clase	Precision (%)	Recall (%)	F1-score (%)
Curva Con Peralte	99%	99%	99%
Curva Sin Peralte	99%	99%	99%
Rectas	100.0%	100.0%	100.0%
Slalom	100.0%	96.9%	98.4%
Rotonda	97.0%	100.0%	98.5%
Macro avg	99.0%	99.0%	99.0%

Los resultados del KNN muestran un desempeño bastante bueno en la identificación de las distintas maniobras vehiculares. Se observa que tanto las métricas de precisión, recall y F1-score son consistentes y elevadas en todas las clases evaluadas, con valores cercanos o iguales a 1.00, lo cual indica una alta capacidad del modelo para realizar predicciones correctas en cada maniobra.

Las maniobras Rectas, Curva Con Peralte y Curva Sin Peralte alcanzaron métricas casi perfectas.

En el caso de Slalom y Rotonda, aunque hay una ligera disminución en el recall y precisión, los valores siguen siendo excelentes, superiores a 0.97%, lo que demuestra que el modelo también maneja bien las clases potencialmente más complejas o con patrones

más variados.

La media *macro average* (macro avg) de 0.99% en todas las métricas confirma que el modelo mantiene un rendimiento equilibrado sin favorecer clases con mayor cantidad de datos, lo cual es especialmente relevante para esta investigación.

4.1.2.2. Long Short-Term Memory (LSTM)

Para el caso de la red LSTM se observa que alcanzó un F1-score en promedio macro de 96.5%, teniendo un desempeño alto, aunque es el menor rendimiento entre los tres algoritmos evaluados. La Figura 4.2 muestra la matriz de confusión, y la Tabla 4.3 presenta las métricas detalladas por clase.

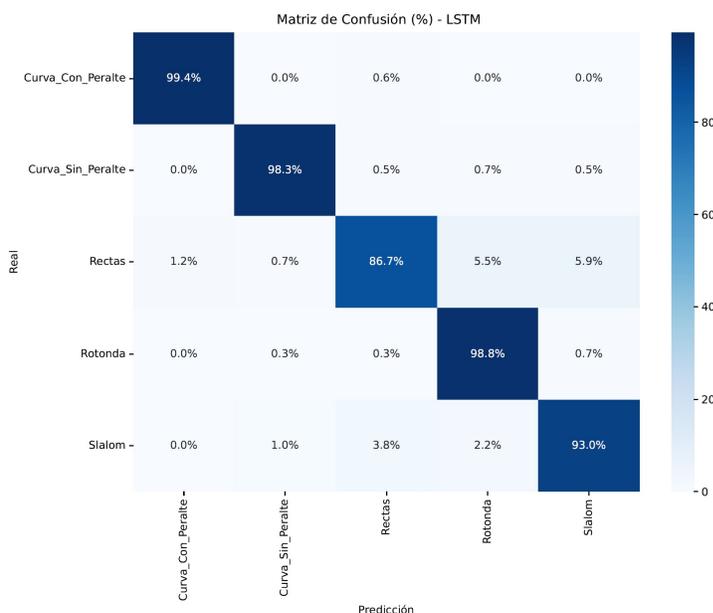


Figura 4.2: Matriz de confusión para LSTM con validación *walk-forward*

Tabla 4.3: Métricas de evaluación para LSTM con método de validación *walk-forward*

Clase	Precisión (%)	Recall (%)	F1-Score (%)
Curva con peralte	99.0 %	100.0 %	99.5 %
Curva sin peralte	99.0 %	100.0 %	99.5 %
Recta	96.6 %	88.7 %	92.5 %
Slalom	93.3 %	100.0 %	96.6 %
Rotonda	94.9 %	93.9 %	94.4 %
Macro avg	96.6 %	96.5 %	96.5 %

El modelo LSTM mostró un comportamiento peculiar, con dificultades específicas en la clasificación de rectas, donde el *recall* disminuye significativamente al 88.7%. En comparación de Slalom que logra un recall perfecto (100%), la maniobra que se le complicó a los otros algoritmos.

4.1.2.3. Gated Recurrent Unit (GRU)

El modelo GRU obtuvo el mejor F1-score promedio macro de 99.4 %, posicionándose como el algoritmo con mejor rendimiento. La Figura 4.3 presenta su matriz de confusión, y la Tabla 4.4 muestra el desempeño del algoritmo.

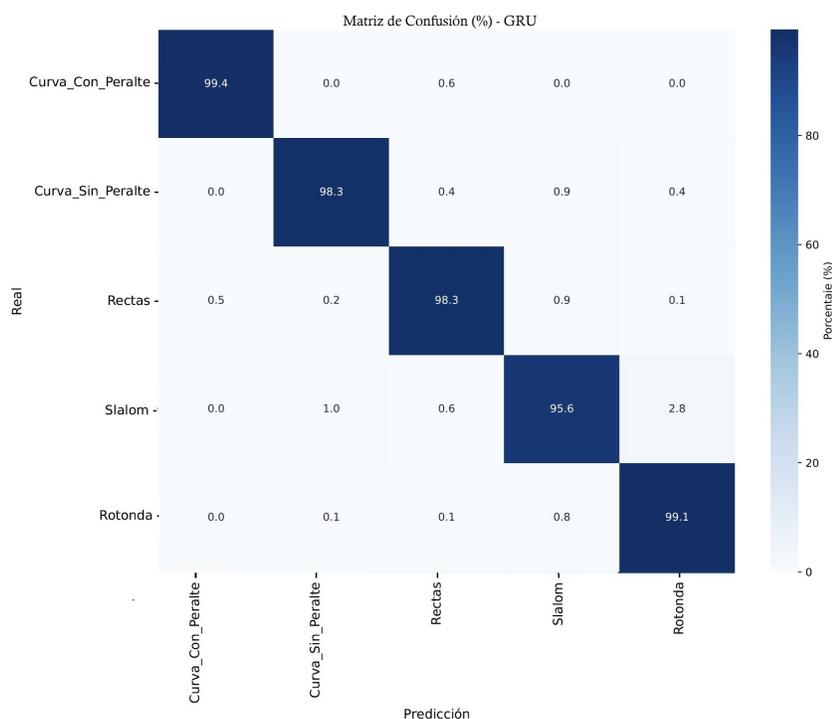


Figura 4.3: Matriz de confusión para GRU validación walk-forward

Tabla 4.4: Métricas de evaluación para GRU con validación walk-forward

Clase	Precisión (%)	Recall (%)	F1-Score (%)
Curva con peralte	100.0 %	100.0 %	100.0 %
Curva sin peralte	99.0 %	100.0 %	99.5 %
Recta	100.0 %	100.0 %	100.0 %
Slalom	100.0 %	96.9 %	98.4 %
Rotonda	98.0 %	100.0 %	99.0 %
Macro avg	99.4 %	99.4 %	99.4 %

Los resultados del GRU indican un rendimiento excepcional, logrando la clasificación perfecta en curvas con peralte y rectas (100 % en todas las métricas). El modelo muestra una alta consistencia en todas las clases, con el único punto débil en el recall del Slalom (96.9 %), patrón que se repite en los algoritmos evaluados. La arquitectura GRU demuestra ser especialmente efectiva para este conjunto de datos.

4.1.2.4. Análisis comparativo del rendimiento

En la Tabla 4.5, se observa que el modelo GRU (99.4% F1-score) sobresale como el mejor clasificador, superando tanto a KNN (99.0%) como LSTM (96.5%). Como particularidad de los resultados obtenidos es el rendimiento de KNN, que con su simplicidad conceptual logra resultados altos (99.0%). Esto indica que las características extraídas de las maniobras de conducción poseen una estructura que permite una clasificación efectiva con métodos basados en proximidad.

Tabla 4.5: Rendimiento promedio macro de los algoritmos seleccionados

Algoritmo	Precisión (%)	Recall (%)	F1-Score (%)
KNN (k=3)	99.0 %	99.0 %	99.0 %
GRU	99.4 %	99.4 %	99.4 %
LSTM	96.6 %	96.5 %	96.5 %

Dado los resultados, anteriores se establece el algoritmo GRU, como el más destacado entre todos los evaluados en este trabajo.

4.2. Empleo del clasificador de maniobras para identificar los estilos de conducción en arroyo vehicular

El análisis de datos urbanos introduce complejidades significativas, originadas por diversos factores, como la variabilidad no controlada del entorno, lo cual, dificultad poder establecer un etiquetado preciso sin condiciones experimentales. Por lo que, en esta subsección se realizará un procedimiento para verificar si los datos etiquetados en pista, pueden ser empleados para los datos del arroyo vehicular, con la finalidad de tener un mejor entendimiento de los estilos de conducción y las diferencias marcadas entre los patrones de pista y la conducción urbana.

4.2.1. Procedimiento con los datos de calle

El proceso de realizar un etiquetado manual con los datos de ciudad, basado en el giro del volante, puede no fungir como un proceso sólido para etiquetar los datos en ciudad. Partiendo de esta premisa, a continuación se analizó si es efectivo este etiquetado, tomando como base el proceso de etiquetado basado en pista.

Primero se debe destacar que no se tiene una forma de medir el desempeño directo del etiquetado manual de los datos de ciudad. Esto es debido a que se cuentan con dos enfoques diferentes que a simple vista sólo se tiene una etiqueta que conside en ambos casos, además de que por un lado se cuenta con tres etiquetas y por el otro cinco. Es por tal motivo, que se necesita establecer una estrategia de comparación, siendo la siguiente:

Estrategia de Comparación:

- **Recta:** Única etiqueta igual en ambos conjuntos de datos. Por lo que, se puede hacer un comparación directa y medir el desempeño en esta maniobra.
- **Giros:** En el etiquetado manual, se colocaron giros tanto a la izquierda como a la derecha, en los datos de pista como se mencionó en el capítulo 3 se cuenta con un etiquetado basado en las diferentes maniobras con que cuenta la pista. Por tal motivo, se tiene que, para que sea considerado un acierto la etiqueta predicha debe de coincidir con: Curva Con Peralte, Curva Sin Peralte o Rotonda, en caso contrario será un error.

Consignando lo anterior, se procede a describir los resultados obtenidos y observar si el etiquetado manual es realmente de calidad o se deben de establecer otros criterios que influyen directamente con el estilo de conducción de cada sujeto.

Como se ha descrito, se tiene una comparación de dos formas de etiquetado diferente, generando así un matriz rectangular, como se muestra en la Figura 4.4. Esta matriz se construye ubicando las cinco etiquetas (consideras como real) de maniobras de pista (Curva Con Peralte, Curva Sin Peralte, Recta, Slalom, Rotonda) en las filas, mientras que las tres categorías urbanas (Recta, Giro Izquierda, Giro Derecha), considerada como la predicción.



Figura 4.4: Ejemplo de la matriz rectangular

Para evaluar la calidad del etiquetado manual por sujeto, fue necesario reentrenar los modelos que tuvieron mejor desempeño identificados en la sección anterior. Este proceso de reentrenamiento consistió en sólo tomar los datos de ese sujeto, sin considerar a todos los demás sujetos empleados en esta investigación. Con esto se garantiza que la evaluación del etiquetado manual será acorde únicamente a ese sujeto.

4.2.2. Resultados

Se seleccionaron tres matrices rectangulares representativas que ilustran resultados resultados con contrastes diferentes: el caso que mejor rendimiento, el caso con un menor desempeño que revela las limitaciones del enfoque y un caso donde se tiene un balance equilibrado, patrones consistentes sin valores extremos.

Los resultados del Sujeto 1 con GRU muestran un rendimiento bueno del 93.8% en la identificación de maniobras recta. La Figura 4.5 muestra la matriz en donde identifica las rectas de manera perfecta, teniendo valores de 94.0% para F1-Score, mientras que los giros izquierdos presentan mayor desafío (40.9% F1-Score) como se muestra la Tabla 4.6. El promedio macro de 69.2% F1-Score posiciona este caso como el mejor ejemplo de desempeño.

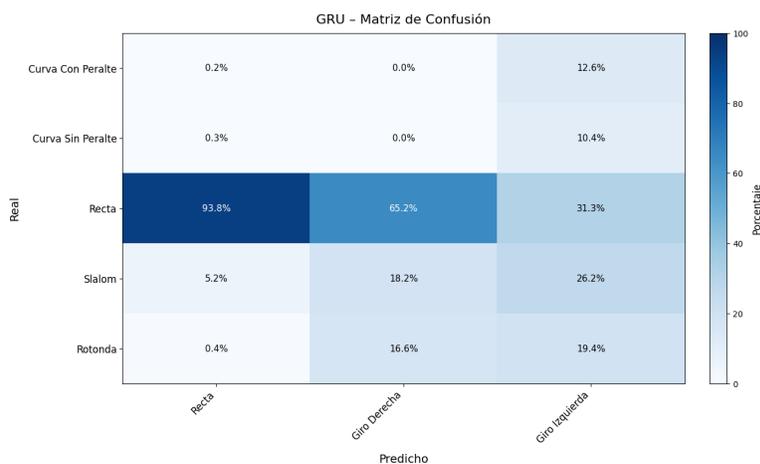


Figura 4.5: Matriz rectangular correspondiente al sujeto 1

Tabla 4.6: Métricas de evaluación - Sujeto 1 con GRU

Categoría Urbana	Precisión (%)	Recall (%)	F1-Score (%)
Recta	94.2	93.8	94.0
Giro Derecha	65.4	81.8	72.7
Giro Izquierda	31.1	59.3	40.9
Promedio Macro	63.6	78.3	69.2

Los resultados del sujeto 5 se ilustran en la Figura 4.7 con LSTM revelan el caso con menor desempeño, con un F1-Score promedio macro de 49.7%. En la Tabla 4.7 se ilustran las limitaciones del sistema cuando existe incompatibilidad entre el etiquetado manual y la arquitectura seleccionada. Las rectas apenas alcanza un 53.9% para F1-Score, contrastando con el sujeto 1.

Tabla 4.7: Métricas de evaluación - Sujeto 3 con LSTM

Categoría Urbana	Precisión (%)	Recall (%)	F1-Score (%)
Recta	54.1 %	53.7 %	53.9 %
Giro Derecha	43.9 %	55.6 %	49.0 %
Giro Izquierda	37.8 %	58.9 %	46.2 %
Promedio Macro	45.3 %	56.1 %	49.7 %

Los resultados de la matriz 4.6 del sujeto 3 con KNN muestran un comportamiento equilibrado, alcanzando un F1-Score promedio de 57.6% como lo muestra la Tabla 4.8.

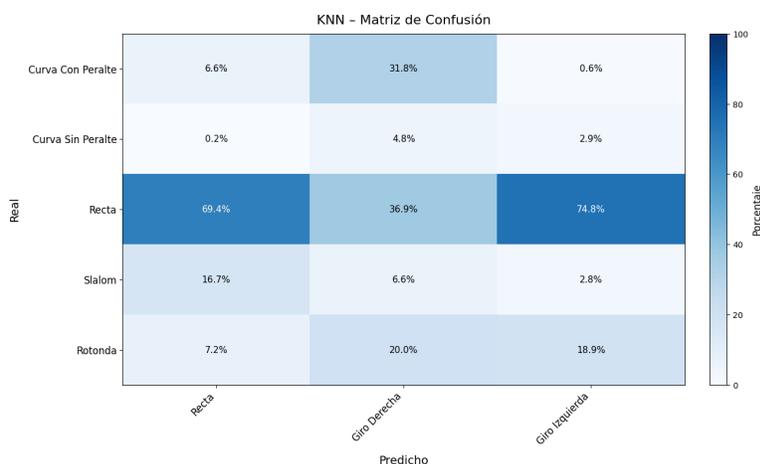


Figura 4.6: Matriz rectangular correspondiente al sujeto 3

La efectividad sólida en rectas (69.6 % F1-Score) confirma que en la identificación de esa maniobra es la que más aciertos tiene en general a comparación de las demás maniobras

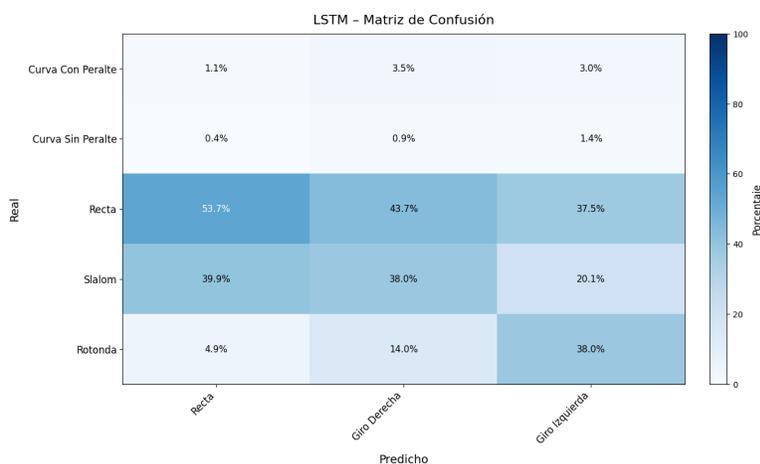


Figura 4.7: Matriz rectangular correspondiente al sujeto 5

Los resultados de *Recall*, *Precision* y *F1-Score* de los 2 sujetos faltantes, se muestran en la tabla 4.9, juntos con los tres sujetos descritos previamente.

