



INSTITUTO POLITÉCNICO NACIONAL

CENTRO DE INVESTIGACIÓN EN CIENCIA APLICADA Y TECNOLOGÍA AVANZADA
UNIDAD QUERÉTARO

MODELOS VEHÍCULO-CONDUCTOR CON TÉCNICAS DE INTELIGENCIA ARTIFICIAL

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

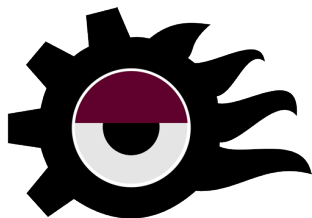
MAESTRO EN TECNOLOGÍA AVANZADA

P R E S E N T A :

ING. HUMBERTO CRUZ RAMÍREZ

DIRECTORES:

DRA. ILSE CERVANTES CAMACHO
DR. CARLOS ANDRÉS PÉREZ RAMÍREZ



CICATA-IPN
QUERÉTARO

QUERÉTARO, QRO, 2022



INSTITUTO POLITÉCNICO NACIONAL

SECRETARIA DE INVESTIGACIÓN Y POSGRADO

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

Ciudad de México, a 21 de junio del 2022

El Colegio de Profesores de Posgrado de **CICATA Unidad Querétaro** en su Sesión **ordinaria** No. **211210** celebrada el día **10** del mes **diciembre** de **2021**, conoció la solicitud presentada por el alumno:

Apellido Paterno:	Cruz	Apellido Materno:	Ramírez	Nombre (s):	Humberto
-------------------	-------------	-------------------	----------------	-------------	-----------------

Número de registro:

A	2	0	0	4	0	1
---	---	---	---	---	---	---

del Programa Académico de Posgrado:

Maestría en Tecnología Avanzada

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

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

Modelos Vehículo-Conductor con técnicas de Inteligencia Artificial.

Objetivo general del trabajo de tesis:

Desarrollo y formulación de un modelo vehículo-conductor que permita encontrar relaciones entre ambos componentes a través de una base de datos y de la aplicación de técnicas de Inteligencia Artificial, para su posible uso en aplicaciones de manejo de energía en vehículos eléctricos o híbridos, o en sistemas de asistencia a la conducción.
--

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

Director

Dra. Ilse Cervantes Camacho

2° Director:

Dr. Carlos Andrés Pérez Ramírez

No aplica:

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

No hubo trabajo experimental debido a la contingencia sanitaria

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 de Tesis

Dra. Ilse Cervantes Camacho

Aspirante

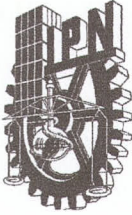
Humberto Cruz Ramirez

2° Director de Tesis

Dr. Carlos Andrés Pérez Ramírez

Presidente del Colegio

Dr. Juan Bautista Humado Ramos
SECRETARÍA DE EDUCACIÓN PÚBLICA
SECRETARÍA DE INVESTIGACIÓN Y POSGRADO
INSTITUTO POLITÉCNICO NACIONAL
CIENCIA Y TECNOLOGÍA AVANZADA
UNIDAD QUERÉTARO
DIRECCIÓN



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

SIP-14
REP 2017

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:

del (la) alumno (a):

Apellido Paterno:	Cruz	Apellido Materno:	Ramírez	Nombre (s):	Humberto
-------------------	------	-------------------	---------	-------------	----------

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 13% 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: Ninguna coincidencia en el reporte rebasa el 1% de similitud. La mayoría de las coincidencias son respecto a palabras comunes o palabras técnicas.

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:

El documento de tesis no constituye plagio y satisface los lineamientos actuales del Instituto.

COMISIÓN REVISORA DE TESIS

Dra. Ilse Cervantes Caracho

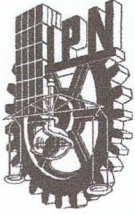
Dr. Carlos Andrés Pérez Ramírez

Dr. Manuel de Jesús Fabela Gallegos

Dr. Juan Bautista Hurtado Ramos

Dr. Joaquín Salas Rodríguez

Dr. Juan Bautista Hurtado Ramos
PRESIDENTE DEL COLEGIO DE PROFESORES
INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN CIENCIA APLICADA Y TECNOLOGÍA AVANZADA
UNIDAD QUERÉTARO
DIRECCIÓN



INSTITUTO POLITÉCNICO NACIONAL

SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

CARTA DE AUTORIZACIÓN DE USO DE OBRA PARA DIFUSIÓN

En la Ciudad de México el día 16 del mes de junio del año 2022, el (la) que suscribe Humberto Cruz Ramírez alumno(a) del programa Maestría en Tecnología Avanzada con número de registro A200401 adscrito(a) a CICATA-QUERÉTARO-IPN manifiesta que es autor(a) intelectual del presente trabajo de tesis bajo la dirección de la Dra. Ilse Cervantes Camacho y del Dr. Carlos Andrés Pérez Ramírez, y cede los derechos del trabajo intitulado "Modelos Vehículo-Conductor con técnicas de Inteligencia Artificial" 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 director(es). Este puede ser obtenido escribiendo a las siguiente(s) dirección(es) de correo humberto1crzram@gmail.com, icervantesc@ipn.mx y carlos.perez@uaq.mx. Si el permiso se otorga, al usuario deberá dar agradecimiento correspondiente y citar la fuente de este.

Ing. Humberto Cruz Ramírez

Estudiante

Dedicatoria

Esta tesis la dedico a mis padres por ser uno de los motores de mi vida.

A mi pareja, Diana, por ser la mujer más maravillosa que he conocido y ser el otro motor de mi vida.

Agradecimientos

Agradezco a Dios por haberme dado la oportunidad de culminar esta etapa de mi vida, pesada pero de mucho fruto profesional y personal, y por haber conocido a excelentes personas.

A mi padre por el apoyo y paciencia brindado hacia mi persona, a mi madre por ser la luz en mi camino y un gran ser humano.

A mis hermanas por sus risas y su apoyo.

A mi pareja por siempre estar ahí a pesar de todo y con todo, que siempre me demuestra que se pueden ver las cosas desde varias perspectivas y me motiva a ser mejor ser humano cada día. ¡Gracias mi Boli. Te amo mucho!

También muestro mi agradecimiento al Instituto Politécnico Nacional por permitirme crecer en sus aulas, que para mí la mayor parte fueron virtuales pero no por ello de menor calidad.

De manera especial, agradezco a mis Directores, la Dra. Ilse Cervantes Camacho y al Dr. Carlos Andrés Pérez Ramírez, por su orientación en todo momento, además de demostrarme que la calidad humana no va peleada con ser un investigador de alto nivel.

A mis Tutores por sus orientaciones y su constante actitud de mejora hacia mi trabajo. De manera especial, agradezco al Dr. Joaquín Salas por las facilidades brindadas en el uso del clúster de GPUs.

Al Dr. Manuel de Jesús Fabela Gallegos y al Instituto Mexicano del Transporte (IMT) por las facilidades brindadas para la experimentación en pista.

Al Ing. Guillermo Esqueda y a los participantes en la fase de experimentación del estudio.

Al Ing. Mauricio Cruz Acevedo por las orientaciones en torno a la forma y condiciones en que se llevó a cabo la experimentación.

Finalmente, y no menos importante, doy gracias a CONACYT por el apoyo económico brindado durante mis estudios de Maestría.

Resumen

Esta tesis propone modelos vehículo-conductor basados en Inteligencia Artificial como estrategia para ganar precisión en la descripción de la relación entre el vehículo y el conductor bajo diferentes escenarios de conducción o maniobra. En el caso de su aplicación en Sistemas Avanzados de Asistencia a la Conducción (ADASs), el objetivo principal de los modelos es la determinación y predicción de la dinámica rotacional del vehículo según el comportamiento del conductor, con el fin de usar esta información en la evaluación de riesgo de accidentes (e.g. vuelco o derrape lateral). Por otro lado, en lo concerniente a las Estrategias de Manejo de Energía (EMSs), lo que se busca es hallar una predicción de la velocidad en el largo plazo, mediante modelos vehículo-conductor que relacionen las entradas ejercidas por el usuario (frenado, aceleración, ángulo y torque del volante, y rapidez) con el perfil de velocidad del vehículo (salida). Lo anterior con miras a calcular el balance de potencia para un vehículo eléctrico o híbrido eléctrico en horizontes de predicción amplios.

Este trabajo usa una recolección de datos experimentales de variables vehiculares provenientes de la conducción de cuatro personas en una pista de manejo que no contenía peatones, ni interacción con otros vehículos. La pista permitía maniobras de vueltas izquierda y derecha, con y sin peralte, así como tramos rectos y codos hacia la izquierda y hacia la derecha. Además, se les pedía hacer una maniobra de evasión de 16 obstáculos (también conocida como *Prueba de Slalom*). Estos datos se usaron en la derivación de los modelos vehículo-conductor para ambas perspectivas, así como para la programación de dichos modelos mediante técnicas de aprendizaje automático (*ML*, *machine learning*). Para las EMSs, los algoritmos empleados son: redes tipo *Gated Recurrent Units* (GRU), *Long-Short Term Memory* (LSTM) Vainilla y Transformer. Para el caso de ADASs, el Transformer se substituyó por LSTM Codificador-Decodificador (o LSTM Encoder-Decoder). La elección de las técnicas de ML fue hecha tomando en consideración las áreas de oportunidad encontradas al revisar el estado del arte.

Los modelos ML propuestos para la perspectiva de EMSs presentan resultados muy notables para los cuatro sujetos a lo largo de un horizonte de predicción de 49 s. Por otra parte, para la perspectiva de ADASs, sólo para un sujeto no se alcanzó un buen nivel de predicción. El sujeto en el que falló la estimación tiene entrenamiento en conducción técnica, por lo que los resultados sugieren la necesidad de usar instrumentaciones adicionales de manera que se predigan ciertas interacciones que generen movimientos abruptos del sistema de tracción.

En cuanto a las contribuciones de la tesis para la perspectiva de EMSs está el considerar las acciones del conductor (*además del ángulo del volante*) como factores que influyen en la predicción del perfil de velocidad (o rapidez) del vehículo; otra es la predicción de la variable en un horizonte de tiempo largo, 49 s, el cual es amplio considerando la duración de las pruebas. Por otro lado, en lo concerniente a ADASs, una de las aportaciones es el manejo de un modelo vehículo-conductor de múltiples entradas y de múltiple salidas, además de que las salidas son referentes a variables relacionadas con el riesgo de derrape lateral o de vuelco de un vehículo. Dichos modelos son escasos en la literatura. Otra aportación para ADASs es el manejo de una mayor cantidad de entradas relacionadas con el conductor que las halladas en el estado del arte.

En lo concerniente a las áreas de oportunidad de la tesis, para la parte de EMSs, se tiene la restricción del tipo de prueba y que no hay tanta diversidad de maniobras. Otra es que el vehículo utilizado es un auto convencional, habría que experimentar con un vehículo híbrido. En lo referente a ADASs, se hallan lo limitado de las pruebas experimentales y el reducido número de participantes. Finalmente, los resultados revelan la necesidad de hacer pruebas adicionales con más sujetos, las cuales no fueron posibles dentro de esta tesis debido a la pandemia de COVID-19.

Abstract

This thesis proposes vehicle-driver models based on Artificial Intelligence as a strategy to gain precision in describing the relationship between the vehicle and the driver under different driving scenarios. The models are foreseen in two main applications: Advanced Driving Assistance Systems (ADASs) and Energy Management Strategies (EMSs). For ADASs, the main objective of the models was to determine and predict the vehicle's rotational dynamics according to driver behavior, which would help assess the risk of accident (e.g., roll-over or side slip). For EMSs, the objective was to relate the action inputs of the driver (braking, acceleration, steering angle, and steering torque) with the vehicle speed profile to calculate the power demand of an electric or hybrid electric vehicle in broad prediction horizons.

Here, an experimental data collection of vehicle variables from four people was used to derive the models. The data was collected from driving conditions on a driving track that did not contain pedestrians or interaction with other vehicles. The track allowed left and right turns and straight sections. The individuals were asked to do an evasive maneuver of 16 obstacles (also known as *Slalom test*). These data were used to derive the vehicle-driver models for both applications using machine learning techniques (*ML*). For EMSs, the algorithms used were *Gated Recurrent Units* (GRU), *Long-Short Term Memory* (LSTM) Vanilla, and Transformer networks. For ADASs, the Transformer was replaced by LSTM Encoder-Decoder. The choice of ML techniques was made taking into consideration the areas of opportunity encountered when reviewing the state of the art.

The ML models proposed for EMSs displayed remarkable results for the four individuals tested up to a prediction horizon of 49 s. While for ADASs only one subject did not achieve a good prediction level. This subject received training on technical driving, so the results suggested the need to use additional instrumentation to predict interactions generating abrupt movements.

The contributions of this dissertation reside in the application. Regarding models for EMSs, the contribution is to consider the driver's actions (*in addition to the angle of the steering wheel*) as factors that influence the vehicle's speed profile prediction. Another contribution is to show that ML models allow reliable prediction up to time horizons of 49 s, which is long considering previous work. Regarding models for ADASs, the contribution is the derivation of vehicle-driver models with multiple inputs and outputs, where the outputs are variables related to the risk assessment of side slip or rollover of a vehicle. Such models are scarce in the literature.

This work revealed research opportunities for EMSs—particularly the need to design experiments with higher diversity of maneuvers. For ADASs, we found that experimental data could be improved using more participants, which was not possible in this work due to the COVID-19 pandemic.

Índice general

Agradecimientos	II
Resumen	III
Abstract	IV
Índice de figuras	VIII
Índice de tablas	X
1. Introducción	1
1.1. Panorama General	1
1.2. Justificación	2
1.3. Estado del arte	4
1.3.1. Sistemas Avanzados de Asistencia a la Conducción	5
1.3.2. Contribuciones en materia de Sistemas de Asistencia a la Conducción	12
1.3.3. Estrategias de Manejo de Energía	15
1.3.4. Contribuciones en materia de Estrategias de Manejo de Energía	18
1.4. Objetivos	19
1.4.1. Objetivo General	19
1.4.2. Objetivos Particulares	19
1.5. Estructura del Documento	20
1.6. Sumario	21
2. Marco Teórico	23
2.1. Sistema de coordenadas del vehículo	23
2.2. Aprendizaje Automático (o Machine Learning)	25
2.3. Elementos Básicos Constituyentes de un Algoritmo ML Supervisado	26
2.3.1. Modelo ML	26
2.3.2. Función costo o de pérdida	28

2.3.3. Optimizador	28
2.4. Redes Neuronales Artificiales	29
2.4.1. Funciones de Activación	29
2.5. Consideraciones Importantes acerca del Manejo de Datos por parte de Algoritmos ML Supervisados	30
2.5.1. Tipos de Conjuntos de Datos	31
2.5.2. Validación Cruzada K-Folds	31
2.5.3. Tipos de Datos	33
2.5.4. Codificación de Variables Categóricas	33
2.5.5. Estandarización de los datos numéricos	34
2.5.6. Normalización de los datos numéricos	34
2.5.7. Filtro de media móvil	35
2.6. Criterios de desempeño de las técnicas ML supervisadas	35
2.7. Sobreajuste y Regularización por Dilución	36
2.8. Redes Neuronales Recurrentes	37
2.9. Redes Long-Short Term Memory (o Redes LSTM)	37
2.9.1. Estados y Compuertas de la Neurona LSTM	39
2.9.2. Configuraciones de Redes LSTM	40
2.10. Redes Gated Recurrent Unit (o <i>Redes GRU</i>)	42
2.10.1. Etapa 1. Compuerta de Actualización para el tiempo actual	42
2.10.2. Etapa 2. Compuerta de Reinicio para el tiempo actual	43
2.10.3. Etapa 3. Memoria Actual	43
2.10.4. Etapa 4. Contenido Final de la Memoria en el tiempo actual	43
2.11. Transformer	44
2.11.1. Arquitectura	44
2.12. Transformer + Time2Vector	45
2.13. Time2Vector	46
2.14. Arquitectura empleada en la presente tesis	46
2.15. Sumario	47
3. Metodología	48
3.1. Descripción de la Base de Datos	50
3.1.1. Lugar del Experimento	50
3.1.2. Equipos para Adquisición de Datos	50
3.1.3. Sujetos	52
3.1.4. Vehículo	52
3.1.5. Condiciones del Experimento	53

3.1.6.	Variables físicas del vehículo y del conductor	54
3.2.	Tratamiento de los datos	56
3.2.1.	Lectura y tratamiento de la información aportada por el Transceptor PCAN-USB	56
3.2.2.	Lectura y tratamiento de la información aportada por el VBOX3i y el IMU03	56
3.2.3.	Compatibilización de Datos	57
3.2.4.	Filtrado de Información	57
3.3.	Construcción de los Modelos Vehículo-Conductor	57
3.3.1.	Estrategias de Manejo de Energía	57
3.3.2.	Sistemas Avanzados de Asistencia a la Conducción	59
3.4.	Programación de los Modelos Vehículo-Conductor	59
3.4.1.	Unidades de Procesamiento Gráfico (GPUs)	60
3.4.2.	Arreglo Experimental	60
3.4.3.	Normalización y Estandarización de los Datos	60
3.4.4.	Procesamiento de las Series de Tiempo	61
3.4.5.	Técnicas de Aprendizaje Automático	64
3.5.	Comparación de Técnicas de Aprendizaje Automático para la delimitación del mejor exponente del Modelo Vehículo-Conductor	65
3.5.1.	Criterios de desempeño para la elección del mejor exponente del Modelo Vehículo-Conductor	65
3.5.2.	Procedimiento de comparación	66
3.6.	Sumario	66
4.	Análisis de Resultados. EMS.	67
4.1.	Sujeto 1	67
4.2.	Sujeto 2	68
4.3.	Sujeto 3	69
4.4.	Sujeto 4	70
4.5.	Discusión de Resultados	71
4.6.	Sumario	72
5.	Análisis de Resultados. ADAS.	73
5.1.	Sujeto 1	74
5.1.1.	Ángulo de Viraje (Ángulo Yaw) o Ángulo de Deslizamiento Lateral (Slip Angle)	74
5.1.2.	Ángulo de Balanceo (Ángulo Roll)	75
5.1.3.	Tasa de cambio del Ángulo de Viraje (o Yaw Rate)	77
5.1.4.	Tasa de cambio del Ángulo de Cabeceo (o Pitch Rate)	78

5.1.5. Tasa de cambio del Ángulo de Balanceo (o Roll Rate)	79
5.2. Sujeto 2	80
5.2.1. Ángulo de Viraje (Ángulo Yaw) o Ángulo de Deslizamiento Lateral (Slip Angle) 80	
5.2.2. Ángulo de Balanceo (Ángulo Roll)	81
5.2.3. Tasa de cambio del Ángulo de Viraje (o Yaw Rate)	82
5.2.4. Tasa de cambio del Ángulo de Cabeceo (o Pitch Rate)	83
5.2.5. Tasa de cambio del Ángulo de Balanceo (o Roll Rate)	84
5.3. Sujeto 3	86
5.3.1. Ángulo de Viraje (Ángulo Yaw) o Ángulo de Deslizamiento Lateral (Slip Angle) 86	
5.3.2. Ángulo de Balanceo (Ángulo Roll)	87
5.3.3. Tasa de cambio del Ángulo de Viraje (o Yaw Rate)	88
5.3.4. Tasa de Cambio del Ángulo de Cabeceo (o Pitch Rate)	89
5.3.5. Tasa de Cambio del Ángulo de Balanceo (o Roll Rate)	89
5.4. Sujeto 4	90
5.4.1. Ángulo de Viraje (Ángulo Yaw) o Ángulo de Deslizamiento Lateral (Slip Angle) 91	
5.4.2. Ángulo de Balanceo (Ángulo Roll)	92
5.4.3. Tasa de Cambio del Ángulo de Viraje (o Yaw Rate)	93
5.4.4. Tasa de Cambio del Ángulo de Cabeceo (o Pitch Rate)	94
5.4.5. Tasa de Cambio del Ángulo de Balanceo (o Roll Rate)	95
5.5. Sumario	96
6. Conclusiones y Prospectivas	98
6.1. Conclusiones	98
6.1.1. Estrategias de Manejo de Energía	98
6.1.2. Sistemas Avanzados de Asistencia a la Conducción	99
6.1.3. Conclusiones Generales	100
6.2. Resumen de Resultados	101
6.2.1. Estrategias de Manejo de Energía	101
6.2.2. Sistemas Avanzados de Asistencia a la Conducción	101
6.3. Prospectivas	101
6.3.1. Estrategias de Manejo de Energía	102
6.3.2. Sistemas Avanzados de Asistencia a la Conducción	102
Anexos	
A. Tablas. EMS	104
B. Tablas. ADAS	105

Índice de figuras

1.1. Clasificación de EMSs	16
1.2. EMSs. Clasificación de técnicas para velocidad	18
1.3. Estructura del Documento.	21
2.1. Sistema de referencia a bordo del vehículo	24
2.2. Sistema de referencia fijo en Tierra.	25
2.3. Clasificación de las técnicas ML	26
2.4. Métodos ML Supervisados.	27
2.5. Elementos Constituyentes de un Método ML Supervisado.	27
2.6. Funciones Costo.	28
2.7. Red Neuronal Artificial.	30
2.8. Funciones de Activación	30
2.9. Tipos de Conjuntos de Datos en ML.	31
2.10. Validación Cruzada.	32
2.11. Regularización por Dilución (Dropout)	37
2.12. Red Neuronal Recurrente.	38
2.13. Neurona LSTM.	38
2.14. LSTM Vainilla	41
2.15. LSTM Codificador-Decodificador	41
2.16. Neurona GRU.	42
2.17. Arquitectura Transformer Tradicional.	44
3.1. Descripción General de la Metodología usada. Parte 1.	48
3.2. Descripción General de la Metodología usada. Parte 2.	49
3.3. Equipos de Adquisición de Datos.	51
3.4. Recorrido en la Pista. Sentido AH.	53
3.5. Maniobras. Sentido AH.	54
3.6. Filtrado de la Señal <i>Rapidez</i>	58

3.7. Procesamiento de las Series de Tiempo	62
4.1. EMSs. Sujeto 1. Rapidez	68
4.2. EMSs. Sujeto 2. Rapidez	69
4.3. EMSs. Sujeto 3. Rapidez	70
4.4. EMSs. Sujeto 4. Rapidez	71
5.1. ADASs. Sujeto 1. Slip Angle	75
5.2. ADASs. Sujeto 1. Roll	76
5.3. ADASs. Sujeto 1. Yaw Rate	77
5.4. ADASs. Sujeto 1. Pitch Rate	78
5.5. ADASs. Sujeto 1. Roll Rate	79
5.6. ADASs. Sujeto 2. Slip Angle	81
5.7. ADASs. Sujeto 2. Roll	82
5.8. ADASs. Sujeto 2. Yaw Rate	83
5.9. ADASs. Sujeto 2. Pitch Rate	84
5.10. ADASs. Sujeto 2. Roll Rate	85
5.11. ADASs. Sujeto 3. Slip Angle	86
5.12. ADASs. Sujeto 3. Roll	87
5.13. ADASs. Sujeto 3. Yaw Rate	88
5.14. ADASs. Sujeto 3. Pitch Rate	89
5.15. ADASs. Sujeto 3. Roll Rate	90
5.16. ADASs. Sujeto 4. Slip Angle	91
5.17. ADASs. Sujeto 4. Roll	92
5.18. ADASs. Sujeto 4. Yaw Rate	93
5.19. ADASs. Sujeto 4. Pitch Rate	94
5.20. ADASs. Sujeto 4. Roll Rate	95

Índice de tablas

1.1. Estimaciones basadas en Modelo y Estimaciones basadas en Técnicas Estadísticas. . .	5
1.2. Clasificación de las Estrategias para la Predicción de Dinámica Vehicular. ADASs. . .	6
1.3. ADASs. Comparación de trabajos	14
1.4. EMSs. Comparación de trabajos	19
2.1. Tipos de codificación de datos categóricos	34
3.1. Características del vehículo	52
3.2. Variables Físicas	55
3.3. EMSs. Modelo Vehículo-Conductor	58
3.4. ADASs. Modelo Vehículo-Conductor	59
3.5. Combinaciones Historial-Horizonte para ADASs	63
3.6. Combinaciones Historial-Horizonte para EMSs	63
3.7. Técnicas de ML para EMSs	64
3.8. Técnicas de ML para ADASs	65
6.1. EMSs. Resumen de resultados	101
6.2. ADASs. Resumen de resultados	101
A.1. Resultados Técnicas ML. EMSs. Todos	104
B.1. Resultados Técnicas ML. ADASs. Sujeto 1	106
B.2. Resultados Técnicas ML. ADASs. Sujeto 2	107
B.3. Resultados Técnicas ML. ADASs. Sujeto 3	108
B.4. Resultados Técnicas ML. ADASs. Sujeto 4	109

Capítulo 1

Introducción

En este capítulo se presentan el panorama general, la justificación, el estado del arte, los objetivos (tanto general como particulares) y la descripción general de la metodología para la presente tesis.

1.1. Panorama General

El aprendizaje máquina (en inglés *Machine Learning*, o *ML*) se refiere a técnicas computacionales que aprenden reglas e interacciones de datos masivos. Actualmente, el aprendizaje máquina es una herramienta poderosa para tareas de predicción y reconocimiento de patrones. Con la digitalización cada vez más fuerte de los sistemas o procesos, el auge de los sistemas embebidos, el crecimiento del Internet de las Cosas (IoT) y de la capacidad de cómputo, el ML se ha estado involucrando en tareas más cotidianas. Los vehículos son una de sus aplicaciones.

Las Técnicas ML han revolucionado dos áreas importantes del ámbito automotriz: los Sistemas Avanzados de Asistencia a la Conducción (o *ADASs*, por sus siglas en inglés) y las Estrategias de Manejo de Energía (*EMSs*, por sus siglas en inglés). Ambas áreas requieren de la predicción de la dinámica del vehículo a partir de las acciones ejecutadas por el conductor. A ello se le conoce como Modelos vehículo-Conductor.

Antiguamente, para automatizar los procesos, el usuario o programador debía establecer las reglas del programa a través de sentencias de código [42]. Con el crecimiento de los sensores y de la digitalización, y con ello, el número de datos de diferentes orígenes, el poder influir en estas relaciones mediante reglas o sentencias de código se complicó [103]. Por ello, el ML se ha transformado en una potencial herramienta que, a través de fundamentos en Inteligencia Artificial, puede proveer de relaciones entre los mismos datos, lo que permite la modelación de fenómenos complejos, además de una predicción de su comportamiento.

Los vehículos actuales tienen sensores, actuadores y un sistema de procesamiento central, con lo que concurren en ellos una inmensa cantidad de información que debe ser manejada de manera

correcta para que el vehículo tenga una función apropiada. Con la investigación enfocada a los vehículos eléctricos, vehículos híbridos eléctricos (*HEVs*, por sus siglas en inglés) y los vehículos autónomos (*AVs*, por sus siglas en inglés), el estudio de técnicas que hallen relaciones y conexiones dentro de los mismos datos, o técnicas *data-driven* (como también se les conoce), se ha convertido en una área de oportunidad para mejorar el desempeño.

Específicamente, en la presente tesis, se hablará de dos áreas en las que el ML puede aportar mucha información relacionada con el vehículo y su entorno: Sistemas Avanzados de Asistencia al Conductor (ADASs) y Estrategias de Manejo de Energía (EMSs). La primera se refiere a aquellos mecanismos que mejoran la seguridad del manejo y que están conformados por sistemas activos o pasivos que perciben el mundo a través de sensores y toman una acción en función de la información que recibieron para salvaguardar la integridad de conductor y pasajeros [112]. Dentro de esta aplicación, la predicción de las variables de dinámica vehicular juega un papel trascendental, puesto que una variación en el rango nominal de las mismas indica un posible riesgo para el vehículo y para sus ocupantes.

La segunda aplicación es la Estrategia Manejo de Energía. En ésta lo que se busca es la óptima distribución de potencia en los sistemas eléctricos eléctrico o híbridos eléctricos en función de la predicción de perfiles de velocidad y aceleración principalmente [71].

1.2. Justificación

En lo concerniente a los vehículos, ML desempeña un papel sustancial por ser predictores para la evaluación de riesgo de accidentes o para el conocimiento del futuro desempeño del vehículo.

En lo que tiene que ver con ADASs, la desviación de ciertos parámetros de la dinámica vehicular (e.g. ángulo de balanceo y el ángulo de deslizamiento lateral) con respecto a valores nominales puede representar un riesgo de accidente, por ende, es indispensable la predicción apropiada de estas variables en un coherente horizonte de predicción. La parte de ADASs tiene el objeto de poder minimizar los accidentes vehiculares, problema que está muy presente en la actualidad. En el mundo, según datos de la Organización Mundial de la Salud del 2017, cada año mueren cerca de 1.3 millones de personas en accidentes de tránsito, los cuales constituyen una de las principales causas de muerte en todos los grupos sociales, especialmente en los jóvenes de 15-19 años. [90] Otro hecho preocupante es que el 93% de las muertes por accidentes vehiculares ocurren en países emergentes, a pesar de que ellos tienen menos de la mitad del padrón vehicular. Según datos de la Administración Federal de Carreteras de Estados Unidos del 2021, una división del Departamento de Transporte, en el 90% de las colisiones la razón crítica es el factor humano [35].

México ocupa el séptimo lugar a nivel mundial y el tercero en la región de Latinoamérica en muertes por siniestros viales, con 22 decesos de jóvenes de entre 15 y 29 años al día, y 24 mil decesos

en promedio al año. Los siniestros viales constituyen la primera causa de muerte en jóvenes entre 5 y 29 años de edad y la quinta entre la población general. Lo anterior según datos del Instituto Nacional de Salud Pública del 2017. [58]

Según el *Anuario estadístico de colisiones en carreteras federales 2020* [55], el 70.67% de los siniestros ocurridos durante el año 2020 en México se debieron al factor humano o al conductor, contabilizando 10,859 de los 15,365 ocurridos. En lo concerniente al tipo de colisión, las cifras nacionales reflejaron que la salida del camino (relacionada al derrape lateral) fue el principal tipo de colisión (con el 22.4% de los percances). El vuelco, por otra parte, fue mucho menos común que la salida del camino, pero siguió estando presente con un 4.28%.

Relacionado con las Estrategias de Manejo de Energía (EMSs), se requieren en vehículos eléctricos o híbridos eléctricos, para una reducción de emisiones, ahorro de combustible, maximización de la autonomía (distancia recorrida por carga de batería o combustible) y una más eficiente distribución de potencia. Un HEV es un vehículo que posee un motor de combustión interna, además de otros motores eléctricos que abren la posibilidad de la repartición de potencia demandada. Estos vehículos tienen un sistema de almacenamiento de energía (provisto de baterías y /o supercapacitores) que funciona como un almacén de energía para la propulsión [100]. La cualidad de tener múltiples fuentes de energía abre la posibilidad de la administración óptima de energía. El mercado mundial de los HEVs acumuló 4,169,000 unidades para 2018, además de que se espera una tasa de crecimiento anual compuesta de 8.9% para el periodo 2018-2025 hasta alcanzar 7,593,000 unidades en 2025. [79]

En México, el número de vehículos eléctricos e híbridos vendidos superó las 39,100 unidades en 2021, representando alrededor del 60% con respecto al año anterior (el COVID-19 tuvo un importante papel en ello). [9] Descartando la pandemia, la proyección nacional de HEVs es próspera, no obstante, también representa una serie de desafíos [118], como la implementación de estaciones de carga, la evaluación de la capacidad de carga de la red eléctrica, el manejo óptimo de su energía de manera que su autonomía o eficiencia se maximice.

En particular, para que el HEV mejore su desempeño en relación a la forma de manejar de los individuos, las EMSs son prioridad, y para la implementación de las EMSs, la predicción de perfiles de aceleración y velocidad es una tarea preponderante. *La razón de ser de los perfiles de estas variables es debido a que la energía de un vehículo se relaciona más con su energía cinética (donde la velocidad y su derivada juegan un papel sustancial) que con la energía potencial (relacionada con la posición) del mismo.*

De esta manera, el ML constituye un instrumento *ad hoc* para el tratamiento de ambos problemas (i.e. el manejo de energía y la asistencia al conductor) por ser capaz de recibir una gran cantidad de información, aprender de los mismos datos, y poder manejar la gran cantidad de no linealidades e incertidumbre propias de la dinámica vehicular. [89] Su potencial, en conjunto con el interés en estos dos problemas importantes para la industria automotriz, es lo que justifica la investigación y

la elaboración de la presente tesis.

1.3. Estado del arte

En esta sección se exponen los antecedentes bibliográficos que sustentan las aportaciones de esta tesis.

Un Modelo de Conducción describe el comportamiento de un vehículo durante la tarea de conducción. Aquí participan diversos agentes, como el conductor, el vehículo, el camino, el ambiente (e.g. tráfico) o una mezcla de ellos [82].

Un caso especial de Modelos de Conducción son los Modelos vehículo-Conductor, que se tratan de aquéllos en los que no sólo se pretende describir la dinámica vehicular, sino su relación con las acciones del conductor. De tal manera que, si el conductor ejecuta alguna acción, dicho movimiento se vea reflejado en el otro agente (i.e., el vehículo). En otras palabras, si el conductor vira a la izquierda mediante el volante, dicha acción se debe ver reflejada en el vehículo. El comportamiento de una persona manejando depende también del tipo de vehículo, pues no es lo mismo manejar un auto que un camión. También, el tipo de maniobras que puede hacer el conductor dependerá de las capacidades del vehículo (i.e., características físicas como el tipo de llanta, de suspensión, la altura del centro de gravedad, entre otras). Así también, cuando el mismo vehículo es manejado por diferentes personas, la respuesta del vehículo y su desempeño en las maniobras dependerá de lo que hace el conductor, lo cual es función entre otros factores de su propia percepción de riesgo, su experiencia y su pericia.

Esto es, **los Modelos vehículo-Conductor son personalizados o por conductor, no se pueden aplicar indiscriminadamente a cualquier operador**. De manera específica, se trata de modelar la relación entre las acciones ejercidas por el conductor (entradas) con el comportamiento del vehículo (salidas) con un determinado objetivo. Para el caso de la presente tesis, la tarea es predecir la dinámica vehicular o las variables de estado del vehículo a través de un Modelo vehículo-Conductor como paso previo al desarrollo de Sistemas Avanzados de Asistencia a la Conducción (ADASs) y Estrategias de Manejo de Energía (EMSs).

Aquí es sustancial aportar una idea. *La mayoría de los vehículos no pueden tener sensores embebidos o externos para cuantificar todas las variables dinámicas directamente ya sea por la dificultad técnica o económica que supone, pero lo que sí se puede hacer es medir algunas con los sensores y estimar otras a partir de aquéllas (este problema también se conoce como **problema de observabilidad**) [132].* De ahí la relevancia de la estimación y futura predicción de las variables dinámicas de un vehículo, a partir de un grupo limitado de variables a medir.

1.3.1. Sistemas Avanzados de Asistencia a la Conducción

En la literatura se encuentran dos grandes enfoques para la estimación de los estados de la dinámica vehicular: estimaciones basadas en modelo y estimaciones data-driven. La primer categoría requiere un modelo matemático del vehículo. En cambio, los modelos data-driven aprenden de los mismos datos, no se requiere de un modelo matemático a priori. Para el caso de la primer categoría, éstos son dependientes de la complejidad del modelo y del conocimiento de los parámetros del vehículo; desafortunadamente tienen la desventaja ante la variación de parámetros o de condiciones [64]. En un ambiente real no hay una estabilidad en las condiciones del camino, vehículo y conductor, son variables y desafiantes.

La primer subcategoría de los modelos data-driven, las estimaciones basadas en técnicas estadísticas, abordan una formalización de las relaciones entre variables en los datos en forma de ecuaciones matemáticas [117]. No se tienen las ecuaciones a priori, sino que se formulan a partir de los mismos datos. En la Tabla 1.1 se muestran trabajos referentes a estas dos técnicas presentadas hasta el momento.

Tabla 1.1: Estimaciones basadas en Modelo y Estimaciones basadas en Técnicas Estadísticas.

Clase de Algoritmo	Técnicas Empleadas	Ejemplos de Trabajo
Modelos Matemáticos.	Modelo Bicicleta (Dinámica Lateral)	[29], [59], [135]
	Modelo de Dinámica Longitudinal	[66], [53], [116]
	Modelo de Dinámica Rotacional	[74], [47], [34]
Técnicas Estadísticas.	Asimetría, Dispersión, Curtosis, Árbol Lógico Probabilístico, Análisis Multivariado, Modelo Logit Condicional.	[130], [124], [44]

La segunda subcategoría de los modelos data-driven son los modelos o estimaciones basadas en Inteligencia Artificial, éstos descienden de la Informática, también aprenden de los datos, pero no requieren suposiciones previas sobre las relaciones subyacentes entre las variables, todo actúa como una *caja negra* que halla relaciones/patrones entre todos los datos [117]. Las principales ventajas de este tipo de modelos es el manejo de no linealidades, no necesitan conocer todos los parámetros de las variables que se involucran en la conducción (*especialmente de los neumáticos*, tema desafiante para los investigadores) y evitan errores de integración, ya que simplemente se alimentan de datos recolectados durante esta tarea. No obstante, su principal limitación es que no pueden generalizar más de lo que fueron entrenados [24]; es decir, si el algoritmo de Inteligencia Artificial se entrenó para un circuito con ocho maniobras, es muy difícil que se pueda aceptar su respuesta ante nuevas condiciones de conducción. Se tendría entonces que re-entrenar el algoritmo con nuevas maniobras o pruebas. Ello es la razón de que sigan siendo la segunda opción para estimar ciertas variables vehiculares. En el siguiente apartado se entrará de lleno a esta categoría.