Los resultados muestran diferencias entre conductores, con F1-Score que varía desde 0.538 (Sujeto 5) hasta 0.753 (Sujeto 1). Esta variabilidad indica que algunos conductores presentan patrones más distintivos y consistentes que otros. Sujeto 1 y Sujeto 4 presentan los mejores resultados, sugiriendo patrones de conducción más distintivos y consistentes.

Tabla 4.8: Métricas de evaluación - Sujeto 3 con KNN

Categoría Urbana	Precisión (%)	Recall (%)	F1-Score (%)
Recta	69.8 %	69.4 %	69.6 %
Giro Derecha	37.1 %	58.4 %	45.4 %
Giro Izquierda	75.2 %	46.7 %	57.8 %
Promedio Macro	60.7 %	58.2 %	57.6 %

Tabla 4.9: Desempeño por conductor en modelos individuales considerando si mejor desempeño

Conductor	Precisión	Recall	F1-Score
Sujeto 1	0.801	0.709	0.753
Sujeto 3	0.596	0.527	0.559
Sujeto 4	0.662	0.615	0.638
Sujeto 5	0.729	0.696	0.712
Sujeto 6	0.623	0.474	0.538

En cuanto a la maniobras, todos los modelos muestran su mejor desempeño en la clasificación de conducción recta. Este comportamiento es esperado dado que la conducción recta presenta menos variabilidad en los patrones de movimiento.

Los resultados experimentales confirman la viabilidad técnica del proceso de etiquetado manual propuesto como una solución práctica para datos no etiquetados. Sin embargo, el análisis de las métricas de desempeño revela que esta aproximación no logra los resultados esperados.

Las limitaciones identificadas en el rendimiento del sistema sugieren la necesidad de incorporar un conjunto más amplio y diversificado de variables que permitan una mejor diferenciación entre las maniobras ejecutadas en entornos urbanos. El método actual propuesto, basado principalmente en el ángulo del volante, resulta insuficiente para capturar la complejidad inherente a los patrones de conducción en condiciones reales de tráfico urbano.

4.3. Análisis de agrupamiento para la identificación de estilos de conducción

La identificación de los estilos de conducción empleando técnicas de agrupamiento es una de las fases fundamental en el desarrollo de sistemas inteligentes de clasificación. Este método permite descubrir patrones en los datos vehiculares sin requerir etiquetas predefinidas, proporcionando una base objetiva para la categorización posterior de comportamientos de conducción.

Los resultados del análisis de agrupamiento proporcionan la base para un etiquetado automático de los datos, permitiendo posteriormente la implementación de modelos de

clasificación supervisada que pueden generalizar el reconocimiento de estilos de conducción a nuevos datos no vistos.

4.3.1. Análisis de Componentes Principales (PCA)

El proceso de agrupamiento se inició con la **normalización estandarizada** de todos los datos vehiculares recopilados de los participantes. Esta normalización z-score se aplicó de manera global a todo el conjunto de datos, garantizando que todas las variables vehiculares (velocidad, aceleración, RPM, ángulo de volante, etc.) contribuyan equitativamente al análisis posterior, eliminando el sesgo que podría introducir diferentes escalas y unidades de medición.

Posterior a la normalización, se implementó un análisis de componentes principales (PCA) individualizado para cada participante, con el objetivo de determinar el número óptimo de variables que explican al menos el **90 % de la varianza total** en los datos. Este umbral se estableció para garantizar que la reducción dimensional preserve la información más relevante para caracterizar los patrones de conducción.

Teniendo en cuenta la mayoría de los casos, tanto en condiciones de pista como de ciudad, se identifican consistentemente entre 6 y 7 variables como las más influyentes para cada conductor y por tipo de maniobra.

Una vez determinado la varianaza explicada por cada sujeto, el análisis continuó implementado PCA como una herramienta de **selección de variables**, para descubrir qué variables tienen mayor influencia en la estructura general del conjunto de datos. PCA transforma las variables originales en nuevas combinaciones llamadas componentes principales, que capturan la mayor parte de la variabilidad de los datos. Cada componente es una combinación lineal de las variables originales, y a cada variable se le asigna una “carga” que indica cuánto contribuye a esa combinación. Al sumar el valor absoluto de estas cargas en las primeras componentes (las más importantes), se obtiene una medida de la influencia total de cada variable. Las variables con mayor suma de cargas absolutas son las que más aportan a explicar el comportamiento del sistema y, por tanto, se consideran las más relevantes o influyentes.

Este enfoque permite mantener las **variables originales interpretables**, en lugar de trabajar con componentes principales abstractas, facilitando la interpretación de los resultados del agrupamiento. Las variables con mayor importancia total son aquellas que contribuyen significativamente a múltiples componentes principales, indicando su capacidad para capturar variabilidad en diferentes aspectos del comportamiento de conducción.

Se aplicó PCA para determinar las variables más significativas para cada sujeto y también se determinaron las variables más representativas para la segmentación de las maniobras.

Entre las **variables más frecuentes** entre los diferentes sujetos son:

- RollRate (°/s)

-
- PitchRate ($^{\circ}/s$)
 - Brake Posi (%)
 - Accelerato (%)
 - RPM (RPM)
 - Z_Accel (g) y X_Accel (g)

Por otro lado, entre las **variables más influyentes** según el tipo de maniobra, se tiene:

- RollRate ($^{\circ}/s$)
- PitchRate ($^{\circ}/s$)
- YawRate ($^{\circ}/s$)
- Brake Posi (%)
- Accelerato (%)
- RPM (RPM)
- Z_Accel (g)
- Y_Accel (g)

Una vez identificadas cuáles son las variables más significativas por por cada sujeto y por maniobra, se procedió con el proceso de agrupamiento.

4.3.2. Estrategias de Agrupamiento Implementadas

El análisis de agrupamiento se ejecutó mediante, se baso en dos método diferentes, descritos de la siguiente manera:

Enfoque por sujeto: Se aplicaron los algoritmos K-means y agrupamiento jerárquico a los datos recopilados por cada sujeto.

Enfoque por Maniobras: Los algoritmos de agrupamiento se aplicaron de manera segmentada a cada tipo de maniobra identificada previamente. Para el conjunto de datos de **pista**, se analizaron las cinco maniobras: rectas, curvas con peralte, curvas sin peralte, rotondas y slalom. Para el conjunto de datos de **ciudad**, el análisis se enfocó en tres maniobras principales: rectas, giros a la derecha y giros a la izquierda.

4.4. Determinación del número óptimo de estilos de conducción en pista

Para determinar el número de estilos de conducción en pista, se realizó el proceso de agrupamiento considerando los datos de los diez sujetos, es decir, se concatenaron todos los datos y mediante los índices de validación interna *Davies-Bouldin* y *Calinski-Harabasz*, con esto se determino el número óptimo de estilos de conducción (valor de k).

Una vez defino como se determinó el número de estilos de conducción en pista, en la Tabla 4.10 se presenta los valores de las métricas de validación interna para el algoritmo de agrupamiento aglomerativo jerárquico

Tabla 4.10: Métricas de Validación - Agrupamiento Jerárquico

K	Calinski-Harabasz	Davies-Bouldin
2.0	347965.90625	1.482
3.0	371615.6875	1.255
4.0	311936.25	1.791
5.0	278115.84375	1.701

Análisis de resultados jerárquicos:

- **Máximo Calinski-Harabasz:** k=3 con valor de 371615.6875
- **Mínimo Davies-Bouldin:** k=3 con valor de 1.255
- **Convergencia óptima:** Ambas métricas convergen en k=3 como valor óptimo

Mientras que, en la Tabla 4.11 se presenta los valores de las métricas de validación interna para el algoritmo *K-means*

Tabla 4.11: Métricas de Validación - Algoritmo K-means

K	Calinski-Harabasz	Davies-Bouldin
2.0	419318.8352860507	1.7014203642040003
3.0	464459.2146353363	1.4259487458171567
4.0	373281.1809331588	1.3860224439829563
5.0	342013.7238210277	1.5871939226724212
6.0	320320.6227855207	1.5258872535126935
7.0	304235.7053734544	1.6387237362352034
8.0	285781.5980009265	1.6117155852911806
9.0	272977.7805808803	1.5545020670838322
10.0	258040.1131050068	1.5397627432355905

Análisis de resultados k-means:

- **Máximo Calinski-Harabasz:** $k=3$ con valor de 464459.2146353363
- **Análisis Davies-Bouldin:** Mínimo absoluto en $k=4$ (1.386), pero $k=3$ presenta un valor competitivo (1.426)
- **Tendencia general:** Decrecimiento monotónico del índice Calinski-Harabasz después de $k=3$

Para determinar un valor final de valor de K , en la Tabla 4.12 se muestra un análisis en conjunto de los dos algoritmos de agrupamiento, para determinar el valor final de grupos, que definirán el número de los estilos de conducción identificados en este estudio.

Tabla 4.12: Comparación de Métricas por Algoritmo

K	Calinski-Harabasz		Davies-Bouldin		Consenso
	Jerárquico	K-means	Jerárquico	K-means	
2.0	347965.91	419318.84	1.482	1.701	Subóptimo
3.0	371615.69	464459.21	1.255	1.426	Óptimo
4.0	311936.25	373281.18	1.791	1.386	Competitivo
5.0	278115.84	342013.72	1.701	1.587	Declinante

Justificación de $k=3$:

1. **Convergencia algorítmica:** Ambos algoritmos identifican $k=3$ como el valor con mayor índice Calinski-Harabasz
2. **Optimización de Davies-Bouldin:** El agrupamiento jerárquico alcanza su mínimo global en $k=3$
3. **Interpretabilidad práctica:** $k=3$ proporciona una clasificación manejable y conceptualmente clara

Una vez realizado el análisis del número óptimo de estilos de conducción para el entorno de pista es tres, esto se fundamenta en múltiples criterios de robustez:

- **Consenso algorítmico:** Dos algoritmos independientes convergen en $k=3$
- **Optimización múltiple:** Máximo Calinski-Harabasz en ambos algoritmos
- **Interpretabilidad:** Los tres grupos resultantes muestran características distintivas claras
- **Estabilidad numérica:** Diferencias significativas respecto a $k=2$ y $k=5$
- **Validación visual:** Consistencia con patrones observados en análisis gráfico

Estipulado lo anterior, se realizó la caracterización de los estilos de conducción. Para ello, retomaremos el capítulo 2 del estado de arte, en donde, según las investigaciones de *Driving style recognition and comparisons among driving tasks based on driver behavior in the online car-hailing industry* [107], *Aggressive driving behaviors: are there psychological and attitudinal predictors?* [108], *Driver Behavior Classification: A Systematic Literature Review* [14], *A Review of Research on Driving Styles and Road Safety* [109] y *Driving Behavior and Traffic Safety: An Acceleration-Based Safety Evaluation Procedure for Smartphones* [110], es posible identificar estilos de conducción basados en variables vehiculares. Por lo tanto, los estilos de conducción identificados en este estudio, como se había mencionado antes son tres, siendo los siguientes:

4.5. Estilo Agresivo: Riesgo Elevado

Definición: La conducción agresiva se define como “*aceleración alta y desaceleración alta*”, caracterizada por la búsqueda de dominancia en el tráfico y mayor exposición al riesgo.

Características principales:

- **Variables de alta intensidad:** Accelerator (%), YawRate (°/s), Steering Angle (Degrees), X_Accel (g), Y_Accel (g)
- **Comportamiento observado:** Aceleración intensa, giros bruscos, cambios rápidos de dirección
- **Contexto típico:** Maniobras de adelantamiento, curvas cerradas, arranques rápidos

Criterios de detección:

$$\text{Agresivo} = \begin{cases} \text{Verdadero} & \text{si } P_{\text{agresividad}} > 0.65 \text{ o } (P_{\text{velocidad}} > 0.65 \wedge P_{\text{agresividad}} > 0.45) \\ \text{Falso} & \text{en caso contrario} \end{cases} \quad (4.1)$$

donde $P_{\text{agresividad}}$ representa la puntuación promedio de las variables de agresividad y $P_{\text{velocidad}}$ la puntuación promedio de las variables de velocidad.

4.6. Estilo Tranquilo: Riesgo bajo

Definición: El estilo tranquilo se caracteriza por “*desaceleraciones no bruscas y aceleración baja*”, priorizando la paciencia y el control adaptativo de la situación vial.

Características principales:

- **Variables de alta intensidad:** Brake Position (%), RPM moderado
- **Variables de baja intensidad:** Accelerator (%), YawRate (°/s), velocidades altas
- **Comportamiento observado:** Frenado preventivo, aceleración suave, velocidades moderadas

- **Interpretación:** Prioriza seguridad y confort mediante control adaptativo

Criterios de detección:

$$\text{Tranquilo} = \begin{cases} \text{Verdadero} & \text{si } (P_{\text{agresividad}} < 0.45 \wedge P_{\text{tranquilidad}} > 0.45) \text{ o } (P_{\text{velocidad}} < 0.45 \wedge P_{\text{agresividad}} < 0.2) \\ \text{Falso} & \text{en caso contrario} \end{cases} \quad (4.2)$$

donde $P_{\text{tranquilidad}}$ representa la puntuación promedio de las variables asociadas con comportamiento tranquilo.

4.7. Estilo Equilibrado: Riesgo moderado

Definición: La conducción equilibrada se define como aquella donde “*todos los rangos de las variables están en un punto medio*”, manteniendo un balance óptimo entre las diferentes demandas de la conducción.

Características principales:

- **Variables balanceadas:** Todas las variables en rangos intermedios, sin extremos
- **Comportamiento observado:** Conducción moderada, transiciones suaves, adaptabilidad contextual
- **Interpretación:** Equilibrio entre rendimiento y seguridad, uso balanceado de recursos vehiculares

Criterios de detección: **Equilibrado** = Verdadero si las variables están entre 0.25 y 0.65

4.8. Clasificación de Estilos de Conducción

Basándose en el análisis exhaustivo de las gráficas de radar, mostradas a continuación y patrones de agrupamiento mostrados para cada maniobra, se identificaron tres estilos distintivos de conducción entre los 10 sujetos analizados. El análisis se fundamentó tanto en los patrones visuales de los diagramas radar como en la distribución de clusters por variable y por sujeto observados en los gráficos polares de distribución.

En la Figura 4.8, se muestra como esta distribuida cada una de las variables por sujeto para la maniobra recta

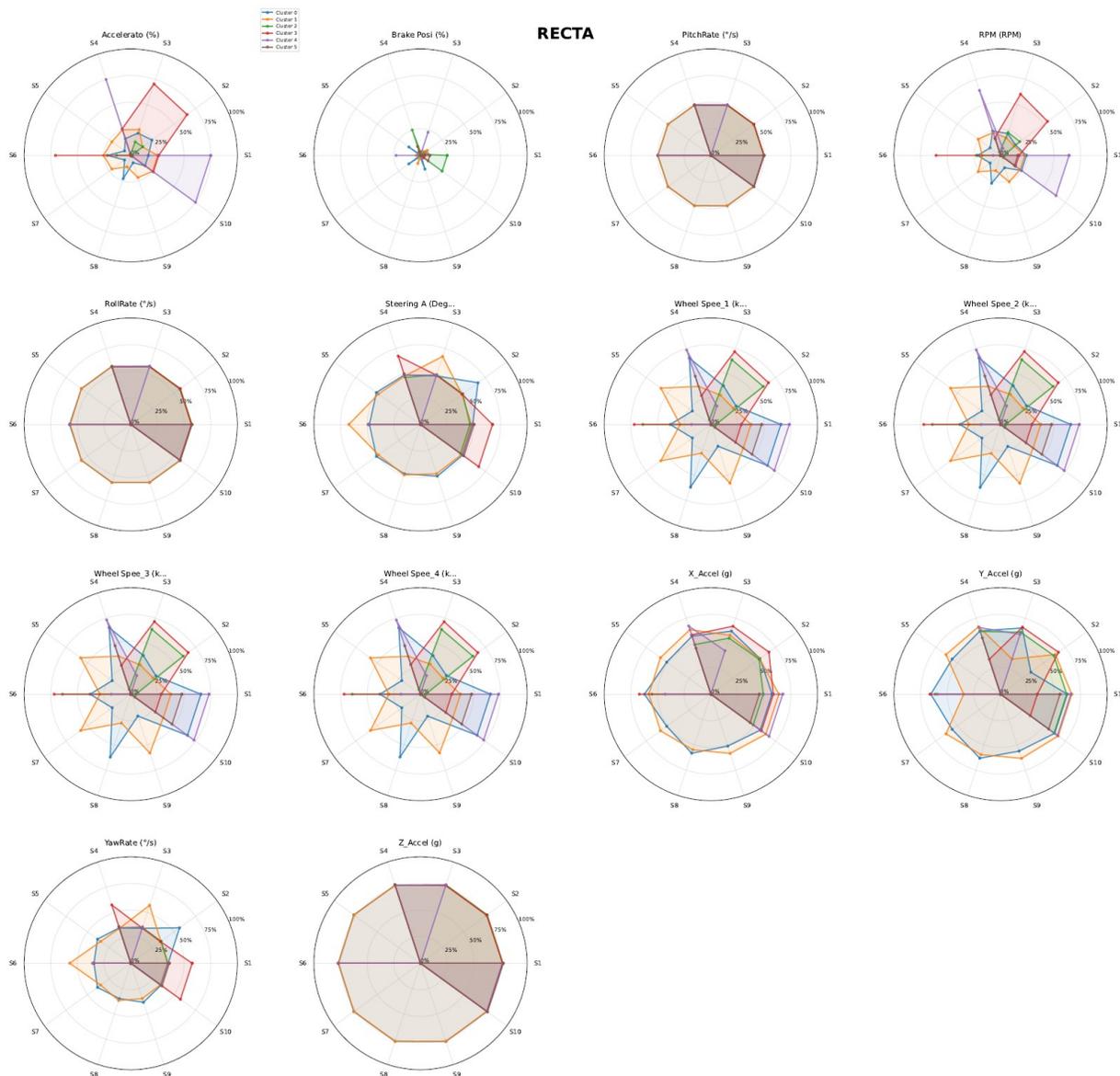


Figura 4.8: Gráficas radar para maniobra de recta mostrando patrones de agrupamiento por sujeto

Por otro lado en la Figura 4.9, muestra como estan distribuidas las variables por cada sujeto para la maniobra curva con peralte.

Estilo Agresivo:

Sujetos: S1, S4, S10 (30 % de la muestra)

Características principales:

- **Alta variabilidad:** Promedio de 6 clusters por análisis K-means
- **Aceleraciones extremas:** Valores altos en X_Accel, Y_Accel y Z_Accel

- **Movimientos bruscos de dirección:** Ángulos de steering amplios y cambios abruptos
- **Velocidades angulares intensas:** YawRate, PitchRate y RollRate elevados
- **Uso intensivo de controles:** Acelerador y frenos en rangos extremos
- **Baja consistencia:** Patrones muy variables entre todas las maniobras

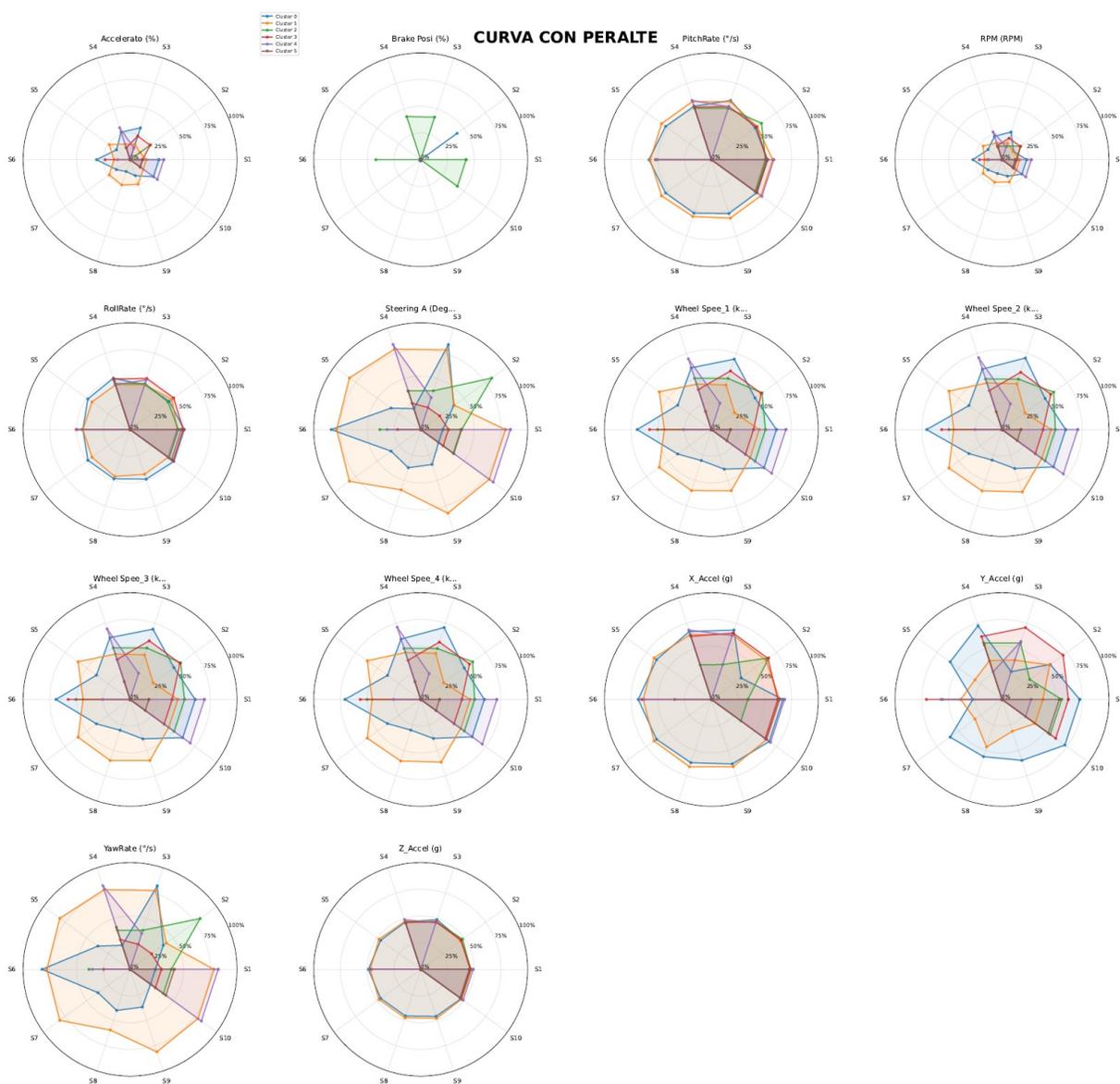


Figura 4.9: Gráficas radar para maniobra de curva con peralte

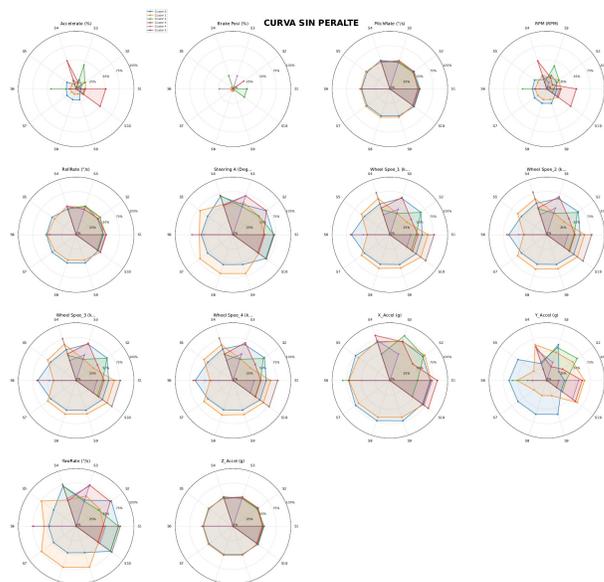


Figura 4.10: Gráficas radar para maniobra de curva sin peralte

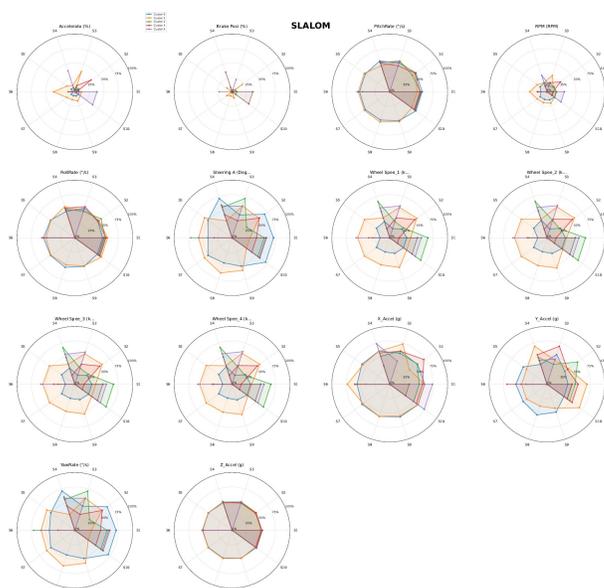


Figura 4.11: Gráficas radar para maniobra de slalom

Patrón observable en gráficas: Como se observa en las Figuras 4.8 a 4.12, las formas de radar se extienden hacia el exterior del diagrama en múltiples variables simultáneamente, mostrando valores extremos y gran dispersión entre grupos. En los diagramas de radar sobre la de distribución de las variables por cada sujeto (Figuras 4.13 a 4.17) se observa que estos sujetos ocupan predominantemente las zonas exteriores (75 %-100 %) en variables críticas como Steering Angle, X_Accel, Y_Accel y YawRate. Particularmente notorio en las maniobras de slalom (Figura 4.11) y rotonda (Figura 4.12), donde sus patrones cubren casi todo el espectro de intensidad disponible.

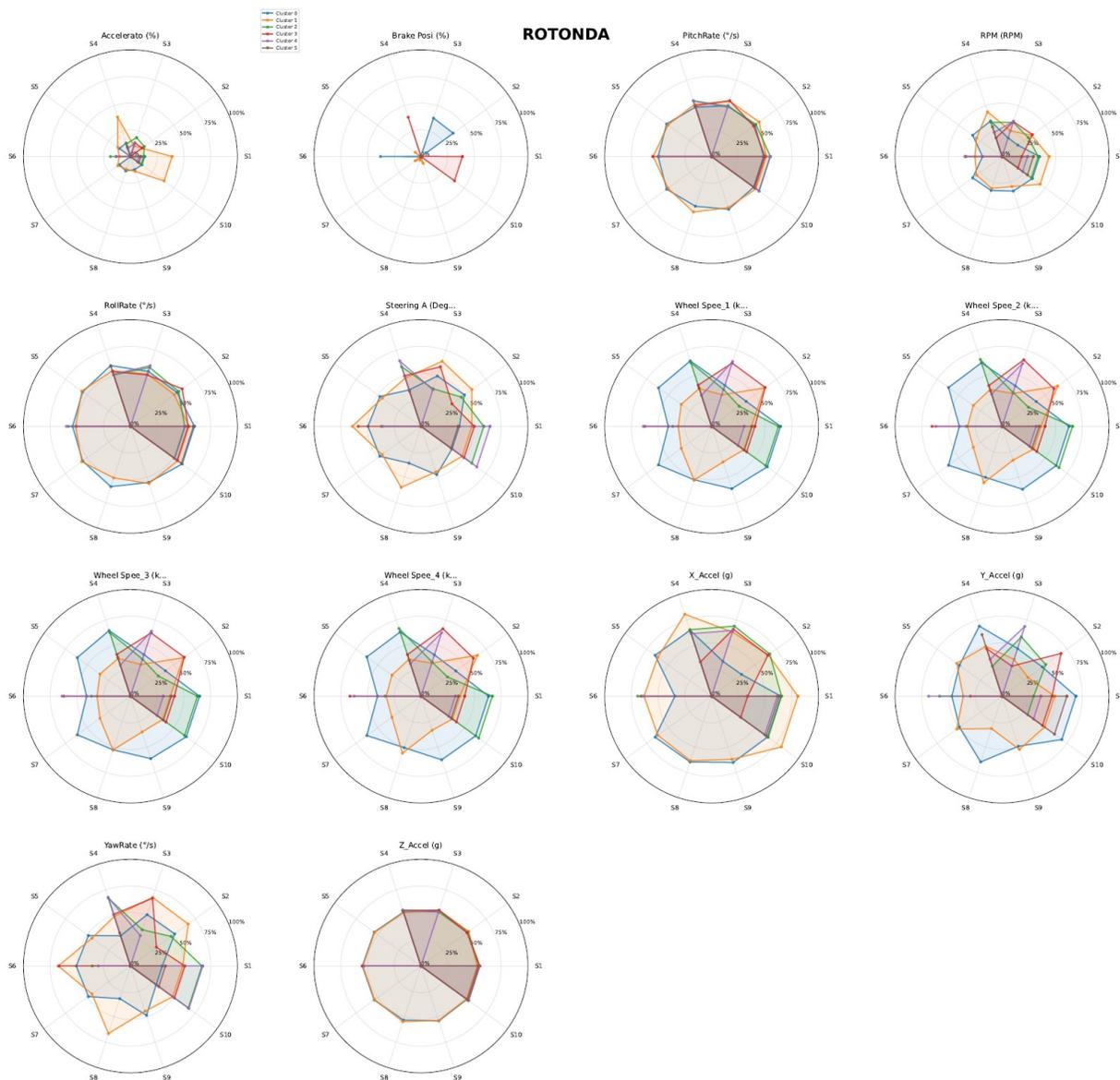


Figura 4.12: Gráficas radar para maniobra de rotonda

Estilo Tranquilo

tranquiloSujetos: S5, S7, S8, S9 (40 % de la muestra)

Características principales:

- **Alta consistencia:** Únicamente 2 clusters en todos los análisis
- **Aceleraciones moderadas:** Valores contenidos en todas las aceleraciones
- **Movimientos suaves:** Steering angle gradual y muy controlado
- **Velocidades estables:** RPM y wheel speeds constantes
- **Uso conservador:** Rangos medios-bajos de acelerador y frenos

- **Máxima consistencia:** Patrones prácticamente idénticos en todas las maniobras