Modelos basados en Inteligencia Artificial

La estimación de estados de la dinámica vehicular se puede efectuar a través de modelos en los que participan diversos agentes. Muchos sólo son modelos del vehículo, otros ya incluyen acciones de conductor, por lo que ya se les considera modelos vehículo-conductor, pero lo sustancial en este punto es que lleven consigo la aplicación de técnicas de Inteligencia Artificial. Esta forma de estimación supera algunas de las dificultades de las estimaciones basadas en modelo, por manejar de buena manera el alto contenido no lineal del fenómeno y evitar las inexactitudes del modelo matemático del vehículo. El algoritmo ML usado por excelencia son las Redes Neuronales Artificiales (por sus siglas en inglés, *ANNs*) o algunas de sus variedades [60]. Lo anterior se confirma en [25] y en [89]. La mayoría de los trabajos encontrados en el estado del arte se refieren a modelos de múltiples entradas que sólo predicen una salida o una variable de la dinámica vehicular, ya que sólo se enfocan en una problemática en particular (riesgo de vuelco o de derrape lateral) [74]. Pocos son los autores que abordan múltiples salidas. Primero se definirán los modelos mono-salida y después los de múltiples salidas.

La clasificación de las estrategias para la predicción de dinámica vehicular, para la perspectiva de Sistemas Avanzados de Asistencia a la Conducción, se expresa en la Tabla 1.2. Se tienen dos grandes categorías dependiendo al número de estimaciones predichas por los modelos. Si es sólo una variable, se trata de Modelos Mono-Salida; y si son varias, Modelos de Múltiples Salidas. Ya dentro de cada categoría, lo único que varían son las distintas variables consideradas.

Tabla 1.2: Clasificación de las Estrategias para la Predicción de Dinámica Vehicular. ADASs.

Perspectiva de Trabajo	Tipo de Modelo de Inteligencia Artificial	VARIABLES PREDICHAS	Trabajos Relacionados
ADASs	Modelos Mono-Salida	Posición / Velocidad / Rapidez / Actitud	[88], [65], [27], [69]
		Ángulo de Balanceo (Ángulo Roll)	[38], [12], [11], [110], [109]
		Ángulo de Deslizamiento Lateral (Slip Angle) del vehículo	[26], [62], [131], [87], [136]
		Parámetros de Neumáticos	[133], [115], [8], [21]
		Parámetros del Camino	[48], [1], [125], [10]
	Modelos Múltiples Salidas	Ángulo de Deslizamiento Lateral (Slip Angle) + Ángulo de Balanceo (Ángulo Roll)	[43], [14], [6]
		Varios (No incluye la categoría superior)	[52], [113], [15]

Modelos de una sola salida

En el área de las estimaciones de una única salida o una variable dinámica, se pueden identificar cinco ámbitos de predicción según el parámetro: aquellos modelos que estiman posición, velocidad, rapidez o actitud del vehículo; los que predicen el ángulo de balanceo (*ángulo roll*); los que estiman

el ángulo de deslizamiento lateral del vehículo (del inglés, *slip angle*); los que estiman parámetros de los neumáticos (fuerzas en los neumáticos, coeficiente de fricción camino-ruedas y el ángulo de deslizamiento lateral de las llantas, *no del vehículo*, principalmente); y finalmente los que estiman un parámetro del camino.

Para la presente tesis sólo importan el segundo y tercer caso, por estar relacionados con el riesgo de vuelco y de derrape lateral, respectivamente.

Modelos que estiman el Ángulo de Balanceo

En lo concerniente a la estimación del *ángulo de balanceo* (o *ángulo roll*), se comenta que este parámetro es un *indicador clave del riesgo de vuelco de un vehículo*, por ello que aparezca en muchas investigaciones que traten de mitigar este problema.

En [121] se evalúa una Red Neuronal de Propagación hacia Adelante (por sus siglas en inglés, *FFNN*) aplicada en un vehículo tipo Van a través de diversas maniobras. Los investigadores toman como entradas las aceleraciones lateral y longitudinal, la tasa de cambio del ángulo de balanceo (en inglés, *Roll Rate*) y la tasa de cambio del ángulo de viraje (en inglés *Yaw Rate*). A la salida de la red neuronal, los autores agregan un Filtro Kalman (*KF*, por sus siglas en inglés) para mejorar la exactitud de los resultados.

Por otro lado, investigadores en [120] estiman el mismo parámetro mediante una ANN; ello a partir de las mismas entradas que las usadas en [121]. Los autores también le agregan un KF con una función de densidad de probabilidad para fines de restricción de los valores hallados.

Desde otro punto de vista, en [38] se expone un trabajo basado en una FFNN con un enfoque hacia el Internet de las Cosas (*IoT*, por sus siglas en inglés) y que se puede emplear en un sistema embebido. La red toma como entradas las mismas que en [121].

Los científicos en [12] proponen un modelo híbrido basado en una ANN y un Observador H_∞ . La ANN tiene una capa oculta de quince nodos; usa como entradas: aceleración lateral, aceleración longitudinal, tasa de cambio del ángulo de viraje y la tasa de cambio del ángulo de balanceo; y arroja como salida un pseudoángulo de balanceo. Posteriormente, el observador H_∞ retoma el pseudoángulo de balanceo y su tasa de cambio, además de que considera la incertidumbre y mitiga las perturbaciones, para dar lugar al ángulo de balanceo final.

En [11] se exhibe un trabajo basado en una Red de Memoria de Largo-Corto Plazo (o *Red LSTM*) embebida en un sistema de control predictivo. La red LSTM emplea como entradas la aceleración lateral y el ángulo del volante. El análisis se realiza en simulación y los resultados se comparan con un modelo matemático.

De otra forma, en [110] se propone un esquema híbrido basado en una red LSTM y un modelo matemático de dinámica rotacional. La red LSTM tiene dos capas con noventa y dos nodos cada una y una capa densa; y considera un historial de diez pasos de tiempo, pero un horizonte de una muestra.

Ambos modelos son acoplados mediante un Filtro Kalman sin Aroma (por sus siglas en inglés, *UKF*). Las entradas que maneja el algoritmo ML son: aceleración lateral, velocidad longitudinal y ángulo del volante. Esta propuesta tiene la ventaja de calcular un nivel de confiabilidad hacia si se debe tomar la predicción del modelo matemático o del algoritmo ML. Dicho nivel de confianza mejora la matriz de covarianza del UKF mejorando los resultados.

Finalmente, en [109] se mejora lo descrito en [110] proponiendo un algoritmo ML con una arquitectura más pesada, cuatro capas LSTM con ciento veintinueve nodos cada una, además de que cambian las variables de entrada de la red a: velocidad, ángulo del volante y tasa de cambio del ángulo de viraje. De igual manera que en [110], los investigadores emplean un UKF para acoplar ambos modelos, y el cálculo del nivel de confiabilidad lo hacen con base en una estructura de cuadrícula dimensional. Los autores se centran en la mejora de las estimaciones, y a la vez, en la confiabilidad de las predicciones.

Modelos que estiman el Ángulo de Deslizamiento Lateral del vehículo

Ahora, en lo que tiene que ver con el estimador del estado dinámico *Ángulo de Deslizamiento Lateral del vehículo (Slip Angle)*, se menciona que la predicción de este parámetro es importante, pues es un *representante inherente de la estabilidad lateral del vehículo*.

En [83] se expone una metodología para estimar el ángulo de deslizamiento lateral del vehículo a través de lo que los autores proponen como una FFNN, pero que más parece una red recurrente por el lazo de retroalimentación basado en una señal de referencia durante el entrenamiento. Ellos usan como entradas la aceleración lateral, la tasa de cambio del ángulo de viraje, la velocidad y el ángulo del volante. Los autores emplean las estimaciones en escenarios de cambios de maniobra y obtienen resultados no tan buenos, algunos críticos piensan que cambiando el algoritmo de aprendizaje pudieran hallarse mejores estimaciones.

Desde otra perspectiva, en [13] se emplea un modelo ML diferente de las ANNs, un Sistema Adaptativo Neuro-difuso (*ANFIS*, por sus siglas en inglés), para estimar el parámetro predeterminado a partir de combinaciones de entradas, entre ellas: aceleración lateral, tasa de cambio del ángulo de viraje, ángulo del volante y velocidad longitudinal del vehículo. Los investigadores hallan que su propuesta mejora las técnicas basadas en modelo y las ANNs.

Luego, en [128] se estima la misma variable de salida a través de una Red Neuronal de Regresión General, sólo que a partir de la tasa de cambio del ángulo de viraje y la aceleración lateral del vehículo. Los investigadores entrenan su arquitectura mediante simulación y la prueban con datos reales de conducción.

Además, en [45] se formula una propuesta basada en Redes de Unidades Recurrentes Cerradas (o *Redes GRU*), que es un tipo de red recurrente, y se usa un conjunto de datos de diversas condiciones de manejo. Las entradas que se manejan son las velocidades angulares de las cuatro ruedas, el ángulo

del volante, la rapidez y aceleraciones (lateral y longitudinal) del vehículo y la tasa de cambio del ángulo de viraje. Se llegan a buenos resultados. La contribución más importante de este trabajo es la comparación de la Red GRU con la misma pero agregándole resultados de un modelo matemático. La segunda opción tuvo mejores resultados.

Por otro lado, en [24] se expone un trabajo *llamativo* porque se emplea una ANN de arquitectura simple de cinco entradas, un nodo oculto y un nodo salida. Las entradas están conformadas por la rapidez del vehículo, ángulo del volante, aceleraciones (lateral y longitudinal) y la tasa de cambio del ángulo de viraje. Se prueba la técnica ML mediante simulación con dos escenarios: el primero basado el cambio de doble carril y el otro basada en impulsos en la dirección.

Después, en [89] se exhibe un trabajo basado en la propuesta de [83], pero agregándole un UKF para corregir el estado del ángulo de deslizamiento lateral. Se toman como entradas de la ANN a las aceleraciones lineales y velocidades angulares, se coloca una capa densa de noventa neuronas con funciones tangente hiperbólica y como salida se estima un pseudoángulo de deslizamiento lateral. Aquí es donde entra el UKF, para corregir dicho valor según un modelo cinemático. Se impulsan mucho en este trabajo las maniobras como medida del rendimiento de la arquitectura. El modelo se prueba con datos numéricos y experimentales, y se obtienen valores notables en cuanto a los criterios de regresión.

En [73] se propone una arquitectura profunda basada en la unión de una Red LSTM con una Red Neuronal Convolutiva (*CNN*, por sus siglas en inglés) y se usan como entradas una fusión de datos de sensores externos e internos. Con internos se refiere a datos de la Unidad de Medición Inercial (por sus siglas en inglés, *IMU*) (aceleraciones lineales y velocidades angulares); y con sensores externos, LIDAR, RADAR y Cámara. El algoritmo se probó con datos experimentales de un vehículo de prueba. La ventaja de esta propuesta es la superación del error de sincronización y la desventaja es su costo computacional.

De la misma manera, en [64] se emplea una arquitectura profunda basada en una Red LSTM y en un ensamble profundo. La primera tiene la función de estimar el ángulo de deslizamiento lateral a partir de las entradas: tasa de cambio del ángulo de viraje, velocidad, ángulo del volante y aceleración lateral; y el segundo estima la incertidumbre en esa estimación. Dichos valores entran al KF no lineal para mejorar la estimación y para actualizar la matriz de covarianza, respectivamente. El desempeño del modelo se prueba mediante simulación y experimentalmente. Lo que los autores buscan es mejorar la capacidad de generalización de los algoritmos ML, sólo que los escenarios evaluados son limitados.

En [26] se comparan los dos enfoques mayormente tratados en la literatura para la estimación del ángulo de deslizamiento lateral del vehículo: KF y las ANNs. La ANN que se propone está basada en una capa con cinco entradas: rapidez, ángulo del volante, aceleraciones lateral y longitudinal y la tasa de cambio del ángulo de viraje; una capa oculta de diez neuronas; y una capa con una salida. Por otra

parte, se utiliza el Filtro Kalman Extendido (*EKF*, por sus siglas en inglés) para casos no lineales. Se diseñan una variedad de maniobras y se prueban ambas técnicas en un vehículo experimental. Lo que se encuentra es algo sumamente interesante, el EKF hace muy bien las predicciones en estado estacionario, pero no con maniobras transitorias; por otro lado, la ANN tiene buen desempeño en ambas modalidades, la limitante es que su rendimiento depende los datos con que se alimente el algoritmo ML.

En [62] se estima el parámetro respectivo mediante una ANN con diez neuronas en la capa oculta con un algoritmo de retro-propagación Levenberg-Marquardt y a partir de cinco entradas: ángulo del volante, velocidad longitudinal, aceleración lateral, tasa de cambio del ángulo de viraje y coeficiente de fricción. Se realizan pruebas en simulador y con ciertas maniobras, sólo que falta extender el número de ellas.

Posteriormente, en [131] se propone la estimación del ángulo de deslizamiento lateral mediante el empleo de dos KFs (uno basado en un Modelo Dinámico y el otro, en un Modelo Cinemático) y de Lógica Difusa, y a partir de datos recolectados por una IMU y un Sistema Satelital de Navegación Global (*GNSS*, por sus siglas en inglés). Las principales variables involucradas son las aceleraciones lineales y las variables angulares. Lo interesante es el factor correctivo que se implementa a una variable que en trabajos anteriores no se tomaba en cuenta, el *rumbo* (del inglés *Heading*) de un vehículo, y dicho factor es la salida del algoritmo de Lógica Difusa. Ello se hace para poder evaluar escenarios de conducción tranquila, pero también de manejo agresivo.

En [87] se compara el desempeño de UKF con el de UKF más una Red de Aprendizaje Profundo (*DNN*, por sus siglas en inglés). Se propone una arquitectura profunda basada en una capa de entrada con ocho neuronas: las velocidades angulares de las cuatro ruedas, aceleraciones lateral y longitudinal, ángulo del volante y la tasa de cambio del ángulo de viraje; una capa GRU con sesenta y cuatro nodos; una capa densa de dos nodos; y una capa de salida de dos nodos (velocidades lateral y longitudinal). La DNN fue implementada en MATLAB y probada en simulador con un modelo de vehículo estimando el ángulo de deslizamiento lateral. Se encuentran resultados superiores con el enfoque UKF+DNN que con sólo UKF, al mejorar las predicciones del ángulo de deslizamiento lateral para ciertas maniobras.

Finalmente, en [136] se emplea una ANN y un KF para estimar la variable tratada. Primero, se usa la ANN entrenada con el algoritmo Levenberg-Marquardt para predecir la rigidez de los neumáticos en las curvas. Y, una vez determinada esa variable, mediante un KF variable en el tiempo, se calcula el ángulo de deslizamiento lateral. Se prueba la propuesta mediante datos experimentales y se halla buena robustez ante diversidad de maniobras.

Modelos de Múltiples Salidas

En relación con los modelos de múltiples salidas, primero se enunciarán aquellos trabajos que estiman el ángulo de balanceo y el ángulo de deslizamiento lateral de manera simultánea (pueden ser más variables estimadas, pero que obligatoriamente se encuentren esas dos). A continuación se mencionan dichos exponentes.

En [43] se desarrolla una DNN densa con cinco capas ocultas, cada una con decenas de neuronas. En este trabajo se busca estimar el ángulo de deslizamiento lateral del vehículo (no de los neumáticos) y el ángulo de balanceo a partir de las aceleraciones lateral y longitudinal, la tasa de cambio del ángulo de balanceo, la tasa de cambio del ángulo de viraje, el ángulo del volante y la rapidez longitudinal.

Luego, en [14] se propone básicamente un esquema híbrido, donde participan una DNN -la misma arquitectura que en [43]- como estimador de pseudoángulos (ángulo de deslizamiento lateral y ángulo de balanceo) simultáneamente, y posteriormente un observador es el encargado de refinar los resultados y mejorarlos. La contribución de este trabajo es la incursión de un sistema de control por red con retardo en las transmisiones, para evitar la saturación de información en el sistema.

Por último, en [6] se declara un enfoque distinto, ya no se busca la mejor arquitectura, sino la mejor estrategia de aprendizaje del algoritmo ML. Se propone una DNN (cinco capas ocultas) con aprendizaje por currículum, en el que los datos son jerarquizados por complejidad para facilitar su aprendizaje, ello a través de funciones autodidactas de puntuación y estimulación. Formalmente, se emplea una arquitectura profunda basada en el Perceptrón Multicapa (MLP, por sus siglas en inglés) y basada en las entradas de aceleraciones lateral y longitudinal, tasa de cambio del ángulo de balanceo, tasa de cambio del ángulo de viraje, ángulo del volante y velocidad longitudinal. La propuesta se evaluó en dieciséis escenarios de conducción, concluyendo que el aprendizaje por currículum da mejores resultados que los métodos no-currículum.

Lo anterior describe los trabajos que simultáneamente predicen el ángulo de balanceo y el ángulo de deslizamiento lateral (aunque pueden ser más variables). Hasta este momento, de lo que se consultó, fueron los trabajos que se pudieron reconocer. No obstante, también hay otras propuestas que predicen múltiples salidas, pero que no estiman ninguna de las dos salidas de interés. Se procede a mostrar algunos de los trabajos hallados durante la revisión de la literatura.

En [99] se propone una metodología ANFIS que estima la posición, velocidad y actitud del vehículo a partir de datos de la IMU: aceleraciones y velocidades angulares en tres ejes; del Sistema de Posicionamiento Global (*GPS*, por sus siglas en inglés) y de odómetros. Todo ello con fines de transportación inteligente.

Desde otro punto de vista, en [2] se estiman algunos parámetros del modelo de neumáticos de Pacejka denominado como *Fórmula Mágica* a partir de fuerzas laterales y longitudinales de las llantas, del ángulo de deslizamiento lateral de las mismas y de la tasa de cambio de ese ángulo.

Además, en [76] se propone una Red Neuronal Artificial Elman (*ENN*, por sus siglas en inglés) mejorada con un Análisis de Componentes Principales (*PCA*, por sus siglas en inglés) que es capaz de estimar la rapidez, el ángulo de deslizamiento lateral del vehículo y la tasa de cambio del ángulo de viraje a partir de un conjunto de características como aceleraciones, rapidez, velocidad, activación y el tiempo en que lo está el acelerador.

Por su parte, en [52] se construye una DNN Convolutiva, la cual admite entradas de tres cámaras, así como datos de la dinámica vehicular, como el ángulo del volante, la aceleración, la desaceleración y la velocidad. Lo que se busca predecir son instantes futuros del ángulo del volante, aceleración y desaceleración.

En [113] se construye una iniciativa basándose en una metodología híbrida edificada, por un lado, en un observador deslizante robusto sustentado en la dinámica vehicular; y por otro, en una ENN. El primero tiene la función de estimar la velocidad lateral del vehículo en condiciones de manejo complicadas, y la segunda tiene la labor de compensar los errores de deriva del IMU. Las entradas que maneja la ENN son: velocidad longitudinal del vehículo y el error de la velocidad del GPS del paso de tiempo anterior. Finalmente, la velocidad lateral estimada por el observador se combina con la salida de la ENN, error de la velocidad en el tiempo actual, mediante un KF para dar las predicciones últimas de la actitud y de la velocidad lateral del vehículo.

Por último, en [15] se exhibe un trabajo para estimar parámetros de vehículo, específicamente referentes al centro de gravedad y los neumáticos, sustentado en inferencias basadas en simulación. Esta reciente propuesta de algoritmo ML arroja funciones de densidad de probabilidad y permite el manejo de salidas simultáneas. Las entradas del algoritmo ML son las aceleraciones lateral y longitudinal, la tasa de cambio del ángulo de viraje y las velocidades angulares de las ruedas. En suma, se llega a una buena identificación de parámetros del vehículo con una prospectiva hacia un número arbitrario.

1.3.2. Contribuciones en materia de Sistemas de Asistencia a la Conducción

De la revisión de la literatura, específicamente acerca de la predicción de estados de la dinámica del vehículo, el algoritmo mayormente empleado es ANN (con sus variedades), no hay mucho de otras técnicas ML que se haya encontrado al momento de la escritura del documento. Se confirma lo defendido en [60] de que las ANNs son el algoritmo ML por excelencia. Esto es, no queda claro el posible desempeño de otras herramientas para predicción de estados, para la tarea específica de la predicción de estados del vehículo, los cuales son altamente cambiantes y dependientes de información no medida (e.g. condiciones de tráfico, condiciones del camino, etc.).

Otro hecho que motiva la elaboración de esta tesis es que los modelos basados en Inteligencia Artificial de múltiples entradas con múltiples salidas no son tan comunes en la literatura y menos los

que predicen simultáneamente dos variables muy importantes para la estabilidad lateral y rotacional del vehículo, a saber: el ángulo de deslizamiento lateral y el ángulo de balanceo.

Por último, pero no menos importante de aclarar, es que las entradas de los modelos basados en Inteligencia Artificial que se hallaron en el estado del arte para poder estimar el ángulo de balanceo, el ángulo de deslizamiento lateral o ambos, a lo más incluyen como acciones del conductor al ángulo del volante. Evaluar otras acciones del operador es una gran contribución de este trabajo.

Estas contribuciones a áreas de oportunidad tienen dos perspectivas muy claras: la posibilidad de predicción de la estabilidad lateral y del riesgo de vuelco de un vehículo.

Finalmente, en la Tabla 1.3 se muestra la comparación de los trabajos encontrados en el estado del arte con respecto a la predicción de estados dinámicos del vehículo-conductor para la perspectiva de ADASs. Además, en la misma figura se muestra la propuesta de la presente tesis para este escenario.

1.3.3. Estrategias de Manejo de Energía

Los HEVs son vehículos que poseen un motor de combustión interna y al menos, un motor eléctrico. Este tipo de vehículos tienen el objetivo de minimizar las emisiones contaminantes hacia el ambiente y reducir su consumo de combustible, al mismo tiempo que buscan optimizar la distribución de potencia entre los motores en escenarios complejos y durante largos horizontes de tiempo [134].

Un caso particular de los HEVs son los vehículos Eléctricos Híbridos de Conexión (*PHEVs*, por sus siglas en inglés), que son HEVs que se pueden conectar a la red eléctrica [102]. En HEVs y PHEVs las EMSs son los métodos de la distribución de potencia entre los diferentes motores y entre las diferentes fuentes de energía, con el fin de maximizar el rendimiento de combustible y mitigar las emisiones contaminantes [92].

El problema de distribución de la potencia, entre las diferentes fuentes, también está presente en vehículos eléctricos, por ejemplo en los que cuentan con baterías y supercapacitores o incluso que tienen celdas de combustible. Es para resolver este problema, que se utilizan las EMSs.

En la literatura se encuentra una diversidad de clasificaciones de las EMSs, pero la que aquí se utilizará es la propuesta en [134] (Figura 1.1). La primera gran categoría son las EMSs fuera de línea (offline), las cuáles se refieren cuando el equipo de procesamiento no está conectado o no está listo para usarse [129].

Dentro de esta gran categoría se encuentran dos tipos de EMSs: las EMSs de Optimización Global y las EMSs basadas en Reglas. La primera alude a técnicas que buscan hallar un óptimo global en la distribución de potencia del vehículo bajo ciertos ciclos de manejo, además de que modifican las EMS en línea, que se describirán a continuación. Una limitante de aquel tipo de EMSs es que son computacionalmente complejas, puesto que buscan un óptimo global, no tienen una implementación de control en tiempo real y requieren información a priori. Dentro de esta clase se encuentran la *Programación Dinámica* y la *Optimización Convexa*, entre otros algoritmos ¹.

Por otro lado, las EMSs basada en Reglas se basan en la predeterminación de una serie de reglas de control definidas durante las pruebas de pre-producción para determinar la mejor división entre los sistemas de propulsión; no obstante, no logran un óptimo global como en la clase previa. A esta clase pertenecen las técnicas de *EMSs basadas en Reglas Determinísticas* y las *EMSs basadas en Lógica Difusa*.

La segunda gran categoría son las EMSs en línea (online), las cuáles aluden a cuando el equipo de procesamiento está conectado o está listo para usarse [129]. Esta categoría de EMSs está basada en la optimización local y tiene la ventaja de la implementación en tiempo real. Dentro de esta otra gran categoría se hallan las clases de: EMSs basadas en Optimización Instantánea, EMSs Predictivas y EMSs basadas en Aprendizaje.

¹No se entrará más a detalle en las técnicas que no están basadas en Inteligencia Artificial, pues no son el objetivo central de esta tesis, sólo se busca brindar un panorama general del estado del arte acerca de EMSs.

Las EMSs basadas en Optimización Instantánea pueden obtener un óptimo local de la minimización del consumo de combustible sin tener necesidad de acceder a todo el ciclo de manejo, sólo localmente. Dentro de esta clase se hallan los algoritmos: *Estrategia de Minimización de Consumo Equivalente*, *Estrategia de Minimización de Consumo Equivalente Adaptativa* y el *Control Robusto*.

En lo concerniente a las EMSs Predictivas, el principal objetivo es optimizar la distribución de potencia en el vehículo según la predicción de su velocidad o aceleración en cierto horizonte de tiempo. Dentro de este estrato se incluyen: *Predicción de los Ciclos de Manejo o Conducción*, *Control Predictivo por Modelo*, *Control Predictivo por Modelo Estocástico* y *Control Predictivo por Modelo basado en Aprendizaje*.

Finalmente, en lo que toca a las EMSs basadas en Aprendizaje, éstas tienen la capacidad de actualizar los parámetros de control a partir del entrenamiento de datos, lo que provee adaptabilidad en las condiciones de manejo [134].

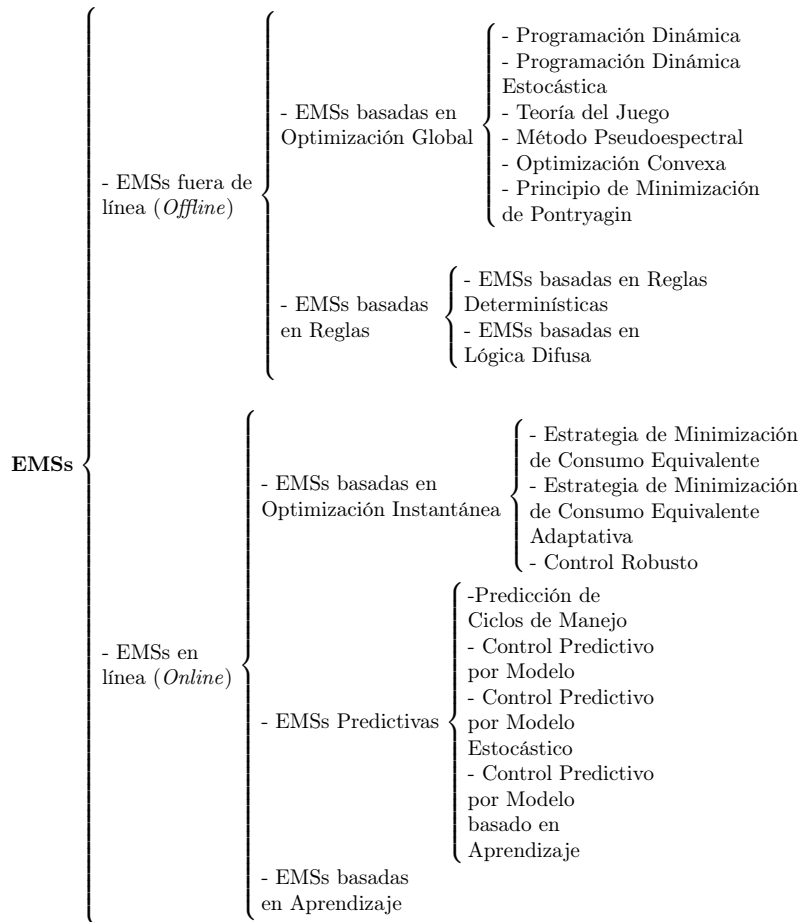


Figura 1.1: Clasificación de las Estrategias de Manejo de Energía [134].

Cabe aclarar que en el ámbito de las EMSs Predictivas enunciadas en [134], los autores se limitan a los ciclos de conducción, que se definen como la forma en que los operadores conducen en cierta región [96]. Ese problema se trata más de una cuestión de clasificación. En [137] se complementa esta información al mencionar que los EMSs Predictivos tienen dos enfoques: *los basados en el reconocimiento y los basados en la experiencia*.

Los primeros son a los que se refieren en [134] a través de los ciclos de manejo o conducción, que básicamente tratan de clasificar la forma que conduce un operador con la región más compatible que corresponda.

Las predicciones de la velocidad del vehículo pueden ser usadas tanto en las EMSs óptimas, como en las basadas en aprendizaje, ya que el perfil de esta variable es la que tiene una repercusión importante en el ahorro de combustible. Esta tesis se centrará en el Control Predictivo por Modelo.

El Control Predictivo por Modelo requiere de la predicción del perfil de velocidad / aceleración de un vehículo en largos o cortos horizontes de tiempo, ahí es donde entran las técnicas de Inteligencia Artificial para poder modelar el fenómeno [75]. Según [134], entre las características que se extraen de los datos con este enfoque están la aceleración, la velocidad y la desaceleración. En [71] se declara que la predicción precisa y exacta de la velocidad es un método clave para una conducción *segura, limpia y eficiente*. También se menciona que el conductor juega un rol clave en las predicciones de velocidad o rapidez.

Técnicas basadas en Inteligencia Artificial y Modelos Probabilísticos

De igual manera que para el caso de ADASs, las técnicas data driven son las que no necesitan de un modelo matemático a priori, sino que hallan relaciones a partir de los mismos datos. Para el enfoque de EMSs, en la literatura se encuentran tres grandes enfoques con los que se ha trabajado en la predicción del perfil de velocidad de un vehículo dentro del Control Predictivo por Modelo: las ANNs/DNNs, las Cadenas de Markov (*MC*, por sus siglas en inglés) y los Modelos Compuestos (combinación de varios) [32]. Esto se puede ver de manera gráfica a través de la Figura 1.2.

En cuanto a la primer categoría (ANNs / DNNs), éstas tienen la ventaja del mapeo no lineal entre predictores y variables de respuesta (en este caso, la velocidad o aceleración), no obstante, padecen de la dependencia de la operación del algoritmo hacia los datos de entrenamiento [75].

En cuanto a la segunda categoría, las Cadenas de Markov (*MCs*, por sus siglas en inglés) se usan para evaluar una sucesión de eventos a partir de la probabilidad y del estado del historial de datos que se tengan [75]. Desafortunadamente, su desempeño decae cuando los nuevos datos que se le presentan al MC son distintos a los del historial [75].

Por último, la tercer categoría, Modelos Compuestos, alude a la combinación de varios tipos de técnicas. Ya que se aclaró esta clasificación, es muy conveniente mostrar los trabajos que se encontraron en el estado del arte en lo concerniente a las técnicas de predicción del perfil de velocidad

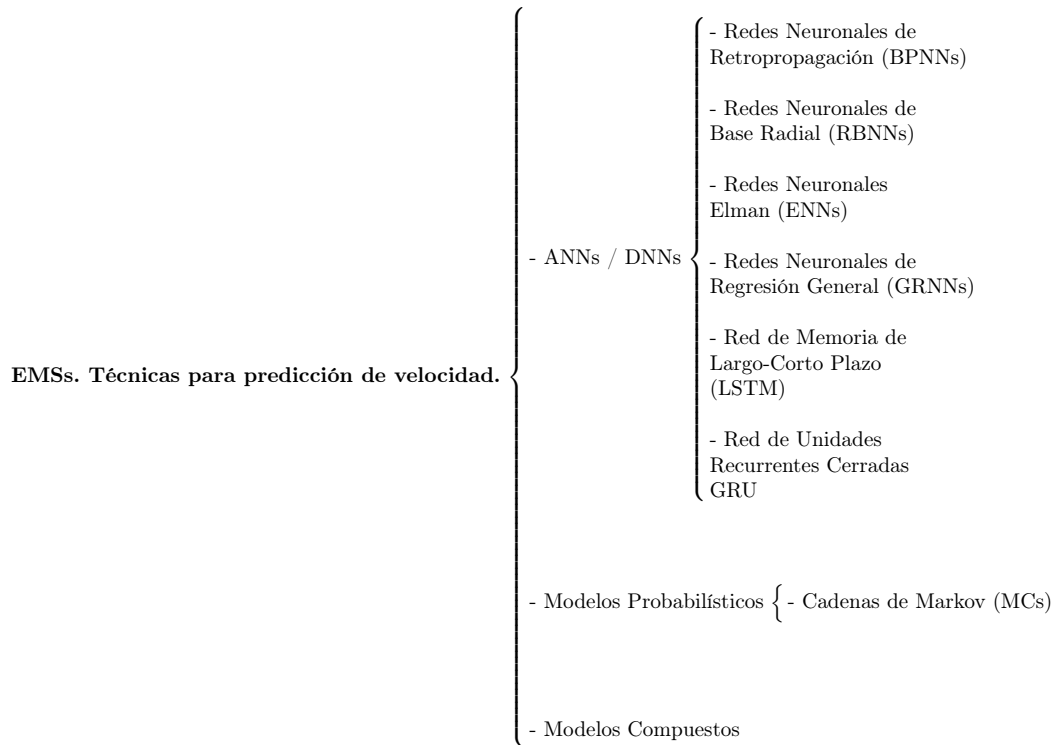


Figura 1.2: EMSs. Clasificación de técnicas para la predicción de perfiles de velocidad en vehículos.

(o aceleración) de un vehículo dentro del contexto del Control Predictivo por Modelo y enfocadas a las EMS. En la Tabla 1.4 se presentan los trabajos hallados con base en la clasificación propuesta en la Figura 1.2, así como la identificación de parámetros elementales de dichas propuestas que permitirá la presentación de la contribución de esta tesis para la perspectiva de EMSs.

1.3.4. Contribuciones en materia de Estrategias de Manejo de Energía

De la revisión bibliográfica, cuyo resumen se muestra en el Tabla 1.4, es posible observar que, si bien ya se ha trabajado en la predicción de la velocidad del vehículo *en el contexto del Control Predictivo por Modelo*, pues es una variable con relación intrínseca al ahorro de combustible y a la energía cinética, el relacionar dicho parámetro de salida del vehículo con las acciones ejecutadas por el conductor (ello a través de un Modelo Vehículo-Conductor) es un campo con muchas oportunidades de investigación. En lo que se revisó del estado del arte, la acción del conductor que a lo más se incluye en los trabajos es el ángulo del volante, faltan explorar otras acciones del operador. Además, existen técnicas ML, que solas o en conjunción con otros algoritmos, no han sido probadas en estas aplicaciones, por lo que no queda claro el potencial de las mismas para sus aplicaciones en ADAS o en EMSs. Esto constituye una oportunidad de contribución. Vale la pena recordar que este rubro de la tesis sólo se limitará a la predicción de la velocidad del vehículo, no a estrategias de control.

Tabla 1.4: Estrategias de Manejo de Energía. Comparación de los trabajos encontrados en el estado del arte con la propuesta de tesis.

Técnica General	Modelo Particular	Trabajos	Entradas del Modelo	Horizonte de Predicción [Pasos de Tiempo / s]
ANN / DNN		[72]	Datos geográficos (latitud, longitud y altitud), inclinación del camino y datos históricos de la velocidad.	No especificado.
		[78]	1. Velocidad. 2. Velocidad, condiciones del camino y del tráfico.	1, 60 y 120
		[61] [127]	Datos históricos velocidad.	5, 10 y 15 s. 1
		[49], [22]	Datos históricos velocidad.	10 s.
		[51]	Velocidad / Aceleración	10
		[126]	Datos históricos velocidad.	5, 10 y 15 s.
		[50] [50]	Datos históricos velocidad.	5, 10 y 15 s. 5, 10 y 15 s.
		[77]	Datos históricos velocidad.	1-10 s.
		[37]	Datos históricos velocidad.	5, 10, 15 y 20 s.
	Modelos Probabilísticos	Cadenas de Markov (MCs)	[138] [32]	Aceleración / Velocidad
[108]			1. Velocidad. 2. Velocidad / Aceleración	1, 5, 10 y 20
[70]			Aceleración / Velocidad / Comportamiento Conductor	10-500 s.
[139]			Aceleración / Velocidad	3, 5 y 10 s.
Modelos Compuestos	MC + C-medias Difuso	[75]	Aceleración, velocidad y rapidez del tráfico,	1
	ENN + Aprendizaje Q Doble	[23]	1. Rapidez. 2. Parámetros relacionados a velocidad y aceleración	3,4,...,15
Propuesta de Tesis	1. GRU. 2. LSTM. 3. Transformer.	Propio	1. Ángulo del volante, 2. Par del volante, 3. Posición del Acelerador. 4. Activación del Freno. 5. Rapidez.	49 s.

1.4. Objetivos

Dados los argumentos anteriormente expuestos, se enuncian los objetivos de este trabajo, tanto el general como los particulares.

1.4.1. Objetivo General

Desarrollar y evaluar modelos vehículo-conductor a partir de una base de datos reales y de técnicas de Inteligencia Artificial, con la finalidad de predecir el comportamiento del vehículo a corto y largo plazo para su potencial uso en Sistemas Avanzados de Asistencia al Conductor y en Estrategias de Manejo de Energía para vehículos.

1.4.2. Objetivos Particulares

Entre los objetivos particulares de esta tesis se enuncian los subsecuentes:

- Establecer la idoneidad de las técnicas de Aprendizaje Máquina (ML) con potencial a ser aplicadas a Sistemas de Asistencia a la Conducción (ADASs) y a Estrategias de Manejo de Energía (EMSs), en términos de la predicción en tiempo de las variables dinámicas del vehículo.
- Seleccionar, apoyándose del fundamento teórico, las variables que fungirán como entradas y salidas de los Modelos vehículo-Conductor, es decir, que relacionarán el comportamiento del conductor con el del vehículo, para ambos escenarios.
- Implementar los algoritmos ML a evaluar, de acuerdo a la potencial contribución de su uso en vehículos.
- Depurar la información de la base de datos para descartar errores de medición, así como clasificarla de acuerdo al conductor.
- Evaluar el rendimiento de las diferentes técnicas ML aplicadas a los conjuntos de datos pertenecientes a cada sujeto y de acuerdo a los Modelos vehículo-Conductor del escenario correspondiente.
- Analizar los resultados, y de acuerdo a ellos, proponer las conclusiones y perspectivas de la tesis.