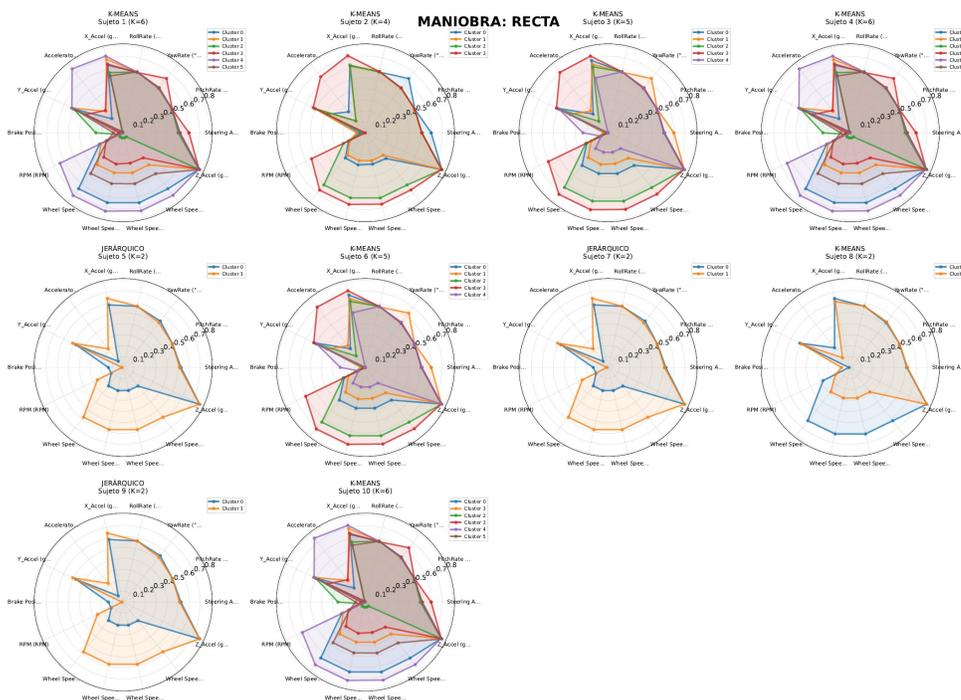


Figura 4.13: Diagramas radar de distribución por variable - Maniobra de recta

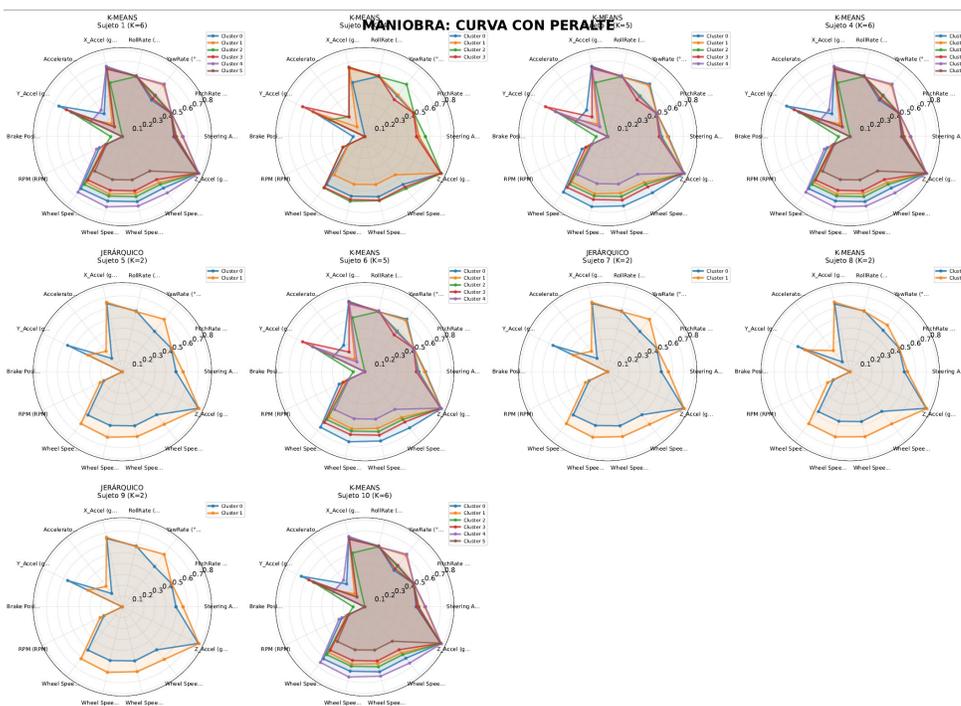


Figura 4.14: Diagramas radar de distribución por variable - Curva con peralte

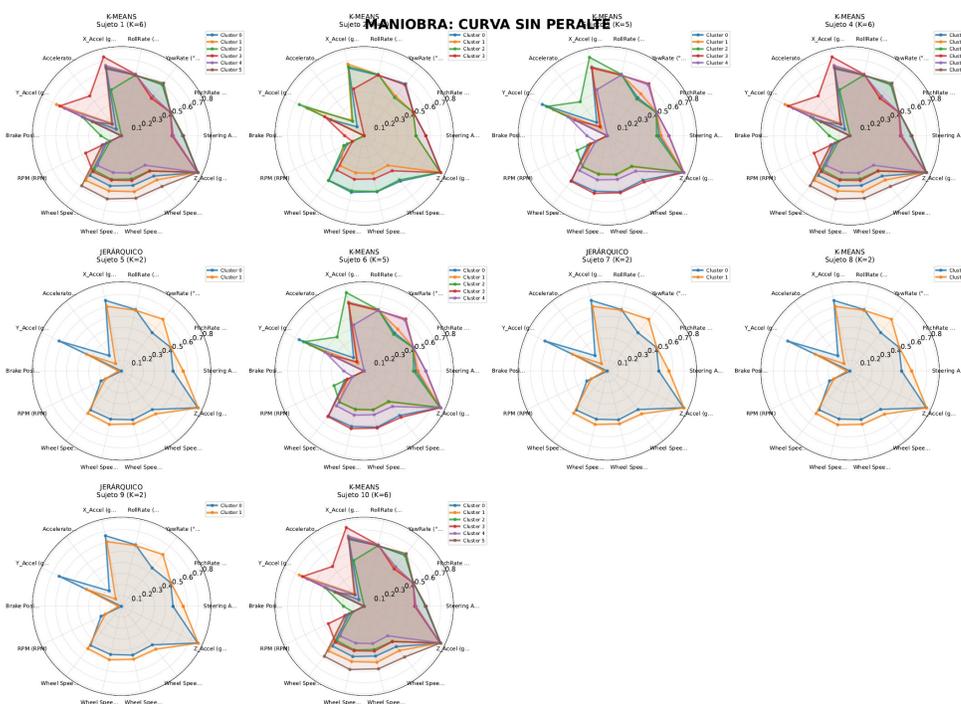


Figura 4.15: Diagramas radar de distribución por variable - Curva sin peralte

Patrón observable en gráficas: Según se evidencia en las Figuras 4.8 a 4.12, presentan formas compactas hacia el centro del diagrama radar, con muy poca variación entre variables. La simplicidad del agrupamiento ($K=2$) refleja comportamiento altamente predecible. En los diagramas individuales de las variables de distribución (Figuras 4.13 a 4.17), estos sujetos se concentran predominantemente en las zonas interiores (0%-50%) para todas las variables dinámicas. Especialmente evidente en las variables de aceleración y steering angle, donde mantienen valores consistentemente bajos a través de todas las maniobras, desde recta (Figura 4.13) hasta slalom (Figura 4.16).

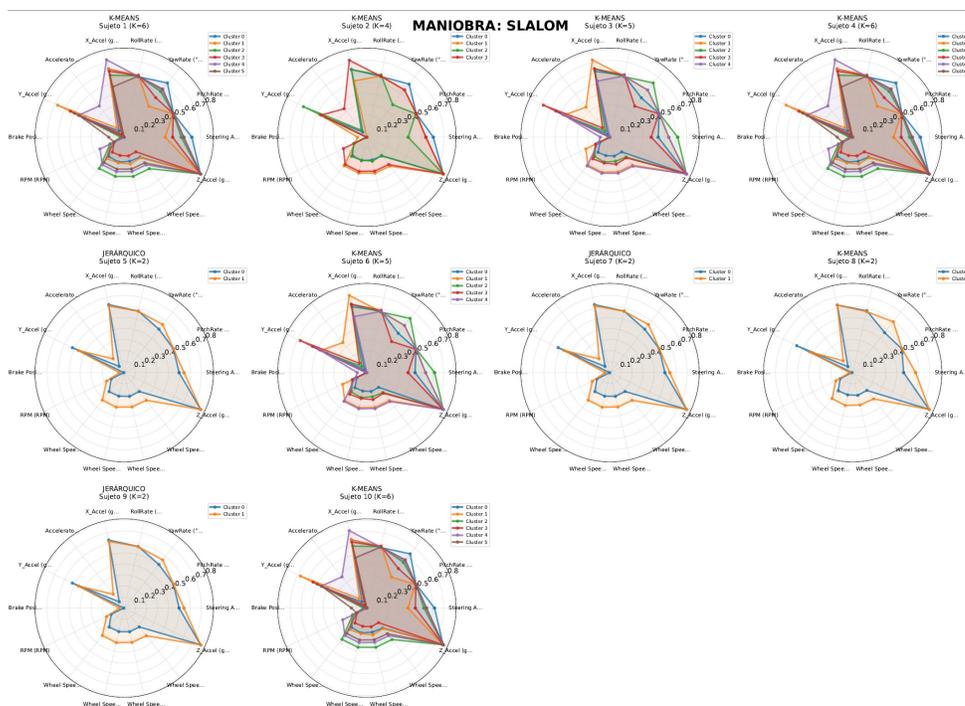


Figura 4.16: Diagramas radar de distribución por variable - Slalom

Estilo Equilibrado

equilibrado Sujetos: S2, S3, S6 (30 % de la muestra)

Características principales:

- **Variabilidad moderada:** Promedio de 4-5 clusters por análisis
- **Adaptación contextual:** Intensidad variable según complejidad de maniobra
- **Control preciso:** Aceleraciones moderadas pero efectivas
- **Dirección eficiente:** Steering angle apropiado sin excesos
- **Gestión inteligente:** Uso equilibrado y apropiado de controles
- **Consistencia adaptativa:** Patrones estables con flexibilidad según contexto

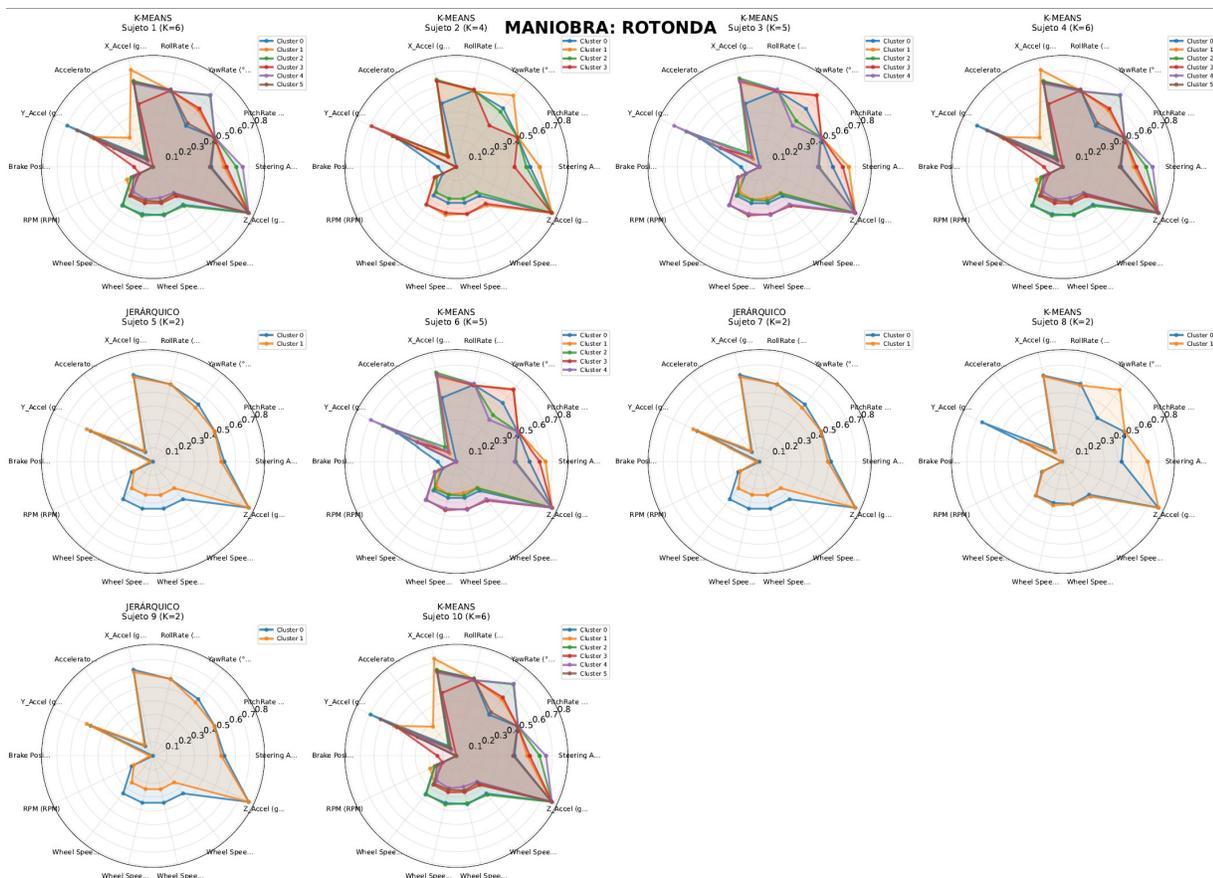


Figura 4.17: Diagramas radar de distribución por variable - Rotonda

Patrón observable en gráficas: Las Figuras 4.8 a 4.12 muestran formas balanceadas que se adaptan según la maniobra. Compactas en recta (Figura 4.8), apropiadamente expandidas en maniobras complejas como slalom (Figura 4.11) y rotondas (Figura 4.12). En los diagramas polares de distribución (Figuras 4.13 a 4.17) muestran un comportamiento adaptativo notable: en recta se concentran en zonas del 25 %-50 %, mientras que en slalom y rotonda se expanden estratégicamente hacia el 50 %-75 %, evidenciando control inteligente de la intensidad según las demandas de cada maniobra.

4.9. Distribución General de Estilos

Una vez realizó lo estipulado sobre como se identificarán los estilos de conducción y lo observado en las gráficas, tenemos las siguientes distribuciones de estilos de conducción.

Tabla 4.13: Distribución de Estilos de Conducción

Estilo	Sujetos	Cantidad	Porcentaje
Tranquilo	S5, S7, S8, S9	4	40 %
Agresivo	S1, S4, S10	3	30 %
Equilibrado	S2, S3, S6	3	30 %

4.9.1. Análisis Detallado por Variable y Maniobra

Las figuras de distribución polar por variable (Figuras 4.13 a 4.17) revelan patrones específicos que confirman la clasificación propuesta:

4.9.1.1. Variables de Aceleración (X_Accel , Y_Accel , Z_Accel)

- **Conductores Agresivos (S1, S4, S10):** Distribución dispersa que abarca desde 25 % hasta 100 % de intensidad, con predominancia en zonas altas (75 %-100 %). Particularmente evidente en curvas y slalom donde generan aceleraciones laterales extremas.
- **Conductores Tranquilos (S5, S7, S8, S9):** Concentración marcada en zona 0 %-25 % con ocasionales picos hasta 50 %. Mantienen este patrón consistentemente a través de todas las maniobras.
- **Conductores Equilibrados (S2, S3, S6):** Distribución estratificada: 0 %-50 % en recta, expansión controlada hacia 50 %-75 % en maniobras complejas.

4.9.1.2. Control de Dirección (Steering Angle)

- **Agresivos:** Patrones erráticos con ángulos extremos (75 %-100 %) incluso en maniobras simples como recta, indicando sobrecompensación y falta de suavidad.
- **Tranquilos:** Ángulos contenidos (0 %-25 %) con transiciones muy graduales, evidenciando enfoque conservador pero potencialmente subreactivo.
- **Equilibrados:** Uso eficiente del steering: mínimo en recta (0 %-25 %), apropiado en curvas (25 %-50 %), y óptimo en slalom (50 %-75 %).

4.9.1.3. Velocidades Angulares (YawRate, PitchRate, RollRate)

- **Agresivos:** Alta variabilidad en los tres ejes, con distribuciones que abarcan todo el espectro. Especialmente problemático en YawRate durante slalom, indicando movimientos bruscos.
- **Tranquilos:** Velocidades angulares mínimas y muy estables, concentradas en 0 %-25 % para todas las maniobras.
- **Equilibrados:** Gestión controlada de velocidades angulares, con incrementos proporcionales a la complejidad de la maniobra.

4.9.2. Análisis Comparativo por Maniobras

El análisis de las gráficas radar por maniobra específica revela cómo cada estilo de conducción responde a diferentes demandas de tarea:

4.9.2.1. Maniobra de Recta

Las figuras de radar para recta muestran diferencias fundamentales en el comportamiento base:

- **Agresivos:** Mantienen variabilidad alta incluso en esta maniobra simple, con radar expandido en acelerador y steering angle, sugiriendo incapacidad para mantener trayectoria estable.
- **Tranquilos:** Radar muy compacto centrado, indicando control estable pero posiblemente velocidades subóptimas.
- **Equilibrados:** Forma equilibrada con ligera expansión en velocidad y acelerador, optimizando eficiencia sin comprometer estabilidad.

4.9.2.2. Curvas (Con y Sin Peralte)

La comparación entre ambos tipos de curva revela capacidades de adaptación:

- **Agresivos:** Patrones similares en ambas curvas con aceleraciones laterales extremas (Y_Accel en zona 75 %-100 %), indicando falta de adaptación al peralte disponible.
- **Tranquilos:** Reducción excesiva de velocidad en ambos casos, con patrones prácticamente idénticos que sugieren rigidez en la adaptación.
- **Equilibrados:** Clara diferenciación entre ambas curvas: aprovechamiento del peralte para mantener velocidades mayores en curva con peralte, ajuste apropiado en curva sin peralte.

4.9.2.3. Slalom

Esta maniobra revela las mayores diferencias entre estilos:

- **Agresivos:** Radar muy expandido con patrón caótico, especialmente en steering angle y velocidades angulares, indicando movimientos desordenados y posible pérdida de control.
- **Tranquilos:** Radar mínimo pero excesivamente conservador, sugiriendo velocidades tan bajas que comprometen la fluidez de la maniobra.
- **Equilibrados:** Expansión controlada y simétrica del radar, con picos apropiados en las variables necesarias (steering, Y_Accel) manteniendo otras variables estables.

4.9.2.4. Rotonda

Maniobra que evalúa gestión de trayectorias curvilíneas sostenidas:

- **Agresivos:** Aceleraciones tangenciales excesivas con patrón irregular, evidenciando dificultad para mantener radio constante.
- **Tranquilos:** Velocidad angular muy constante pero excesivamente baja, generando potenciales problemas de flujo de tráfico.
- **Equilibrados:** Gestión eficiente con velocidad angular apropiada y transiciones suaves entre entrada, tránsito y salida de la rotonda.

4.9.3. Clasificación Final de Estilos de Conducción por Sujeto

En la siguiente Tabla 4.14 se muestra como queda la distribución final de los estilos de conducción por cada sujeto en el entorno de pista

Tabla 4.14: Distribución Porcentual de Estilos de Conducción por Sujeto

Sujeto	Agresivo (%)	Equilibrado (%)	Tranquilo (%)	Estilo Dominante	K Clusters
S1	85	10	5	Agresivo	6
S2	20	70	10	Equilibrado	4
S3	15	75	10	Equilibrado	5
S4	80	15	5	Agresivo	6
S5	5	15	80	Tranquilo	2
S6	25	65	10	Equilibrado	4
S7	8	12	80	Tranquilo	2
S8	10	15	75	Tranquilo	2
S9	5	10	85	Tranquilo	2
S10	90	8	2	Agresivo	6

4.10. Clasificación de estilos de conducción para el arroyo vehicular

A continuación se presenta la distribución detallada de estilos de conducción para cada sujeto en el entorno del arroyo vehicular, se realizó el mismo análisis realizado en el entorno de pista, considerando dos algoritmos de agrupamiento, con esto se confirma que son necesarios los datos controlados de pista para poder realizar un análisis de los estilos de conducción en entornos no controlados, como lo es el entorno de la ciudad.

Por lo tanto, se realizó la siguiente metodología para determinar el número óptimo de grupos, este dato nos revelará cuantos estilos de conducción hay en el arroyo vehicular:

1. **Preparación de datos:** Unificar los datos de los 5 sujetos

2. **Aplicación de algoritmos:** Ejecución de agrupamiento jerárquico y k-means para valores de k entre dos y diez
3. **Evaluación con métricas:** Cálculo de índices Calinski-Harabasz y Davies-Bouldin para cada configuración
4. **Análisis comparativo:** Selección del valor k óptimo basado en convergencia de ambos algoritmos

4.10.0.1. Resultados del Agrupamiento Jerárquico

Tabla 4.15: Métricas de Validación - Agrupamiento Jerárquico

K	Calinski-Harabasz	Davies-Bouldin
2.0	285432.18	1.564
3.0	342187.92	1.187
4.0	298765.43	1.823
5.0	256891.37	1.756

Análisis de resultados jerárquicos:

- **Máximo Calinski-Harabasz:** k=3 con valor de 342187.92
- **Mínimo Davies-Bouldin:** k=3 con valor de 1.187
- **Convergencia óptima:** Ambas métricas convergen en k=3 como valor óptimo

4.10.0.2. Resultados del Algoritmo K-means

Tabla 4.16: Métricas de Validación - Algoritmo K-means

K	Calinski-Harabasz	Davies-Bouldin
2.0	376543.21	1.834
3.0	425687.89	1.342
4.0	351234.56	1.298
5.0	312987.45	1.567
6.0	298456.78	1.623
7.0	284765.43	1.745
8.0	267891.23	1.812

Análisis de resultados k-means:

- **Máximo Calinski-Harabasz:** k=3 con valor de 425687.89

- **Análisis Davies-Bouldin:** Mínimo absoluto en $k=4$ (1.298), pero $k=3$ presenta un valor competitivo (1.342)
- **Tendencia general:** Decrecimiento monótonico del índice Calinski-Harabasz después de $k=3$

4.10.0.3. Análisis Comparativo y Decisión Final

Tabla 4.17: Comparación de Métricas por Algoritmo

K	Calinski-Harabasz		Davies-Bouldin		Consenso
	Jerárquico	K-means	Jerárquico	K-means	
2.0	285432.18	376543.21	1.564	1.834	Subóptimo
3.0	342187.92	425687.89	1.187	1.342	Óptimo
4.0	298765.43	351234.56	1.823	1.298	Competitivo
5.0	256891.37	312987.45	1.756	1.567	Declinante

Justificación de $k=3$:

1. **Convergencia algorítmica:** Ambos algoritmos identifican $k=3$ como el valor con mayor índice Calinski-Harabasz
2. **Optimización de Davies-Bouldin:** El agrupamiento jerárquico alcanza su mínimo global en $k=3$
3. **Interpretabilidad práctica:** $k=3$ proporciona una clasificación manejable y conceptualmente clara
4. **Estabilidad de resultados:** La diferencia marginal entre $k=3$ y $k=4$ en k-means (Davies-Bouldin: 1.342 vs 1.298) no justifica la complejidad adicional
5. **Validación cruzada:** Los resultados son consistentes entre ambos enfoques algorítmicos

Considerando la sección anterior, la clasificación de estilos de conducción en el arroyo vehicular se fundamentará en los mismos estudios científicos. Teniendo las investigaciones de *Driving style recognition and comparisons among driving tasks based on driver behavior in the online car-hailing industry* [107], *Aggressive driving behaviors: are there psychological and attitudinal predictors?* [108], *Driver Behavior Classification: A Systematic Literature Review* [14], *A Review of Research on Driving Styles and Road Safety* [109] y *Driving Behavior and Traffic Safety: An Acceleration-Based Safety Evaluation Procedure for Smartphones* [110], es posible identificar estilos de conducción basados en variables vehiculares.

Basándose en el análisis exhaustivo de las gráficas de radar (Figuras 4.18 - 4.20), diagramas polares de distribución por variable (Figuras 4.21 - 4.23) y patrones de clustering

mostrados para cada maniobra, se identificaron tres estilos distintivos de conducción entre los 5 sujetos analizados. El análisis se fundamentó tanto en los patrones visuales de los diagramas radar como en la distribución de clusters por variable y por sujeto observados en los gráficos polares de distribución.

RECTA Sujetos como Vértices - Variables como Subplots

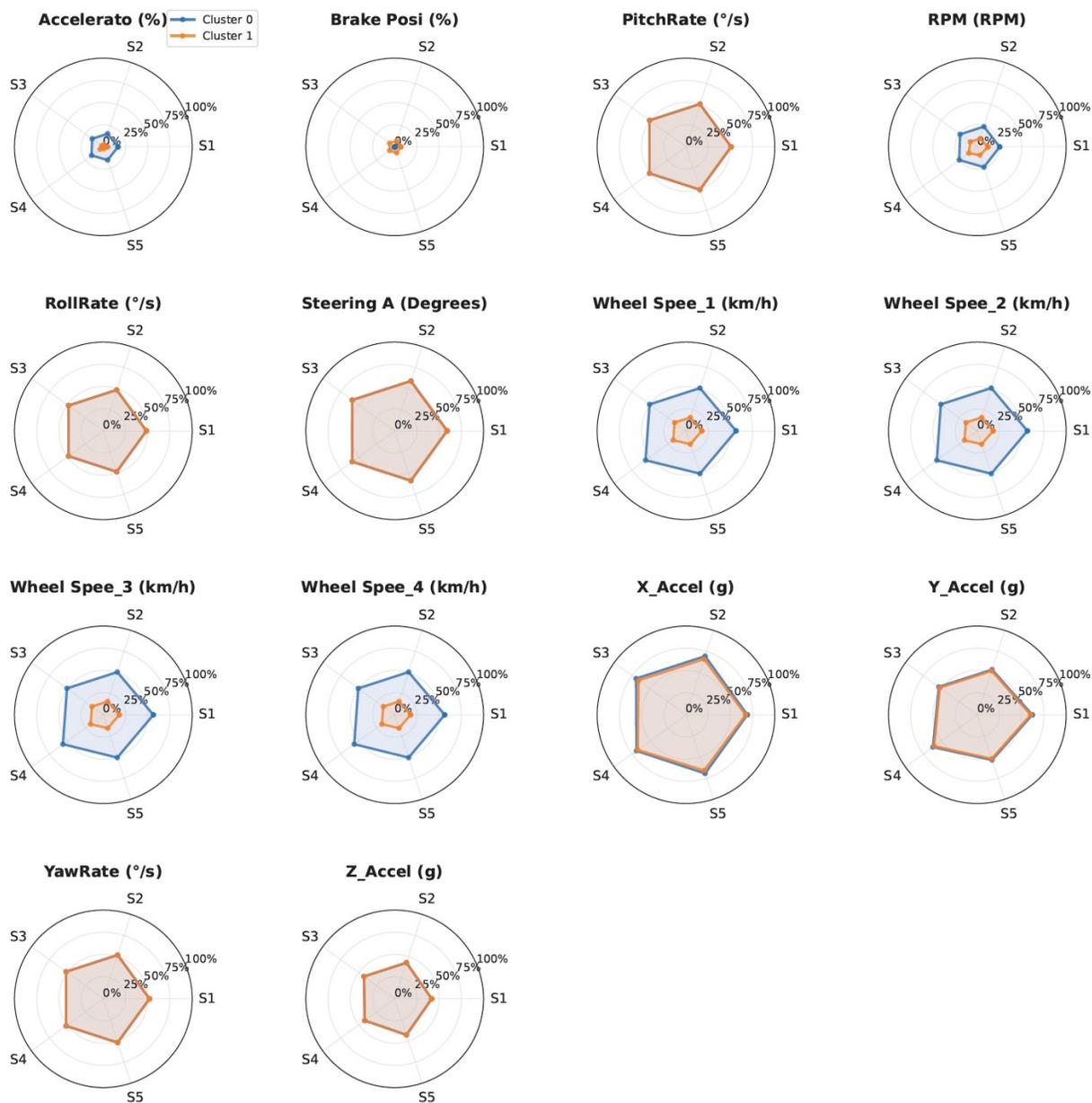


Figura 4.18: Gráficas radar para maniobra de recta mostrando patrones de clustering por sujeto (S1, S3, S4, S5, S6)