1.5. Estructura del Documento

La Figura 1.3 muestra de manera gráfica la metodología general para alcanzar los objetivos de la presente tesis.

Conforme con lo expuesto en la Figura 1.3, es importante destacar que dicho proceso se aplica a cada escenario y por cada sujeto. En términos generales, la metodología propuesta en esta tesis consta de 7 pasos.

El paso 1 consiste en la revisión del estado del arte en lo concerniente a la aplicación de técnicas ML para la predicción de variables del vehículo orientado a dos potenciales aplicaciones: Sistemas Avanzados de Asistencia a la Conducción y de Manejo de Energía. El siguiente paso (paso 2) es en el que se realiza la preparación de los datos en el entorno MATLAB, para que puedan ser útiles para las técnicas ML. Aquí se incluye la lectura, compatibilización y el pre-procesamiento de la información.

El paso 3 es la selección de las variables que aportan más información para la descripción de los fenómenos, ello con base en la comprensión de la física de movimiento. La etapa subsecuente (paso 4) se refiere a la construcción de los modelos vehículo-conductor a partir de la revisión de la teoría y del discernimiento entre las variables que actuarán como entradas y salidas del modelo. Después, una vez identificados los modelos vehículo-conductor, se procede a la programación de dichos modelos (paso 5) usando las distintas técnicas de ML revisadas en la literatura a través del lenguaje Python.

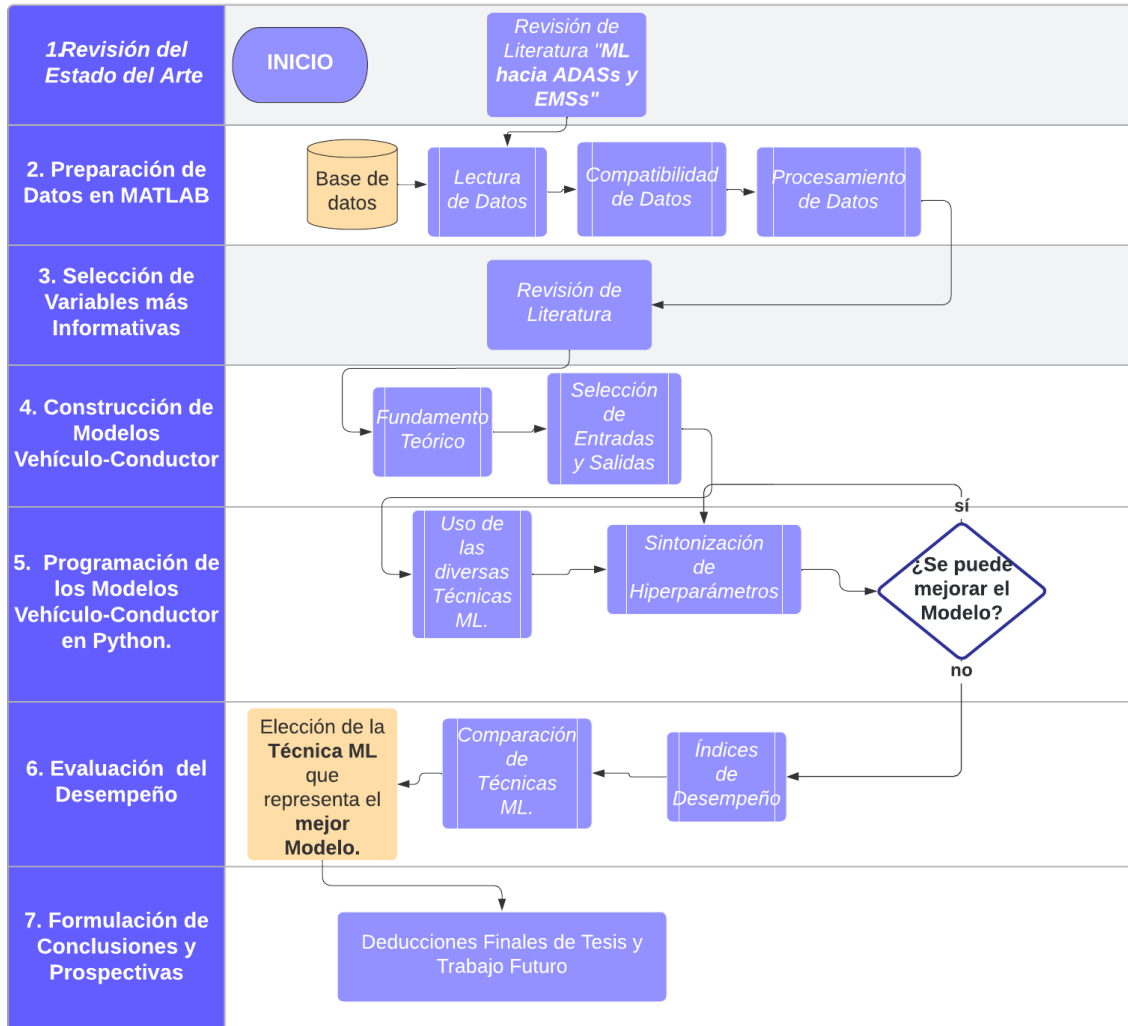


Figura 1.3: Estructura del Documento.

Este paso, el paso 5, se repite hasta que se tengan los mejores resultados de cada algoritmo de ML, apoyándose de la sintonización de hiperparámetros². Ahora, considerando cada sujeto y cada escenario, se procede a identificar la técnica de ML que mejor representa el modelo para dicho conductor y enfoque (paso 6); ello a través de la evaluación de los índices de desempeño y de otros criterios. Finalmente, en la última fase (paso 7) se incluyen la discusión de resultados y la formulación de conclusiones y de prospectivas.

1.6. Sumario

En este capítulo se expuso la motivación de esta tesis y la descripción de trabajos previos relacionados con la derivación de modelos vehículo-conductor. En ellos, fue posible observar una opor-

²a veces a este proceso se le suele llamar tuneo y constituye un anglicismo no formal que será evitado en esta tesis.

tunidad de aportación en la descripción basada en datos reales de la relación entre el vehículo y el conductor usando técnicas de inteligencia artificial. Ello para ambas perspectivas de trabajo, tanto Sistemas Avanzados de Asistencia a la Conducción como Estrategias de Manejo de Energía.

Se establecieron los objetivos de esta tesis y la estructura del documento. En capítulos subsecuentes se expresará con mayor detalle la metodología que se usará en el presente trabajo y se explicarán los antecedentes teóricos necesarios para su comprensión.

Capítulo 2

Marco Teórico

En este capítulo se define el fundamento científico e ingenieril que permite la comprensión, desarrollo y evolución de la tesis. Primeramente se introduce el marco de referencia para el movimiento del vehículo, luego se establece las herramientas necesarias para la descripción de las técnicas de inteligencia artificial usadas.

2.1. Sistema de coordenadas del vehículo

En el análisis de la dinámica rotacional de un vehículo resulta muy importante ubicarse en el espacio, y para ello es necesario un sistema de coordenadas. Gillespie (1992) propone en [41] un sistema de referencia fijo ortogonal derecho que tiene su origen en el centro de gravedad del vehículo, y es el que adoptamos en esta tesis. Las coordenadas son (Fig. 2.1):

- x . Hacia adelante y sobre el plano longitudinal de simetría.
- y . Lateral y crece hacia fuera de lado derecho del vehículo.
- z . Sobre la vertical y crece hacia abajo con respecto al vehículo.
- *Ángulo de balanceo (o ángulo roll)*. Rotación con respecto al eje x .
- *Ángulo de cabeceo (o ángulo pitch)*. Rotación con respecto al eje y .
- *Ángulo de viraje (o ángulo yaw)*. Rotación con respecto al eje z .

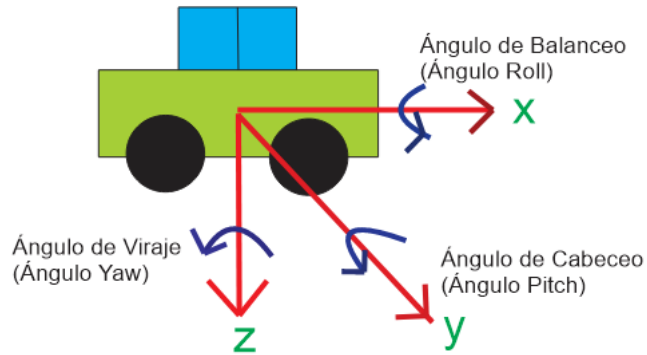


Figura 2.1: Sistema de referencia a bordo del vehículo [41].

El movimiento del vehículo en el espacio está definido por las velocidades y sus respectivas direcciones considerando el sistema de referencia a bordo del vehículo. Sin embargo, dicho sistema de referencia del vehículo se cuantifica con respecto a un sistema de referencia fijo sobre Tierra.

La actitud -*orientación en tres dimensiones de un vehículo con respecto a un sistema de referencia*- y la trayectoria de un vehículo a lo largo de una maniobra se describen mejor apoyándose del sistema de referencia derecho fijo centrado en la Tierra. Formalmente lo que se hace es hacer coincidir el sistema de referencia sobre la Tierra con el sistema de referencia vehicular en el punto que empieza la maniobra. Las coordenadas que permiten la descripción del movimiento (Fig. 2.2) se enuncian como sigue:

- X. Movimiento hacia adelante.
- Y. Movimiento hacia la derecha.
- Z. Movimiento vertical (sentido hacia abajo como positivo).
- Rumbo (en inglés *Heading Angle*). Ángulo entre X y x en el plano terrestre.
- Ruta/Curso (en inglés *Course Angle*). Ángulo entre el vector velocidad del vehículo y el eje X .
- Ángulo de deslizamiento lateral del vehículo (en inglés *Sideslip/Slip Angle*). Ángulo entre el vector velocidad del vehículo y el eje x .

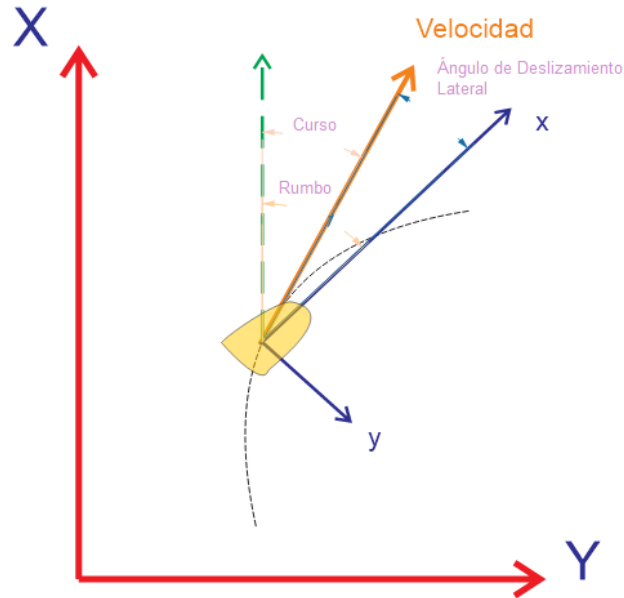


Figura 2.2: Sistema de referencia fijo en Tierra ([41], [4]).

2.2. Aprendizaje Automático (o Machine Learning)

El Aprendizaje Automático (por sus siglas en inglés, *ML*) es una rama científica de la Inteligencia Artificial que permite que las computadoras adquieran conocimiento y mejoren la forma en que realizan sus tareas de manera progresiva. En ML lo que se busca es mapear entre unas variables predictoras y unas variables de respuesta, donde se aprende y se infiere sobre el contexto [101]. Las Técnicas ML se pueden dividir como se muestra en la Figura 2.3; no obstante, para esta tesis sólo interesan los *métodos supervisados*, en los cuales el proceso de generación de conocimiento se realiza con ejemplos o etiquetas (también denominadas *true targets* o *ground truth*) [46].

La Figura 2.4 muestra la forma en que trabajan los métodos supervisados. Este tipo de métodos requieren datos etiquetados, y lo que buscan es hallar una función que permita mapear unas variables de entrada (predictores) en las variables de salida o etiquetas deseadas. En este tipo de métodos, el algoritmo ML se entrena con un conjunto de datos y, de esta manera, puede inferir ante nuevos datos que se le presenten. La razón por la que se les nombra *supervisados* a este tipo de métodos es porque se les debe proveer de las entradas y las salidas deseadas para que el algoritmo pueda aprender [97].

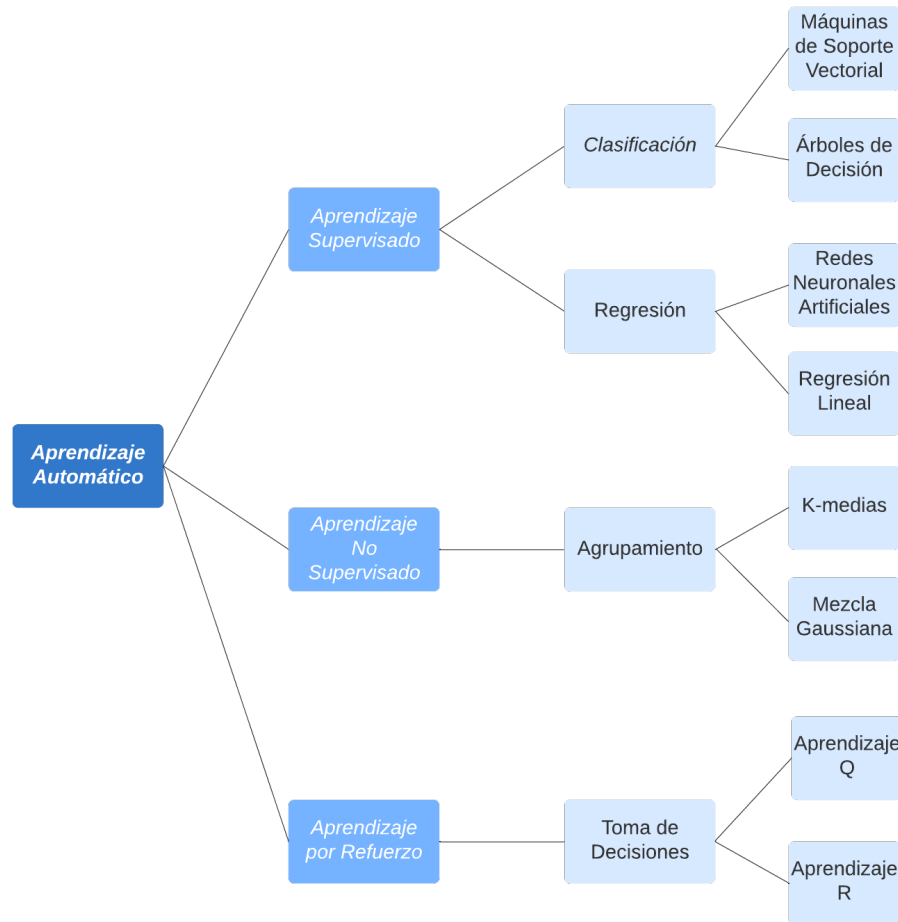


Figura 2.3: Clasificación de las técnicas ML [46].

2.3. Elementos Básicos Constituyentes de un Algoritmo ML Supervisado

Un algoritmo ML supervisado de manera genérica contiene los componentes que se muestran en la Figura 2.5. Se tiene una técnica ML (o modelo ML), que es el encargado de mapear las entradas en salidas. Luego, esa salida del modelo ML es comparada con el valor real a través de una función de pérdida o de costo. Posterior a ello, y con base en ese valor de pérdida, se ajustan los hiperparámetros y parámetros del modelo ML para reducir dicha valor de costo. Lo descrito previamente es el funcionamiento elemental de un algoritmo ML supervisado.

2.3.1. Modelo ML

El algoritmo ML o modelo ML se refiere a la técnica empleada para hallar la función que mapea las entradas (o predictores) en las salidas (o etiquetas) [97]. Entre los diferentes tipo de modelos ML supervisados se encuentran las Máquinas de Soporte Vectorial (*SVM*, por sus siglas en inglés), las

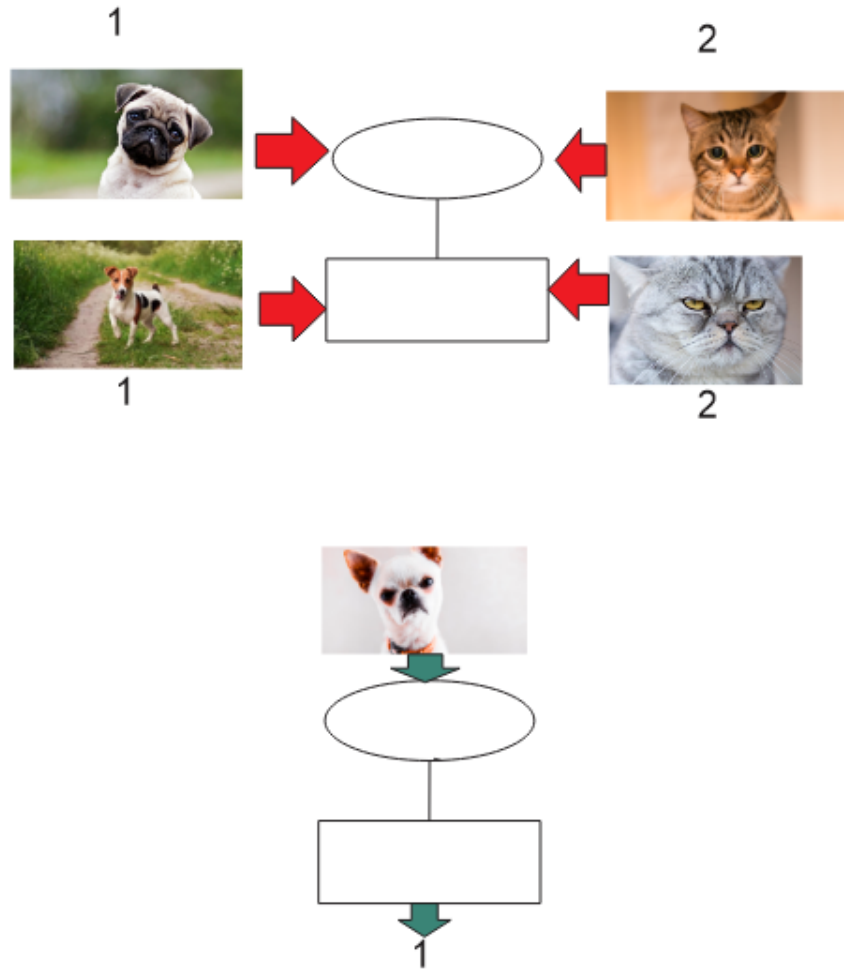


Figura 2.4: Métodos ML Supervisados.

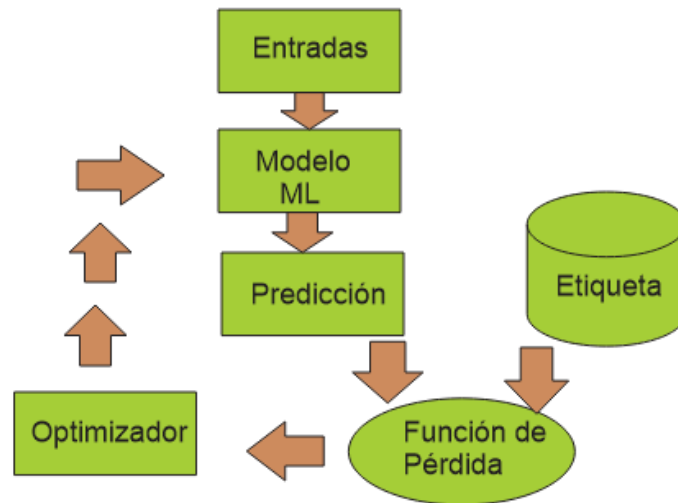


Figura 2.5: Elementos Constituyentes de un Método ML Supervisado.

ANNs, entre otras técnicas.

2.3.2. Función costo o de pérdida

Esta función compara el resultado de la técnica ML con los resultados reales y determina qué tan malo o bueno es el modelo ML en la estimación del conjunto de datos [68]. Algunas de las funciones de pérdida más usadas en la literatura se muestran en la Figura 2.6.

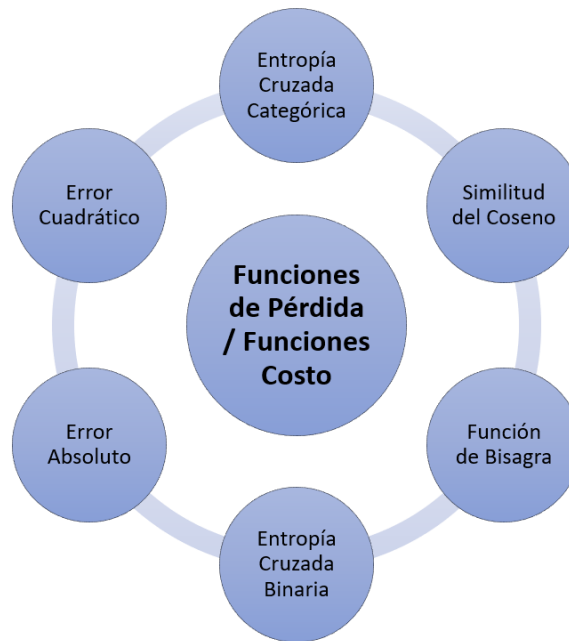


Figura 2.6: Funciones Costo.

Para tareas de regresión o predicción las funciones costo encontradas en la literatura son el error cuadrático, el error absoluto y la similitud de coseno; y para las tareas de clasificación, la entropía cruzada binaria (clasificación binaria), la entropía cruzada categórica (clasificación multi-clase) y la función de bisagra [111].

2.3.3. Optimizador

El optimizador es un agente que permite la modificación de los *parámetros del modelo* (biases y pesos) de la técnica ML, de manera que se minimice la función costo o de pérdida. Para la presente tesis sólo interesa *Adam* porque, según lo revisado en la literatura, es un optimizador robusto y adaptativo, características fundamentales para el desarrollo de los algoritmos ML de la presente tesis. Además, otra razón es porque este optimizador es capaz de trabajar con gradientes dispersos y con ruido [17]. Adam es un optimizador en la que la tasa de aprendizaje puede variar según los datos. El algoritmo Adam utiliza el momento de primer orden del gradiente y el momento de segundo orden

para diseñar tasas de aprendizaje adaptativo independientes para diferentes parámetros, lo que mejora el rendimiento en gradientes dispersos y la tasa de aprendizaje se reserva adaptativamente para cada parámetro. Específicamente, el algoritmo calcula el promedio móvil exponencial del gradiente, y los hiperparámetros β_1 y β_2 controlan la tasa de disminución de estos promedios móviles. El valor inicial de la media móvil y los valores de β_1 y β_2 están cerca de 1 (valor recomendado), por lo que la desviación de la estimación de momento está cerca de 0. La desviación se mejora calculando primero la estimación con desviación y luego calculando la estimación después de la corrección de desviación [95].

Ya una vez definidos los componentes básicos de los métodos ML supervisados, conviene enfocarse en una técnica ML que servirá de referencia para entender modelos que se presentarán más adelante.

2.4. Redes Neuronales Artificiales

Las Redes Neuronales Artificiales (ANNs) son un modelo o técnica ML que tiene como unidad fundamental a la neurona, agente encargado de mapear las entradas en las salidas. Un conjunto de neuronas se denomina *capa*, y las ANNs poseen 3 capas [80].

La *capa de entrada* es el conjunto de neuronas que reciben directamente la información proveniente de las fuentes externas a la red.

Por otra parte, la *capa oculta* es interna a la ANN, puede estar interconectada de distinta manera, lo que genera las diferentes topologías existentes.

Finalmente, la *capa de salida* transfiere la información de la red hacia el exterior.

En la Figura 2.7 se muestra una ANN con tres neuronas en la capa de entrada, cinco en la capa oculta y una en la capa de salida. Si la ANN tiene más de una capa oculta, se le denomina Red Neuronal Profunda (*DNN*, por sus siglas en inglés), aunque esta percepción puede variar entre los autores [80].

2.4.1. Funciones de Activación

Las neuronas o capas en las ANNs, para que mapeen entradas en salidas, necesitan de un agente que se denomina en términos de ML *función de activación*.

Una función de activación decide si la neurona se activa o no, es decir, decide si la entrada que se le aplica es importante para la predicción [114]. Algunas funciones de activación tienen el objetivo de agregar una no linealidad al proceso de aprendizaje o inferencia, como la función sigmoide (σ), la función tangente hiperbólica (\tanh) y la función ReLU. Otro tipo de función de activación, como la lineal, se usa para casos de regresión y con el motivo de que la neurona o capa genere un valor único de salida [36].

La Figura 2.8 muestra las funciones de activación mencionadas.

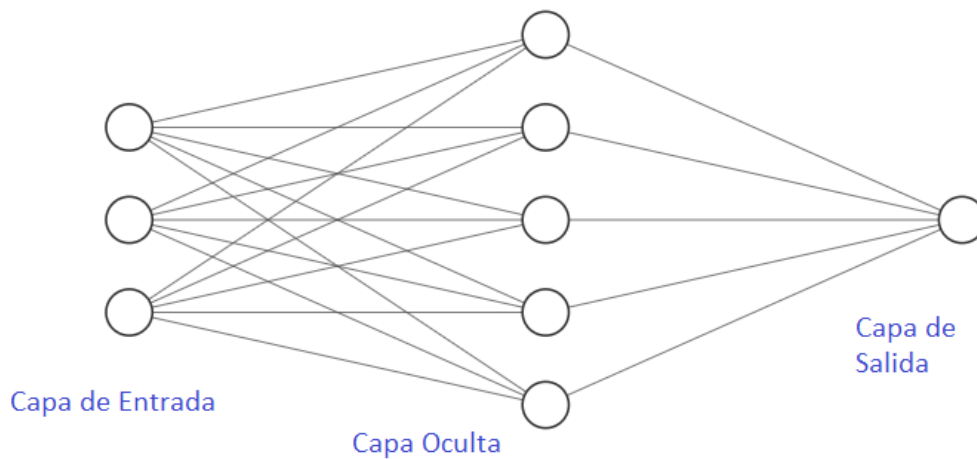


Figura 2.7: Red Neuronal Artificial.

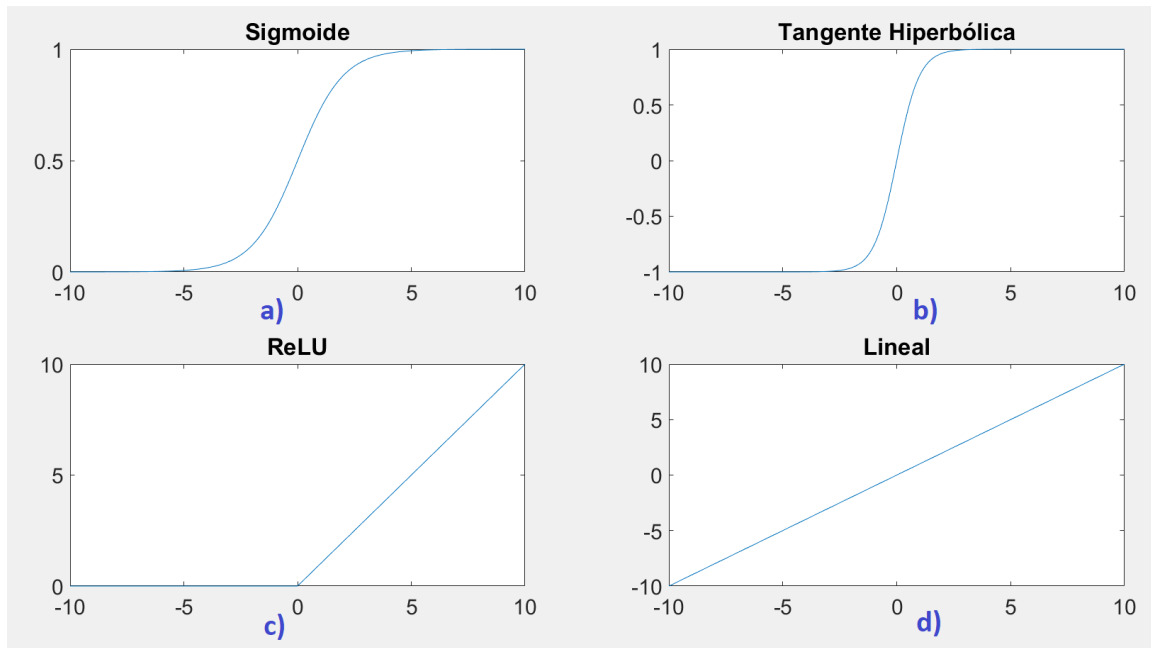


Figura 2.8: Funciones de Activación. a) Función Sigmoide, b) Función Tangente Hiperbólica, c) Función ReLU y d) Función Lineal.

2.5. Consideraciones Importantes acerca del Manejo de Datos por parte de Algoritmos ML Supervisados

En esta sección se aclararán los tipos de conjuntos de datos que manejan las técnicas ML supervisadas, el tipo de datos que pueden contener las variables, así como el procesamiento de la información que se requiere para que la técnica ML supervisada tenga el mejor desempeño.

2.5.1. Tipos de Conjuntos de Datos

En ML hay tres tipos de conjuntos de datos o *Datasets* [54]:

- **Conjunto de Entrenamiento (Train Set).** Son los datos con los que se entrena el modelo o algoritmo. Éste aprende de las relaciones entre las entradas y las salidas proporcionadas por estos datos. *Generalmente, conforme con lo revisado en la teoría, constituye el 80 % del Conjunto Total de Datos.* [19]
- **Conjunto de Validación (Validation Set).** Son los datos con los que se evalúa el modelo durante la etapa de ajuste, y permiten el ajuste de hiperparámetros (*parámetros ajustables*, como el número de capas ocultas por ejemplo) o la selección de características. Generalmente es un subconjunto de los de datos de entrenamiento. Con el apoyo de este conjunto de datos se ve si hay sobreajuste (en inglés, *Overfitting*) o Infraajuste (en inglés, *Underfitting*). *Si lo hay, conforme con lo revisado en la teoría, constituye el 10 % del Conjunto Total de Datos.* [7]
- **Conjunto de Prueba (Test Set).** Estos datos se mantienen aislados, no los ve la técnica ML sino una vez entrenada, y con ellos se evalúa el rendimiento del modelo. *Generalmente constituye el 20 % del conjunto total de datos.* [19]

La propuesta anterior (ver Fig. 2.9) es la más común y es válida siempre y cuando *los datos no estén limitados* [91].

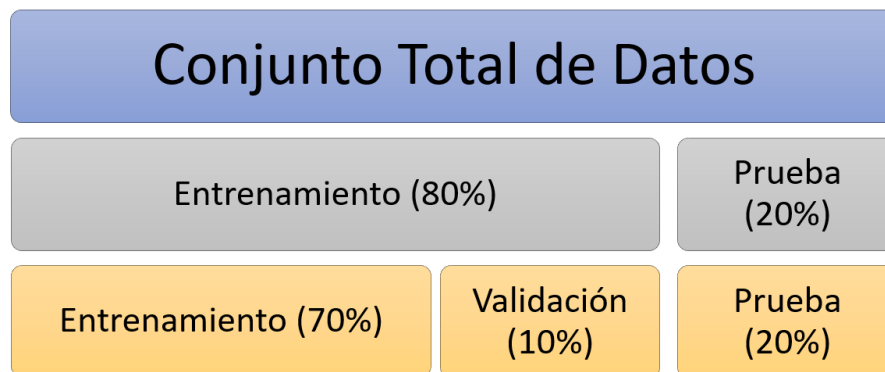


Figura 2.9: Tipos de Conjuntos de Datos en ML.

2.5.2. Validación Cruzada K-Folds

Si los datos son limitados, puede darse el problema de que la información que no se contiene en el conjunto de entrenamiento sea importante y ello provoque un sesgo [91]. Para corregir esta inclinación de la balanza hacia un lado, lo que se emplea en ML es lo que se denomina como *Validación Cruzada (Cross Validation)*.

La Validación Cruzada es una técnica estadística que evalúa el rendimiento de un modelo ML cuando se tienen datos limitados. Formalmente, ésta analiza los resultados del modelo ML para garantizar que sean independientes de las particiones de los datos entre entrenamiento y prueba. Se calcula la media aritmética de las puntuaciones sobre las diferentes divisiones de datos [119]. Aquí hay que mencionar algo muy importante. *Previo a la aplicación de la validación cruzada, ya se debió haber separado el conjunto de prueba. Éste no puede entrar a la validación cruzada y después evaluarse en la etapa de operación, eso sería un tipo de trampa. En otras palabras, ya una vez separado el conjunto de prueba, los datos restantes son los que van a entrar a la validación cruzada.*

El Método K-Folds es una técnica de validación cruzada que es ampliamente usada en la literatura para mitigar este sesgo [28]. Su funcionamiento consiste básicamente en particionar un conjunto de datos de manera aleatoria en K pliegues (o folds). El modelo ML se entrena empleando $K - 1$ pliegues, y se deja el restante para validar el modelo. El proceso se realiza hasta que se cada K-fold (K iteraciones) se utilice y finalmente se obtiene la media aritmética de las puntuaciones obtenidas [91].

El valor de K no debe ser ni muy alto, ni muy bajo, generalmente es entre 5 y 10 dependiendo a los datos. Un valor K elevado puede llevar a la minimización del sesgo, pero a un crecimiento de la varianza [91]. La Figura 2.10 muestra un ejemplo de funcionamiento de la validación cruzada K-Folds con $K = 5$.

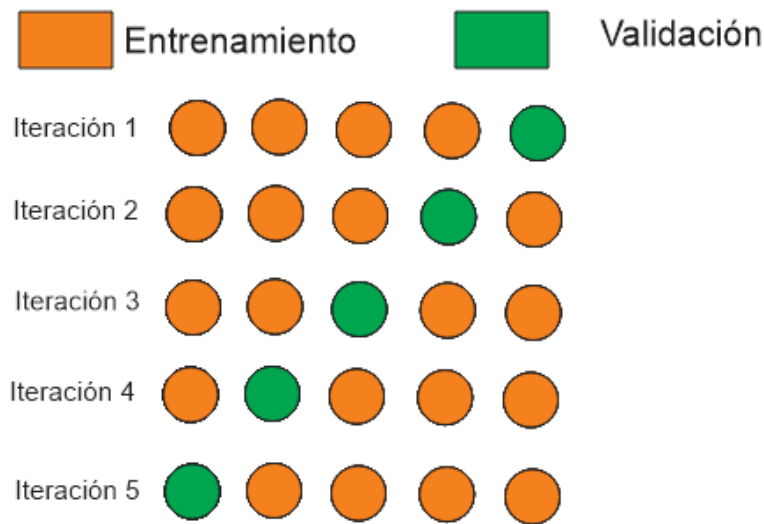


Figura 2.10: Validación Cruzada.

El sobreajuste (u *overfitting*) es un fenómeno que se mitiga con la validación cruzada, por lo que ésta es una técnica muy utilizada para mejorar la generalización de los algoritmos ML.

2.5.3. Tipos de Datos

A grandes rasgos cualquier conjunto de datos puede tener dos tipos de variables [98]:

1. **Variables Numéricas.** Son aquellas que describen una característica mediante un valor numérico o cantidad. *Se pueden hacer operaciones aritméticas con ellas.* Además, se pueden identificar dos tipos de variables numéricas:
 - **Variables Numéricas Discretas.** Variables numéricas que tienen un número contable de valores entre dos valores cualesquiera [85]. Ejemplos de ellas son el número de reportes de una empresa, número de motores con fallas en una fábrica.
 - **Variables Numéricas Continuas.** Variables numéricas que tienen un número infinito de valores entre dos valores cualesquiera. Ejemplos de ellas son la masa, la temperatura, entre otras.[85]
2. **Variables Categóricas.** Aquéllas que toman valores limitados y que asignan a cada observación una clase, una categoría o un grupo. *No se pueden realizar operaciones aritméticas con ellas.* Ejemplos de ellas son la raza y el sexo de personas.

2.5.4. Codificación de Variables Categóricas

Un algoritmo ML supervisado no puede procesar datos que no sean numéricos. No obstante, ello no implica que no se puedan asimilar los datos categóricos, sino es que con un trato especial, la *Codificación*. La codificación simplemente es el proceso de convertir una categoría en un número. Se hablará de las dos formas de codificación empleadas en la presente tesis.

La primera es la *Codificación por Etiqueta* (en inglés *Label Encoding*). Lo que hace esta codificación es básicamente hallar las diferentes categorías de una variable y asignarles un número distinto (entre cero y el número de categorías menos uno). Ello resulta muy benéfico, pero la desventaja de este tipo de codificación es que respeta una relación de orden entre los números, lo que implicaría, ahondándose un poco en el modelo ML, que habría más penalización si el algoritmo se equivoca con una categoría que con otra, y eso sólo interesa cuando importa el orden entre las categorías. Si no interesa la relación de orden entre las clases, entonces esta codificación no funciona [40].

La segunda codificación es la *Codificación Binaria* (en inglés *One Hot Encoding*). En este tipo de codificación se crea un vector de longitud igual al número de categorías existentes, se coloca 0 en la posición de las categorías incorrectas, y 1 en la posición de la correcta. Este tipo de codificación es útil para escenarios donde no importa el orden entre las categorías [20].

En la Tabla 2.1 se muestra en un ejemplo relacionado a la tesis la aplicación de ambas técnicas de codificación. Supóngase que hay cuatro maniobras que efectúa un vehículo en un estudio: zig-zag, recta, vuelta a la derecha y vuelta a la izquierda. Si esas cadenas de caracteres (*strings*) se

introducen al algoritmo ML, no se puede hacer nada, deben ser datos numéricos y ello a través de la codificación.

Primero se considerará la codificación por etiqueta. Para este caso, a la categoría zig-zag se le asigna el valor 0; a recta, 1; a vuelta a la derecha, 2; y a vuelta a la izquierda, 3.

Por otra parte, para el caso de la codificación binaria, a zig-zag se le asigna el entramado “1000”; a recta, “0100”; a vuelta a la derecha, “0010”; y a vuelta a la izquierda, “0001”.