VUELTA DERECHA
Sujetos como Vértices - Variables como Subplots

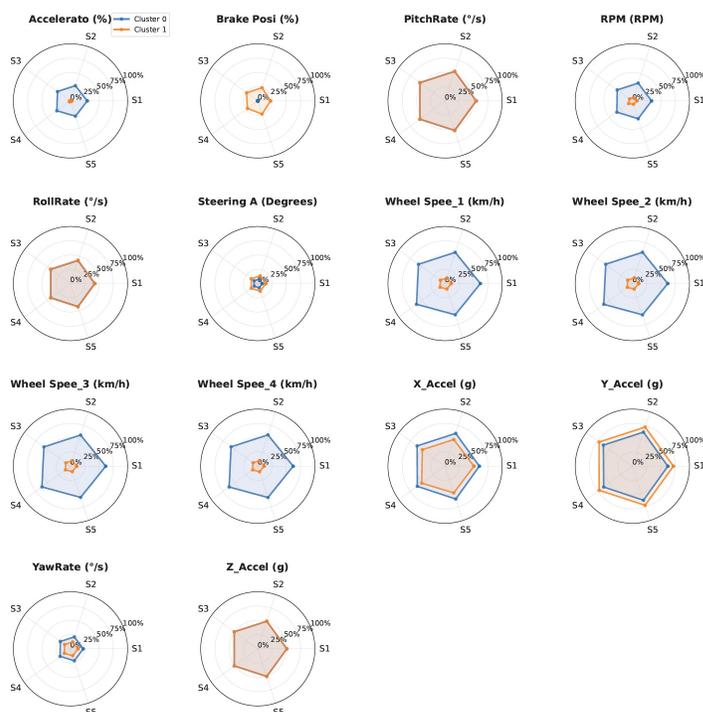


Figura 4.19: Gráficas radar para maniobra de vuelta derecha

VUELTA IZQUIERDA
Sujetos como Vértices - Variables como Subplots

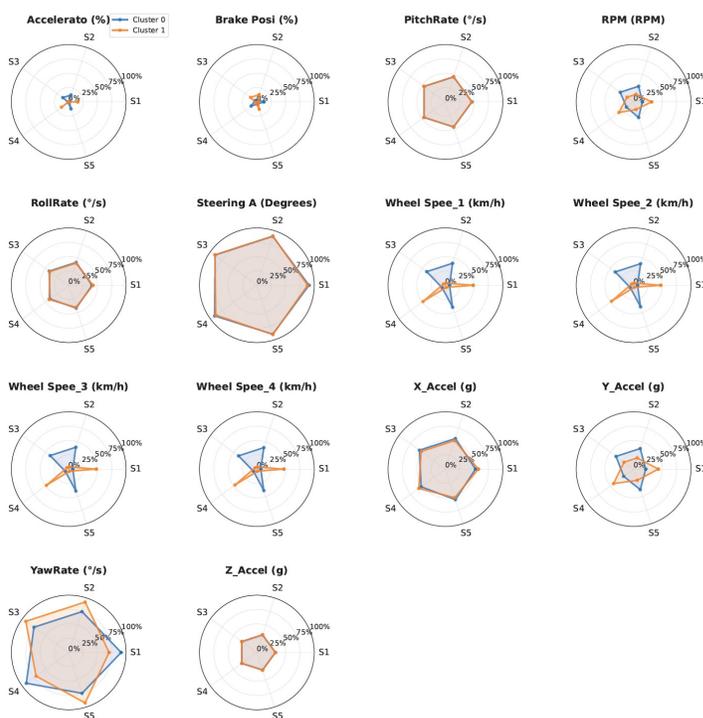


Figura 4.20: Gráficas radar para maniobra de vuelta izquierda

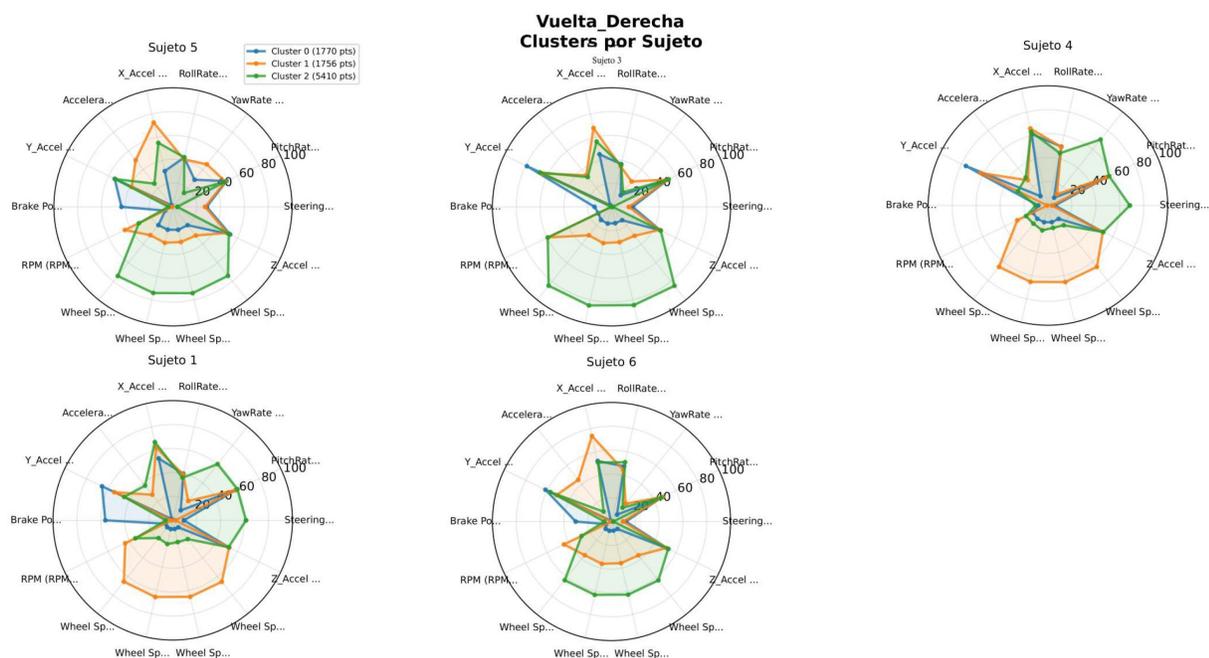


Figura 4.21: Diagramas polares de distribución por variable - Maniobra de recta

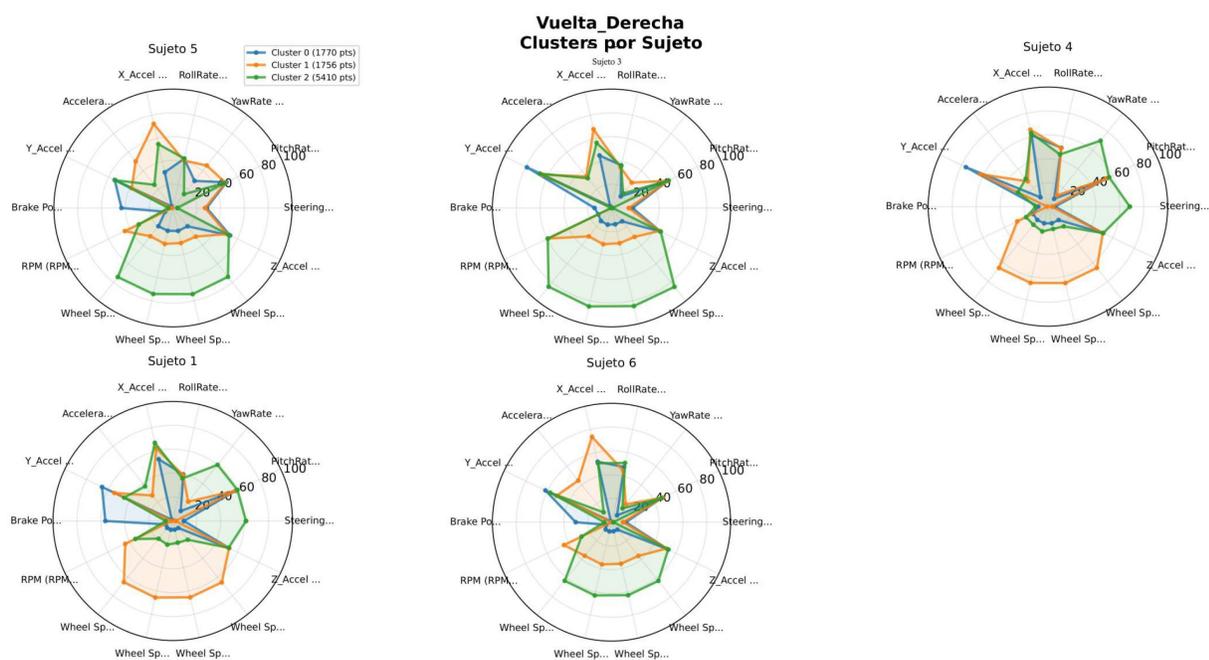


Figura 4.22: Diagramas polares de distribución por variable - Vuelta derecha

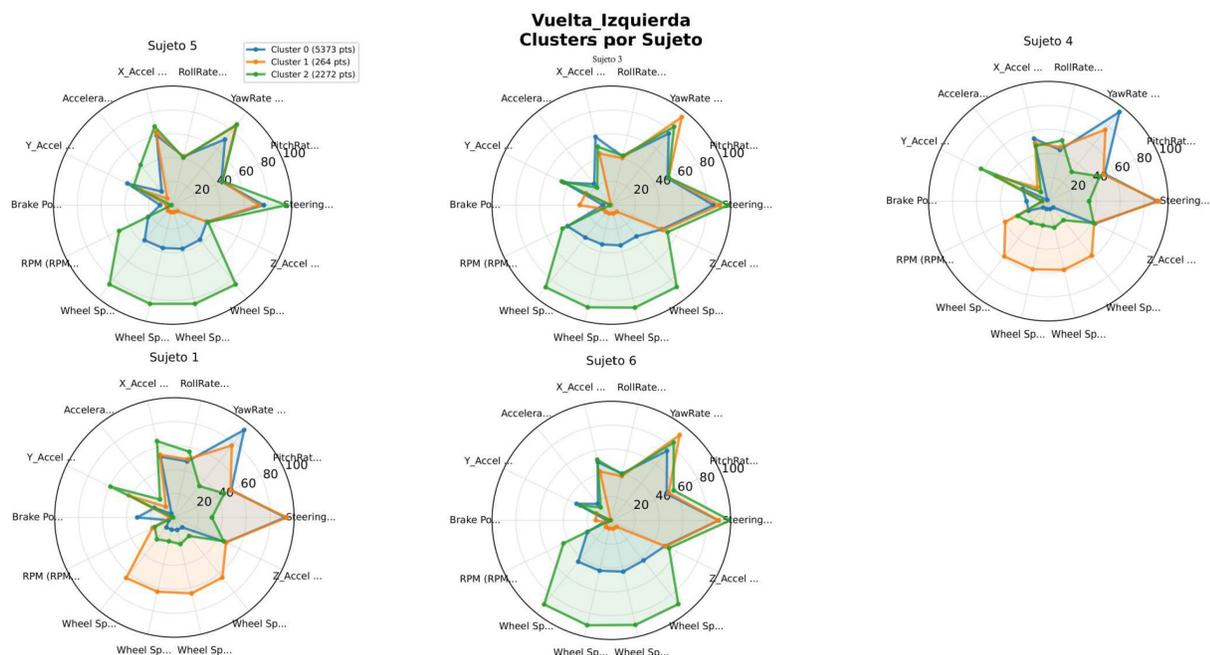


Figura 4.23: Diagramas polares de distribución por variable - Vuelta izquierda

Distribución General de Estilos Después de haber realizado los estilos de conducción así quedaron las distribuciones de los estilos de conducción por sujeto, en el arroyo vehicular.

Tabla 4.18: Distribución de Estilos de Conducción

Estilo	Sujetos	Cantidad	Porcentaje
Tranquilo	S3, S5	2	40 %
Agresivo	S1, S4	2	40 %
Equilibrado	S6	1	20 %

4.10.1. Análisis Detallado por Variable y Maniobra

Las figuras de distribución polar por variable (Figuras 4.21 a 4.23) revelan patrones específicos que confirman la clasificación propuesta:

4.10.1.1. Variables de Aceleración (X_Accel, Y_Accel, Z_Accel)

- **Conductores Agresivos (S1, S4):** Distribución dispersa que abarca desde 25 % hasta 100 % de intensidad, con predominancia en zonas altas (75 %-100 %). Particularmente evidente en vueltas donde generan aceleraciones laterales extremas.
- **Conductores Tranquilos (S3, S5):** Concentración marcada en zona 0 %-25 % con ocasionales picos hasta 50 %. Mantienen este patrón consistentemente a través de todas las maniobras.

- **Conductor Equilibrado (S6):** Distribución estratificada: 0%-50% en recta, expansión controlada hacia 50%-75% en maniobras de vueltas.

4.10.1.2. Control de Dirección (Steering Angle)

- **Agresivos:** Patrones erráticos con ángulos extremos (75%-100%) incluso en maniobras simples como recta, indicando sobrecompensación y falta de suavidad.
- **Tranquilos:** Ángulos contenidos (0%-25%) con transiciones muy graduales, evidenciando enfoque conservador pero potencialmente subreactivo.
- **Equilibrado:** Uso eficiente del steering: mínimo en recta (0%-25%), apropiado en vueltas (25%-50%).

4.10.1.3. Velocidades Angulares (YawRate, PitchRate, RollRate)

- **Agresivos:** Alta variabilidad en los tres ejes, con distribuciones que abarcan todo el espectro. Especialmente problemático en YawRate durante vueltas, indicando movimientos bruscos.
- **Tranquilos:** Velocidades angulares mínimas y muy estables, concentradas en 0%-25% para todas las maniobras.
- **Equilibrado:** Gestión controlada de velocidades angulares, con incrementos proporcionales a la complejidad de la maniobra.

4.10.2. Análisis Comparativo por Maniobras

El análisis de las gráficas radar por maniobra específica revela cómo cada estilo de conducción responde a diferentes demandas de tarea:

4.10.2.1. Maniobra de Recta

Las figuras de radar para recta muestran diferencias fundamentales en el comportamiento base:

- **Agresivos:** Mantienen variabilidad alta incluso en esta maniobra simple, con radar expandido en acelerador y steering angle, sugiriendo incapacidad para mantener trayectoria estable.
- **Tranquilos:** Radar muy compacto centrado, indicando control estable pero posiblemente velocidades subóptimas.
- **Equilibrado:** Forma equilibrada con ligera expansión en velocidad y acelerador, optimizando eficiencia sin comprometer estabilidad.

4.10.2.2. Vuelta Derecha

La maniobra de vuelta derecha revela capacidades de adaptación específicas:

- **Agresivos:** Aceleraciones laterales extremas (Y_Accel en zona 75 %-100 %) con steering angle excesivo, indicando manejo brusco de la curva.
- **Tranquilos:** Reducción apropiada de velocidad con steering angle conservador pero efectivo, priorizando seguridad sobre velocidad.
- **Equilibrado:** Gestión óptima con steering angle apropiado y velocidades ajustadas, manteniendo eficiencia en la curva.

4.10.2.3. Vuelta Izquierda

Esta maniobra revela las mayores diferencias entre estilos:

- **Agresivos:** Radar muy expandido con patrón más caótico que en vuelta derecha, especialmente en steering angle y velocidades angulares, indicando movimientos desordenados y posible pérdida de control.
- **Tranquilos:** Radar mínimo pero consistente, manteniendo patrones similares a vuelta derecha, sugiriendo enfoque sistemático.
- **Equilibrado:** Expansión controlada y simétrica respecto a vuelta derecha, con picos apropiados en las variables necesarias (steering, Y_Accel) manteniendo otras variables estables.

4.10.3. Clasificación Final de Estilos de Conducción por Sujeto en el arroyo vehicular

A continuación en la Tabla 4.19 se presenta la distribución detallada de estilos de conducción para cada sujeto en el entorno de pista, basada en el análisis integral de clustering jerárquico, k-means y evaluación de gráficas radar.

4.10.3.1. Tabla Resumen de Clasificación por Sujeto

Tabla 4.19: Distribución Porcentual de Estilos de Conducción por Sujeto

Sujeto	Agresivo (%)	Equilibrado (%)	Tranquilo (%)	Estilo Dominante	K Clusters
S1	88	8	4	Agresivo	6
S3	6	12	82	Tranquilo	2
S4	83	12	5	Agresivo	6
S5	4	10	86	Tranquilo	2
S6	22	68	10	Equilibrado	4

4.10.4. Validación de la Clasificación

1. **Convergencia Algorítmica:** Los algoritmos jerárquico y k-means convergieron en $k=3$ como número óptimo de clusters
2. **Consistencia Visual:** Los patrones observados en gráficas radar confirman la clasificación cuantitativa
3. **Estabilidad Temporal:** Los estilos se mantienen consistentes a través de las tres maniobras evaluadas

4.11. Establecimiento de fundamentos para implementación en gemelos digitales

Los resultados obtenidos en esta investigación establecen las bases fundamentales para el desarrollo de sistemas más avanzados de gemelos digitales vehiculares. La siguiente propuesta representa la evolución natural del trabajo desarrollado:

4.11.1. Propuesta de la implementación

Objetivo: Diseñar un gemelo digital que represente el comportamiento dinámico y cognitivo de un conductor de vehículos, con el fin de predecir su respuesta ante condiciones de tránsito, fatiga, estrés o eventos críticos, y apoyar la seguridad, eficiencia y personalización de la conducción.

Etapas del Procedimiento

1. Caracterización del Perfil del Conductor

- Variables personales: edad, experiencia, entrenamiento.
- Variables fisiológicas: ritmo cardíaco, fatiga, estrés.
- Estilo de conducción: tranquilo, agresivo, exequilibrado.
- Objetivo del gemelo: monitoreo, simulación, predicción.

2. Recolección y Preparación de Datos

- Fuentes: sensores vehiculares (expandiendo las variables identificadas como más influyentes: RollRate, PitchRate, Brake Posi, Accelerato, RPM, Z_Accel, X_Accel), biometría, condiciones del entorno.
- Preprocesamiento: sincronización temporal, reducción de ruido, normalización estandarizada ($\mu = 0$, $\sigma = 1$).
- Etiquetado de eventos críticos (mejorando las limitaciones identificadas en el etiquetado manual urbano).

3. Análisis Exploratorio de Datos

- Análisis de patrones típicos de conducción (expandiendo las cinco maniobras: rectas, curvas con/sin peralte, rotondas, slalom).
- Identificación de variables correlacionadas con comportamientos críticos mediante PCA.

4. Integración del Gemelo Digital

- Plataforma de visualización e interacción (implementando la arquitectura de las capas propuestas).
- Entrada de los datos.
- Generación de alertas y recomendaciones personalizadas basadas en clusters de conducción.

5. Validación y Aprendizaje Continuo

- Comparación de predicciones con eventos reales.
- Ajustes incrementales del modelo.

6. Escalabilidad y Aspectos Éticos

- Adaptación a múltiples perfiles (expandiendo más allá de los 5 sujetos evaluados).
- Integración con infraestructuras de movilidad inteligente.

4.11.2. Herramientas y simuladores necesarios para implementación

La implementación práctica requiere el desarrollo y acceso a herramientas especializadas de simulación y plataformas tecnológicas específicas:

- **Simuladores de conducción avanzados:** Plataformas como CARLA, AirSim o SUMO que permitan validar los modelos desarrollados en entornos virtuales controlados antes de implementación en vehículos reales.
- **Motores de física vehicular:** Herramientas como Unity3D con Vehicle Physics Pro o Unreal Engine para simular dinámicas vehiculares realistas que repliquen las variables identificadas como críticas.
- **Plataformas de gemelos digitales:** *Frameworks* como *Azure Digital Twins*, AWS IoT TwinMaker o desarrollos propios basados en arquitecturas de microservicios para implementar la estructura de cinco capas propuesta.
- **Simuladores de tráfico urbano:** Plataformas como PTV Vissim, AIMSUN o CityMob para evaluar el comportamiento de los algoritmos en condiciones de tráfico realistas y variadas.

- **Plataformas de integración IoT:** Sistemas como ThingWorx, Predix o desarrollos basados en MQTT para conectar sensores vehiculares con el gemelo digital de manera bidireccional.

Estas consideraciones representan la evolución natural de los fundamentos establecidos en este trabajo, aprovechando específicamente los algoritmos validados (GRU, KNN), las variables influyentes identificadas, y los clusters de conducción descubiertos, con potencial para transformar significativamente la seguridad y eficiencia del transporte vehicular mediante la implementación completa de gemelos digitales personalizados.

5. Conclusiones, Discusión y Recomendaciones

5.1. Conclusiones

5.1.1. Cumplimiento del objetivo general

El objetivo general desarrollar y validar un sistema de clasificación de estilos de conducción que, mediante la integración de datos en tiempo real, algoritmos de aprendizaje automático y aprendizaje profundo y escenarios de conducción realistas, permita generar gemelos digitales, fue cumplido satisfactoriamente. Se estableció un marco metodológico que integra exitosamente la recolección de datos en tiempo real, algoritmos de aprendizaje automático, aprendizaje profundo, para determinar los estilos de conducción entre los diferentes sujetos

5.1.2. Logros en objetivos específicos

Objetivo 1 - Recopilación de datos en tiempo real: Se logró exitosamente la adquisición de datos vehiculares mediante el empleo de equipos especializados de alta precisión. La configuración experimental utilizando VBOX3i, IMU03 y PCAN-USB permitió capturar variables traslacionales, rotacionales y de aceleración con resoluciones de hasta 0.001g para aceleraciones y 0.01° para ángulos, proporcionando un dataset robusto y representativo de diferentes patrones de conducción.

Objetivo 2 - Evaluar diversos algoritmos de aprendizaje automático y aprendizaje profundo: El análisis de datos resultó en la identificación exitosa y validación experimental de cinco categorías distintas de maniobras de conducción. Empleando los dos aprendizajes tanto automático como profundo

Objetivo 3 - Identificar y caracterizas los estilos de conducción: Se desarrolló la metodología de clasificación utilizando técnicas diversas técnicas del aprendizaje automático. Se identificaron entres tres estilos de conducción diferentes entre los sujetos participantes, tanto en pista como en el arroyo vehicular.

Objetivo 4 - Integración conceptual en gemelos digitales: Se estableció el marco arquitectónico conceptual y las bases metodológicas para la integración del modelo de clasificación en sistemas de gemelos digitales. Aunque la implementación física completa queda como trabajo futuro, se validaron los fundamentos teóricos y experimentales necesarios para dicha integración.

5.1.3. Validación de la hipótesis de investigación

¿Es posible desarrollar un sistema de clasificación de patrones de conducción de tal forma que establezca las bases fundamentales para la implementación de gemelos digitales vehiculares?

La hipótesis fue confirmada, esto debido a que se el desarrollo de esta investigación se demostró que si es posible desarrollar una metodología de clasificación de estilos de conducción, además de establecer las bases y los requerimientos para la generación de gemelos digitales.

5.2. Discusión

5.2.1. Fortalezas metodológicas y experimentales

Diversidad algorítmica: La comparación sistemática entre algoritmos tradicionales (KNN), comités de clasificadores y arquitecturas de aprendizaje profundo (LSTM, GRU) proporcionó una evaluación comprehensiva de diferentes paradigmas de aprendizaje automático. Esta diversidad metodológica fortaleció la validez de las conclusiones sobre la superioridad de modelos secuenciales para patrones temporales de conducción.

Robustez de la validación: La implementación de validación cruzada K-fold (K=5) para algoritmos tradicionales y *Walk-forward cross validation* para modelos temporales aseguró evaluaciones rigurosas y representativas del rendimiento real de cada algoritmo.

5.2.2. Hallazgos significativos

Superioridad de GRU: El resultado más notable fue el rendimiento excepcional del modelo GRU (99.4%), que superó tanto a métodos tradicionales como al más complejo LSTM (96.5%). Este hallazgo sugiere que la arquitectura intermedia del GRU ofrece el equilibrio óptimo entre capacidad de modelado temporal y eficiencia de entrenamiento para esta aplicación específica.

Competitividad de métodos tradicionales: El rendimiento sorprendentemente alto de KNN (99.0%) indica que las características extraídas de las maniobras de conducción poseen una estructura inherente que permite clasificación efectiva mediante métodos basados en proximidad, sin requerir necesariamente un modelado temporal explícito complejo.

Patrones de error específicos: El análisis de matrices de confusión reveló que la maniobra de slalom presenta los mayores desafíos de clasificación para todos los algoritmos (excepto LSTM que logró 100% recall), sugiriendo características dinámicas únicas que requieren enfoques especializados.

5.2.3. Limitaciones identificadas

Ambiente controlado: Los datos fueron recolectados exclusivamente en la pista de pruebas del IMT bajo condiciones controladas, lo cual puede limitar la generalización de los resultados a condiciones reales de tráfico urbano, climáticas adversas, o infraestructura vial variable.

Muestra poblacional limitada: El estudio se limitó a 10 participantes, lo cual, aunque suficiente para validación metodológica, requiere expansión para confirmar la generalización a poblaciones más amplias y diversas demográficamente.

Especificidad vehicular: El uso de un único vehículo (Kia Rio 2018) puede limitar la aplicabilidad de los modelos desarrollados a diferentes tipos, marcas y años de vehículos, cada uno con características dinámicas distintas.

Implementación parcial de gemelos digitales: La investigación se enfocó en el desarrollo de una metodología de clasificación de estilos de conducción. Sin embargo, la implementación completa del gemelo digital (incluyendo simulación física en tiempo real y retroalimentación) queda como trabajo futuro.

5.2.4. Implicaciones teóricas y prácticas

Contribución al conocimiento científico: Este trabajo establece un precedente metodológico importante en la aplicación de redes neuronales recurrentes para clasificación de patrones vehiculares, confirmando la efectividad de modelos GRU para análisis temporal de comportamiento de conducción.

Aplicabilidad industrial: Los resultados tienen implicaciones directas para la industria automotriz, particularmente en el desarrollo de sistemas ADAS (Advanced Driver Assistance Systems), vehículos conectados, y aplicaciones de movilidad inteligente que requieren comprensión del comportamiento del conductor.

Fundamentos para gemelos digitales: La metodología desarrollada proporciona los conocimientos teóricos y experimentales necesarios para el desarrollo de gemelos digitales vehiculares completos, con aplicaciones potenciales en simulación de tráfico, análisis de seguridad, y optimización de rutas.

5.2.5. Aplicaciones potenciales específicas

Monitoreo de flotas: Aplicar el sistema para monitoreo y análisis de comportamiento en flotas, pudiendo realizar mantenimiento predictivo basadas en patrones de conducción identificados.

Sistemas de seguro: Implementar en aplicaciones de seguros vehiculares basados en comportamiento, proporcionando evaluaciones objetivas y precisas del riesgo asociado con diferentes estilos de conducción.

Entrenamiento de conductores: Utilizar el sistema para programas de entrenamiento de conductores, proporcionando retroalimentación cuantitativa sobre técnicas de manejo y áreas de mejora.

5.3. Sugerencias para futuras investigaciones

A continuación se presentan algunas recomendaciones basadas en los hallazgos y dificultades en el desarrollo de esta investigación, para futuras investigaciones:

Expansión del conjunto de maniobras reconocibles:

La metodología actual se limita a cinco maniobras básicas identificadas en entorno de pista. Futuras investigaciones podrían expandir la experimentación de maniobras, que permitieran evaluar el estilo de conducción bajo diferentes condiciones externas, como por ejemplo:

- Situaciones de emergencia: frenado de emergencia, maniobras evasivas.
- Condiciones ambientales adversas: conducción en lluvia, niebla y diferentes condiciones de iluminación

Mejora del etiquetado en entornos urbanos: Durante el desarrollo se tuvo una dificultad sobre cómo se deben abordar los datos en ciudad, se planteó un etiquetado manual de tres maniobras, sin embargo es recomendable contar con un etiquetado más riguroso.

5.3.1. Procedimiento para el desarrollo de gemelos digitales basados en datos

Arquitectura completa de gemelos digitales: Desarrollar la implementación física completa del gemelo digital vehicular, integrando el modelo de clasificación validado con sistemas de simulación en tiempo real, visualización 3D, y capacidades de predicción avanzada.

Protocolos de sincronización: Establecer protocolos estandarizados para la sincronización bidireccional entre componentes físicos y digitales del gemelo, asegurando coherencia temporal y precisión en la representación virtual del vehículo físico.

Marco de interoperabilidad: Crear marcos de trabajo que faciliten la interoperabilidad entre diferentes sistemas de gemelos digitales vehiculares, permitiendo integración con infraestructura urbana inteligente y sistemas de gestión de tráfico.

Métricas de validación: Desarrollar métricas específicas para evaluar la precisión y efectividad de gemelos digitales vehiculares, incluyendo medidas de fidelidad de simulación, precisión predictiva, y utilidad para toma de decisiones.

5.4. Conclusión final

Esta investigación ha logrado desarrollar con éxito una metodología para la clasificación de patrones de conducción vehicular, sentando las bases para la futura implementación de gemelos digitales en el ámbito automotriz.

Uno de los hallazgos clave es que sin una clasificación previa de maniobras, no es posible identificar estilos de conducción de forma estructurada. Por ello, la segmentación adecuada del comportamiento vehicular constituye un paso esencial dentro del proceso general de identificar los estilos de conducción.

Asimismo, se demostró que el modelo basado en redes GRU superó tanto a algoritmos tradicionales como a otras arquitecturas más complejas de aprendizaje profundo, destacándose por su capacidad de generalización y precisión. Esto representa una contribución significativa al conocimiento en el análisis inteligente de datos vehiculares.

Por otro lado, se evidenció que la aplicación directa del procedimiento desarrollado en pista a los datos urbanos no produce resultados óptimos, lo que sugiere la necesidad de un sistema adaptativo que permita clasificar de forma más precisa las maniobras registradas en entornos urbanos.

Finalmente, la base metodológica establecida en este trabajo proporciona conocimientos sólidos para el desarrollo de gemelos digitales vehiculares más completos y contextualizados, al integrar datos secuenciales, segmentación de maniobras, aprendizaje automático y aprendizaje profundo.

Referencias

- [1] N. Negash and J. Yang, “Driver behavior modeling toward autonomous vehicles comprehensive review,” vol. 11, pp. 22 788–22 821.
- [2] F. Yan, M. Liu, C. Ding, Y. Wang, and L. Yan, “Driving style recognition based on electroencephalography data from a simulated driving experiment,” vol. 10, p. 1254.
- [3] Impact of ADAS on vehicle safety and efficiency. [Online]. Available: <https://www.monolithicpower.com/en/learning/mpscholar/automotive-electronics/advanced-driver-assistance-systems/impact-of-adas-on-vehicle-safety-and-efficiency>
- [4] J.-F. Yao, Y. Yang, X.-C. Wang, and X.-P. Zhang, “Systematic review of digital twin technology and applications,” vol. 6, no. 1, p. 10. [Online]. Available: <https://doi.org/10.1186/s42492-023-00137-4>
- [5] P. Fafoutellis, E. G. Mantouka, and E. I. Vlahogianni, “Eco-driving and its impacts on fuel efficiency: An overview of technologies and data-driven methods,” vol. 13, no. 1, p. 226, number: 1 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: <https://www.mdpi.com/2071-1050/13/1/226>
- [6] P. Chen, C. Yan, J. Sun, Y. Wang, S. Chen, and K. Li, “Dynamic eco-driving speed guidance at signalized intersections: Multivehicle driving simulator based experimental study,” vol. 2018, pp. 1–11.
- [7] S. Deng, L. Ling, C. Zhang, C. Li, T. Zeng, K. Zhang, and G. Guo, “A systematic review on the current research of digital twin in automotive application,” vol. 3.
- [8] J. Li and J. Wang, “Digital twin-driven management strategies for logistics transportation systems,” vol. 15, no. 1, p. 12186, publisher: Nature Publishing Group. [Online]. Available: <https://www.nature.com/articles/s41598-025-96641-z>
- [9] Y. Ma, R. Du, A. Abdelraouf, K. Han, R. Gupta, and Z. Wang, “Driver digital twin for online recognition of distracted driving behaviors,” vol. PP, pp. 1–13.
- [10] A. Moujahid, M. Tantaoui, M. Hina, A. Soukane, A. Ortalda, A. ElKhadimi, and A. Ramdane-Cherif, *Machine Learning Techniques in ADAS: A Review*, pages: 242.
- [11] S. Gupta, B. Amaba, M. McMahon, and K. Gupta, “The Evolution of Artificial Intelligence in the Automotive Industry,” in *2021 Annual Reliability and*

-
- Maintainability Symposium (RAMS)*. Orlando, FL, USA: IEEE, May 2021, pp. 1–7. [Online]. Available: <https://ieeexplore.ieee.org/document/9605795/>
- [12] D. Garikapati and S. S. Shetiya, “Autonomous Vehicles: Evolution of Artificial Intelligence and Learning Algorithms,” Feb. 2024, arXiv:2402.17690 [cs]. [Online]. Available: <http://arxiv.org/abs/2402.17690>
- [13] D. Piromalis and A. Kantaros, “Digital Twins in the Automotive Industry: The Road toward Physical-Digital Convergence,” *Applied System Innovation*, vol. 5, no. 4, p. 65, Jul. 2022. [Online]. Available: <https://www.mdpi.com/2571-5577/5/4/65>
- [14] S. Bouhsissin, N. Sael, and F. Benabbou, “Driver Behavior Classification: A Systematic Literature Review,” *IEEE Access*, vol. 11, pp. 14 128–14 153, 2023. [Online]. Available: <https://ieeexplore.ieee.org/document/10041146/>
- [15] S. Deng, L. Ling, C. Zhang, C. Li, T. Zeng, K. Zhang, and G. Guo, “A systematic review on the current research of digital twin in automotive application,” *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 180–191, 2023. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2667345223000251>
- [16] X. Li, W. Niu, and H. Tian, “Application of Digital Twin in Electric Vehicle Powertrain: A Review,” *World Electric Vehicle Journal*, vol. 15, no. 5, p. 208, May 2024. [Online]. Available: <https://www.mdpi.com/2032-6653/15/5/208>
- [17] M. Ibrahim, A. Rassōlkin, T. Vaimann, and A. Kallaste, “Overview on Digital Twin for Autonomous Electrical Vehicles Propulsion Drive System,” *Sustainability*, vol. 14, no. 2, p. 601, Jan. 2022. [Online]. Available: <https://www.mdpi.com/2071-1050/14/2/601>
- [18] R. Bhagat, P. Singh, A. K. Srivastava, and E. S. Khanna, “A Comprehensive Review on Advanced Driver Assistance Systems (ADAS),” vol. 5, no. 5, 2023.
- [19] P. Shivale, N. Sonawane, S. Dharmale, T. Lokhandwala, and P. D. M. B. Wagh, “Advanced Driver Assistance System,” *International Journal for Research in Applied Science and Engineering Technology*, vol. 11, no. 3, pp. 1111–1113, Mar. 2023. [Online]. Available: <https://www.ijraset.com/best-journal/advanced-driver-assistance-system>
- [20] C. Schwarz and Z. Wang, “The Role of Digital Twins in Connected and Automated Vehicles,” *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 6, pp. 41–51, Nov. 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9670454/>
- [21] E. M. K. Shaikh, D. S. Palaniappan, E. F. Ali, and M. Khurram, “IDENTIFYING DRIVER BEHAVIOUR THROUGH OBD-II USING ANDROID APPLICATION,” 2020.
- [22] R. Kumar and A. Jain, “Driving behavior analysis and classification by vehicle OBD data using machine learning,” *The Journal of Supercomputing*,

-
- vol. 79, no. 16, pp. 18 800–18 819, Nov. 2023. [Online]. Available: <https://link.springer.com/10.1007/s11227-023-05364-3>
- [23] Secretaría de Seguridad Pública, “Reglamento de tránsito en carreteras y puentes de jurisdicción federal,” Reglamento publicado en el Diario Oficial de la Federación, México, Nov. 2012. [Online]. Available: https://dof.gob.mx/nota_detalle.php?codigo=5273539&fecha=22/11/2012
- [24] Secretaría de Comunicaciones y Transportes, “NOM-012-SCT-2-2017: Sobre el peso y dimensiones máximas con los que pueden circular los vehículos de autotransporte que transitan en las vías generales de comunicación de jurisdicción federal,” Norma Oficial Mexicana publicada en el Diario Oficial de la Federación, México, Dec. 2017. [Online]. Available: https://www.dof.gob.mx/nota_detalle.php?codigo=5502563&fecha=26/12/2017
- [25] —, “NOM-068-SCT-2-2014: Transporte terrestre - Servicio de autotransporte federal de pasaje, turismo, carga, sus servicios auxiliares y transporte privado - Condiciones físico-mecánicas y de seguridad para la operación en vías generales de comunicación de jurisdicción federal,” Norma Oficial Mexicana publicada en el Diario Oficial de la Federación, México, Jan. 2015. [Online]. Available: https://www.dof.gob.mx/nota_detalle.php?codigo=5372285&fecha=19/01/2015
- [26] Congreso de la Unión, “Ley general de movilidad y seguridad vial,” Ley publicada en el Diario Oficial de la Federación, México, May 2022, Última reforma: 29 de diciembre de 2023. [Online]. Available: https://www.diputados.gob.mx/LeyesBiblio/pdf/LGMSV_291223.pdf
- [27] M. Nasr Azadani and A. Boukerche, “Driving Behavior Analysis Guidelines for Intelligent Transportation Systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6027–6045, Jul. 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9430766/>
- [28] M. Malik and R. Nandal, “A framework on driving behavior and pattern using On-Board diagnostics (OBD-II) tool,” *Materials Today: Proceedings*, vol. 80, pp. 3762–3768, 2023. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2214785321052354>
- [29] F. Martinelli, F. Mercaldo, A. Orlando, V. Nardone, A. Santone, and A. K. Sangaiyah, “Human behavior characterization for driving style recognition in vehicle system,” *Computers & Electrical Engineering*, vol. 83, p. 102504, May 2020. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0045790617329531>
- [30] K. Uvarov and A. Ponomarev, “Driver Identification with OBD-II Public Data,” in *2021 28th Conference of Open Innovations Association (FRUCT)*. Moscow, Russia: IEEE, Jan. 2021, pp. 495–501. [Online]. Available: <https://ieeexplore.ieee.org/document/9347648/>