Tabla 2.1: Tipos de codificación de datos categóricos. Ejemplo.

Maniobra	Codificación por Etiqueta	Codificación Binaria			
		Zig-Zag	Recta	Vuelta a la Derecha	Vuelta a la Izquierda
Zig-Zag	0	1	0	0	0
Recta	1	0	1	0	0
Vuelta a la Derecha	2	0	0	1	0
Vuelta a la Izquierda	3	0	0	0	1

2.5.5. Estandarización de los datos numéricos

La estandarización de los datos numéricos consiste en procesar los datos de manera que tengan una media 0 y una desviación estándar de 1. La fórmula se muestra en la ecuación 2.1 [107].

$$z = \frac{(x - \mu)}{s}, \quad (2.1)$$

donde z es el valor estandarizado, x la muestra original de un conjunto de datos, μ la media de dicho conjunto, y s la desviación estándar del mismo.

La estandarización es útil cuando las diferentes variables tienen distintas escalas, y permite una convergencia más rápida durante el mecanismo de aprendizaje del modelo.

2.5.6. Normalización de los datos numéricos

La normalización de un conjunto de datos numéricos consiste en un escalado de los mismos dentro de un rango deseado (min, max). Las expresiones matemáticas que permiten el escalado de un conjunto de datos se muestran mediante las fórmulas [106]:

$$X_{std} = \frac{(X - X_{min})}{(X_{max} - X_{min})}, \quad (2.2)$$

$$X_{sc} = X_{std} \times (max - min) + min. \quad (2.3)$$

donde X_{min} es el valor mínimo del conjunto de datos, X_{max} es el valor máximo y X es la muestra actual.

La normalización permite la estabilidad numérica del aprendizaje de los datos por parte del algoritmo ML.

2.5.7. Filtro de media móvil

Aparte de la normalización y la estandarización, el filtrado es otro tipo de tratamiento de los datos numéricos. Un filtro de media móvil o un filtro promediador es un método usado para suavizar una señal ruidosa. Este filtro hace uso de una ventana de cierto número de datos, WS , los cuales promedia y arroja un resultado, y así va recorriendo un vector de entrada para un paso en específico, $x(n)$. La fórmula matemática que representa este filtro está dada mediante la ecuación ?? [81]:

$$y(n) = \left(\frac{1}{WS}\right)(x(n) + x(n-1) + \dots + x(n - (WS - 1))). \quad (2.4)$$

2.6. Criterios de desempeño de las técnicas ML supervisadas

Recuérdese que para el caso de las técnicas ML supervisadas hay dos etapas en el proceso de ejecución, a saber:

1. *Entrenamiento.* Durante el entrenamiento y a partir del conjunto de datos con el mismo nombre, el Modelo ML halla relaciones entre las entradas y las salidas brindadas.
2. *Operación.* En este punto el desempeño de la técnica ML supervisada entrenada se prueba frente a un conjunto de datos que el modelo no ha visto, el conjunto de prueba.

Entonces se deben tener métricas que evalúen cómo fue el desempeño de la técnica ML supervisada en los ambientes de entrenamiento y operación. Para el caso de tareas de regresión, se comparan las salidas del modelo con las salidas verdaderas (también conocidas como **ground truth**) y se cuantifica dicha similitud o diferencia mediante los siguientes criterios de desempeño:

- **Error Cuadrático Medio**

El Error Cuadrático Medio (en inglés, MSE) se define como se muestra en la ecuación 2.5.

$$MSE = (\hat{y} - \vec{y})^t (\hat{y} - \vec{y}) \quad (2.5)$$

donde \hat{y} es un vector de n predicciones, y \vec{y} es el vector de valores verdaderos. Lo ideal es que el valor del MSE de un modelo tienda a 0.

- **Raíz Cuadrada del Error Cuadrático Medio**

La raíz cuadrada del error cuadrático medio (en inglés, $RMSE$) se define como la raíz cuadrada de la variable definida en el punto anterior. Matemáticamente se expresa en la ecuación 2.6.

$$RMSE = \frac{1}{\sigma} \sqrt{(\hat{\mathbf{y}} - \vec{y})^t (\hat{\mathbf{y}} - \vec{y})} \quad (2.6)$$

donde $\hat{\mathbf{y}}$ es un vector de n predicciones, \vec{y} es el vector de valores verdaderos y σ es la desviación estándar de $\hat{\mathbf{y}}$. Lo ideal es que el valor del RMSE de un modelo tienda a 0.

▪ **Coefficiente de Determinación o R-cuadrado (R^2)**

El coeficiente de determinación (o R^2) es una medida estadística de la proporción de la variación en la variable de respuesta que se explica por los predictores [101]. En notación matemática se expresa en la ecuación 2.7.

$$R^2 = 1 - \frac{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - \vec{y}_i)^2}{\frac{1}{n} \sum_{i=1}^n (\vec{y}_i - \bar{y})^2} \quad (2.7)$$

donde $\hat{\mathbf{y}}$ es un vector de n predicciones, \vec{y} es el vector de valores verdaderos, y \bar{y} es la media de los valores verdaderos.

El numerador de la ecuación 2.7 alude al cálculo del MSE o también conocido como *suma residual* (en inglés, *SSR*), mientras que el denominador se conoce como la *suma total* (en inglés, *SST*). El resultado es un valor entre 0.0 y 1.0 (o en porcentaje), donde el primer valor describe a un modelo cuyas salidas no se ajustan a los valores verdaderos (no fiable); y el segundo, a un modelo bien ajustado (o fiable).

Ya declarado todo lo anterior, ahora sí es tiempo de adentrarse en los Algoritmos ML Supervisados a utilizar en la presente tesis, para ambos escenarios de trabajo.

2.7. Sobreajuste y Regularización por Dilución

Cuando se entrena un modelo ML, específicamente una ANN o sus derivadas, en algunas ocasiones la técnica aprende tan bien las relaciones entre entradas y salidas del conjunto de entrenamiento que, cuando se le presentan nuevos datos para generalizar, no puede hacerlo de buena manera. A ello se le conoce como Sobreajuste (en inglés, *Overfitting*). Para combatir el sobreajuste, han aparecido las Técnicas de Regularización. Una de esas técnicas es la Dilución (o *Dropout*).

La Regularización por Dilución (o Dropout) lo que hace es que, por cada nueva entrada a la ANN en fase de entrenamiento, se desactivará aleatoriamente un porcentaje de las neuronas en cada capa oculta acorde a una probabilidad definida (Figura 2.11). Ello induce a que ningún nodo memorice información, se encuentren mejores relaciones entre los datos de entrenamiento, y se permita una mejor generalización con nuevos datos de entrada.

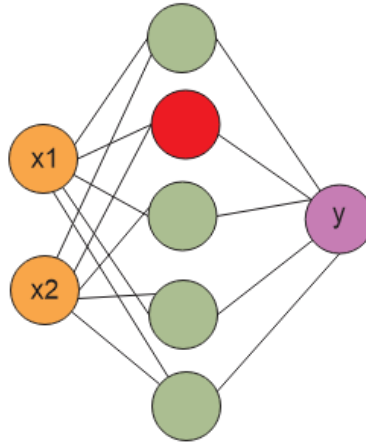


Figura 2.11: Regularización por Dilución (Dropout). Factor de 0.2. El color rojo indica las neuronas desactivadas, y el verde las activadas.

2.8. Redes Neuronales Recurrentes

Las Redes Neuronales Recurrentes (*RNNs*, por sus siglas en inglés) toman en cuenta el orden de los datos o consideran secuencias de datos (si toman en cuenta directamente el tiempo, se habla de *series de tiempo*) (Figura 2.12). Las RNNs se enfocan en tareas de predicción, surgieron para poder dar ese seguimiento de los sucesos ordenados o temporales; no obstante, los primeros modelos formulados sufrían del problema de que los gradientes de las redes *explotaban* (con magnitud muy grande) o se *desvanecían* (cero), lo que no permitía hallar las relaciones entre los datos de buena manera [3]. Para solucionar ello, aparecieron diferentes arquitecturas de RNNs, entre ellas las Redes Long-Short Term Memory (o *Redes LSTM*, por sus siglas en inglés) y las Gated Recurrent Unit (o *Redes GRU*).

2.9. Redes Long-Short Term Memory (o Redes LSTM)

Las Redes LSTM permiten el flujo de información no sólo hacia adelante, como las FFNNs, también hacia las neuronas laterales, lo que permite que se generen bucles de realimentación. Además, las salidas a un cierto tiempo se vuelven entradas en un tiempo posterior. Por lo que se va generando una memoria que aprende del contexto y a través del tiempo [16].

La unidad básica de las redes LSTM se denomina *célula de memoria*, o simplemente, *neurona*. En la Figura 2.13 se muestra el esquema de una neurona LSTM, donde se perciben las compuertas. La neurona LSTM está compuesta de pesos y compuertas, que son los elementos que permiten la estimación del estado sobre la sucesión de eventos. En lo que tiene que ver con los pesos se tienen los 1) *Pesos de las Entradas*, los cuales ponderan las entradas en el paso de tiempo actual; 2) *Pesos de las Salidas*, que ponderan las salidas en el paso de tiempo previo; y 3) *Estado Interno*, que se usa

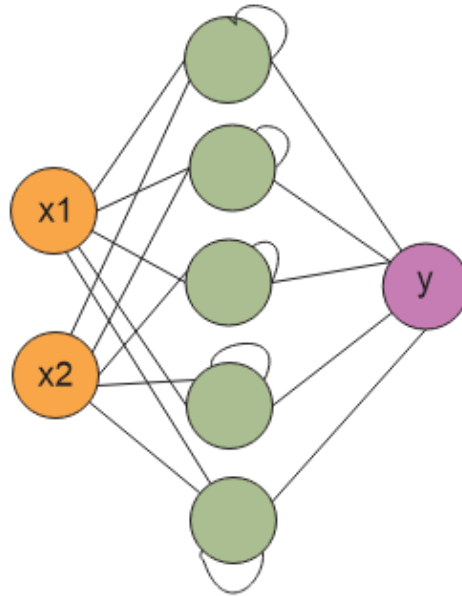


Figura 2.12: Red Neuronal Recurrente.

para el cálculo de las salidas en el paso de tiempo actual [18]. Y con respecto a las compuertas, se definirán en el siguiente apartado, pues requieren una atención más detallada.

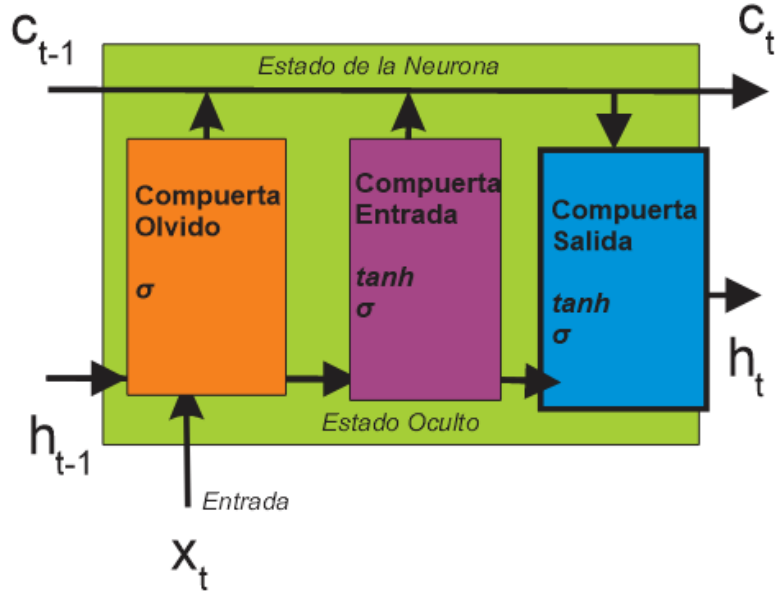


Figura 2.13: Neurona LSTM.

2.9.1. Estados y Compuertas de la Neurona LSTM

Como se puede percibir en la Figura 2.13, en la neurona LSTM hay tres elementos muy importantes: Estado de la Neurona (en inglés *Cell State*), el Estado Oculto (en inglés, *Hidden State*) y las Compuertas o *Gates* (del Olvido, Entrada y Salida). El Estado de la Neurona se refiere a la codificación de la información de todos los pasos de tiempo previos que se han procesado; por otra parte, el Estado Oculto codifica la caracterización de información del paso de tiempo *inmediatamente previo*, no de todos los pasos de tiempo anteriores. Ahora, en lo concerniente a las compuertas, hay de tres tipos: Compuerta Olvido (*Forget Gate*), Compuerta de Entrada (*Input Gate*), y Compuerta de Salida (*Output Gate*) [30]. Cada una de esas compuertas lleva consigo una o dos ecuaciones que la definen. A continuación se muestran esas dos ecuaciones:

$$g_t = \sigma((W_{hg} h_{t-1}) + (W_{xg} x_t) + b_g), \quad (2.8)$$

$$g_t = \tanh((W_{hg} h_{t-1}) + (W_{xg} x_t) + b_g), \quad (2.9)$$

donde h_{t-1} es el estado oculto del paso de tiempo previo, x_t es la entrada en el paso de tiempo actual, W_{hg} es la matriz de pesos correspondiente a la compuerta de que se trate y al estado oculto, W_{xg} es la matriz de pesos correspondiente a la compuerta y a la entrada, b_f es el bias de la predefinida compuerta.

La parte común de ambas ecuaciones (2.8) y (2.9) son el estado oculto del paso de tiempo previo, la entrada del paso de tiempo actual y un sesgo o bias. Tanto el estado oculto del paso de tiempo anterior como la entrada actual van multiplicadas por una matriz de pesos correspondiente a la compuerta de que se trate y a si es el estado oculto o la información de entrada. Esos dos resultados se suman y se le agrega el bias o sesgo. A ello es lo que se le aplica una función de activación. Como ya se exhibió en las ecuaciones anteriores, esa función de activación puede ser de dos tipos: *tangente hiperbólica* (\tanh) o *sigmoide* (σ). Si se trata de una función de activación sigmoide, el resultado es simplemente un conjunto de escalares cuyo propósito es incrementar/decrementar la importancia de la información por la que se multiplique, esto es, explotar/desvanecer los datos, respectivamente. Si el resultado de esa función de activación sigmoide son escalares cercanos a 1, cuando se multiplique por información de la neurona, ello querrá decir que esa información va a prevalecer; por el contrario, si el conjunto de escalares son valores cercanos a cero, al multiplicarse por la información de la neurona, ello va a implicar que dicha información desaparecerá o no tendrá importancia para la célula. Tocando la otra función de activación, la tangente hiperbólica, esta tiene un rango $(-1,1)$, por lo que se encarga de realizar una transformación de los datos a una codificación normalizada [18].

Ya una vez aclaradas las fórmulas y la interpretación de los resultados, considérese las distintas

Compuertas.

Compuerta Olvido (o *Forget Gate*)

Esa compuerta lleva una función de activación sigmoide, por lo que se refiere a un escalado de los datos por los cuales se multiplique, a saber: el estado de la neurona. Si, según los datos de entrada en el paso de tiempo actual la compuerta arroja datos cercanos a 1, ello quiere decir que ve importante seguir conservando información del estado de la neurona; por el contrario, si los valores son cercanos a cero, ello quiere decir que la información del estado de la neurona no es relevante y debe olvidarse [31].

Compuerta de Entrada (o *Input Gate*)

Para el caso de la Compuerta de Entrada, ésta hace uso de las dos ecuaciones de compuertas ya definidas. La Compuerta de Entrada tiene una función de activación sigmoide, por lo que su función es análoga al caso de la Compuerta Olvido pero obviamente con sus propias matrices de pesos. A diferencia de la Compuerta Olvido, esta también tiene una función de activación tangente hiperbólica, la cual codifica y normaliza la información proveniente del estado oculto previo y de la entrada de datos actual. La salida de la función tangente hiperbólica se multiplica con la salida de la función sigmoide. En otras palabras, la función tangente hiperbólica actúa como un extractor de características de los datos provenientes del estado oculto previo y de la entrada actual, y la función sigmoide es la encargada de decir cuáles de esas características vale la pena recordar y cuáles no. El producto de ambas funciones se multiplica por el estado de la célula. Ello lleva dos finalidades: 1) agregar al estado de la neurona información importante para la red, y 2) recordar/desechar datos [31].

Compuerta de Salida (*Output Gate*)

Finalmente, la Compuerta de Salida hace uso de ambas funciones de activación con los mismos propósitos (escalado y normalización), pero con dos diferentes enfoques: 1. incorporar información al estado de la célula, y 2. formar un estado oculto de salida para predicciones o para ser retroalimentado a la neurona LSTM en el siguiente paso de tiempo. Hasta este momento, el estado de la neurona posee información de todos los pasos de tiempo previos, pero también ya del paso de tiempo actual [94].

2.9.2. Configuraciones de Redes LSTM

En la revisión de la teoría se pueden encontrar diversas configuraciones de Redes LSTM [18], pero para la presente tesis sólo interesan dos:

LSTM Vainilla

Esta arquitectura es la más simple y convencional, y está compuesta por una capa de entrada, una capa LSTM, una capa oculta densa y una capa de salida, como se ilustra en la Figura 2.14 [18].

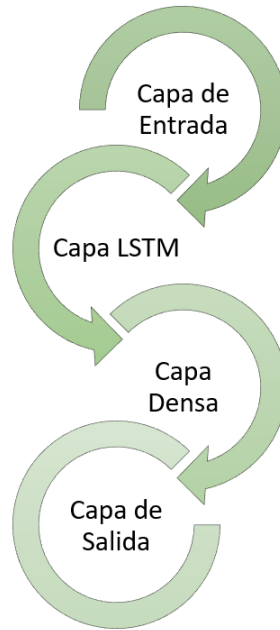


Figura 2.14: Vainilla LSTM [18].

LSTM codificador-decodificador (o LSTM Encoder-Decoder)

Esta arquitectura está compuesta por dos modelos: *Codificador* y un *Decodificador*. El primero lee las secuencias de entrada y las codifica en un vector de tamaño fijo, y el segundo tiene la labor de decodificar dicho vector de tamaño fijo y extraer las secuencias de salida. Esta arquitectura de codificador-decodificador tiene varias aplicaciones, entre ellas: traducción automática, clasificación de movimientos, generación de descripciones de texto para imágenes, entre otras [18]. En la Figura 2.15 se muestra la estructura del modelo LSTM codificador-decodificador.



Figura 2.15: LSTM Codificador-Decoder [18].

2.10. Redes Gated Recurrent Unit (o *Redes GRU*)

Las Redes Gated Recurrent Unit (o *Redes GRU*, por sus siglas en inglés), son otro tipo de redes recurrentes que se utilizan para tareas de predicción. Las Redes GRU fueron creadas en el 2014 [104] con el propósito de combatir la desaparición/explosión de los gradientes, problema típico de las tradicionales RNNs. La forma de trabajo de las Redes GRU es muy semejante a la de las Redes LSTM, sólo que las primeras sólo tienen dos compuertas, no tienen el estado de la célula (o *cell state*) y aparece un nuevo operador elemento a elemento, el operador *Producto Hadamard* [67].

Las Redes GRU tienen dos compuertas de trabajo: la Compuerta de Actualización (o *Update Gate*) y la Compuerta de Reinicio (o *Reset Gate*). Y lo más interesante de ello es que se sigue aplicando la lógica que se propone en [94] acerca de las funciones de activación.

De manera subsecuente se muestran los pasos de la forma de trabajo de la unidad GRU (Figura 2.16).

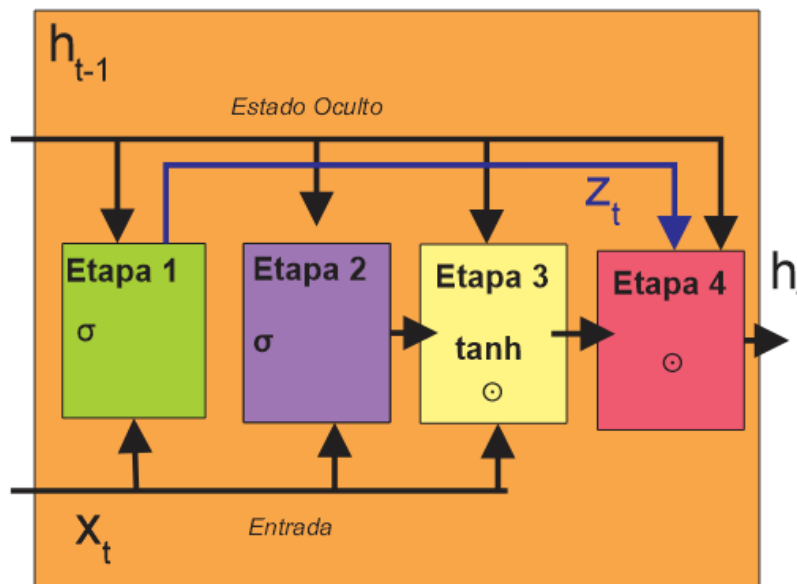


Figura 2.16: Neurona GRU.

2.10.1. Etapa 1. Compuerta de Actualización para el tiempo actual

La salida que arroja la compuerta de actualización se parece a la Compuerta Olvido de la Red LSTM. En ambas participan la entrada en el tiempo actual y el estado oculto del tiempo previo. A ambas se les multiplica por su matriz de pesos, se les suma y se les aplica la función sigmoide, pero en la unidad GRU no se le suma el sesgo [39]. Matemáticamente lo anterior se expresa mediante (2.10)

$$z_t = \sigma((W^{(z)} \times x_t) + (U^z \cdot h_{t-1})) \quad (2.10)$$

, donde $\mathbf{W}^{(z)}$ es la matriz de pesos correspondiente a las entradas del tiempo actual, \mathbf{x}_t , y \mathbf{U}^z es la matriz de pesos correspondiente al estado oculto del tiempo previo, \mathbf{h}_{t-1} . De manera puntual, la función de la Compuerta de Actualización es *ver cuánta información del pasado (pasos de tiempo previos) es útil para el futuro*.

2.10.2. Etapa 2. Compuerta de Reinicio para el tiempo actual

La compuerta de reinicio para el tiempo actual tiene la función de *decidir cuánta información del pasado (pasos de tiempo previos) debe olvidarse* [39]. La deducción matemática se muestra en (2.11).

$$r_t = \sigma((W^{(r)} x_t) + (U^r h_{t-1})), \quad (2.11)$$

, donde $\mathbf{W}^{(r)}$ es la matriz de pesos correspondiente a las entradas del tiempo actual, \mathbf{x}_t , y \mathbf{U}^r es la matriz de pesos correspondiente al estado oculto del tiempo previo, \mathbf{h}_{t-1} .

2.10.3. Etapa 3. Memoria Actual

Este nuevo contenido de memoria hace uso de la compuerta de reinicio, además de una función tangente hiperbólica y del operador Producto de Hadamard. Matemáticamente, este contenido de memoria se representa mediante (2.12).

$$h'_t = \tanh(Wx_t + r_t \odot Uh_{t-1}), \quad (2.12)$$

, donde \mathbf{W} es la matriz de pesos correspondiente a las entradas del tiempo actual, \mathbf{x}_t , y \mathbf{U} es la matriz de pesos correspondiente al estado oculto del tiempo previo, \mathbf{h}_{t-1} . La interpretación del Producto de Hadamard es *evaluar cuánta información debe eliminarse de los pasos de tiempo previos* [39].

2.10.4. Etapa 4. Contenido Final de la Memoria en el tiempo actual

Este es el último paso en el funcionamiento de la unidad GRU, y lo que se busca es recopilar toda la información que se tiene hasta el paso de tiempo actual. Se requiere el uso del resultado de la compuerta de actualización, y formalmente el propósito de esta última etapa es *evaluar qué usar del contenido de la memoria actual (etapa previa) y qué de los pasos de tiempo previos* [67]. Lo anterior en notación matemática queda como lo muestra (2.13)

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t, \tag{2.13}$$

donde z_t es el resultado de la compuerta de actualización; h_{t-1} es información de pasos de tiempo previos; y h'_t es el contenido actual de la memoria de la unidad GRU.

2.11. Transformer

El *Transformer* es un modelo de aprendizaje automático muy potente que tiene su fundamento de trabajo en un mecanismo denominado *Atención* (en inglés, *Self-Attention*) [122]. En las arquitecturas recurrentes se toma una muestra o paso de tiempo a la vez, en el Transformer son todos los pasos a la vez, lo que le permite una mejor paralelización y entrenamiento.

2.11.1. Arquitectura

La arquitectura del modelo original de Transformer se describe en [122]. Para visualizar la constitución del modelo ML, remítase a la Figura 2.17.

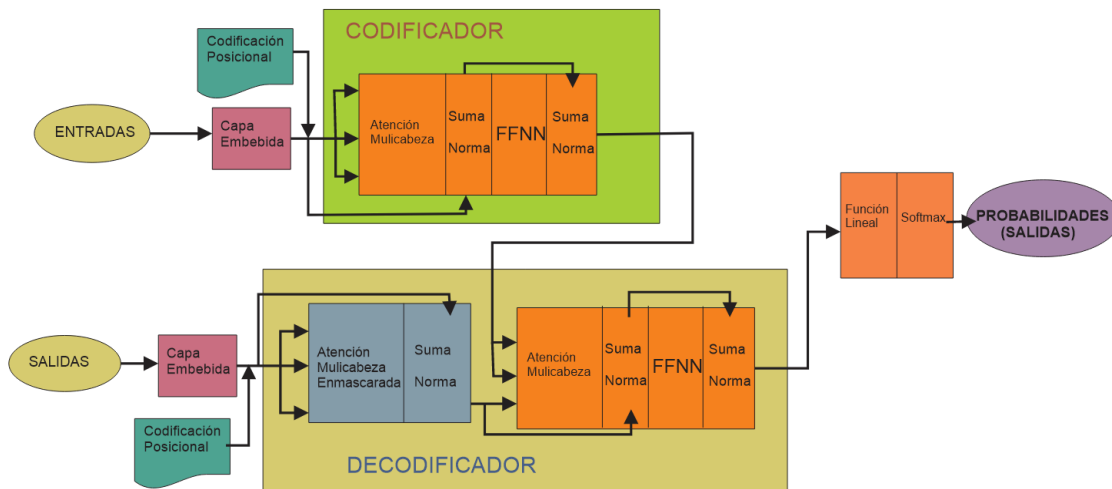


Figura 2.17: Arquitectura Transformer Tradicional.

Esta arquitectura estaba enfocada a tareas de procesamiento de lenguaje natural, por ello asume que se están procesando palabras o tokens. Los tokens, para las entradas como para las salidas, deben convertirse en una representación vectorial, y ello se logra a través de la Capa de *Embedding*. Al tener una representación vectorial, se pierde el orden en las entradas y salidas, por lo que es necesario agregar una *Codificación Posicional*. Los principales elementos de este modelo se enuncian de manera subsecuente.

Codificador (o Encoder)

El Encoder está conformado por varias capas apiladas, cada una de ellas compuesta por una capa de Multi-Atención (en inglés *Multi-Head Attention*) y una Capa Densa o Convolutiva 1D. La primera tiene el objetivo de obtener las relaciones temporales y espaciales de las secuencias de entrada, y la segunda la de adaptar los vectores de atención para que puedan ser aceptables para otras capas del encoder [122].

Decodificador (o Decoder)

El Decoder está conformado por varios estratos, cada uno constituido por una capa multi-atención para las secuencias de salida, otra para la conjunción de las secuencias de entrada con las de salida, y por último una capa densa o convolutiva 1D [122].

Mecanismo de Atención (o *Self-Attention*)

El mecanismo de atención es la base de esta arquitectura y está compuesto de tres elementos, las Consultas (en inglés, *Queries (Q)*), las Claves (en inglés, *Keys (K)*) y los Valores (en inglés, *Values (V)*). Remitiéndose a la Figura 2.17, se puede percibir que a cada bloque multi-atención llegan tres flechas, las cuales corresponden a los tres elementos ya declarados. Lo que se hace básicamente es comparar una consulta con todas las claves mediante el *producto punto* para determinar la matriz de pesos que multiplicará a los valores. Para evitar que, tras esa multiplicación, los valores se hagan muy grande, se aplica un factor de normalizado, para posteriormente aplicar una función *softmax* que tiene el objetivo de hallar la función de probabilidad de las relaciones entre los datos [122].

El mecanismo de atención se define como (2.14)

$$\text{Atención}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (2.14)$$

Atención Multi-Cabeza (o *Multi-Head Attention*)

La Atención Multi-Cabeza (en inglés, *Multi-Head Attention*) es una generalización del mecanismo de atención (definido en el punto previo) aplicada h veces para permitir la paralelización de la información.

2.12. Transformer + Time2Vector

La principal aplicación del Transformer es el procesamiento de lenguaje natural [84]. No obstante, para secuencias temporales esta arquitectura no es suficiente. Y la razón es que las secuencias de tiempo el Transformer las maneja como un todo, es decir, pierde las dependencias espaciales y tem-

porales. Para solucionar ello, en [105] se expone la idea de unir el Transformer con una representación vectorial de tiempo, que se retoma del trabajo de [63].

2.13. Time2Vector

La idea detrás de esta representación vectorial de tiempo, conocida como *Time2Vector*, es poder codificar el tiempo para el Transformer. Es algo semejante a una Capa de Inscrustación/Encaje (en inglés, *Embedding Layer*) para las ANNs. En [105] se destacan dos puntos sustanciales del trabajo de [63]. El primero es que la representación vectorial del tiempo debe incluir *patrones periódicos* y *patrones no periódicos*. El segundo se refiere a que la representación vectorial del tiempo debe ser *invariante* a su reescalado, lo que se traduce en que la representación no puede verse afectada por incrementos de tiempo o largos horizontes.

Para un escalar de tiempo, τ , ambas sentencias se expresan matemáticamente como:

$$t2v(\tau)[i] = \begin{cases} \omega_i \tau + \varphi_i, & \text{if } i = 0, \\ \mathcal{F}(\omega_i \tau + \varphi_i), & \text{if } 1 \leq i \leq k, \end{cases} \quad (2.15)$$

donde $t2v(\tau)[i]$ es el i -ésimo elemento de $t2v(\tau)$, \mathcal{F} es una función periódica o *no lineal* de activación, y ω_i s y φ_i s son parámetros aprendibles.

La primer sentencia de la función a trozos mostrada en (2.15) representa la porción no periódica o lineal de la representación vectorial del tiempo. Si se pone atención, es análoga a la ecuación de un regresor lineal, $y = mx + b$, donde ω es una matriz que define la pendiente de la serie de tiempo y φ es la ordenada al origen. Por otro lado, la segunda sentencia representa la porción periódica o no lineal del tiempo, y tiene sentido ya que al argumento se le aplica una función de activación no lineal. En el trabajo de [63] se destaca la función de activación *seno* con mejores resultados sobre otras [105].

2.14. Arquitectura empleada en la presente tesis

El modelo Transformer + Time2vec es el que se utilizará en la presente tesis. Este modelo no tiene decoder (como el modelo BERT) y el conjunto consulta, clave y valor está dado por las salidas, entradas y la representación vectorial del tiempo (*time2vec*), respectivamente. Otro punto interesante es que no hay codificación posicional, sino la representación vectorial del tiempo [105].

2.15. Sumario

Este capítulo describió el fundamento teórico de lo que se verá en la Metodología de la presente tesis. Primero, se empezó por el sistema coordinado vehicular, pues es necesario ubicarse en el espacio y conocer las variables dinámicas vehiculares de las cuales se hablará en los capítulos que vienen. Después, se explicaron los fundamentos del ML, empezando con términos generales aplicables a todas las técnicas ML para después pasar a las técnicas específicas en sí. Es un capítulo un tanto largo, pero necesario para la comprensión de la Metodología y Resultados. De esta manera, se da el paso a la descripción de la Metodología, ámbito que se definirá en el siguiente capítulo.

Capítulo 3

Metodología

Este capítulo detalla la metodología (Figura 3.1 y Figura 3.2) a seguir referente a la aplicación de técnicas ML para la configuración de Modelos Vehículo-Conductor. Es de destacar que, en términos generales, esta misma metodología aplica para ambos enfoques. Sin embargo, sufre pequeñas variaciones dependiendo de la aplicación: Estrategias de Manejo de Energía o Sistemas de Asistencia a la Conducción.

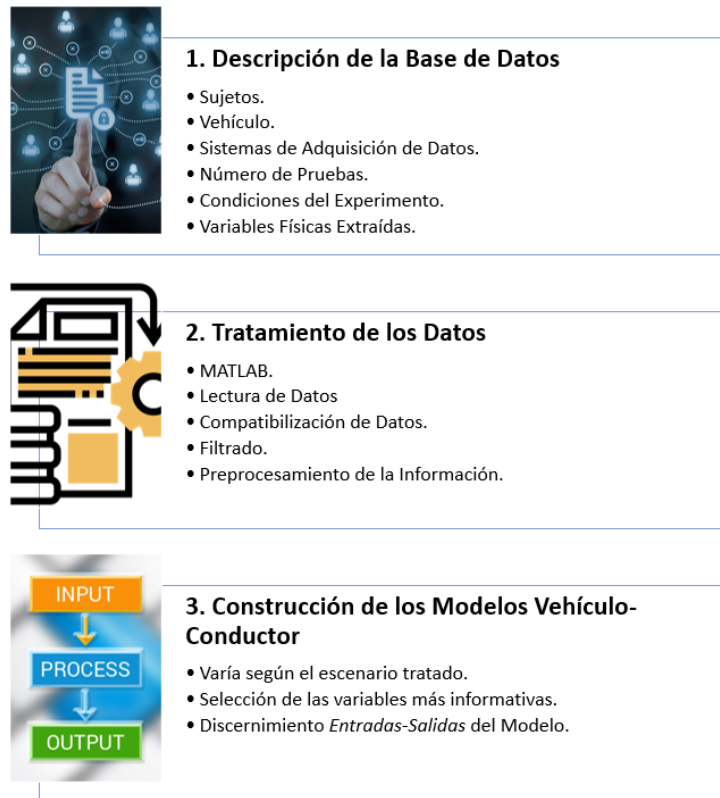


Figura 3.1: Descripción General de la Metodología usada. Parte 1.

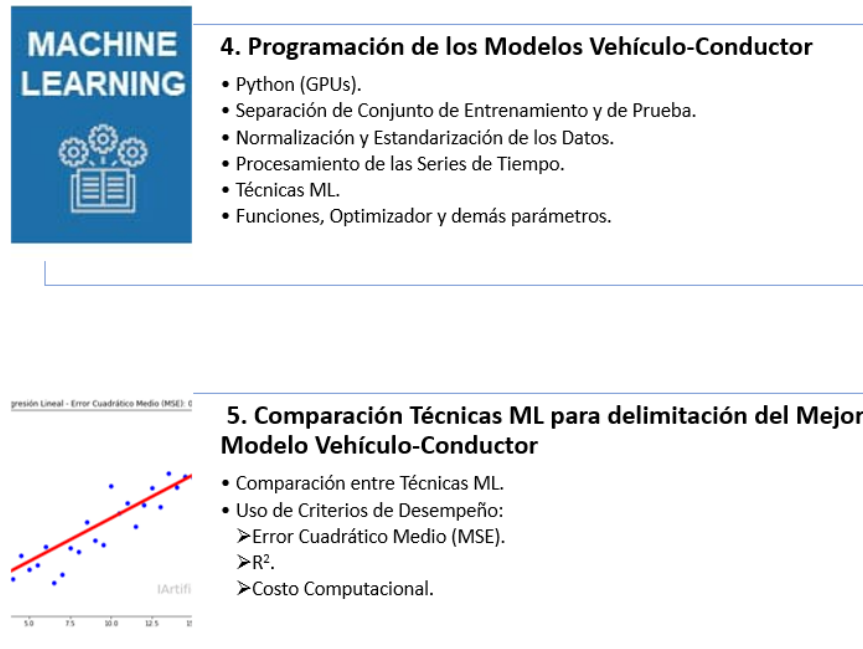


Figura 3.2: Descripción General de la Metodología usada. Parte 2.

La Metodología General de la tesis se divide en cinco etapas, las cuales se enuncian a continuación:

1. Descripción de la Base de Datos. En este punto se describen los lineamientos en que se realizaron los experimentos para obtener los datos, a saber: número de sujetos, vehículo, condiciones para el experimento, equipos para adquisición de datos y variables físicas cuantificadas.

2. Tratamiento de los Datos. Una vez extraídos los datos mediante las fuentes de adquisición, se procede a leerlos mediante ordenador con apoyo del software *MATLAB*, compatibilizarlos y, de manera general, realizar un pre-procesamiento de la información para que quede lo más "limpia" posible para su utilización en los pasos subsecuentes.

3. Construcción de los Modelos Vehículo-Conductor. Según el escenario abordado se seleccionan las variables que mejor describen el fenómeno y que más aportan información; ello con apoyo de la literatura. Además, en esta etapa, se identifican las variables físicas que fungirán como entradas y salidas del modelo.

4. Programación de los Modelos Vehículo-Conductor. Una vez que se construyeron los Modelos Vehículo-Conductor, se programan en el lenguaje de *Python*. Para ello, lo primero es segmentar las entradas y salidas de manera que se obtengan los Conjuntos de Entrenamiento (en inglés *Train Set*) y de Prueba (en inglés *Test Set*). Después, se procede a la normalización y la estandarización de la información. Luego, viene la parte del Procesamiento de las Series de Tiempo, donde se acomodan las variables dependiendo al horizonte de predicción que se busca y al historial que se desea tomar. Ya teniendo esto, se aplican todas las técnicas de ML que se encontraron según el enfoque abordado. Es muy importante hacer notar que se buscan los mejores resultados de cada

técnica de ML empleada, por ello es imprescindible la afinación de hiperparámetros y parámetros del modelo.

5. Comparación de las técnicas ML para la determinación del mejor modelo para cada sujeto y cada escenario. Situándose en un escenario en particular y con determinado sujeto, lo que se hace es comparar los resultados de cada técnica ML empleada, dados por los Criterios de Desempeño o Evaluación (MSE, R^2) y por el Costo Computacional. *Se elige como representante del mejor modelo la técnica ML con el mayor R^2 y menor costo computacional de todas.* Este último rubro sirve de escalón para la formulación de las conclusiones y prospectivas del trabajo.

3.1. Descripción de la Base de Datos

En este rubro se describirá lo relacionado al experimento y a la obtención de los datos. Se aclara de manera muy puntual que el autor de esta tesis no realizó la experimentación ni la adquisición de la información, pero sí se trató de informar de la manera más completa posible acerca todo lo relacionado a aquéllas. Para una mejor comprensión de esta sección, se desglosará la información en varios subtemas.

3.1.1. Lugar del Experimento

Los experimentos en campo se realizaron en la pista de pruebas del Instituto Mexicano del Transporte, 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 [56],[57].

3.1.2. Equipos para Adquisición de Datos

Para la adquisición de los datos se emplearon tres equipos: un Registrador de Datos, una Unidad de Medición Inercial (o *IMU*) y un Transceptor PCAN-USB. En la Figura 3.3 se muestra la conjunción de dichos dispositivos.

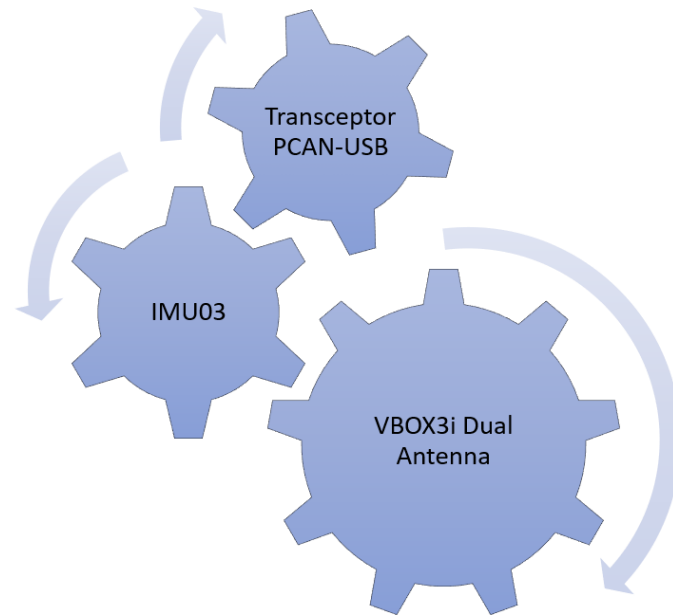


Figura 3.3: Equipos de Adquisición de Datos.

Registrador de Datos

El primero de los equipos de adquisición es el *VBOX3i Dual Antenna*, un registrador de datos vehiculares de la marca *RACELOGIC*. El dispositivo tiene las siguientes características [123]:

- Medida simultánea del ángulo de deslizamiento lateral (o *slip angle*), ángulos pitch y roll, la tasa de cambio del ángulo yaw (o *yaw rate*), el rumbo verdadero (o *true heading*), latitud, longitud y rapidez.
- Baja latencia (8.5 ± 1 ms).
- 4 entradas analógicas y 2 salidas digitales.
- Sistema de adquisición de datos inalámbrico de 100 Hz (*frecuencia de muestreo*) basado en el GPS.
- RS232/USB/Bluetooth.
- Medida de algunas variables físicas con apoyo de la IMU.

Unidad de Medición Inercial (o IMU)

El segundo equipo es una IMU 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*, 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 (o CAN)*, y posteriormente enviarlo a un ordenador.

3.1.3. Sujetos

Para las pruebas se consideraron cuatro sujetos: tres hombres y una mujer. No se consideraron restricciones de experiencia de manejo, sólo que fueran mayores de edad.

3.1.4. Vehículo

Para la ejecución de las pruebas se empleó un vehículo Mazda 3 s Sport, modelo 2008. En la Tabla 3.1 se muestran algunas de las características del vehículo usado.

Tabla 3.1: Características del vehículo [33].

Característica / Propiedad	Valor
Transmisión	Caja automática de 5 velocidades.
Tracción	Delantera.
Cilindros	4 en línea
Número de Asientos	5
Capacidad de Combustible	65.92 l.
Alcance en Km. (Ciudad / Carretera)	513.38 / 676.73 Km.
Tamaño del Motor	2.3 l.
Caballos de Fuerza	156 hp @ 6,500 rpm
Torque	150 lb-ft @ 4,500 rpm
Número de Válvulas	16
Vía Delantera	1.53 m.
Altura	1.46 m.
Largo	4.51 m.
Ancho (sin retrovisores)	1.75 m.
Vía Trasera	1.51 m.
Batalla	2.64 m.

Es importante mencionar las características del vehículo, pues un factor fundamental en el Modelo Vehículo-Conductor.

3.1.5. Condiciones del Experimento

Una prueba consistió en una vuelta al circuito por cada conductor y en determinado sentido. A los sujetos se les pidió que manejaran de manera natural y de forma variable durante los ensayos. Se tienen dos sentidos: *sentido antihorario (o AH)* y *sentido horario (o H)*. En la Figura 3.4 se ilustra el sentido AH del recorrido en pista (la estrella indica el inicio/fin de la trayectoria).



Figura 3.4: Recorrido en la Pista. Sentido AH.

La conducción de cada sujeto en una prueba, ya sea en sentido H o AH, debía cumplir algunas maniobras. Las maniobras que se realizaron son seis: Vuelta a la Derecha (VD); Vuelta la Izquierda (VI); Codo hacia la Izquierda (CI); Codo hacia la Derecha (CD); Recta (R); y Maniobra de Evasión de 16 Obstáculos o *Prueba de Slalom (ZIG-ZAG)*. En la Figura 3.5 se muestran las maniobras presentes en una prueba AH.

Es importante hacer notar lo siguiente: Cada conductor realizó múltiples pruebas en ambos sentidos, pero sólo se consideraron las diez mejores pruebas (ó diez mejores vueltas a la pista): cinco en sentido AH, y cinco en sentido H. En cada prueba se obtuvieron datos del conductor y del vehículo mediante los tres sistemas de adquisición de datos. Con mejores pruebas se alude a que en algunos ensayos se perdieron datos por desconexión de los equipos, saturación de sensores, vibraciones, y esas se descartaron. Otra situación que se necesita aclarar es que, *para la adquisición de los datos se sincronizaron los tres equipos de manera que empezaran a tomar datos al mismo tiempo.*

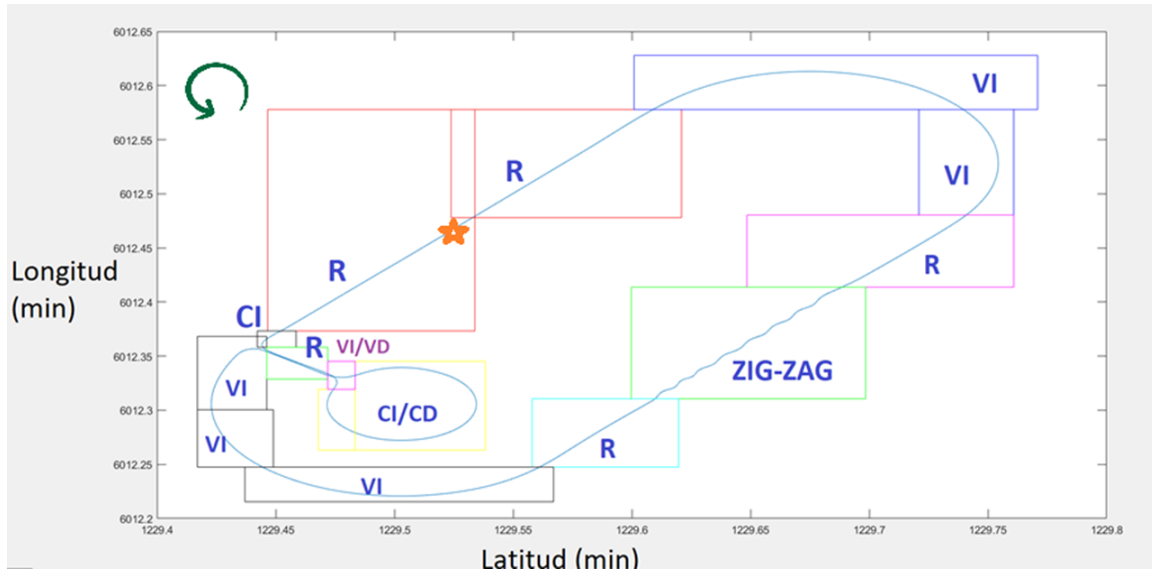


Figura 3.5: Maniobras. Sentido AH. La estrella marca el inicio-fin de la prueba. Las unidades están en minutos (min).

3.1.6. Variables físicas del vehículo y del conductor

Los Modelos vehículo-Conductor incluyen variables físicas de ambos agentes. En la Tabla 3.2 se presentan las variables físicas medidas o calculadas por los tres equipos de adquisición de datos, así como su categoría y resolución respectivas.

Cabe notar dos situaciones. La primera es que, por la forma de la colocación de las antenas del VBOX, no se pudo medir el ángulo Pitch (razón por la que aparece en rojo en la Tabla 3.2). Y la segunda es que, también debido a la configuración de las antenas, el Ángulo de Deslizamiento Lateral (o Slip Angle) coincidió con el Ángulo de Viraje (o Ángulo Yaw).

La mayoría de las variables pueden tomar valores en el dominio continuo, a excepción de las variables de los pedales, que son discretas. Para el caso del acelerador, el rango de valores de es 0-200 unidades con incrementos de 1 unidad, no todos los número reales. Y para el caso de la activación del freno, sólo hay dos valores posibles: 0 (freno desactivado) y 64 (freno activado).

3.2. Tratamiento de los datos

Una vez que se realizó la descripción de la base de datos, la siguiente etapa es el tratamiento de la información. Se debe leer la información aportada por los equipos de adquisición de datos mediante el ordenador, compatibilizarla y prepararla de la mejor manera para que pueda ser empleada por las técnicas ML. A continuación se mencionan los pasos dentro de esta etapa.

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

El equipo arroja datos *en bruto* (o datos *raw*), que finalmente son almacenados en un archivo excel, ello por cada prueba (para todos los sujetos y para ambos sentidos). *Con datos en bruto se refiere a que son datos sin procesar*. El archivo excel contiene información de los distintos sistemas del vehículo, a lo que se le asocia un número como Identificador (*ID*). Recuérdese que el transceptor PCAN-USB recibe información del vehículo a través del Protocolo CAN. Cada identificador tiene una longitud de ocho bytes; no obstante, no todos esos bytes son útiles para determinar información del correspondiente sistema. Complementando lo anterior, algunos sistemas o variables del vehículo pueden provenir del mismo Identificador (*ID*), pero no de la misma combinación de bytes. De manera específica, sólo interesan dos sistemas o variables:

- **Posición del Acelerador**(*ID: 513*). Esta variable viene definida por el byte 7 del respectivo *ID*.
- **Activación del Freno** (*ID: 400*). Este parámetro equivale al byte 3 del correspondiente *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 me-

diante ordenador, la empresa fabricante de los equipos ofrece algunas opciones, entre ellas la lectura mediante el software *MATrix LABoratory* (o *MATLAB*). Y esa es la opción que se utilizó. Los programas proporcionados por RACELOGIC tienen la desventaja de que sólo permiten leer un archivo .vbo a la vez. Para solucionar ese problema, lo que se le añadió fue la opción de poder leer varios archivos a la vez para acelerar el proceso.

3.2.3. Compatibilización de Datos

Los tres equipos de adquisición de datos 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., la única frecuencia de muestreo que se ajustó fue la del PCAN-USB a 100 Hz.

Para este momento los datos provenientes de las tres fuentes son iguales en tamaño, por ello se pueden leer y concatenar para que queden guardados en un mismo archivo.

3.2.4. Filtrado de Información

Hasta el momento las variables del vehículo y del conductor, proporcionadas por los tres sistemas de adquisición, no están filtradas, sólo un tanto procesadas. Para filtrarlas, se les aplica un filtro de media móvil de diez muestras. La razón de ello es que se debe buscar un compensación entre una señal limpia y una señal que no reduzca tanto su amplitud, ese es el compromiso. Al probar con diferentes parámetros de muestras, se encontró que diez es el valor más adecuado que respeta ese acuerdo.¹

En la Figura 3.6 se muestra un ejemplo de una señal original con su correspondiente señal filtrada. Para fines ilustrativos se muestra la variable *Rapidez* correspondiente a la Prueba 1 del Sujeto 1 en sentido AH.

Dicho proceso de filtrado se aplica a las variables obtenidas mediante los equipos VBOX3i e IMU03. Con el mecanismo de Filtrado se finaliza la etapa de Tratamiento de los Datos.

3.3. Construcción de los Modelos Vehículo-Conductor

Este estrato de Metodología se dividirá en dos grandes rubros según el enfoque abordado, las *Estrategias de Manejo de Energía* y *Sistemas Avanzados de Asistencia a la Conducción*.

3.3.1. Estrategias de Manejo de Energía

Conforme a lo revisado en la literatura, además de las áreas de oportunidad que se encontraron en dicha revisión (que se pueden encontrar en el capítulo 1 de esta tesis), las variables del Modelo

¹No se entrará más a detalle de los experimentos que se realizaron para hallar este valor óptimo por dos razones, la extensión del documento y porque no es un objetivo central de esta tesis.

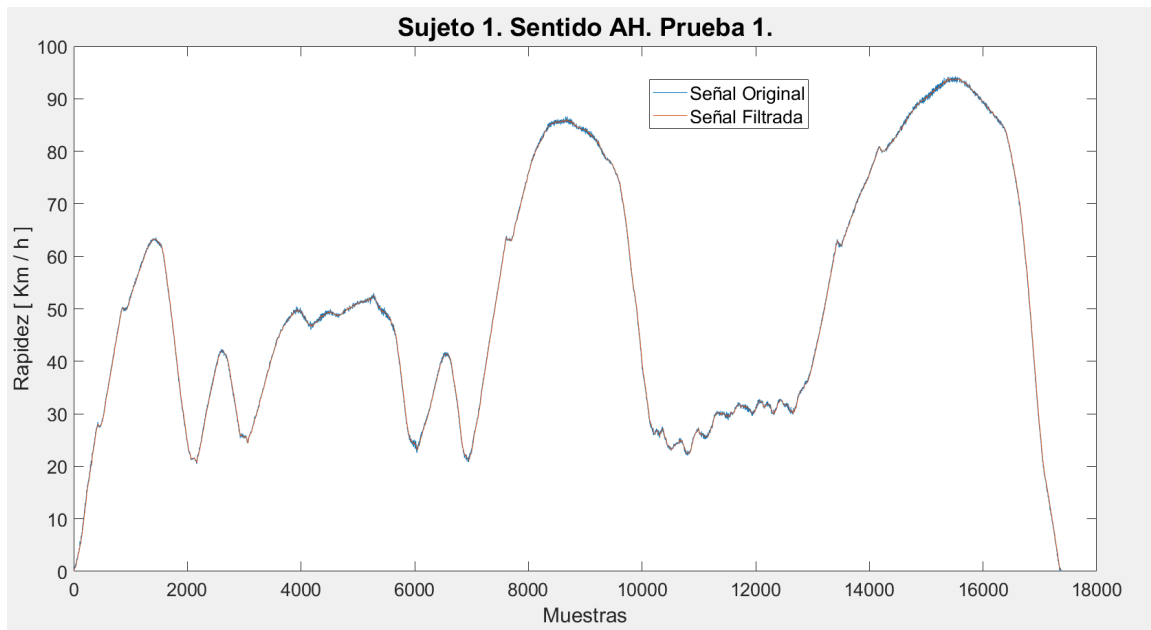


Figura 3.6: Filtrado de la Señal *Rapidez* de la Prueba 1 del Sujeto 1 en sentido AH.

Vehículo-Conductor que más aportan información, además de su discernimiento entre entradas y salidas, se muestran en la Tabla 3.3. La Rapidez es una variable sumamente importante en las Estrategias de Manejo de Energía, ya que lo que se busca para esta perspectiva es maximizar la transmisión de potencia en un vehículo eléctrico híbrido mediante la predicción de los perfiles de velocidad en largos horizontes de tiempo. Después de aclarado lo anterior, conviene destacar el hecho de que la rapidez también se tomó como entrada en el modelo, y ello es porque también se analizó la forma en que el individuo percibía esa magnitud de velocidad, así como sus cambios, una especie de retroalimentación. Además de dicha entrada, se consideraron las acciones que el conductor ejercía durante su tarea de manejo, a saber: ángulo del volante, par (torque) del volante, posición del acelerador y activación del freno.

Tabla 3.3: Estrategias de Manejo de Energía. Modelo Vehículo-Conductor.

ENTRADAS	SALIDAS
Ángulo del Volante [°]	Rapidez [Km/h]
Par (Torque) del Volante [N-m]	
Posición del Acelerador	
Acción de Frenado	
Rapidez [Km/h]	

3.3.2. Sistemas Avanzados de Asistencia a la Conducción

Como en el caso previo, tras la revisión de la literatura y de la identificación de potenciales áreas de contribución, las variables que más aportan información del fenómeno de ADASs, así como su papel en el modelo, se muestran en la Tabla 3.4. Se busca relacionar las acciones del conductor (entradas) con ciertas variables del vehículo (salidas). Para el caso de ADASs, lo que se busca es relacionar los movimientos hechos por el conductor con las variables del vehículo referentes a su maniobrabilidad, o en otras palabras, con las variables que describen su comportamiento lateral y rotacional.

Tabla 3.4: Sistemas Avanzados de Asistencia a la Conducción. Modelo Vehículo-Conductor.

ENTRADAS	SALIDAS
Ángulo del Volante [°]	Ángulo de Deslizamiento Lateral (en inglés, Slip Angle) o Ángulo de Viraje (en inglés, Yaw) [°]
Par (Torque) del Volante [N-m]	Ángulo de Balanceo (en inglés, Roll) [°]
Posición del Acelerador	Cambio del Ángulo de Viraje respecto al tiempo (en inglés, Yaw Rate) [°/s]
Acción de Frenado	Cambio del Ángulo de Cabeceo respecto al tiempo (en inglés, Pitch Rate) [°/s]
Rapidez [Km/h]	Cambio del Ángulo de Balanceo respecto al tiempo (en inglés, Roll Rate) [°/s]

3.4. Programación de los Modelos Vehículo-Conductor

Para esta fase de la Programación de los Modelos vehículo-Conductor se utilizó el lenguaje de programación *Python* a través del Entorno de Desarrollo Integrado (*IDE*, por sus siglas en inglés) de *Jupyter Hub*. Se mencionan una librería y una Interfaz de Programación de Aplicaciones (*API*, por sus siglas en inglés) que fueron sustanciales para el desarrollo de este rubro de la Metodología. La librería es la de *Tensorflow*, versión 2.6.1, la cual es una librería de código abierto enfocada al Aprendizaje Automático y desarrollada por Google, que ofrece mucha flexibilidad de trabajo, así como una amplia variedad de modelos ML para su aplicación a conjuntos de datos o *Datasets* (Tensorflow, 2021). Dentro de *Tensorflow* se encuentra una API, *Keras* versión 2.6.0, de carácter modular y flexible desarrollada inicialmente por *ONEIROS* (*Open-ended Neuro-Electronic Intelligent*

Robot Operating System), y la que tiene la finalidad de facilitar la experimentación con datos (Keras, 2021).

3.4.1. Unidades de Procesamiento Gráfico (GPUs)

Para poder cumplir con el objetivo de esta etapa se hizo uso del Clúster de Unidades de Procesamiento Gráfico (por sus siglas en inglés, *GPUs*) de CICATA-IPN-QRO. El Clúster cuenta con cuatro GPUs NVIDIA Titan X con Arquitectura Pascal y 12 GB de Memoria Ram de Video (o Memoria VRAM). Para este trabajo se emplearon dos GPUs.

3.4.2. Arreglo Experimental

Una vez que ya se definió el hardware que se utilizó, así como las librerías y APIs más importantes, se procede a la delimitación del arreglo experimental o la forma de división de las pruebas. Rememorando el rubro de *Descripción de la Base de Datos* (sección 3.1), hay que recalcar que, para cada individuo, se extrajeron diez pruebas, cinco en sentido AH y cinco en sentido H. Entonces, en este punto lo que se hizo es lo siguiente.

Se diseñó un algoritmo en python de manera que seleccionara aleatoriamente de entre los diez ensayos por sujeto, dos pruebas (una AH y una H) para conformar el conjunto de prueba. Las pruebas restantes ingresaron al proceso de validación cruzada con $K=8$ por lo señalado en [91]. *No se manejó la técnica tradicional de división entrenamiento-validación-prueba por porcentajes debido a las escasas pruebas que se tenían, por ello se optó por la técnica de validación cruzada K-folds.*

Una ventaja de la validación cruzada K-Folds es que permite combatir y mitigar el sobreajuste (u overfitting), situación que para el caso de la presente tesis es sustancial.

Para esta sección de la metodología, se extrajeron y manipularon las variables que constituyen los Modelos Vehículo- Conductor (entradas y salidas), ello para ambos escenarios.

3.4.3. Normalización y Estandarización de los Datos

Como las variables de los Modelos Vehículo-Conductor correspondientes al conjunto de entrenamiento tienen diferentes rangos, hay que normalizarlas o escalarlas en el rango $[-1,1]$ *por variable*, por lo expuesto en la literatura y porque en los datos hay valores tanto positivos como negativos. La función empleada para este propósito es *MinMaxScaler()* de la librería *Scikit learn*. Además, también debe aplicarse una estandarización a los mismos *por variable* para tener una media de cero y una desviación estándar de uno. Lo anterior se hace mediante la función *StandardScaler()* de la misma librería. Ambos procesos se realizan por variable, puesto que cada variable presenta una dinámica en particular; aplicar estas estrategias a la información considerándola como un todo conllevaría a la pérdida de importantes relaciones dentro y entre las mismas.

Es de suma importancia hacer notar que el conjunto de prueba también se normalizó y estandarizó, pero con base en los parámetros obtenidos del conjunto de entrenamiento (no del mismo de prueba); no se pueden obtener nuevas medias y desviaciones estándar por variable para el caso de la estandarización, o nuevos rangos máximo-mínimo por parámetro para el caso de la normalización, puesto que ello sesgaría el conjunto de prueba.

3.4.4. Procesamiento de las Series de Tiempo

Este punto es sumamente importante para los dos escenarios tratados en esta tesis, *Estrategias de Manejo de Energía y Sistemas Avanzados de Asistencia a la Conducción*.

Todas las variables (entradas y salidas) de los Modelos Vehículo-Conductor, para ambos escenarios, constituyen series de tiempo, pues representan la evolución de una señal tomada con el transcurso del tiempo. Para poder ingresar esas señales a las técnicas de ML, se les debe aplicar un proceso especial, que consiste en organizarlas según ciertos parámetros. Aquí aparecen tres parámetros muy importantes por definir: 1. *Horizonte de predicción de las salidas* (en inglés *Look Ahead Times*), que se refiere al número de pasos de tiempo (o número de muestras) que de una o varias salidas se pretende predecir; 2. *Historial de las entradas* (en inglés *Look Back Times*), que alude al número de pasos de tiempo (o número de muestras) que de las entradas se quiere tomar en consideración; 3. *Desfase entre secuencias* (en inglés *Shift*), que se define como el desajuste entre secuencias entrada-salida consecutivas. En la Figura 3.7 se muestra un ejemplo ilustrativo del proceso para una entrada y una salida. En el caso de la primera se maneja un historial de 23 pasos de tiempo (o muestras), y para la salida un horizonte de predicción del mismo tamaño que el historial. El desfase es de dos, lo que indica que tras una secuencia entrada-salida, la siguiente va desfasada dos pasos de tiempo (o muestras) respecto a la previa (como se puede ver en la misma Figura). Esa misma lógica se puede extender para un caso de múltiples entradas y múltiples salidas.

De forma genérica, para el caso de *múltiples entradas-múltiples salidas*, después del proceso, el tamaño de las entradas es de:

$$\text{Entradas} : [\text{Tamaño Lote}, \text{Historial}, \text{Número Entradas}] \quad (3.1)$$

y de las salidas es de:

$$\text{Salidas} : [\text{Tamaño Lote}, \text{Horizonte Predicción}, \text{Número Salidas}] \quad (3.2)$$

Tamaño de Lote se calcula de la siguiente manera:

$$\text{Tamaño Lote} = \text{Número Muestras} - \text{Historial} - \text{Horizonte} + 1. \quad (3.3)$$

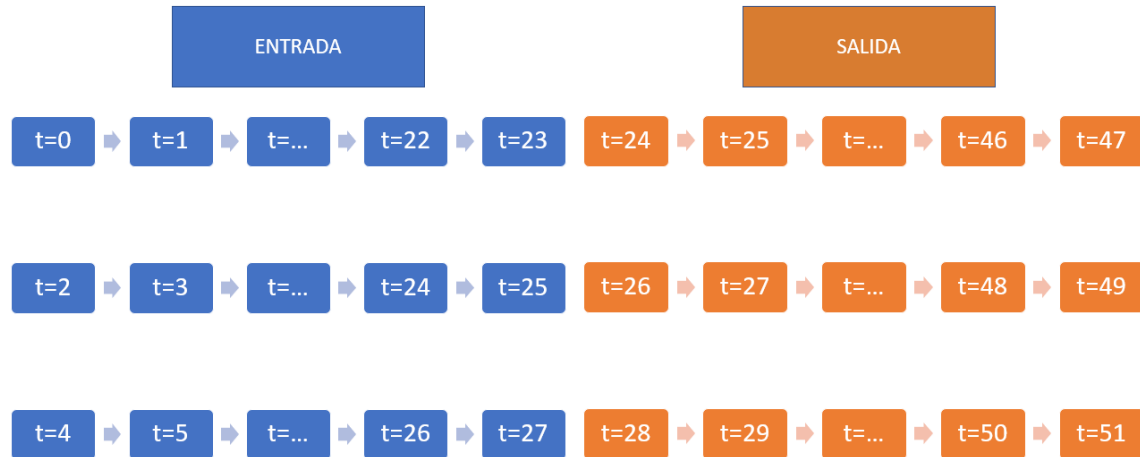


Figura 3.7: Procesamiento de las Series de Tiempo. Caso 1 Entrada - 1 Salida con un Historial (Entrada) y Horizonte (Salida) de 23 pasos, además de un desfase de 2 pasos.

Finalmente, el *Número de Muestras* se refiere al tamaño del conjunto de datos de que se trate (conjunto de entrenamiento o de prueba).

Para la presente tesis este rubro es sustancial, puesto que se busca el analizar cómo, fijado un horizonte de predicción y usando una determinada técnica ML, el desempeño de la técnica va cambiando al variar el historial, de manera que se encuentre la combinación historial-horizonte que maximice el desempeño y minimice el costo computacional; ello por escenario, sujeto y técnica ML.

Sistemas Avanzados de Asistencia a la Conducción

Para esta perspectiva se analizaron las combinaciones historial-horizonte que se muestran en la Tabla 3.5. Es muy importante mencionar que el desfase se manejó de 1, por lo que 100 muestras o pasos de tiempo (ya sea en horizonte o historial) representan 1 s.; o de modo análogo, cada paso de tiempo o muestra representa 0.01 s. (ello por la frecuencia de muestreo de los equipos de adquisición).

Estrategias de Manejo de Energía

Para este escenario se analizaron las combinaciones historial-horizonte que se modelan en la Tabla 3.6. Cabe destacar que el horizonte de predicción que se utilizó fue la tercera parte de la mínima duración de todas las pruebas para todos los conductores, a saber la tercera parte de 166.59 s. (i.e. 49 s.). La razón de la tercera parte de la prueba más corta se fundamenta en métodos heurísticos. Otro hecho muy importante de aclarar para este escenario es que el desfase no se pudo fijar en 1, puesto que, debido al amplio horizonte de predicción, se saturaba la memoria VRAM de video de la GPU (12 GB). El horizonte de predicción es mucho más largo que el máximo de ADASs (49 s. frente a 3 s., respectivamente). Ello desembocó en que el desfase se fijara en 100, lo que quiere decir que *de cada señal del Modelo Vehículo-Conductor se tomó un valor cada 100 muestras, es decir, se tomó*

Tabla 3.5: Combinaciones Historial-Horizonte para la perspectiva de Sistemas Avanzados de Asistencia a la Conducción.

Horizonte	Historial (Proporción con respecto al Horizonte)	Desfase	Caso
100 pasos de tiempo / muestras (1.0 s.).	0.6 (60 pasos ó 0.6 s.).	1	C1
	0.8 (80 pasos ó 0.8 s.).		C2
	1.0 (100 pasos ó 1.0 s.).		C3
	2.0 (200 pasos ó 2.0 s.).		C4
	3.0 (300 pasos ó 3.0 s.).		C5
200 pasos de tiempo / muestras (2.0 s.).	0.6 (120 pasos ó 1.2 s.).		C6
	0.8 (160 pasos ó 1.6 s.).		C7
	1.0 (200 pasos ó 2.0 s.).		C8
	2.0 (400 pasos ó 4.0 s.).		C9
	3.0 (600 pasos ó 6.0 s.).		C10
300 pasos de tiempo / muestras (3.0 s.).	0.6 (180 pasos ó 1.8 s.).		C11
	0.8 (240 pasos ó 2.4 s.).		C12
	1.0 (300 pasos ó 3.0 s.).		C13
	2.0 (600 pasos ó 6.0 s.).		C14
	3.0 (900 pasos ó 9.0 s.).		C15

un valor cada segundo, no 100 como en el caso de ADASs. El ajuste anterior se hizo para poder correr los códigos superando las limitaciones del Hardware.

Tabla 3.6: Combinaciones Historial-Horizonte para la perspectiva de Estrategias de Manejo de Energía.

Horizonte	Historial (Proporción con respecto al Horizonte)	Desfase	Caso
49 pasos de tiempo / muestras (49 s.).	0.6 (29 pasos ó 29 s.).	100	C1
	0.8 (39 pasos ó 39 s.).		C2
	1.0 (49 pasos ó 49 s.).		C3
	2.0 (98 pasos ó 98 s.).		C4

3.4.5. Técnicas de Aprendizaje Automático

En este rubro se definen las Técnicas ML implementadas para ambos escenarios, así como sus parámetros respectivos.

Estrategias de Manejo de Energía

Para este escenario se muestran en la Tabla 3.7 las técnicas ML que se usaron para programar el Modelo Vehículo-Conductor en python, además de sus parámetros. Para todos esos algoritmos ML se emplearon las siguientes métricas en común, a saber: optimizador *Adam*, función de pérdida *MSE*, 6 épocas y un tamaño de lote de 32.

Para evitar el fenómeno de sobreajuste (*overfitting*), se usó regularización por dilución (*dropout*) en los tres modelos ML propuestos.

Tabla 3.7: Técnicas de aprendizaje automático para el escenario de Estrategias de Manejo de Energía.

Técnica ML	Estructura/ Configuración	Función de Activación
LSTM Vainilla	Capa Entrada: [Tamaño Lote, Historial, Número de Entradas]. 2 Capas de 300 Unidades. Regularización por Dilución (Dropout) de 0.2. Capa Salida-Densa: [Horizonte, Número de Salidas].	Tangente Hiperbólica. Lineal (Salidas).
GRU	Capa Entrada: [Tamaño Lote, Historial, Número de Entradas]. 2 Capas de 300 Unidades. Regularización por Dilución (Dropout) de 0.2. Capa Salida-Densa: [Horizonte, Número de Salidas].	
Transformer	Capa Entrada: [Tamaño Lote, Historial, Número de Entradas]. Atención Multi-cabeza de 3 capas con 3 elementos: Llaves (Keys): 96. Valores (Values): 96. Consulta (Query): 96. Regularización por Dilución (Dropout) de 0.2. Capa de Normalización. Capa Convolutiva 1D de 96 filtros con kernel de 1. Capa Convolutiva 1D de 7 filtros con kernel de 1. Capa de Dilución de 0.2. Capa de Normalización. Capa Salida-Densa: [Horizonte, Número de Salidas].	ReLU. Lineal (Salidas).

Sistemas Avanzados de Asistencia a la Conducción

De igual manera, se muestran en la Tabla 3.8 las técnicas ML que se emplearon para programar el Modelo vehículo-Conductor en python, así como sus parámetros. Para todos esos algoritmos ML se emplearon las siguientes métricas en común, a saber: optimizador *Adam*, función de pérdida *MSE*, 6 épocas y un tamaño de lote de 32.

De igual manera que para EMSs, para evitar el sobreajuste (*overfitting*), se utilizó una regularización por dilución (*dropout*) en las técnicas de ML, como se indica en la Tabla 3.8.

Tabla 3.8: Técnicas de Aprendizaje Automático para el escenario de Sistemas Avanzados de Asistencia a la Conducción.

Técnica ML	Estructura/ Configuración	Función de Activación
LSTM Vainilla	Capa Entrada: [Tamaño Lote, Historial, Número de Entradas]. 1 Capa de 300 Unidades. Regularización por Dilución (Dropout) de 0.2. Capa Densa-Salida: [Horizonte, Número de Salidas].	
GRU	Capa Entrada: [Tamaño Lote, Historial, Número de Entradas]. 1 Capa de 300 Unidades. Regularización por Dilución (Dropout) de 0.2. Capa Densa-Salida: [Horizonte, Número de Salidas].	
LSTM Codificador- Decodificador (LSTM Encoder- Decoder)	Capa Entrada: [Tamaño Lote, Historial, Número de Entradas]. 1 Capa de 300 Unidades (Encoder). Regularización por Dilución (Dropout) de 0.2. 1 Capa de Repetición de Vectores (Horizonte). 1 Capa de 300 Unidades (Decoder). Regularización por Dilución (Dropout) de 0.2. Capa Distribución de Tiempo-Densa (Número de Salidas).	Tangente Hiperbólica. Lineal (Salidas).

3.5. Comparación de Técnicas de Aprendizaje Automático para la delimitación del mejor exponente del Modelo Vehículo-Conductor

Esta fase de la Metodología General se refiere a la evaluación de los algoritmos ML ya entrenados ante el conjunto de prueba y a la elección de la técnica ML que mejor representa el Modelo vehículo-Conductor, basándose en los criterios de desempeño, para el sujeto y para el escenario estudiado.

3.5.1. Criterios de desempeño para la elección del mejor exponente del Modelo Vehículo-Conductor

Para la perspectiva de Estrategia de Manejo de Energía se considera como mejor exponente del Modelo Vehículo-Conductor para un determinado sujeto el algoritmo ML con el menor RMSE o mayor R^2 , además de la menor carga computacional. Generalmente, el RMSE y el R^2 van hermanados y son inversamente proporcionales. Mientras más alto el RMSE, más bajo el R^2 , y viceversa. En caso de que dos algoritmos tengan valores de los criterios muy similares, se optará por el de menor complejidad o de arquitectura más simple.

Para la perspectiva de Sistemas Avanzados de Asistencia a la Conducción, se tienen cinco salidas en el Modelo Vehículo-Conductor, por lo que una opción sería obtener el RMSE y el criterio R^2 por cada salida y promediarlo, pero se optó mejor por obtener la norma L2 de RMSE y la norma L2 del Criterio R^2 . De esta manera, para la perspectiva de Sistemas Avanzados de Asistencia a la Conducción el mejor exponente del Modelo Vehículo-Conductor es la técnica ML con la menor norma L2 de RMSE o la mayor norma L2 de R^2 , además de la menor complejidad computacional.

En términos prácticos, se hará uso del *Criterio R^2* como parámetro fundamental de identificación

del desempeño del algoritmo ML; no obstante, en los Anexos se podrán consultar los resultados tanto del RMSE como del R^2 para todos los sujetos y ambas perspectivas.

3.5.2. Procedimiento de comparación

De manera específica, en esta etapa de la tesis se realiza el siguiente procedimiento:

1. Se considera un determinado escenario y cierto sujeto para su análisis y estudio.
2. Se elige una técnica ML y se halla la *mejor combinación historial-horizonte* dentro de dicha técnica usando los criterios de desempeño descritos previamente. Este proceso se repite para cada algoritmo ML que se tenga.
3. Una vez que se tienen los mejores *exponentes* de cada técnica ML, se procede a la comparación entre los anteriores para hallar el *mejor algoritmo ML*. De igual manera, se usan los criterios de desempeño. este punto ya se tiene la mejor Técnica ML que representa el Modelo Vehículo-Conductor para cierto sujeto y escenario.
4. El procedimiento previo se repite para los cuatro sujetos y los dos escenarios, pues recuérdese que los Modelos Vehículos-Conductor son personalizados y según dos enfoques.

Esta última fase propicia la formulación de las conclusiones del trabajo y de las perspectivas de la tesis.

3.6. Sumario

Este capítulo describió los pasos a seguir para la culminación de los objetivos de la presente tesis para ambas perspectivas. Es un capítulo largo, pero trataron de abordarse todos los temas, todos los detalles, de manera que fuera comprensible la forma en que se obtuvieron los resultados de los Modelos Vehículo-Conductor para los ámbitos de ADASs y EMSs. Para los dos escenarios se sigue la misma metodología de cinco pasos, sólo hay unas pequeñas variaciones en ciertos rubros para cada uno (a las cuales se les dio aclaración).

En los siguientes dos capítulos se expondrán los resultados de la aplicación de las técnicas de Inteligencia Artificial a los Modelos Vehículo-Conductor para ambas perspectivas, primero EMSs y después para ADASs.

Capítulo 4

Análisis de Resultados. Estrategias de Manejo de Energía

En este capítulo se expondrán los resultados concernientes a las técnicas ML empleadas para la representación del Modelo Vehículo-Conductor para cada sujeto desde la perspectiva de Estrategias de Manejo de Energía. En este escenario se toman como entradas del Modelo Vehículo-Conductor el ángulo del volante, torque (o par) del volante, posición del acelerador, activación del freno y la rapidez, y como salida sólo la rapidez.