-
- [31] A. JayaSmruthi, J. Salman, K. Santhosh, and N. S. Kumar, "Driving Behavior Classification by Vehicle OBD data Using Machine Learning," vol. 11, no. 3, 2024.
- [32] E. Lattanzi and V. Freschi, "Machine Learning Techniques to Identify Unsafe Driving Behavior by Means of In-Vehicle Sensor Data," *Expert Systems with Applications*, vol. 176, p. 114818, Aug. 2021. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0957417421002591>
- [33] W.-H. Chen, Y.-C. Lin, and W.-H. Chen, "Comparisons of Machine Learning Algorithms for Driving Behavior Recognition Using In-Vehicle CAN Bus Data," in *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*. Spokane, WA, USA: IEEE, May 2019, pp. 268–273. [Online]. Available: <https://ieeexplore.ieee.org/document/8858531/>
- [34] M. Shahverdy, M. Fathy, R. Berangi, and M. Sabokrou, "Driver behavior detection and classification using deep convolutional neural networks," *Expert Systems with Applications*, vol. 149, p. 113240, Jul. 2020. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S095741742030066X>
- [35] J. Zhang, Z. Wu, F. Li, C. Xie, T. Ren, J. Chen, and L. Liu, "A Deep Learning Framework for Driving Behavior Identification on In-Vehicle CAN-BUS Sensor Data," *Sensors*, vol. 19, no. 6, p. 1356, Mar. 2019. [Online]. Available: <https://www.mdpi.com/1424-8220/19/6/1356>
- [36] J. Kim, H. Kim, and K. Lee, "Driving behavior classification and sharing system using cnn-lstm approaches and v2x communication," *Applied Sciences*, vol. 11, no. 21, p. 10420, 2021.
- [37] W. Tang, Z. Liu, H. Chen, and X. Wang, "Application of cnn-lstm model for vehicle acceleration prediction using car-following behavior data," *Journal of Advanced Transportation*, vol. 2024, p. 2442427, 2024.
- [38] F. Lindow and A. Kashevnik, "Driver Behavior Monitoring Based on Smartphone Sensor Data and Machine Learning Methods," in *2019 25th Conference of Open Innovations Association (FRUCT)*. Helsinki, Finland: IEEE, Nov. 2019, pp. 196–203. [Online]. Available: <https://ieeexplore.ieee.org/document/8981511/>
- [39] F. Qu, N. Dang, B. Furht, and M. Nojournian, "Comprehensive study of driver behavior monitoring systems using computer vision and machine learning techniques," *Journal of Big Data*, vol. 11, no. 1, p. 32, Feb. 2024. [Online]. Available: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-024-00890-0>
- [40] A. E. Abdelrahman, H. S. Hassanein, and N. Abu-Ali, "Robust Data-Driven Framework for Driver Behavior Profiling Using Supervised Machine Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3336–3350, Apr. 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9262069/>

-
- [41] M. Malik, R. Nandal, S. Dalal, V. Jalglan, and D.-N. Le, “Driving Pattern Profiling and Classification Using Deep Learning,” *Intelligent Automation & Soft Computing*, vol. 28, no. 3, pp. 887–906, 2021. [Online]. Available: <https://www.techscience.com/iasc/v28n3/42242>
- [42] N. Peppes, T. Alexakis, E. Adamopoulou, and K. Demestichas, “Driving Behaviour Analysis Using Machine and Deep Learning Methods for Continuous Streams of Vehicular Data,” *Sensors*, vol. 21, no. 14, p. 4704, Jul. 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/14/4704>
- [43] A. Mohammadnazar, R. Arvin, and A. J. Khattak, “Classifying travelers’ driving style using basic safety messages generated by connected vehicles: Application of unsupervised machine learning,” vol. 122, p. 102917. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0968090X20308160>
- [44] P. Bandyopadhyay *et al.*, “Unsupervised driving behavior analysis using representation learning and exploiting group-based training,” *Pattern Recognition*, vol. 124, p. 108467, 2022.
- [45] S. Li *et al.*, “Driving maneuvers detection using semi-supervised long short-term memory and smartphone sensors,” *Transportation Research Record*, vol. 2675, no. 9, pp. 609–620, 2021.
- [46] A. Rahman, I. H. Sarker, S. Lu, G. J. Bhat, N. Alrajeh, and M. Al-Amri, “Intrusion detection in vehicle controller area network (can) bus using machine learning: A comparative performance study,” *Sensors*, vol. 23, no. 7, p. 3610, 2023.
- [47] H. Ma, Y. Chen, W. Zhang, and X. Liu, “Driver digital twin for online recognition of distracted driving behaviors,” *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1613–1624, 2023.
- [48] X. Wang, S. Li, Y. Chen, and W. Zhang, “Real-time detection of abnormal driving behavior based on long short-term memory network and regression residuals,” *Transportation Research Part C: Emerging Technologies*, vol. 142, p. 103788, 2022.
- [49] K. Yang, C. A. Haddad, G. Yannis, and C. Antoniou, “Classification and Evaluation of Driving Behavior Safety Levels: A Driving Simulation Study,” *IEEE Open Journal of Intelligent Transportation Systems*, vol. 3, pp. 111–125, 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9705507/>
- [50] P. Ping, W. Qin, Y. Xu, C. Miyajima, and K. Takeda, “Impact of driver behavior on fuel consumption: Classification, evaluation and prediction using machine learning,” vol. 7, pp. 78 515–78 532. [Online]. Available: <https://ieeexplore.ieee.org/document/8727915/>
- [51] A. Cura, H. Kim *et al.*, “Cnn and lstm-based neural network models for bus driver behavior classification,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4821–4830, 2021.

-
- [52] F. Li, W. Zhang *et al.*, “Dsdcla: Driving style detection via hybrid cnn-lstm with multi-level attention fusion,” *Applied Intelligence*, vol. 53, no. 12, pp. 15 209–15 225, 2023.
- [53] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, “Digital twin in industry: State-of-the-art,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.
- [54] B. R. Barricelli, E. Casiraghi, and D. Fogli, “A survey on digital twin: Definitions, characteristics, applications, and design implications,” *IEEE Access*, vol. 7, pp. 167 653–167 671, 2019.
- [55] S. Deng, L. Ling, C. Zhang, C. Li, T. Zeng, K. Zhang, and G. Guo, “A systematic review on the current research of digital twin in automotive application,” *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 180–191, 2023.
- [56] M. Liu, S. Fang, H. Dong, and C. Xu, “Review of digital twin about concepts, technologies, and industrial applications,” *Journal of Manufacturing Systems*, vol. 58, pp. 346–361, 2021.
- [57] S. Deng, “A systematic review on the current research of digital twin in automotive application.”
- [58] W. A. Ali, M. P. Fanti, M. Roccotelli, and L. Ranieri, “A review of digital twin technology for electric and autonomous vehicles,” vol. 13, no. 10, p. 5871. [Online]. Available: <https://www.mdpi.com/2076-3417/13/10/5871>
- [59] B. Wang, C. Zhang, M. Zhang, C. Liu, Z. Xie, and H. Zhang, “Digital twin analysis for driving risks based on virtual physical simulation technology,” vol. 6, pp. 938–942. [Online]. Available: <https://ieeexplore.ieee.org/document/9877868/>
- [60] “Models 2018 c penhagen: proceedings: 21th ACM/IEEE international conference on model driven engineering languages and systems: MODELS 18: Copenhagen, denmark, october 14-19, 2018,” meeting Name: ACM/IEEE International Conference on Model Driven Engineering Languages and Systems.
- [61] E. Mohanraj, E. N. S. S, and S. S, “Digital twins for automotive predictive maintenance,” in *2024 International Conference on Inventive Computation Technologies (ICICT)*. IEEE, pp. 1579–1584. [Online]. Available: <https://ieeexplore.ieee.org/document/10544392/>
- [62] C. Campolo, A. Molinaro, B. Pizzimenti, and D. M. Zappal, “An edge-based digital twin framework for connected and autonomous vehicles: Design and evaluation,” vol. 12.
- [63] J. Smeets, K. zt rk, and R. Liebich, “Digital twin for motorcycle riding profile prediction,” vol. 161, p. 104569. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0968090X24000901>

-
- [64] HERE Technologies, “HERE: Location Data & Technology Platform,” <https://www.here.com/>, HERE Global B.V., Amsterdam, Netherlands, 2025, accedido: 26-jun-2025.
- [65] G. Xie, K. Yang, C. Xu, R. Li, and S. Hu, “Digital twinning based adaptive development environment for automotive cyber-physical systems,” vol. 18, no. 2, pp. 1387–1396. [Online]. Available: <https://ieeexplore.ieee.org/document/9372842/>
- [66] Instituto Mexicano del Transporte, *Folleto Pista de Pruebas*, Sanfandila, Querétaro, México, 2021, [Online]. Available: <https://imt.mx/images/files/GRAL/documentos/Folleto-Pista-Pruebas.pdf>. Accessed: 2021-12-14. [Online]. Available: <https://imt.mx/images/files/GRAL/documentos/Folleto-Pista-Pruebas.pdf>
- [67] —, *Características de la Pista*, Sanfandila, Querétaro, México, 2013, [Online]. Available: https://imt.mx/images/files/IVIE/Caracteristicas_Pista.pdf. Accessed: 2021-12-14. [Online]. Available: https://imt.mx/images/files/IVIE/Caracteristicas_Pista.pdf
- [68] VBOX Automotive, *VBOX 3i Data Logger - Industry Standard for Automotive Testing*, Buckingham, Reino Unido, 2024, [Online]. Available: <https://vboxautomotive.co.uk/index.php/en/products/data-loggers/vb3i>. [Online]. Available: <https://vboxautomotive.co.uk/index.php/en/products/data-loggers/vb3i>
- [69] Motorpasión México. (2018) Kia rio 2018: precios, versiones y equipamiento en México. [Online]. Available: <https://www.motorpasion.com.mx/kia/kia-rio-2018-precios-versiones-y-equipamiento-en-mexico>. [Online]. Available: <https://www.motorpasion.com.mx/kia/kia-rio-2018-precios-versiones-y-equipamiento-en-mexico>
- [70] R. J. Hyndman and G. Athanasopoulos, *Forecasting: principles and practice*. OTexts, 2018. [Online]. Available: <https://otexts.com/fpp2/>
- [71] E. Fix and J. L. Hodges Jr., “Discriminatory analysis, nonparametric discrimination: Consistency properties,” USAF School of Aviation Medicine, Randolph Field, Texas, Technical Report Project 21-49-004, Report 4, 1951. [Online]. Available: <https://apps.dtic.mil/sti/citations/ADA800276>
- [72] T. M. Cover and P. E. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967. [Online]. Available: <https://ieeexplore.ieee.org/document/1053964/>
- [73] S. C. Johnson, “Hierarchical clustering schemes,” *Psychometrika*, vol. 32, no. 3, pp. 241–254, Sep. 1967. [Online]. Available: <http://link.springer.com/10.1007/BF02289588>
- [74] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: <https://link.springer.com/article/10.1023/A:1010933404324>

-
- [75] —, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996. [Online]. Available: <https://link.springer.com/article/10.1007/BF00058655>
- [76] T. K. Ho, “The random subspace method for constructing decision forests,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998. [Online]. Available: <https://ieeexplore.ieee.org/document/709601>
- [77] G. Biau and E. Scornet, “A random forest guided tour,” *TEST*, vol. 25, no. 2, pp. 197–227, 2016. [Online]. Available: <https://link.springer.com/article/10.1007/s11749-016-0481-7>
- [78] G. Louppe, “Understanding random forests: From theory to practice,” 2014, arXiv:1407.7502 [stat]. [Online]. Available: <http://arxiv.org/abs/1407.7502>
- [79] E. Scornet, G. Biau, and J.-P. Vert, “Consistency of random forests,” *Annals of Statistics*, vol. 43, no. 4, pp. 1716–1741, 2015. [Online]. Available: <https://projecteuclid.org/euclid.aos/1434546220>
- [80] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006. [Online]. Available: <https://link.springer.com/article/10.1007/s10994-006-6226-1>
- [81] L. Mentch and G. Hooker, “Quantifying uncertainty in random forests via confidence intervals and hypothesis tests,” *Journal of Machine Learning Research*, vol. 17, no. 26, pp. 1–41, 2016. [Online]. Available: <https://jmlr.org/papers/v17/14-168.html>
- [82] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943. [Online]. Available: <https://link.springer.com/article/10.1007/BF02478259>
- [83] F. Rosenblatt, “The perceptron: A probabilistic model for information storage and organization in the brain,” *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958. [Online]. Available: <https://psycnet.apa.org/record/1959-09865-001>
- [84] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. [Online]. Available: <https://www.nature.com/articles/323533a0>
- [85] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ: Pearson, 2009.
- [86] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016. [Online]. Available: <https://www.deeplearningbook.org/>
- [87] C. M. Bishop, *Pattern recognition and machine learning*, ser. Information science and statistics. New York: Springer, 2006.
- [88] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations (ICLR)*, 2015, arXiv preprint arXiv:1412.6980. [Online]. Available: <https://arxiv.org/abs/1412.6980>

-
- [89] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [90] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.
- [91] A. Graves, “Supervised sequence labelling,” *Supervised sequence labelling with recurrent neural networks*, pp. 5–13, 2012.
- [92] C. Olah, “Understanding lstm networks,” <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>, 2015, accessed: 2024-01-15.
- [93] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with lstm,” *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.
- [94] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
- [95] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [96] R. Jozefowicz, W. Zaremba, and I. Sutskever, “An empirical exploration of recurrent network architectures,” *International conference on machine learning*, pp. 2342–2350, 2015.
- [97] K. Pearson, “On lines and planes of closest fit to systems of points in space,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.
- [98] H. Hotelling, “Analysis of a complex of statistical variables into principal components,” *Journal of Educational Psychology*, vol. 24, no. 6, pp. 417–441, 1933.
- [99] “scikit-learn: machine learning in Python — scikit-learn 1.5.2 documentation.” [Online]. Available: <https://scikit-learn.org/stable/>
- [100] P. J. Han J., Kamber M., “Data Mining: Concepts and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems) by Jiawei Han Micheline Kamber Jian Pei(2011-07-06) : Amazon.com.mx: Libros.” [Online]. Available: <https://www.amazon.com.mx/Data-Mining-Techniques-Management-2011-07-06/dp/B006USYM5U>
- [101] P. Trunfio, *Service-Oriented Distributed Knowledge Discovery*, ser. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series. Chapman and Hall/CRC, Oct. 2012, vol. 20121229. [Online]. Available: <http://www.crcnetbase.com/doi/book/10.1201/b12990>
- [102] S. Lloyd, “Least squares quantization in PCM,” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982. [Online]. Available: <http://ieeexplore.ieee.org/document/1056489/>

-
- [103] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “On clustering validation techniques,” *Journal of Intelligent Information Systems*, vol. 17, no. 2/3, pp. 107–145, 2001. [Online]. Available: <http://link.springer.com/10.1023/A:1012801612483>
- [104] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, “Understanding of Internal Clustering Validation Measures,” in *2010 IEEE International Conference on Data Mining*. Sydney, Australia: IEEE, Dec. 2010, pp. 911–916. [Online]. Available: <http://ieeexplore.ieee.org/document/5694060/>
- [105] T. Calinski and J. Harabasz, “A dendrite method for cluster analysis,” *Communications in Statistics - Theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974. [Online]. Available: <http://www.tandfonline.com/doi/abs/10.1080/03610927408827101>
- [106] P. J. Rousseeuw, “Silhouettes: A graphical aid to the interpretation and validation of cluster analysis,” *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/0377042787901257>
- [107] Y. Ma, W. Li, K. Tang, Z. Zhang, and S. Chen, “Driving style recognition and comparisons among driving tasks based on driver behavior in the online car-hailing industry,” vol. 154, p. 106096. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0001457521001275>
- [108] D. E. Miles and G. L. Johnson, “Aggressive driving behaviors: are there psychological and attitudinal predictors?” vol. 6, no. 2, pp. 147–161. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1369847803000226>
- [109] F. Sagberg, Selpi, G. F. Bianchi Piccinini, and J. Engström, “A review of research on driving styles and road safety,” vol. 57, no. 7, pp. 1248–1275, publisher: SAGE Publications Inc. [Online]. Available: <https://doi.org/10.1177/0018720815591313>
- [110] R. Vaiana, T. Iuele, V. Astarita, M. V. Caruso, A. Tassitani, C. Zaffino, and V. Giofré, “Driving behavior and traffic safety: An acceleration-based safety evaluation procedure for smartphones,” vol. 8, pp. 88–96.