Es importante aclarar que por cuestiones de la extensión del documento y por la variedad de las simulaciones elaboradas, sólo se presentarán los resultados de la técnica ML que mejor representó el Modelo Vehículo-Conductor para cierto escenario y determinado sujeto. Los resultados para los demás casos se pueden ver en el Anexo A *Análisis de Resultados. Estrategias de Manejo de Energía*. En dicha tabla, de cada sujeto se destaca el mejor caso y la mejor técnica ML mediante un sombreado para una mejor visualización.

4.1. Sujeto 1

Para este sujeto, el Modelo Vehículo-Conductor quedó mejor representado por el modelo *LSTM Vainilla* con un horizonte de predicción de 49 s. y un historial de 39 s. (*Caso 2*). El RMSE reportado fue de 0.033 con un R^2 de 0.993, algo muy cercano al valor ideal de 1.0. En palabras más intuitivas, el modelo representado por el caso 2 de LSTM Vainilla es capaz de arrojar, con las entradas empleadas, una buena predicción de salida comparada con su valor verdadero. En la Figura 4.1 se muestra el grado de predicción del modelo y se compara con el valor de la salida proporcionado por el conjunto de prueba (o **ground truth**).

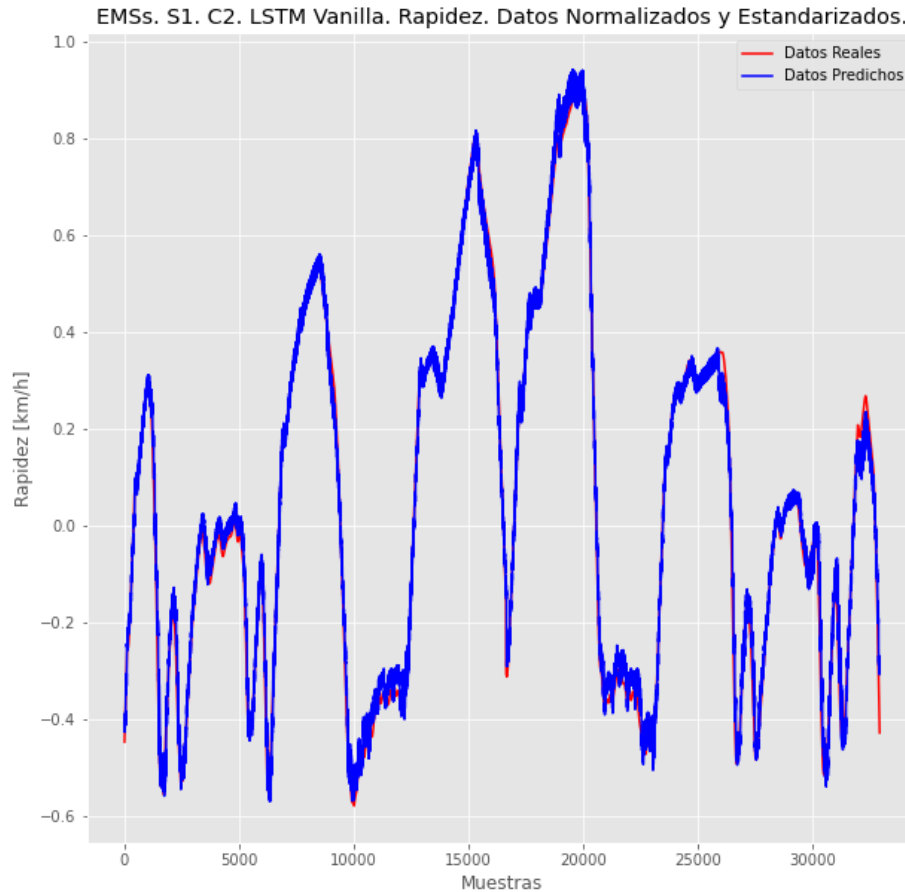


Figura 4.1: Estrategias de Manejo de Energía. Sujeto 1. Rapidez.

Otra situación muy interesante es que el sujeto 1 posee una *conducción técnica o educada*. Ello implica que conoce las capacidades del vehículo y trata de maximizarlas. A pesar de y con ello, el nivel de predicción del algoritmo ML fue sumamente bueno, lo que refleja la gran relación que hay entre las acciones del conductor (entradas del Modelo Vehículo-Conductor) elegidas en la presente tesis y la velocidad del vehículo (salida del Modelo Vehículo-Conductor).

4.2. Sujeto 2

Para el sujeto 2, un conductor con un estilo de manejo más cauto (lo que se percibe en la gráfica del perfil de velocidad), el modelo *LSTM Vainilla Caso 1* fue el mejor exponente del Modelo Vehículo Conductor. El RMSE y el R^2 reportados fueron de 0.028 y 0.995, respectivamente, valores sumamente buenos en el nivel de predicción; el R^2 reportado es muy cercano al 1.0 ideal.

En la Figura 4.2 se puede comparar la salida del algoritmo ML con el valor verdadero de la variable *Rapidez*. Como se puede percatar, parece que ambas gráficas están traslapadas, lo que quiere decir que los datos se parecen mucho entre sí y el algoritmo está haciendo bien su función.

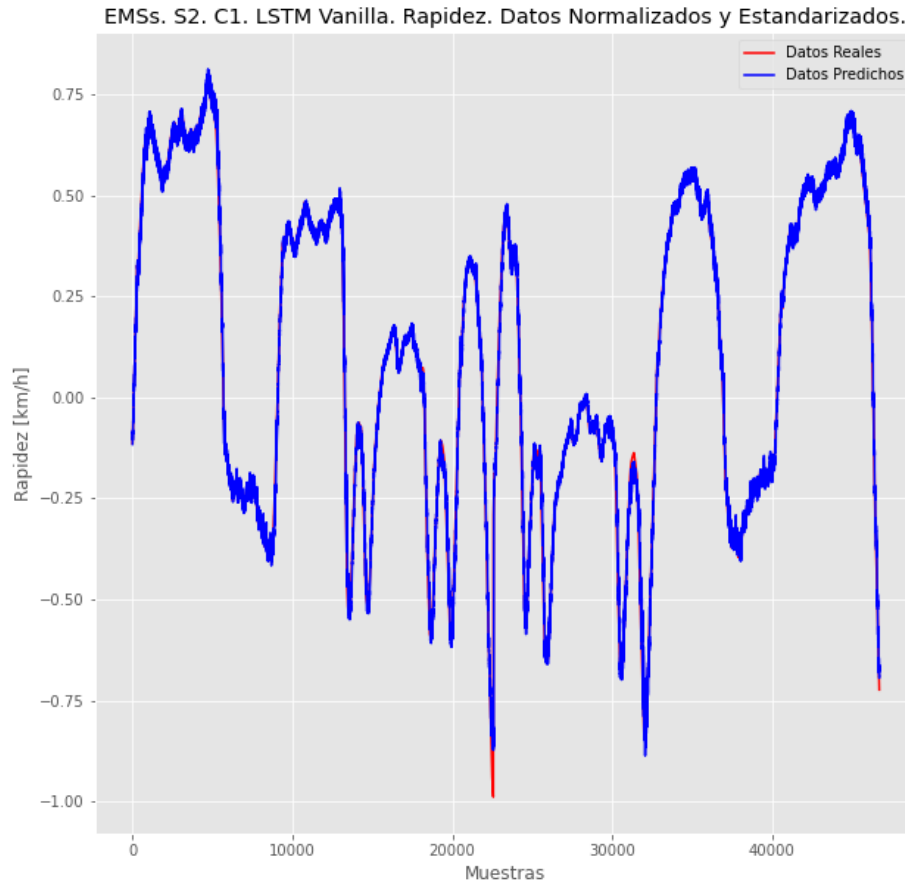


Figura 4.2: Estrategias de Manejo de Energía. Sujeto 2. Rapidez.

De la misma manera que para el sujeto 1, las entradas del Modelo Vehículo-Conductor para EMSs tienen una relación muy marcada con el perfil de velocidad (o rapidez); en otras palabras, las acciones del conductor tienen mucho que ver con la velocidad del vehículo.

4.3. Sujeto 3

Para el sujeto 3, el modelo *LSTM Vanilla Caso 1* fue el algoritmo ML que mejor representó el Modelo Vehículo-Conductor. Lo reportado por las simulaciones es un R^2 y un RMSE de 0.990 y 0.045, respectivamente. De la misma manera que los sujetos anteriores, las predicciones son sumamente buenas, muy cercanas a lo ideal. Si uno ve la Figura 4.3 cuesta el poder discernir los valores predichos por el algoritmo ML de los valores reales proporcionados por el conjunto de prueba, que es lo que se espera.

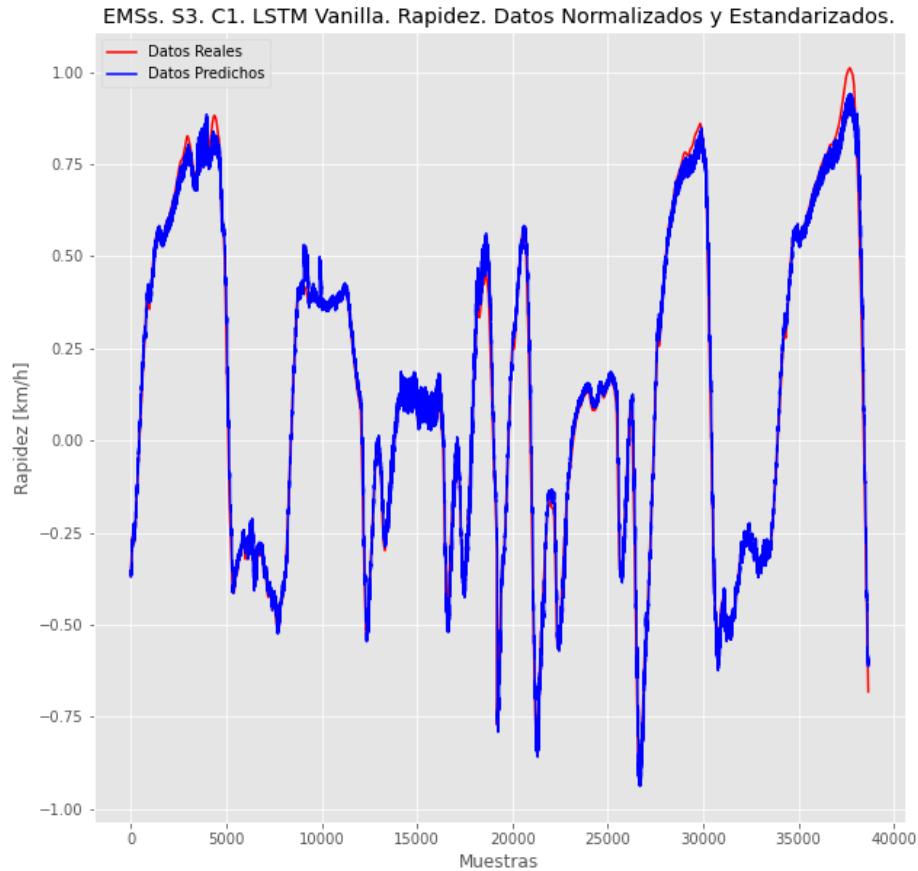


Figura 4.3: Estrategias de Manejo de Energía. Sujeto 3. Rapidez.

Otro hecho sumamente interesante, aplicable para este y todos los sujetos, es que *el algoritmo ML es capaz de determinar la velocidad del vehículo, dadas ciertas acciones del conductor, de manera sobresaliente independientemente de la maniobra de que se trate. Para todas las maniobras el algoritmo ML predice notablemente el perfil de velocidad.*

4.4. Sujeto 4

Para este último sujeto, la tendencia se repitió, quedando como mejor modelo ML el *LSTM Vainilla Caso 2*, con un RMSE y un R^2 de 0.023 y 0.994 de manera respectiva. La predicción es sumamente buena y eso se ve en la Figura 4.4.

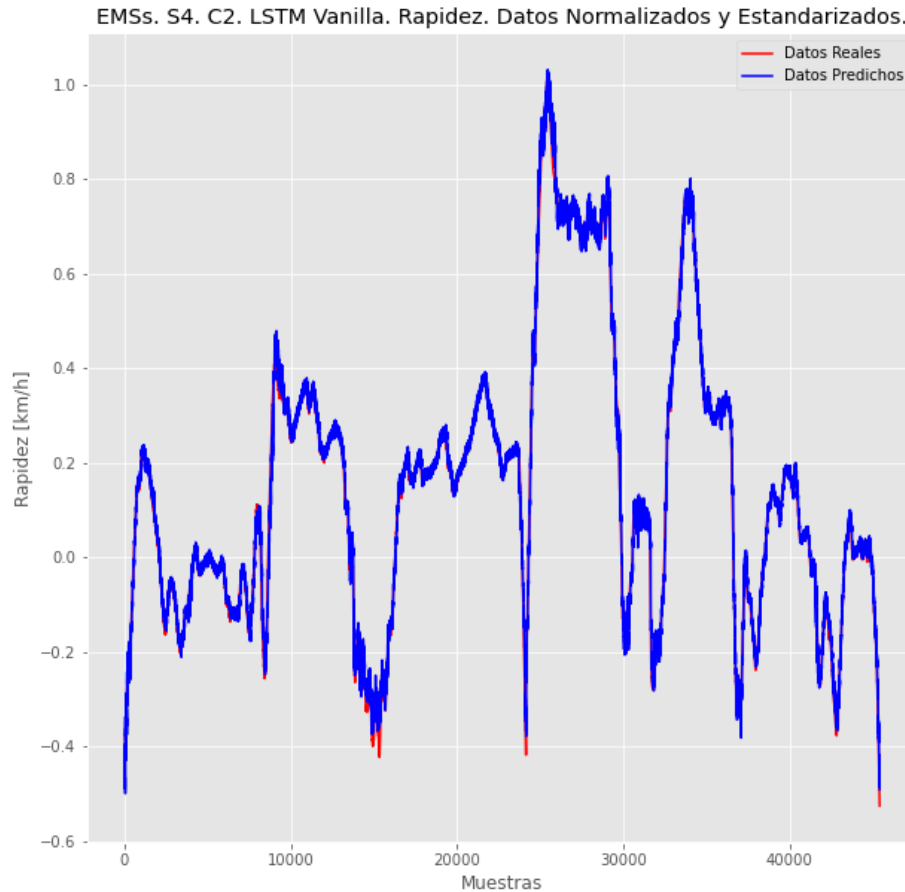


Figura 4.4: Estrategias de Manejo de Energía. Sujeto 4. Rapidez.

Otra situación que conviene comentarse acerca de este conductor es que su estilo de conducción está caracterizado por acciones precipitadas, a diferencia del sujeto 1 que tiene una conducción técnica. Hay que diferenciar entre los estilos de conducción de estos dos sujetos. Independientemente de que su forma de conducción sea más abrupta que los sujetos 2 y 3, el algoritmo ML halla muy buenas relaciones entre los datos, lo que se demuestra en la Figura 4.4.

4.5. Discusión de Resultados

Esta perspectiva de trabajo, Estrategias de Manejo de Energía, se cierra haciendo las siguientes deducciones:

1. De manera contundente, el modelo ML que mejor representó el Modelo Vehículo-Conductor para esta perspectiva y para todos los sujetos fue la *LSTM Vanilla*. Ello se defiende por el hecho del manejo de sus tres compuertas de información, lo que le permite el olvido y el recuerdo de secuencias temporales de una manera eficiente. El Transformer es un algoritmo muy poderoso que se pensaría que obtendría los mejores resultados, pero no fue así; ello se deduce porque

sólo se ocupó la parte del encoder de la arquitectura propuesta en [122]. Formalmente, entre los tres modelos ML propuestos en esta tesis, no hay una mejora drástica de alguno de ellos respecto a los otros, todos tuvieron desempeños sobresalientes, lo que indica la robustez del Modelo Vehículo-Conductor propuesto.

2. En los sujetos 1 y 4, correspondientes a los sujetos con un estilo de conducción más técnico o de acciones precipitadas, respectivamente, el mejor algoritmo estuvo representado por el caso 2 y tiene sentido físico, puesto que su comportamiento es un poco más variable que para los sujetos 2 y 3, en los que el caso 1 fue la mejor representación del Modelo Vehículo-Conductor. En otras palabras, se necesita conocer más información de la conducción de los sujetos 1 y 4 para poder predecir el perfil de velocidad, por ello un caso 2, a diferencia de los otros sujetos con un perfil de velocidad más predecible por su manejo cauto (caso 1).
3. A pesar de que intuitivamente se pensaría que, fijado el horizonte de predicción, mientras más grande sea el historial (mientras más datos se tomen hacia atrás), mejor, eso no siempre se cumple, pues se puede dejar de recordar las cosas importantes y darle relevancia a lo que no lo es, ello por la gran cantidad de datos que se están tomando en el historial. Por lo que *más datos en consideración en el historial no necesariamente implica una mejor calidad del horizonte de predicción.*
4. Los sujetos 1 y 4 mostraron un comportamiento más “abupto” que los conductores 2 y 3. A pesar de ello, los modelos lograron unas predicciones del perfil de velocidad muy notables.

4.6. Sumario

Este capítulo se cierra haciendo notar los destacables resultados que se obtuvieron para esta perspectiva, Estrategias de Manejo de Energía, y no sólo para los mejores exponentes de cada persona. De manera general, para todos los casos y todos los modelos ML, se obtuvieron resultados muy buenos por sujeto, valores R^2 por encima de 0.900. Ya los mejores exponentes de cada sujeto superaban el 0.980.

Otra situación notable es que el modelo ML que representó mejor los Modelos Vehículo-Conductor para los cuatro sujetos, fue la arquitectura *LSTM Vainilla*. Se intuyen los mejores resultados por el manejo de las tres compuertas: entrada, salida y olvido, lo que impide que se abandone lo útil y se conserve la información que no sirve. A pesar de ello, las variaciones con respecto a los otros algoritmos ML fueron mínimas.

Capítulo 5

Análisis de Resultados. Sistemas Avanzados de Asistencia a la Conducción

Este escenario es más elaborado que el de Estrategias de Manejo de Energía, ya que el Modelo Vehículo-Conductor es más complejo. *En este escenario se toman como entradas del Modelo Vehículo-Conductor a el ángulo del volante, torque (o par) del volante, posición del acelerador, activación del freno y la rapidez, y como salidas el ángulo de viraje (ángulo yaw) / ángulo de deslizamiento lateral (slip angle), ángulo de balanceo (o ángulo roll), tasa de cambio del ángulo de viraje (o yaw rate), tasa de cambio del ángulo de cabeceo (o pitch rate) y tasa de cambio del ángulo de balanceo (o roll rate).*

Es importante aclarar que por cuestiones de la extensión del documento, variedad de simulaciones y complejidad del Modelo Vehículo-Conductor, sólo se presentarán gráficamente los resultados de la técnica ML que mejor representó dicho modelo en el horizonte de predicción más amplio, 3 s. (Casos 11-15) para cierto escenario y determinado sujeto. Los resultados para los demás casos y para los horizontes de predicción de 1 y 2 s. (Casos 1-10) se pueden consultar en el Anexo B *Análisis de Resultados. Sistemas Avanzados de Asistencia a la Conducción*. Las celdas amarillas-naranjas indican la técnica y caso con mejor representación del Modelo Vehículo-Conductor para el horizonte de predicción de 1 s.; las de color verde, para el de 2 s.; y las de color azul, para el horizonte de predicción de 3 s.

La elección de 3 s. fue porque se obtuvieron resultados interesantes para 1 s. y 2 s. de ventana de predicción (si se tienen dudas al respecto, consúltese el Anexo B), por lo que se optó por ampliar ésta; no obstante, más allá de 3 s. no se pudo evaluar por limitaciones de hardware (saturación de

los 12 GB. de Memoria VRAM).

Con base en los criterios de desempeño expuestos en una sección previa, el mejor modelo ML (con su respectivo caso) que representó el Modelo Vehículo-Conductor para cierto sujeto está dado por aquél que tiene la arquitectura más simple y el mayor valor de la norma L2 de R^2 considerando un horizonte de predicción fijo. De esta manera, a continuación se presentan los resultados para todos los individuos.

5.1. Sujeto 1

Antes de empezar la discusión de resultados acerca de este sujeto, vale la pena señalar que éste es un individuo con conducción técnica, especializado en la desestabilización de vehículos.

Para el horizonte de predicción de 3 s., la técnica ML con mejores resultados fue la GRU del Caso 14, es decir, *GRU con relación historial-horizonte de 6 s. - 3 s. (600 pasos de tiempo hacia atrás-300 pasos de tiempo hacia adelante)*. La técnica GRU del caso 14 tuvo la mayor norma L2 de R^2 de los casos de horizonte de 3 s. (casos 11-15), a saber: 1.563. Como el escenario de ADASs es de múltiples salidas, conviene ver el comportamiento de cada salida de manera particular.

5.1.1. Ángulo de Viraje (Ángulo Yaw) o Ángulo de Deslizamiento Lateral (Slip Angle)

Esta variable tuvo un R^2 bajo, llegando a un valor de 0.385. El valor ideal de R^2 es 1.0 como se declara en [5], y dicho valor muestra la proporción de la variación en la variable de respuesta que se explica por los predictores [101]. O, el R^2 indica qué tan cerca están los datos de la línea de regresión ajustada [86]. A pesar de que el modelo ML no fue capaz de obtener un valor alto del parámetro respectivo, en la Figura 5.1 se puede ver cómo la predicción sí sigue o *trackea* la señal objetivo.

Un punto muy importante de señalar es que esta variable de salida, el ángulo de deslizamiento lateral (o slip angle) es un factor que cambia con muy alta frecuencia en el transcurso de la conducción. En otras palabras, un pequeño cambio de las acciones del conductor se ve reflejado de manera inmediata en el comportamiento del vehículo. Si a ello se le añade que el conductor tenía un manejo técnico, ya es más razonable el por qué el modelo no llegó a las amplitudes de la señal objetivo (o ground truth). Como el criterio R^2 está más relacionado a las amplitudes de las señales, ello explica el por qué del valor bajo.

Hasta aquí parece que funcionó mal el modelo ML para este sujeto, pero hay que ir más allá. Si uno ve la Figura 5.1, se puede percatar que no hubo desfase de la predicción respecto al ground truth. En otras palabras, si la maniobra que se presentaba en determinado instante de tiempo era la Prueba Slalom (zig-zag), la predicción debía estimar dicha maniobra y no otra, como recta o vuelta a la derecha. Y ello lo hace muy bien el modelo ML, entonces, como trabajo futuro, se debe

buscar otro tipo de criterio que, además de la amplitud de las señales, tome en consideración las componentes frecuenciales.

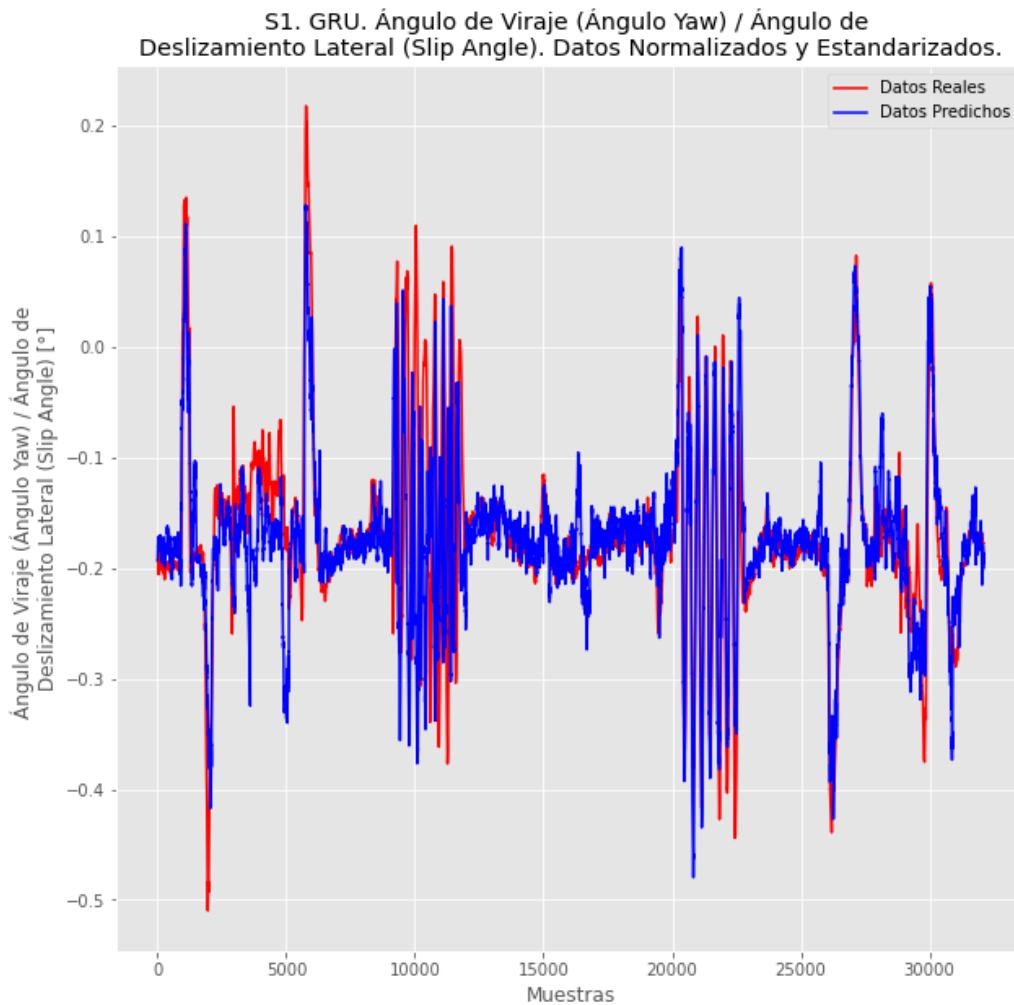


Figura 5.1: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 1. Ángulo de Viraje (Ángulo Yaw) o Ángulo de Deslizamiento Lateral (Slip Angle).

5.1.2. Ángulo de Balanceo (Ángulo Roll)

El Ángulo de Balanceo (Ángulo Roll) tiene un comportamiento muy bueno al alcanzar un R^2 de 0.928. Incluso, se esperarían muy buenos resultados de esta variable para horizontes de predicción mucho más amplios, si se pudieran alcanzar.

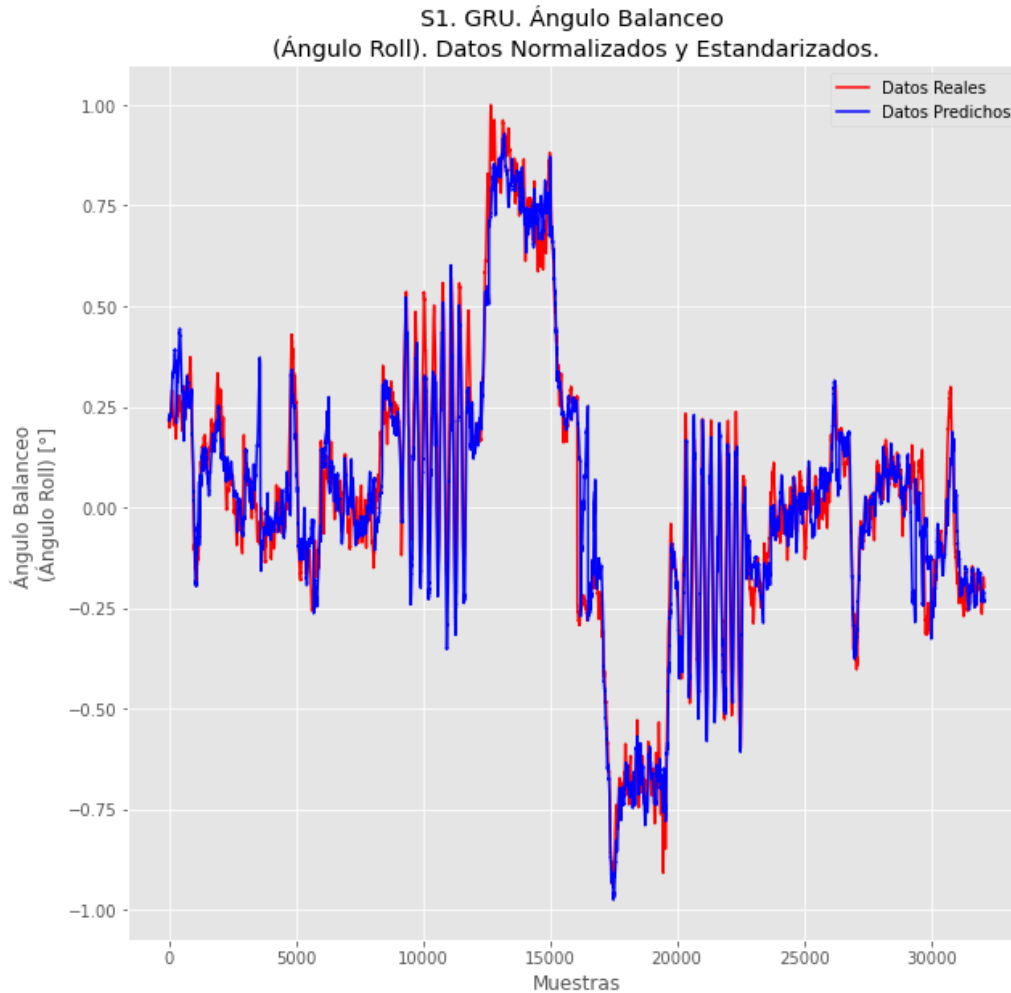


Figura 5.2: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 1. Ángulo de Balanceo (Ángulo Roll).

La gráfica de esta variable se muestra en la Figura 5.2. Para este caso, se observa cómo la predicción y la señal objetivo son muy parecidas, lo que habla de un buen modelo ML que representa el modelo vehículo-conductor en lo concerniente al ángulo de balanceo (ángulo roll).

El ángulo de balanceo (o ángulo roll) es una variable vehicular que cambia con baja frecuencia durante la tarea de conducción. Es por ello que, a pesar de que este individuo tenía un conducción técnica, ello no se vió reflejado en la estimación de este ángulo. De la misma manera que para el caso del ángulo de deslizamiento lateral, hay un claro respeto a las diferentes maniobras que se efectuaron. Se admite que en algunas partes de la gráfica, las amplitudes de la predicción no llegaron a las de la señal objetivo, pero fueron mínimas. Lo que más importa para la perspectiva de ADASs es la correcta identificación de maniobras, y eso lo hizo bien el modelo ML.

5.1.3. Tasa de cambio del Ángulo de Viraje (o Yaw Rate)

Para esta señal, igual que para la previa, la técnica GRU en el caso 14 fue muy buena al lograr un R^2 de 0.861. De manera semejante al ángulo roll, se esperarían muy buenos resultados si el horizonte de predicción se incrementara. La señal se puede ver de manera ilustrativa mediante la Figura 5.3.

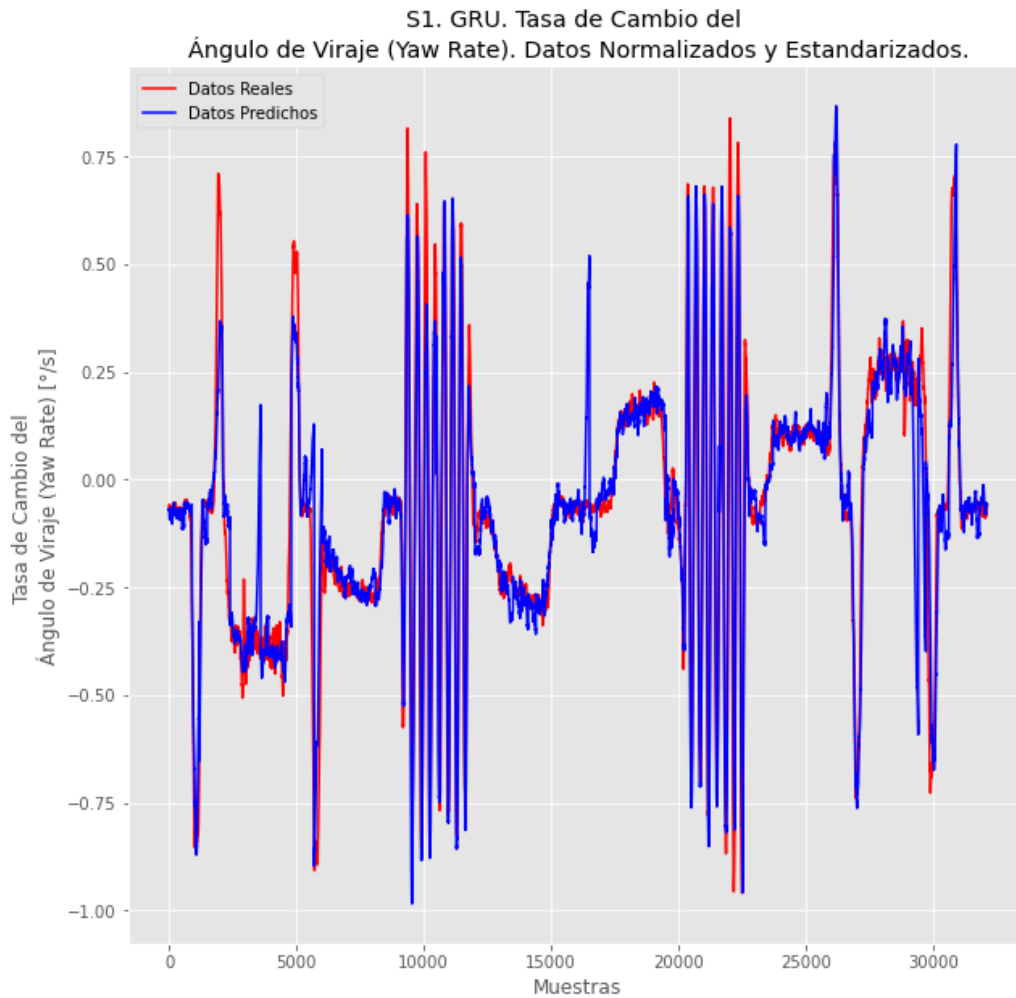


Figura 5.3: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 1. Tasa de cambio del Ángulo de Viraje (o Yaw Rate).

En la Figura 5.3 se muestra cómo las maniobras ocurrieron en el momento adecuado como lo marca el ground truth. Se deduce que esta variable no tiene tan alta frecuencia como su integral, el ángulo de deslizamiento lateral (o slip angle); de ahí que fuera predicha de muy buena manera a pesar del estilo de conducción de este sujeto. No obstante, en ciertas zonas de la gráfica las amplitudes de la predicción quedaron cortas con respecto a la señal objetivo.

5.1.4. Tasa de cambio del Ángulo de Cabeceo (o Pitch Rate)

Acerca de esta variable resulta muy interesante mencionar que, si bien el ángulo pitch no se pudo medir con el VBOX3i por la disposición de las dos antenas, se tenía cierta incertidumbre con respecto a la tasa de cambio del ángulo respectivo. Por la forma de la señal, que no es la que se espera recabar de una medición, se intuye que es ruido. A pesar de la aleatoriedad de la señal, el valor de R^2 logrado es de 0.326. Obviamente no cubre todas las señales del ground truth, pero sí hay un seguimiento de forma, como se muestra en la Figura 5.4.

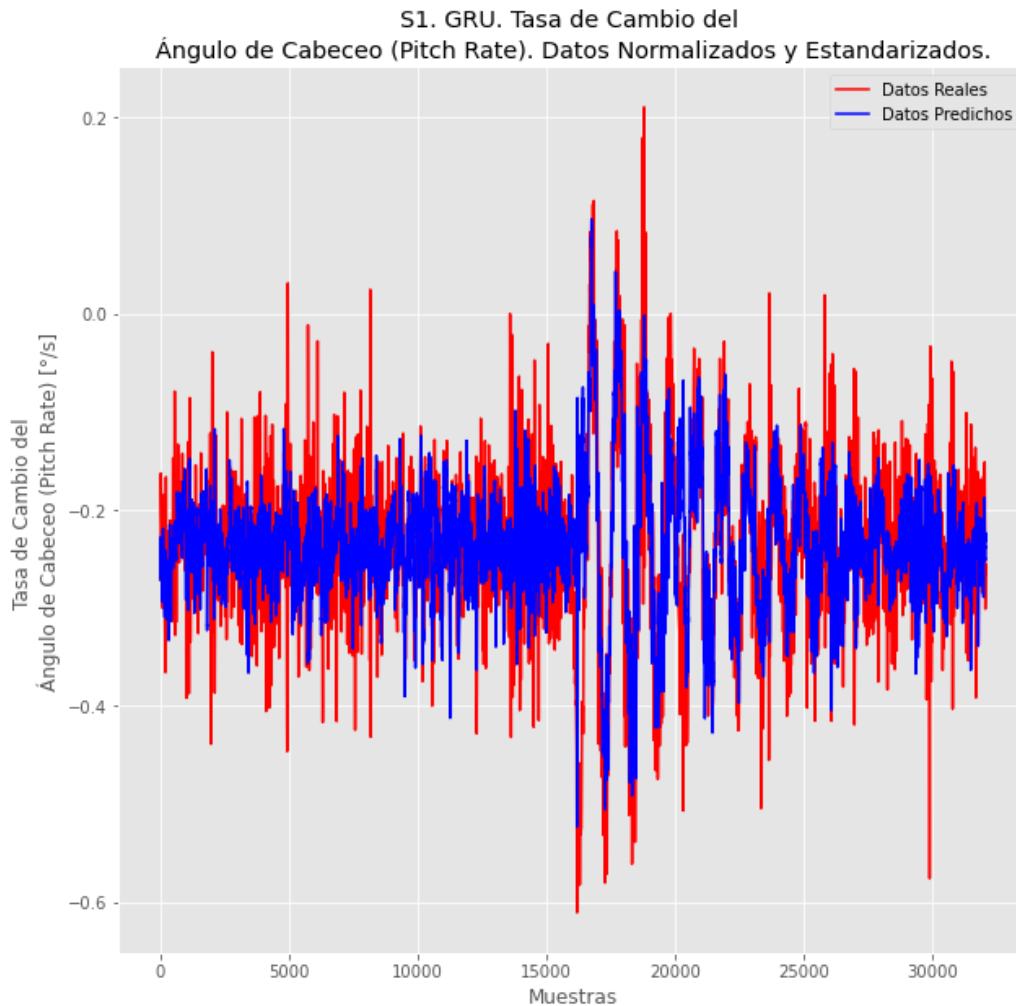


Figura 5.4: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 1. Tasa de cambio del Ángulo de Cabeceo (o Pitch Rate).

Esta señal no arroja información útil alguna. Por ende, hay que considerar en un futuro otras opciones de colocación de los sistemas de adquisición, de manera que se puedan medir las señales deseadas.

5.1.5. Tasa de cambio del Ángulo de Balanceo (o Roll Rate)

Para la técnica GRU y el caso 14, la predicción de la variable tasa de cambio del ángulo de balanceo (o roll rate) fue medianamente buena, con un R^2 de 0.766. La Figura 5.5 muestra la señal de esta variable de manera gráfica.

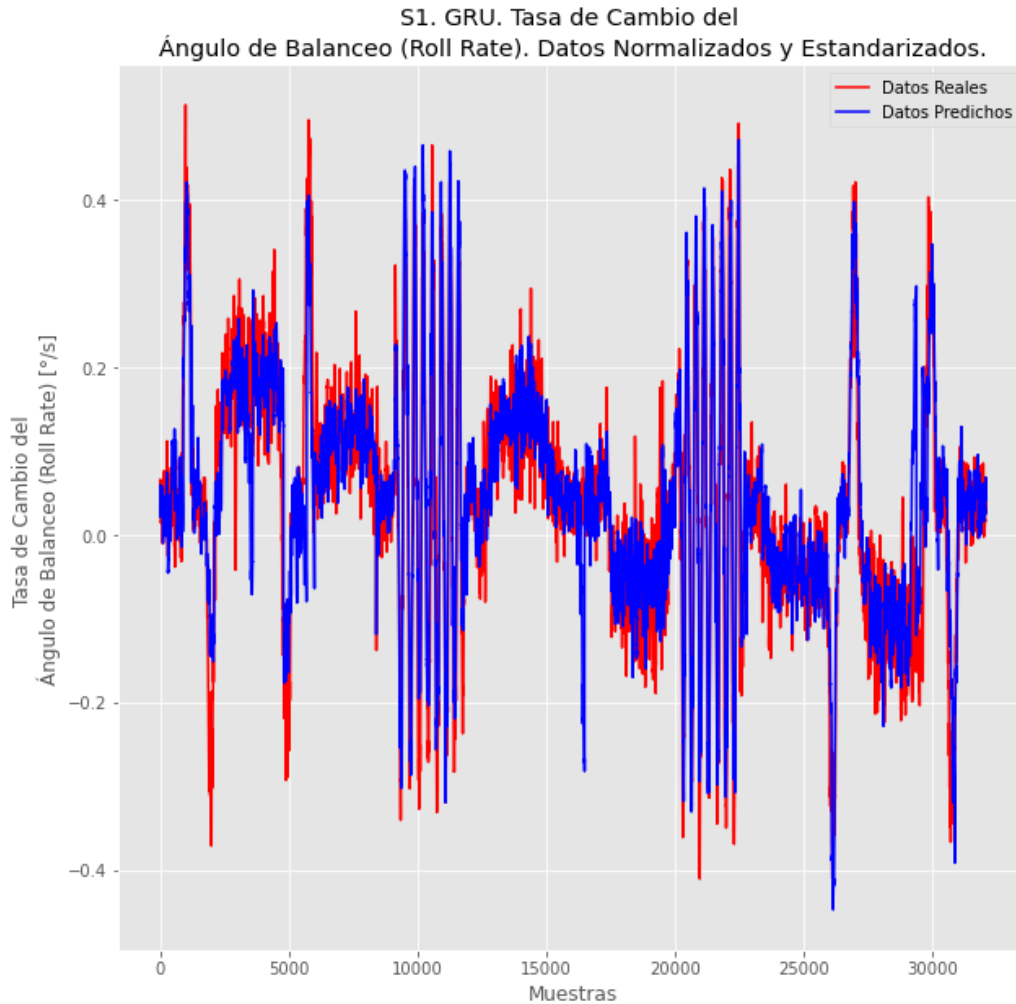


Figura 5.5: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 1. Tasa de cambio del Ángulo de Balanceo (o Roll Rate).

Como se ve en la Figura 5.5, las predicciones no llegaron a cubrir todas las zonas de la señal objetivo, mayormente fallaron en las zonas correspondientes a la maniobra de slalom (zig-zag) o a las vueltas o codos (cambios abruptos). Pero, lo que es destacable, al igual que para las otras variables, es la correcta estimación de las maniobras en el instante adecuado. Ello sigue siendo sumamente relevante para el presente trabajo y más para el caso de ADASs. Si la predicción hubiese estimado una maniobra de recta cuando era una prueba slalom o algún caso análogo, ahí sí habría un problema porque hace una incorrecta identificación de maniobras, pero este modelo ML no tiene

ese problema.

Finalmente, se debe remarcar algo muy interesante acerca de este sujeto, además de su pericia. Si se compara la arquitectura del modelo ML para este sujeto con las de los demás sujetos, se halla que ésta es la más pesada, se trata de un caso 14 (consúltese la Metodología), es decir, que el historial es dos veces el horizonte de predicción. En otras palabras, se está tomando el doble de pasos de tiempo hacia atrás que para la predicción hacia adelante, o se debe conocer más del fenómeno para hacer una predicción. Es intuitivo que, para una persona con gran variabilidad en sus datos, el algoritmo deba aprender los patrones considerando muchos más datos atrás.

5.2. Sujeto 2

Para el horizonte de predicción de 3 s., el modelo ML con mejores resultados fue el GRU del Caso 11, es decir, *GRU con relación historial-horizonte de 1.8 s. - 3 s. (180 pasos de tiempo hacia atrás-300 pasos de tiempo hacia adelante)*. El modelo GRU del caso 11 tuvo la mayor norma L2 de R^2 de los casos de horizonte de 3 s. (casos 11-15) para el sujeto 2, alcanzando un valor de 1.799, y tiene la menor complejidad (arquitectura más sencilla) para dicho horizonte.

Este sujeto, por lo visto en los perfiles de velocidad del capítulo pasado, tiene un manejo más cauto (magnitudes de velocidad bajas), y ello se muestra en el rendimiento del modelo ML para cada una de las salidas relacionadas con ADASs, como se ilustra a continuación.

5.2.1. Ángulo de Viraje (Ángulo Yaw) o Ángulo de Deslizamiento Lateral (Slip Angle)

A diferencia del sujeto 1, para el sujeto 2 se obtiene un muy buen valor de R^2 , específicamente de 0.845, que tiene una buena prospectiva para un horizonte de predicción más amplio una vez que se pueda implementar. La Figura 5.6 muestra esta salida de una manera gráfica.

El estilo de manejo de este sujeto era mucho más tranquilo/cauto que el del sujeto 1 (véase el perfil de velocidad para el sujeto 2 del capítulo pasado). En efecto, el ángulo de deslizamiento lateral está caracterizado por cambiar con una frecuencia alta, pero, debido a que este sujeto tenía una conducción muy tranquila, no hubo tanta variabilidad en sus datos, y ello le permitió al algoritmo ML predecir la variable de manera notable.

Hay unos detalles en cuanto al alcance de las amplitudes por parte de la predicción, pero en general las maniobras se respetan en el momento que deben ocurrir respecto a la señal verdadera (ground truth), y ello es lo que más importa. Obviamente, el rendimiento de R^2 se ve afectado porque la predicción no llega a las amplitudes del ground truth, pero hay que buscar otros criterios de desempeño en los que se involucren otro tipo de parámetros.

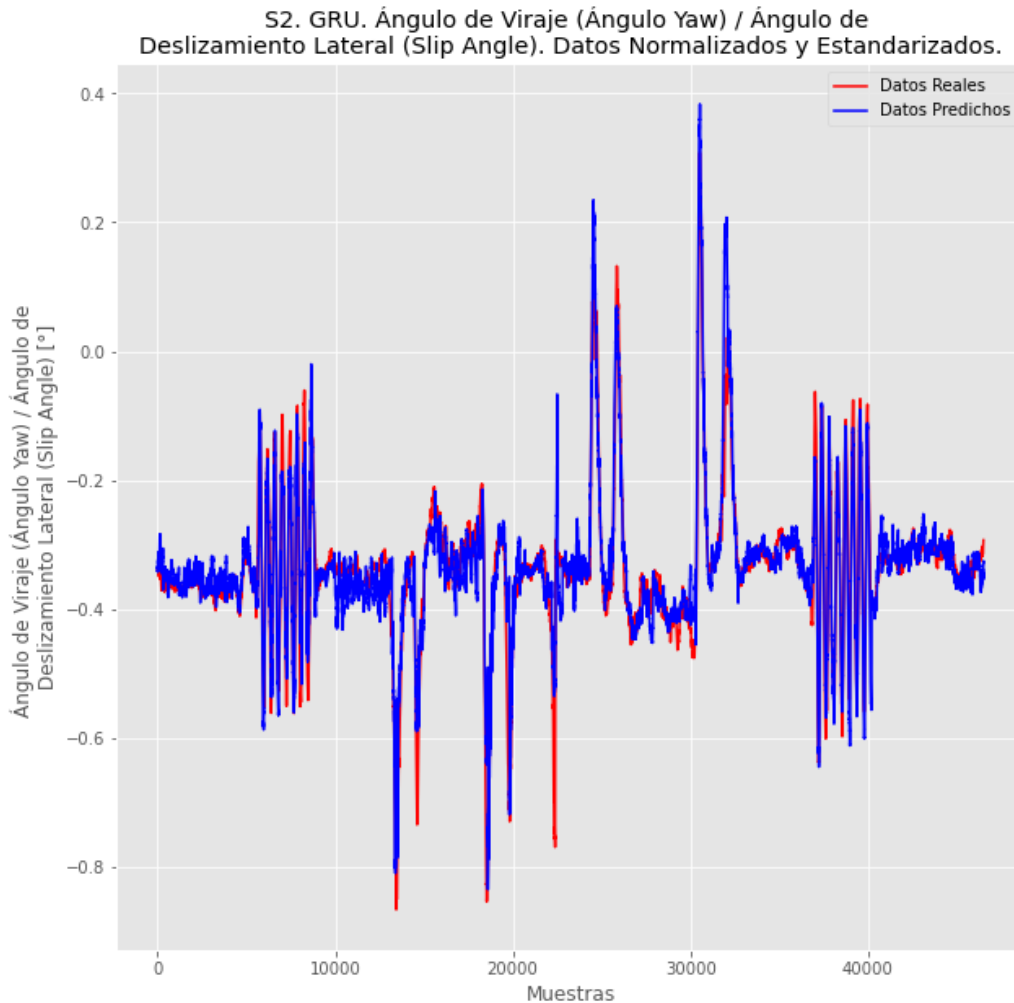


Figura 5.6: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 2. Ángulo de Viraje (Ángulo Yaw) o Ángulo de Deslizamiento Lateral (Slip Angle).

5.2.2. Ángulo de Balanceo (Ángulo Roll)

La predicción del ángulo de balanceo (ángulo roll) fue buena, cuantitativamente alcanzó un R^2 de 0.961, incluso mejor que para el sujeto 1. El algoritmo siguió de una manera sobresaliente la señal objetivo, aunque, como no hay un algoritmo ML 100% eficiente, sigue teniendo pequeños detalles, específicamente en la zona de las maniobras de Slalom (zig-zag), pero son muy mínimos. La gráfica de esta variable se muestra en la Figura 5.7.

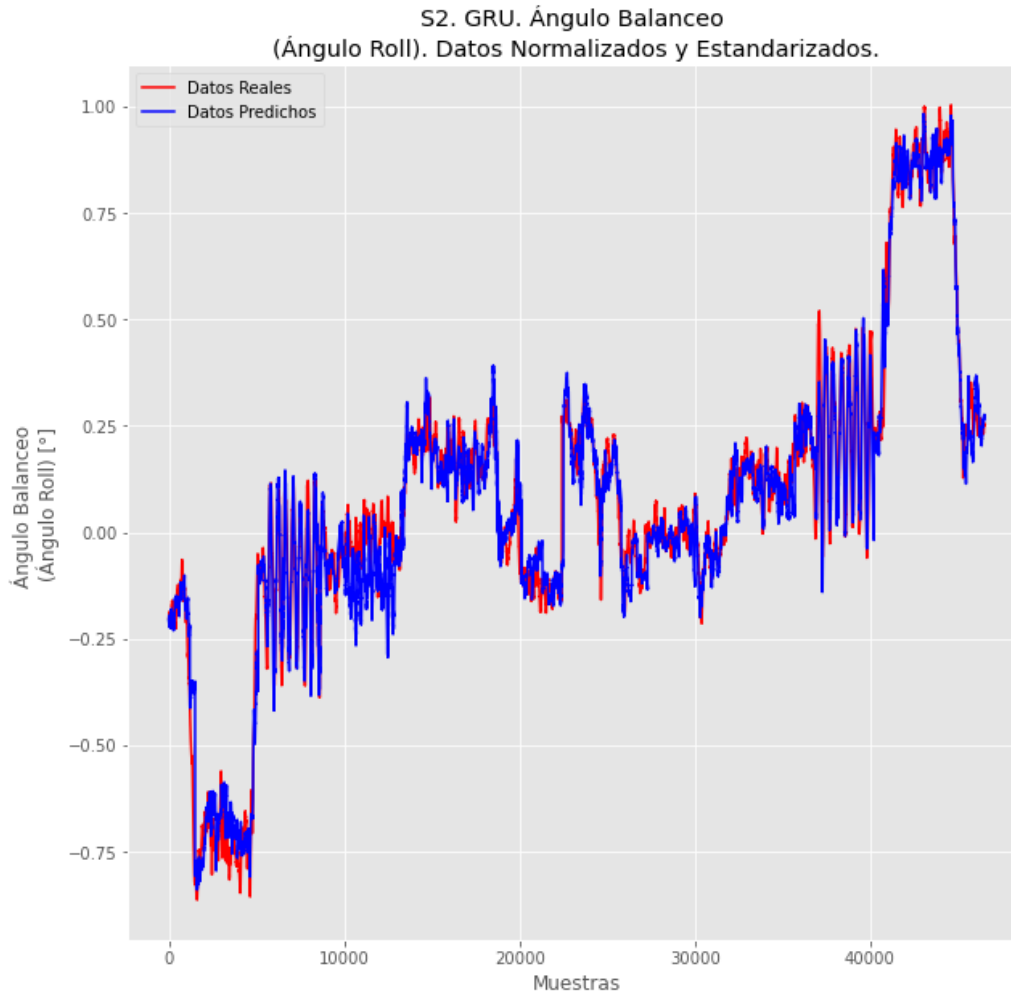


Figura 5.7: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 2. Ángulo de Balanceo (Ángulo Roll).

En la Figura 5.7 se muestra cómo el modelo es capaz de predecir todas las maniobras (inmersas en dicha gráfica) de manera muy notable. Se respeta la ocurrencia de todas las maniobras, pero también el tiempo en que deben suceder. Además de que las amplitudes son muy semejantes entre la predicción y la señal objetivo.

5.2.3. Tasa de cambio del Ángulo de Viraje (o Yaw Rate)

Para esta señal, igual que para la previa, la técnica GRU en el caso 11 logró un R^2 de 0.925. De manera semejante al ángulo roll, se esperarían muy buenos resultados si el horizonte de predicción se incrementara. La señal se puede ver mediante la Figura 5.8.

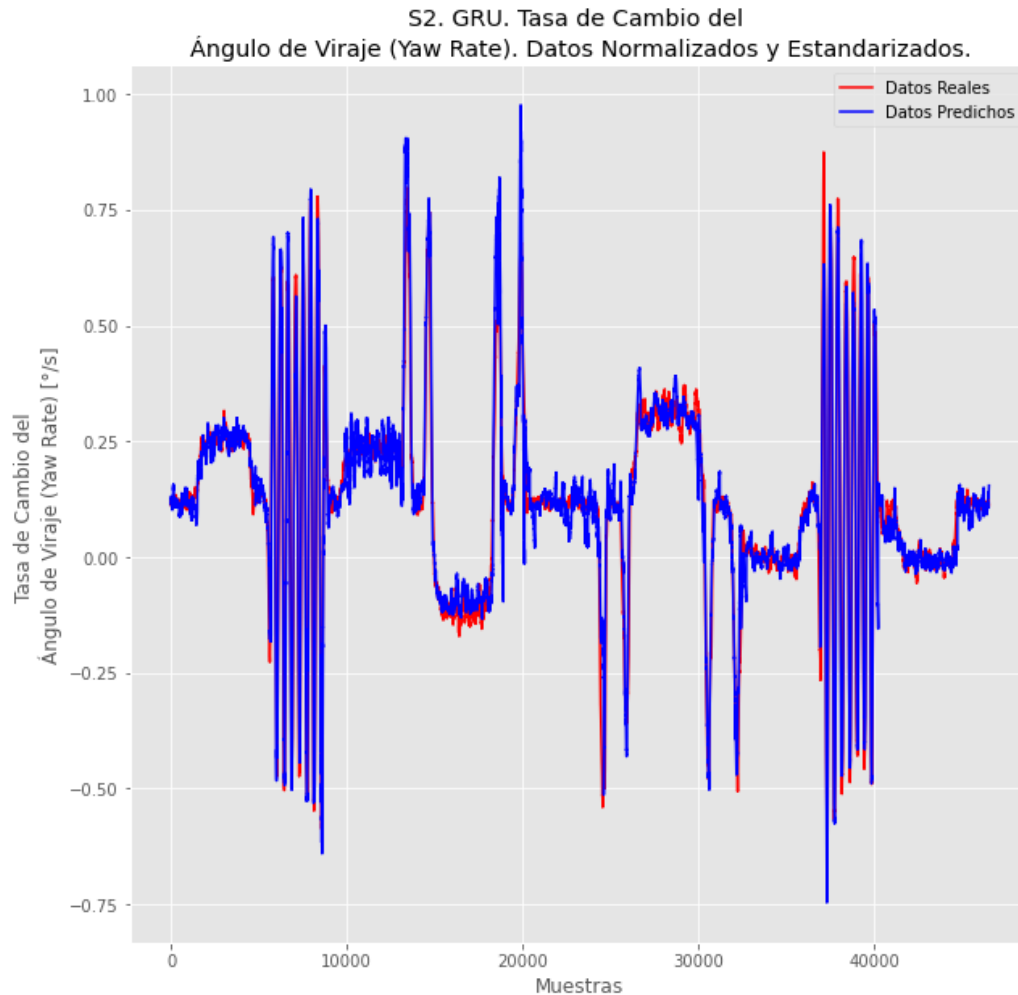


Figura 5.8: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 2. Tasa de cambio del Ángulo de Viraje (o Yaw Rate).

Visualizando la Figura 5.8, se puede notar el grado de predicción, la gráfica habla por sí sola. Como el ángulo de deslizamiento lateral, que para este caso coincidió con el ángulo de viraje (ángulo yaw), para este sujeto tuvo muy buena predicción, es obvio que su tasa de cambio también debe tener un buen rendimiento, y en efecto eso se refleja en los resultados.

5.2.4. Tasa de cambio del Ángulo de Cabeceo (o Pitch Rate)

Como ya se aclaró para el sujeto 1, esta variable se intuye que está embebida en mucha señal espuria, o es ruido. A partir de la Figura 5.9, se advierte que la predicción siguió el target, pero, para nada, le es suficiente. El desempeño del modelo GRU (caso 11) concerniente a la tasa de cambio del ángulo de cabeceo (o pitch rate), fue muy pobre, un R^2 de 0.063.

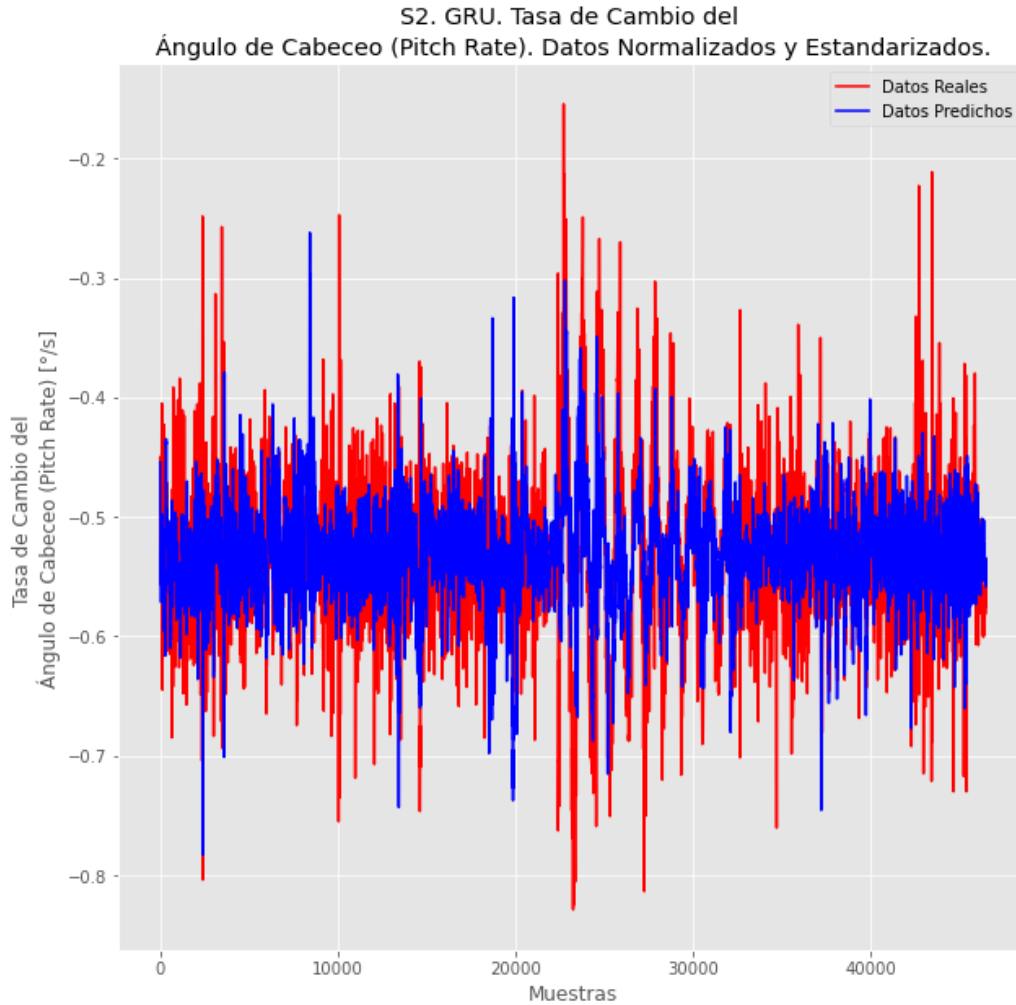


Figura 5.9: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 2. Tasa de cambio del Ángulo de Cabeceo (o Pitch Rate).

Se debe verificar de mejor manera el acomodo de los equipos de adquisición, para no obtener señales espurias.

5.2.5. Tasa de cambio del Ángulo de Balanceo (o Roll Rate)

Finalmente, para esta variable se obtiene un buen desempeño del modelo GRU Caso 11; matemáticamente el valor de R^2 es de 0.860. Y de forma gráfica, la salida se ve en la Figura 5.10.

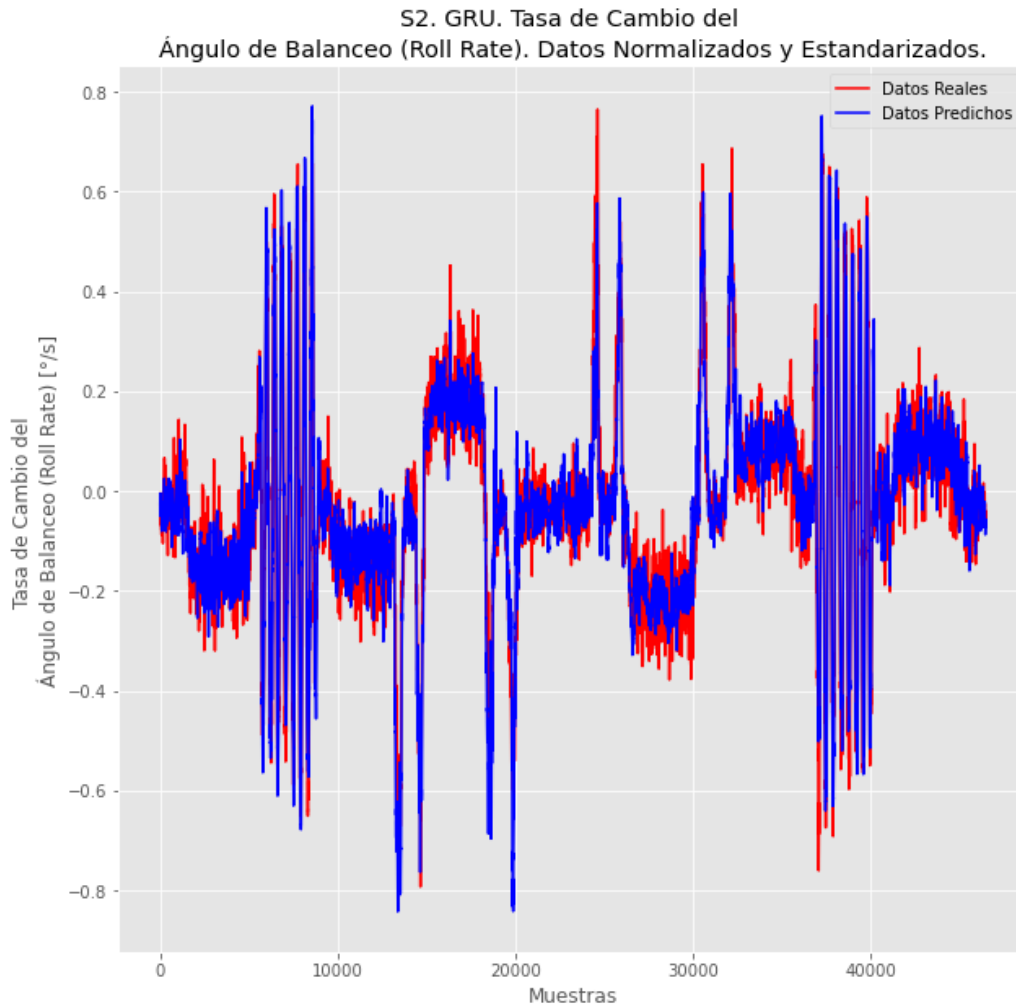


Figura 5.10: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 2. Tasa de cambio del Ángulo de Balanceo (o Roll Rate).

A diferencia del sujeto 1, para este participante el modelo ML es capaz de predecir de mejor manera la sección correspondiente a la evasión de obstáculos (zig-zag). La predicción no se muestra desfasada respecto al ground truth, lo que muestra que el algoritmo ML está trabajando de buena manera y que las maniobras se están respetando.

Se finaliza este sujeto comentando que, tras la revisión de los datos, su conducción fue más cauta y tranquila que el sujeto 1. Muestra de ello es que una arquitectura sencilla fue capaz de obtener los mejores resultados para este sujeto, a excepción de la tasa de cambio del ángulo de cabeceo (*pitch rate*), variable que no se adquirió correctamente.

5.3. Sujeto 3

Para el horizonte de predicción más grande, 3 s., el modelo ML con mejores resultados que representa el Modelo Vehículo-Conductor para el sujeto 3 fue el *LSTM Codificador-Decodificador (LSTM Encoder-Decoder) Caso 12* o referente a la relación historial-horizonte de 2.4 s. - 3 s. (240 pasos de tiempo hacia atrás-300 pasos de tiempo hacia adelante). Esta técnica ML alcanzó una norma L2 de R^2 de 1.708.

La conducción de este sujeto fue más tranquila que la del sujeto 1, por lo que no hubo tanta variabilidad de los datos, y ello ayudó a que las predicciones de las variables de salida sean notables.

5.3.1. Ángulo de Viraje (Ángulo Yaw) o Ángulo de Deslizamiento Lateral (Slip Angle)

Para el Sujeto 3 se concluyen resultados buenos para esta variable en el horizonte de predicción de 3 s. El valor de R^2 es de 0.869, por lo que la variable se comporta de buena manera y se prevén buenas predicciones para un horizonte todavía más largo. La Figura 5.11 muestra el comportamiento de esta variable de salida.

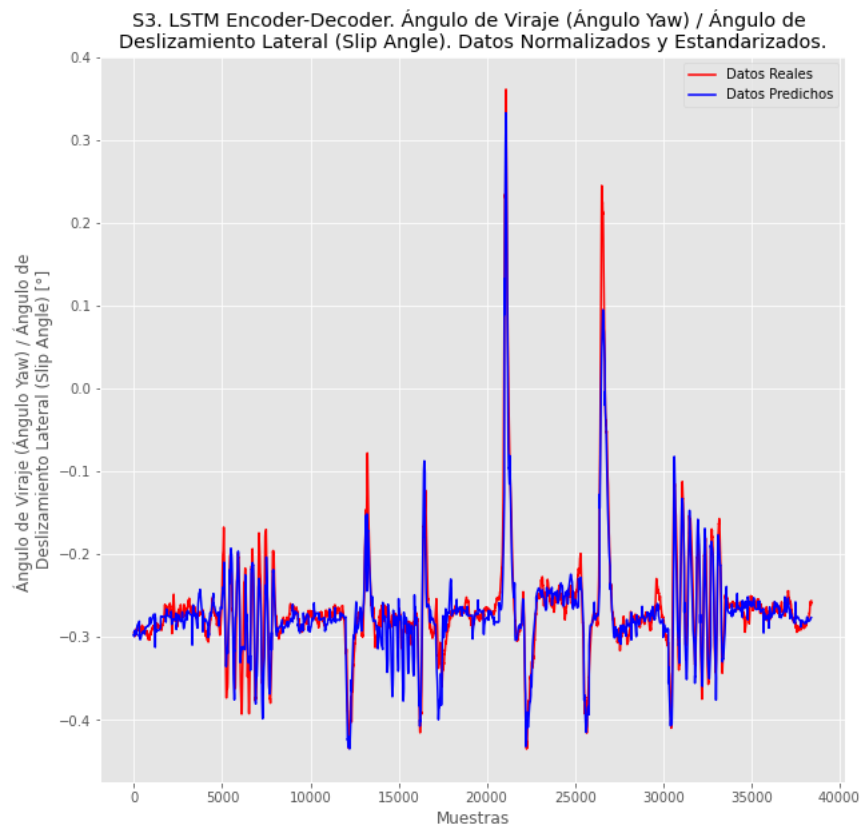


Figura 5.11: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 3. Ángulo de Viraje (Ángulo Yaw) o Ángulo de Deslizamiento Lateral (Slip Angle).

El sujeto 3, al igual que los sujetos 2 y 4, son conductores no técnicos, ello explica por qué el modelo ML, sencillo en arquitectura, puede tener tan buenas predicciones. La predicción del ángulo de deslizamiento lateral respeta las maniobras de la señal objetivo; no obstante, falta un mejor procesamiento de la información para poder llegar a los extremos del ground truth. En ello hay que trabajar para éste y los demás sujetos.

5.3.2. Ángulo de Balanceo (Ángulo Roll)

El ángulo de balanceo (ángulo roll) tiene un comportamiento bueno al alcanzar un R^2 de 0.907 para el horizonte de 3 s. La Figura 5.12 muestra el comportamiento de esta variable. Se observa cómo la predicción y la señal objetivo son muy parecidas durante la mayor parte del tiempo.

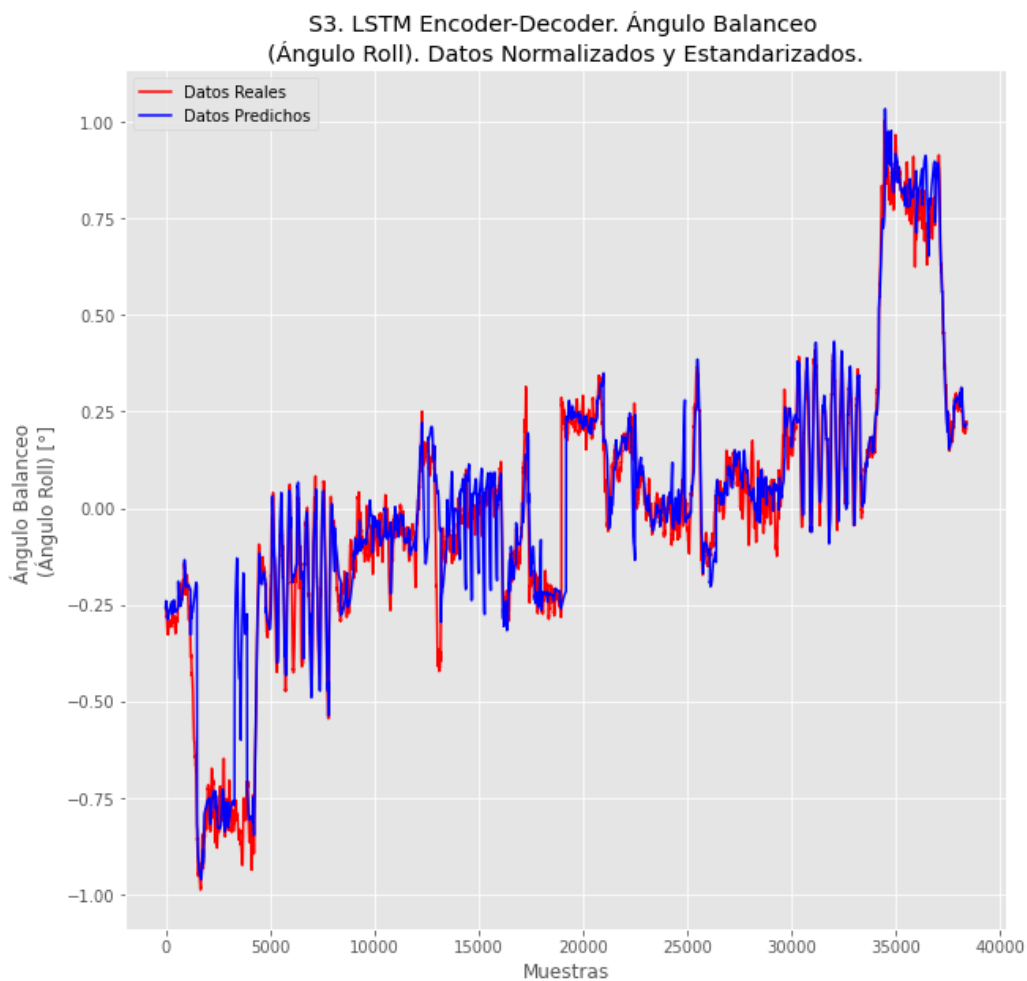


Figura 5.12: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 3. Ángulo de Balanceo (Ángulo Roll).

La variable del ángulo de balanceo, como anteriormente se ha definido, es una cantidad que cambia muy lentamente con el tiempo, o de frecuencia baja. Ello sumado al hecho de que este sujeto

no tiene conducción técnica, ayudó a que las predicciones fueran notables para un horizonte un tanto amplio, 3 s.

5.3.3. Tasa de cambio del Ángulo de Viraje (o Yaw Rate)

EL valor de R^2 para esta variable es de 0.862. Se sigue una buena predicción para estas primeras tres variables por parte de este modelo ML y este caso. La señal se puede ver mediante la Figura 5.13.

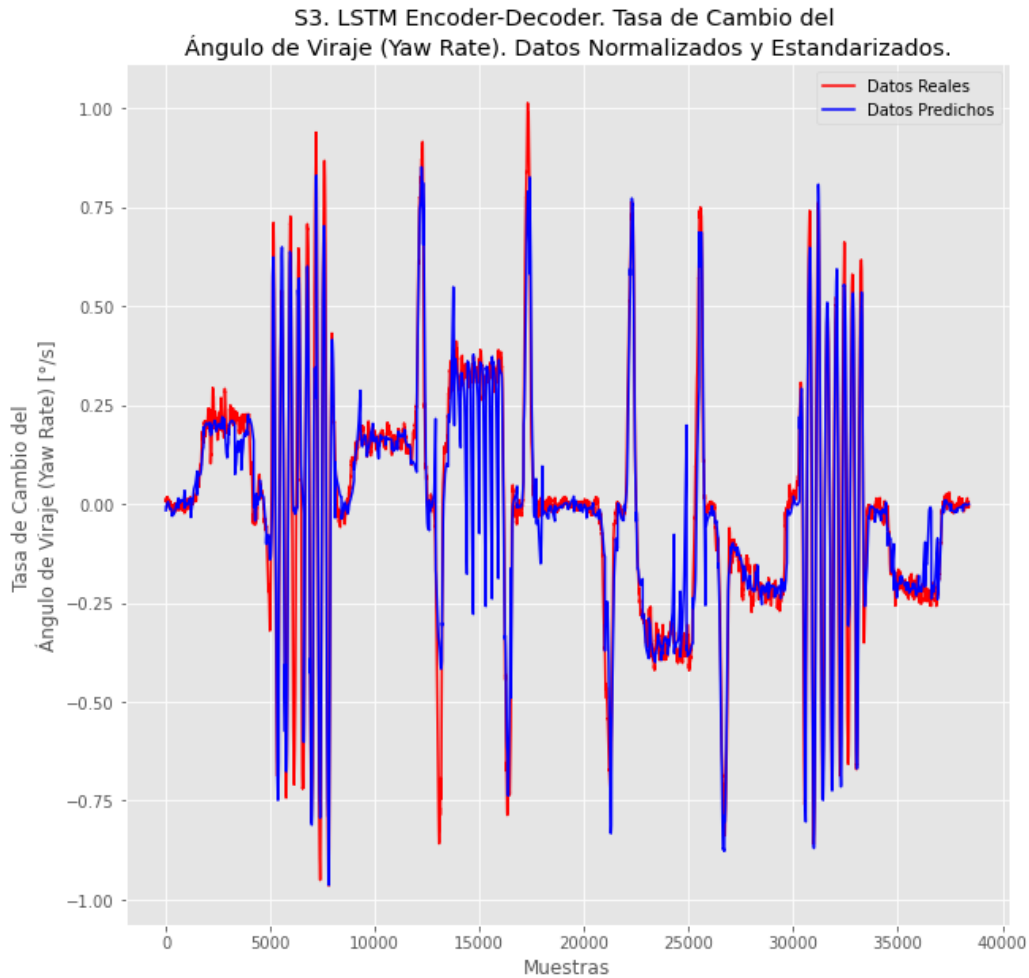


Figura 5.13: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 3. Tasa de cambio del Ángulo de Viraje (o Yaw Rate).

Como se ve de la Figura 5.13, la predicción del modelo ML hizo un seguimiento o *tracking* de la señal objetivo notable. En cuanto a los cambios bruscos, en muchos casos la estimación no llegó al valor real, pero la identificación de las maniobras se conservó en todo momento. Ello deriva en un modelo que da resultados con sentido físico, pero que aún falta por mejorarse para alcanzar los valores extremos.

5.3.4. Tasa de Cambio del Ángulo de Cabeceo (o Pitch Rate)

Con respecto a esta variable, del mismo modo que los sujetos previos, no parece haber señal inteligente o que lleve información. Con esa señal espuria, el modelo ML es capaz de hacer una predicción con un valor R^2 de -0.069 . Ello implica que se podría dar el valor de la media y tener mejor resultado que con la predicción [101].

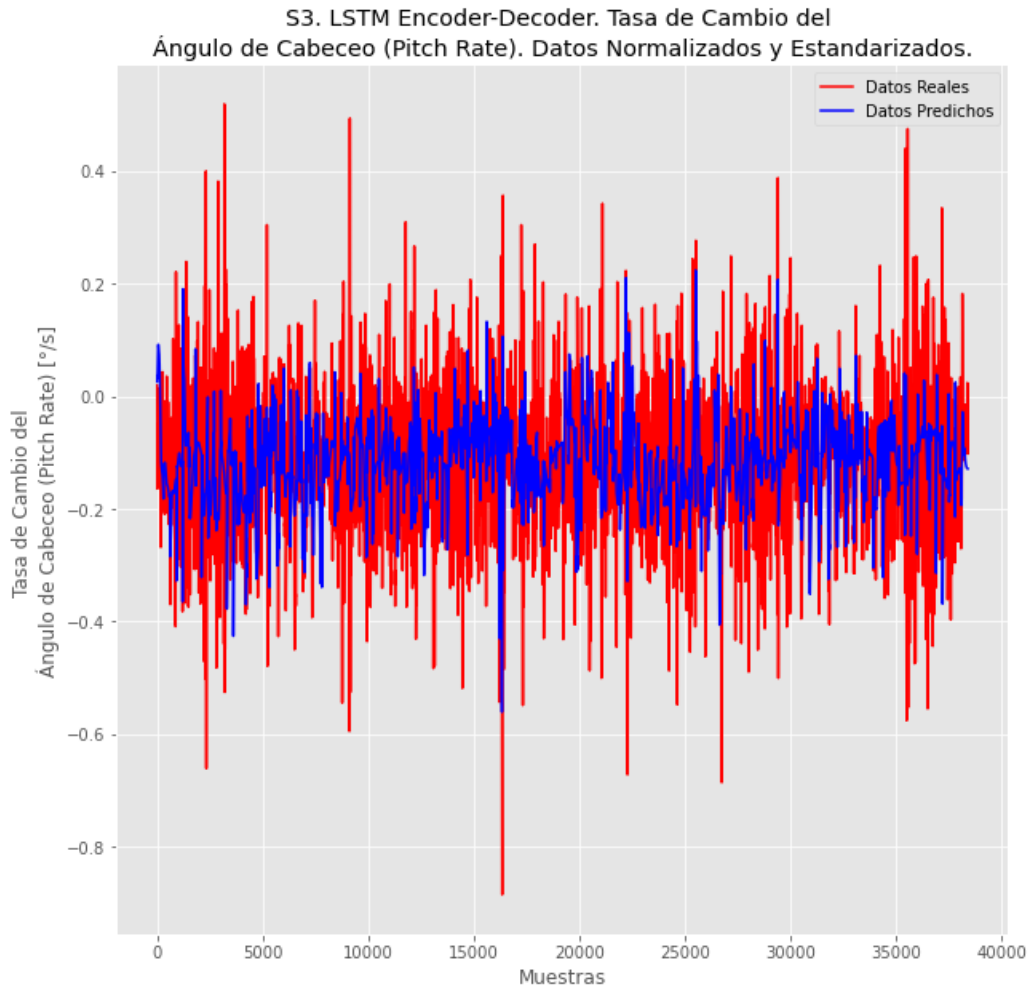


Figura 5.14: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 3. Tasa de cambio del Ángulo de Cabeceo (o Pitch Rate).

La Figura 5.14 demuestra cómo el modelo ML no es capaz de mapear las entradas con la salida.

5.3.5. Tasa de Cambio del Ángulo de Balanceo (o Roll Rate)

Para esta última variable, el valor de R^2 es de 0.769 , lo que implica un comportamiento medianamente bueno del modelo ML. La Figura 5.15 muestra la forma en que este modelo ML (en el respectivo caso) mapea las entradas a la salida.

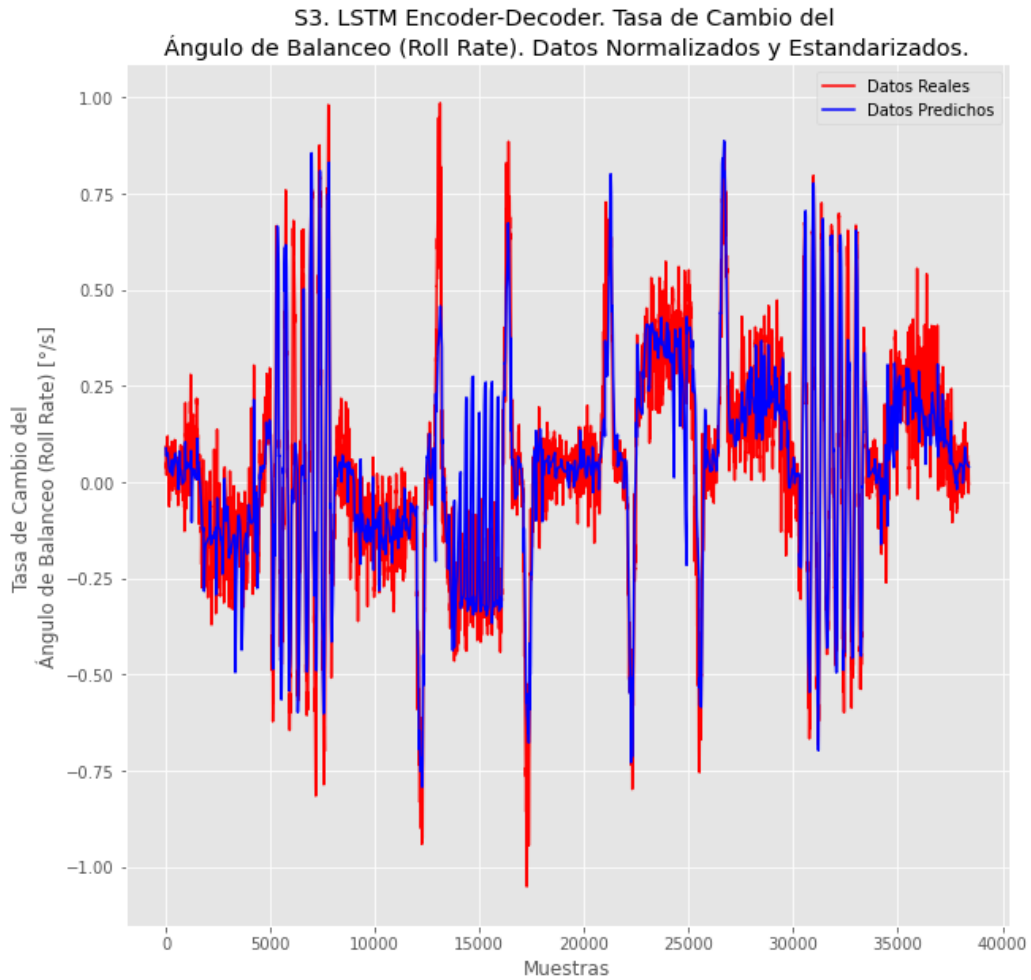


Figura 5.15: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 3. Tasa de cambio del Ángulo de Balanceo (o Roll Rate).

Para esta variable de salida, la predicción sigue la forma del ground truth, lo que implica una correcta identificación de maniobras; no obstante, no puede estimar bien los extremos. A pesar de ello, para un horizonte de predicción de 3 s., dicho valor de R^2 es un valor interesante.

Finalmente se concluye este sujeto diciendo que era un conductor normal, sin entrenamiento alguno y mencionando que el modelo ML estimó de buena manera las variables que más interesan para la evaluación de riesgo de vuelco y de derrape lateral, ángulo de balanceo (ángulo roll) y ángulo de deslizamiento lateral (slip angle), respectivamente.

5.4. Sujeto 4

El sujeto 4, por la forma en que realizó las pruebas de campo, se puede calificar como un conductor con acciones precipitadas y sin entrenamiento alguno.

Para el horizonte de predicción más grande, 3 s., el modelo ML que representa de mejor manera

el Modelo Vehículo-Conductor para el sujeto 4 es la *LSTM Vanilla Caso 11* o referente a la relación historial-horizonte de 1.8 s. - 3 s. (180 pasos de tiempo hacia atrás-300 pasos de tiempo hacia adelante). Este modelo ML en el respectivo caso alcanzó una norma L2 de R^2 de 1.600. Se procede a mostrar el desempeño en las predicciones que tuvo esta técnica ML en el respectivo caso para cada una de las salidas.

5.4.1. Ángulo de Viraje (Ángulo Yaw) o Ángulo de Deslizamiento Lateral (Slip Angle)

Para el sujeto 4 se concluyen resultados medianamente buenos para esta variable en el horizonte de predicción de 3 s. El valor de R^2 es de 0.704. La Figura 5.16 muestra el comportamiento de esta variable de salida.

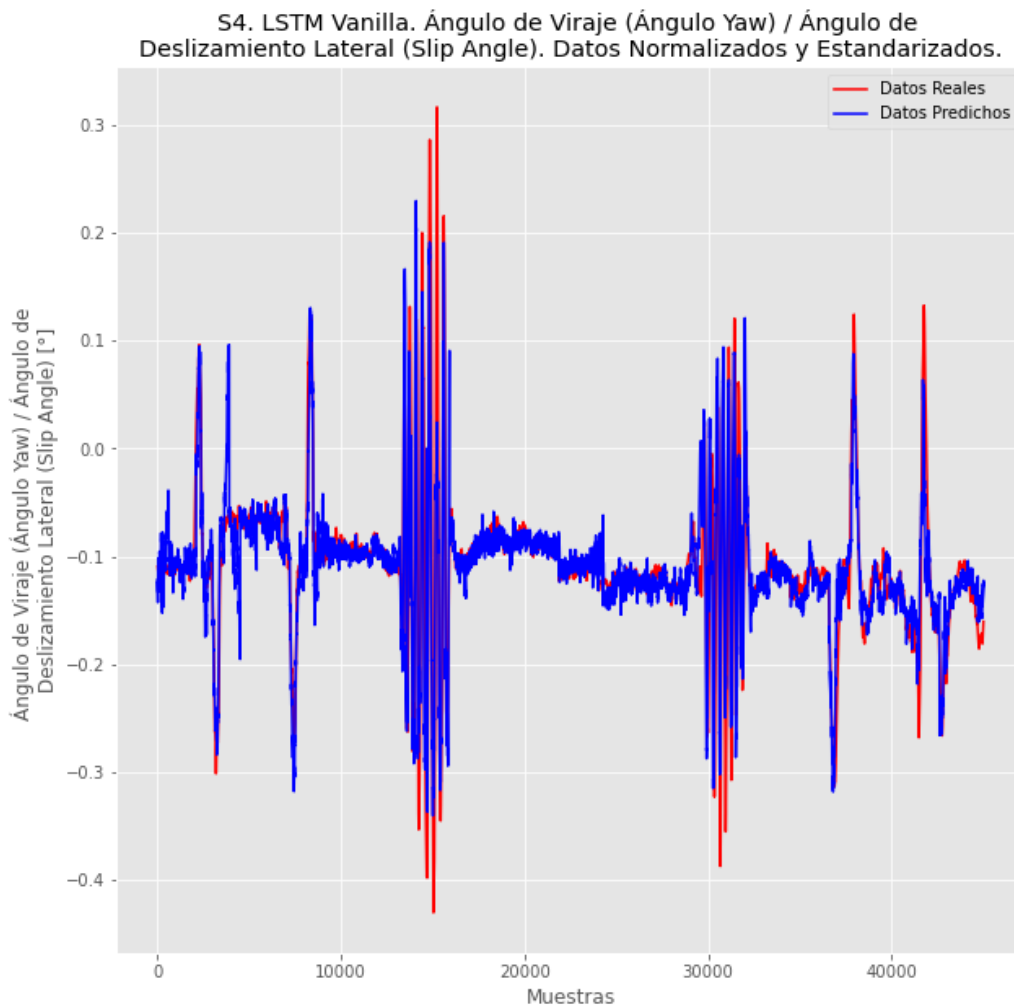


Figura 5.16: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 4. Ángulo de Viraje (Ángulo Yaw) o Ángulo de Deslizamiento Lateral (Slip Angle).

Se debe recordar que el ángulo de deslizamiento lateral, relacionado al riesgo de derrape lateral, cambia constantemente, por ende, para un conductor con acciones precipitadas (sujeto 4), el comportamiento de dicha variable cambia aún más, pero sin llegar al caso del sujeto 1. Debido a esa gran frecuencia de variación de los datos del ángulo de deslizamiento lateral, se le complica más al modelo ML hacer buenas predicciones. Por otro lado, aunque la predicción no estimó de buena manera los valores extremos, la identificación de maniobras se respeta.

5.4.2. Ángulo de Balanceo (Ángulo Roll)

El ángulo de balanceo (ángulo roll) tiene un comportamiento bueno al alcanzar un R^2 de 0.954 para el horizonte de 3 s. La Figura 5.17 muestra el comportamiento de esta variable. Se observa cómo la predicción y la señal objetivo son muy parecidas durante la mayor parte del tiempo.

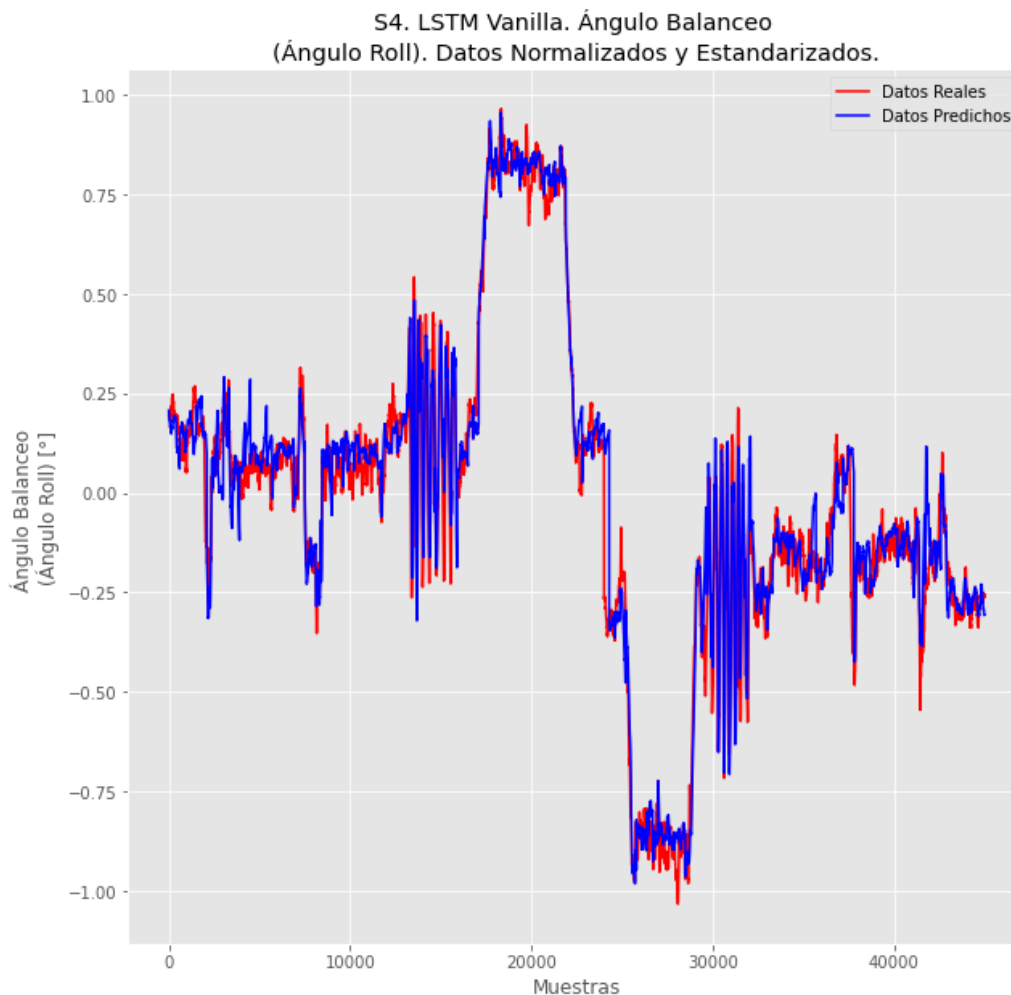


Figura 5.17: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 4. Ángulo de Balanceo (Ángulo Roll).

Como anteriormente ya se ha aclarado, el ángulo de balanceo es una variable vehicular que cambia

muy poco con el tiempo, entonces es fácilmente predecible. Y ello se reflejó para los cuatro sujetos, pero más para los que tenían manejo cauto. Queda de más el comentar la clara identificación de maniobras respecto al ground truth. Todo se asimila al visualizar la Figura 5.17.

5.4.3. Tasa de Cambio del Ángulo de Viraje (o Yaw Rate)

El valor de R^2 para esta variable es de 0.806. La señal se puede ver mediante la Figura 5.18.

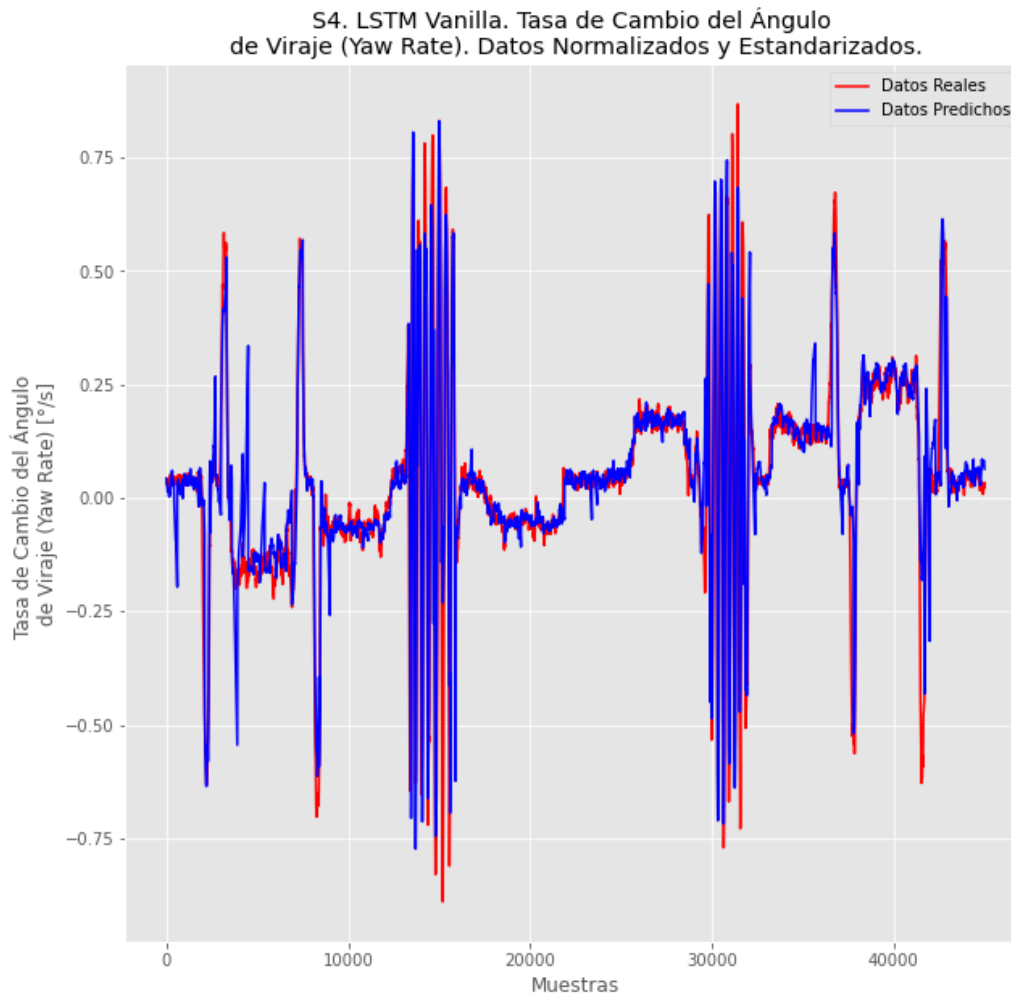


Figura 5.18: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 4. Tasa de cambio del Ángulo de Viraje (o Yaw Rate).

La predicción se ve bien, se estima la prueba de slalom (zig-zag) cuando debe de ocurrir, así como las demás maniobras. También es notable que se debe trabajar más para que el modelo ML pueda estimar mejor los valores extremos o cambios abruptos de la señal, pero de manera general, la identificación de maniobras ocurre de manera relevante.

El criterio R^2 no es tan alto porque depende más de la amplitud que de otros factores de las señales. Así que hay que buscar también otros criterios de desempeño.

5.4.4. Tasa de Cambio del Ángulo de Cabeceo (o Pitch Rate)

Con respecto a esta variable, del mismo modo que para los tres sujetos previos, parece una señal espuria. Con ese ruido, el modelo ML es capaz de hacer una predicción con un valor R^2 de 0.052. Ello implica un rendimiento muy bajo del modelo ML en el respectivo caso para esta salida. La Figura 5.19 muestra la predicción de esta variable.

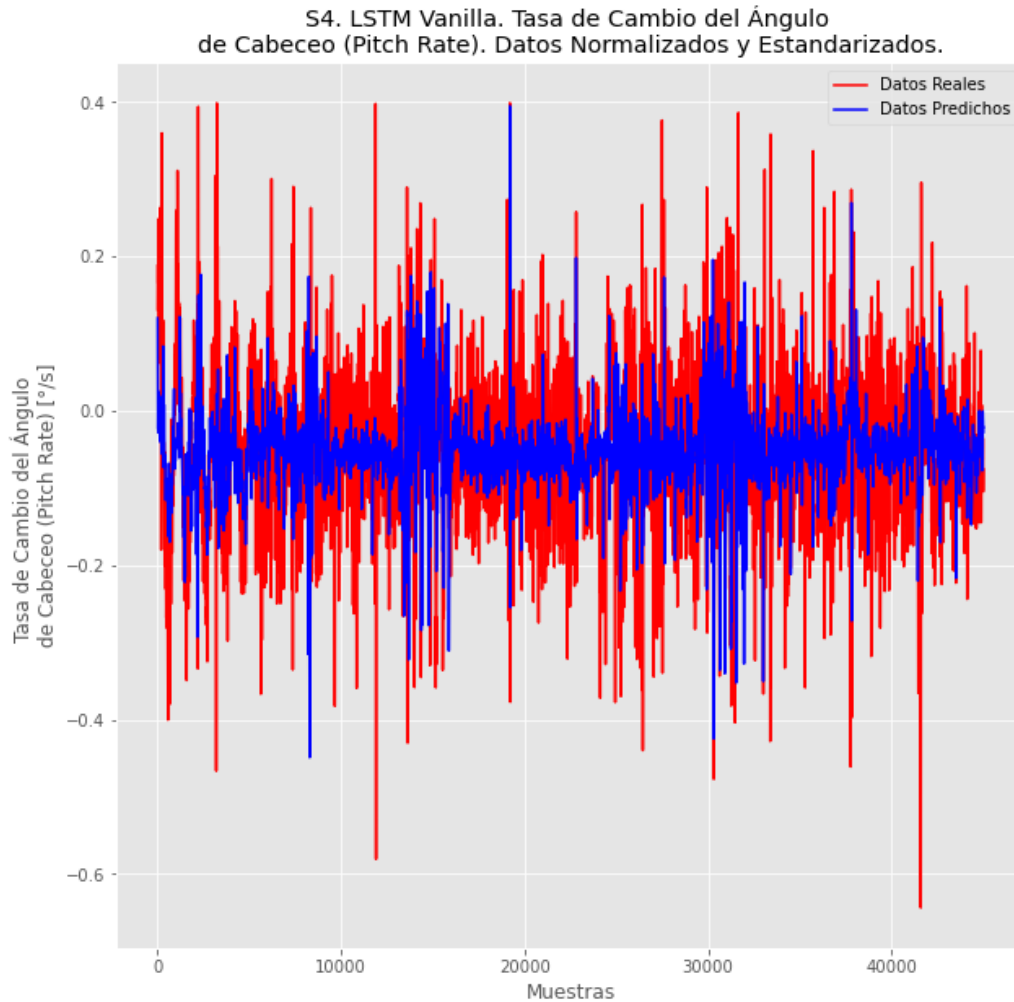


Figura 5.19: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 4. Tasa de cambio del Ángulo de Cabeceo (o Pitch Rate).

Es muy claro que no hay un seguimiento de la señal objetivo, de ahí que el desempeño del modelo para esta variable sea muy bajo. Sin embargo, ese bajo rendimiento se relaciona más con los propios datos que con el funcionamiento del modelo ML. Queda como prospectiva mejorar la adquisición de esta variable.

5.4.5. Tasa de Cambio del Ángulo de Balanceo (o Roll Rate)

Para esta última variable, el valor de R^2 es de 0.700, lo que implica un comportamiento medianamente bueno del modelo ML. La Figura 5.20 muestra el desempeño de este modelo ML en lo concerniente a la variable analizada.

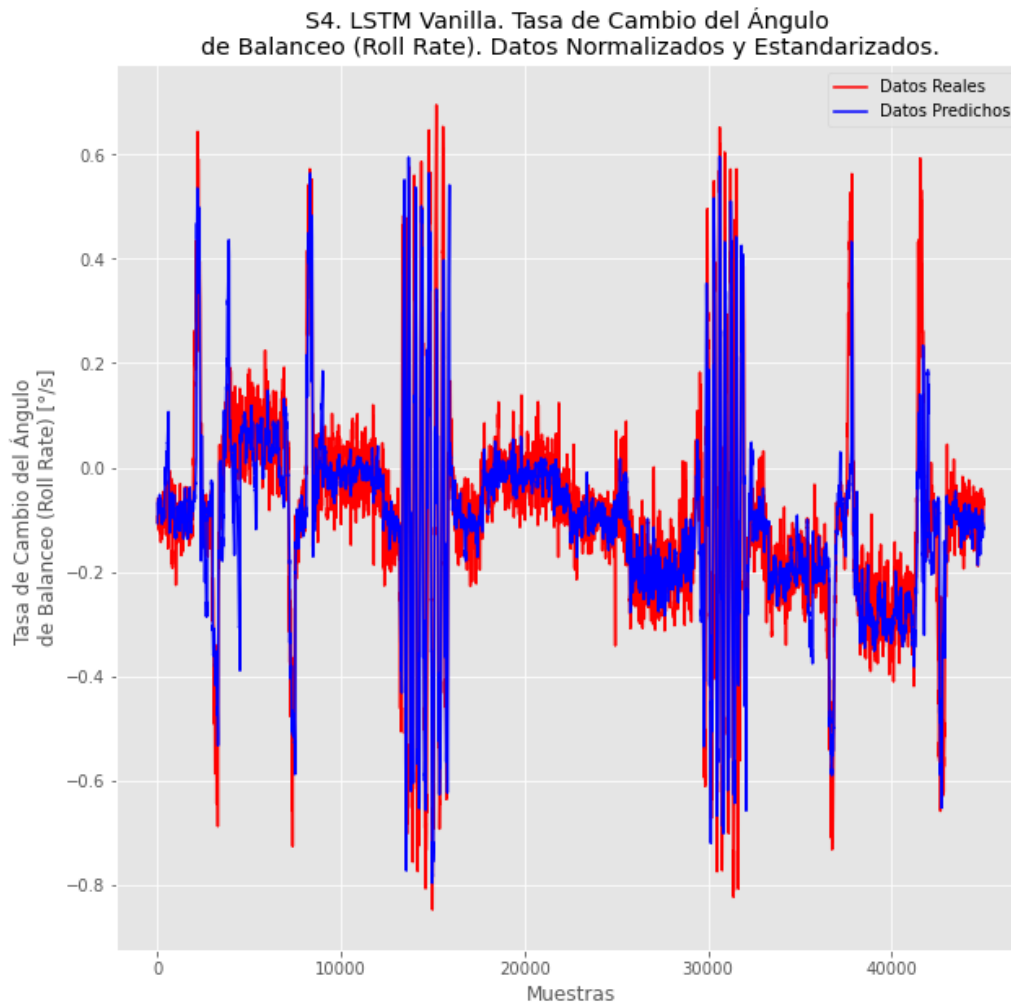


Figura 5.20: Sistemas Avanzados de Asistencia a la Conducción. Sujeto 4. Tasa de cambio del Ángulo de Balanceo (o Roll Rate).

El seguimiento de la predicción con respecto a la señal objetivo es claro, sólo que el modelo ML no termina de captar los cambios abruptos de la dinámica vehículo-conductor. Sin embargo, resulta sorprendente verificar cómo el modelo ML es capaz de predecir las maniobras en el instante correcto, y comprobar que no hay desfase de la predicción con respecto al ground truth.

De esta manera, se acaba la discusión de resultados con respecto a la perspectiva de ADASs para todos los sujetos. A continuación se hará un sumario de lo argumentado en este capítulo.

5.5. Sumario

Para fines muy puntuales, conviene hablar de los resultados de este capítulo, pero sólo los relacionados a las dos variables que se estiman más en la literatura mediante el Problema de Observabilidad: *ángulo de deslizamiento lateral (slip angle)* y *ángulo de balanceo (ángulo roll)*. Ello por la relación intrínseca que guardan con la estabilidad lateral del vehículo y el riesgo de vuelco, respectivamente. Los resultados se enuncian a continuación:

1. El ángulo de deslizamiento lateral (slip angle) cambia de manera muy rápida con respecto al tiempo, a diferencia del ángulo de balanceo (ángulo roll), que lo hace muy lento.
2. Para los sujetos 2 y 3, ambas variables de salida son predichas o estimadas de manera muy notable para un horizonte de predicción de 3 s. Ello se atribuye a la naturaleza *cauta* de la conducción de este tipo de sujetos. Las entradas propuestas en el Modelo Vehículo-Conductor bastan para la predicción de la dinámica del vehículo (dos salidas), siempre y cuando el estilo de conducción sea cauto o “seguro”.
3. En lo concerniente al sujeto 4, su estilo de conducción se caracteriza por acciones precipitadas. De este modo, el ángulo de balanceo (o ángulo roll) es estimado de manera notable gracias a que cambia muy lento respecto al tiempo; sin embargo, para el ángulo de deslizamiento lateral (slip angle) ya hay un poco más de falla en cuanto a la amplitud de la estimación respecto al ground truth. Por otro lado, la identificación de las maniobras se realiza correctamente y en el momento adecuado, además de que no hay desfase entre dichas señales. El R^2 es un criterio de desempeño que está más relacionado a las amplitudes de las señales, por lo que sería muy conveniente tomar algún otro que considere otros factores.
4. Con respecto al sujeto 1, su conducción era técnica o educada, es decir, trataba de maximizar el rendimiento del vehículo. La estimación del ángulo de balanceo (ángulo roll) fue íntegra por lo ya descrito; sin embargo, las predicciones del ángulo de deslizamiento lateral (slip angle) fueron muy pobres, un R^2 de 0.385. Son dos las principales razones que se le dan a este fenómeno. La primera tiene que ver con la tasa de cambio de esta variable, que es muy alta; y la segunda, con la variabilidad de los datos derivada del estilo de conducción de este sujeto. En conjunto dan lugar a una alta modificación de la variable vehicular, lo que dificulta la tarea de predicción por parte del modelo ML.
5. Para los sujetos con conducción técnica o caracterizada por acciones precipitadas (sujetos 1 y 4, respectivamente), ya no se podría concluir que las entradas del Modelo Vehículo-Conductor propuestas en esta tesis bastan para la predicción de la dinámica vehículo-conductor. Por ende, deben, primero, recabarse más pruebas o realizar diferentes experimentos; si ello no funciona, implementar otras técnicas ML o de pre-procesamiento de información; y si tampoco

eso funciona, irse al caso más radical, proponer otro Modelo Vehículo-Conductor (distintas entradas y salidas) con diferentes variables provenientes de sensores a bordo del vehículo o externos.

Capítulo 6

Conclusiones y Prospectivas

En este capítulo se expondrán las conclusiones del trabajo de tesis, además de las prospectivas o áreas de oportunidad si se desea continuar en esta línea de investigación.

6.1. Conclusiones

Las conclusiones, primero, se dividirán en dos rubros según los dos escenarios trabajados en esta tesis, y después se darán las conclusiones globales.

6.1.1. Estrategias de Manejo de Energía

1. El primer punto que debe quedar claro en esta perspectiva es que se trató de una primera aproximación a las Estrategias de Manejo de Energía utilizando técnicas de Inteligencia Artificial. De manera específica, lo que se buscó es proponer un Modelo Vehículo-Conductor que empleara más acciones del conductor (entradas del modelo) que las halladas en los trabajos del estado del arte, para predecir el perfil de velocidad (rapidez) de un vehículo. Además, dicho modelo se validó y construyó con datos experimentales, no con simulaciones. Ese es la principal contribución de la presente tesis, para este momento no interesan implementaciones en línea o en tiempo real, eso quedará como prospectiva.
2. Para este escenario se obtuvieron resultados notables para los cuatro sujetos, un R^2 por sujeto arriba de 0.9. No obstante, es menéster aclarar que el desfase utilizado en el procesamiento de las series de tiempo fue de 100 muestras, lo que en realidad hace que no se tomen ciertas secuencias de datos. Esta restricción fue a causa de las limitaciones de hardware, memoria VRAM de las GPUs.
3. Las Estrategias de Manejo de Energía son aplicables a vehículos eléctricos híbridos. El vehículo utilizado durante la experimentación en esta tesis no es un vehículo de este tipo. La razón

del uso de esta perspectiva en un vehículo convencional es porque sólo se limitó la tesis a la *predicción de variables vehiculares*, en este caso, la rapidez del vehículo, dadas ciertas entradas del conductor. Entonces, sólo fue una primera aproximación de la aplicación de las técnicas ML a cierta área específica de las Estrategias de Manejo de Energía, la predicción del comportamiento vehicular.

6.1.2. Sistemas Avanzados de Asistencia a la Conducción

Este rubro es más amplio que el de Estrategias de Manejo de Energía, pues en vez de predecir una variable vehicular, se predicen cinco. De forma general, se mencionan las siguientes conclusiones:

1. El primer punto que debe quedar claro en esta perspectiva es que la contribución del trabajo es proponer un Modelo Vehículo-Conductor que empleara más acciones del conductor (entradas del modelo) que las halladas en los trabajos del estado del arte, para predecir el comportamiento rotacional del vehículo, específicamente de dos variables: ángulo de deslizamiento lateral (slip angle) y ángulo de balanceo (ángulo roll). Además, dicho modelo se validó y construyó con datos experimentales y a partir de técnicas de Inteligencia Artificial (o Machine Learning).
2. De las cinco variables de salida, para todos los sujetos la señal de *tasa de cambio del ángulo de cabeceo (pitch rate)* se deduce se trató de ruido. Por lo que con ninguna técnica ML y caso respectivos se obtuvieron buenas predicciones. Una forma de solucionar ello en un futuro sería con una diferente disposición de las antenas del VBOX3i.
3. Con respecto a las salidas que interesan para la parte de la evaluación del riesgo de derrape lateral y de vuelco, a saber: ángulo de deslizamiento lateral (slip angle) y ángulo de balanceo (ángulo roll), para los sujetos 2 y 3 se obtuvieron resultados muy buenos. Estos resultados tuvieron una conducción *cauta*, por lo que su variabilidad de datos fue poca, desembocando en un buen desempeño del modelo ML. El Modelo Vehículo-Conductor propuesto se desempeñó de manera satisfactoria.
4. Por otra parte, para los sujetos 1 y 4, lo declarado en el punto anterior no puede aplicar. Las predicciones fueron buenas para el ángulo de balanceo (ángulo roll), pero fallaron para el ángulo de deslizamiento lateral (slip angle). Ello se atribuye al estilo de manejo pero también a la tasa de variación de la variable respecto al tiempo. El ángulo de balanceo (ángulo roll) cambia muy lento respecto al tiempo; por otra parte, el ángulo de deslizamiento lateral (slip angle) cambia de manera muy rápida. Entonces, para este último la variabilidad de los datos fue mayor, lo que obstaculizó el rendimiento del modelo ML. Esta situación es mucho más marcada para el sujeto 1 por su conducción técnica. Para tratar de corregir lo anterior, primero, podrían recabarse más pruebas por conductor o realizarse diferentes experimentos; si ello no funciona, implementar

otros modelos ML o de pre-procesamiento de información; y si tampoco eso funciona, irse al caso más radical, proponer otro Modelo Vehículo-Conductor (distintas entradas y salidas) con diferentes variables, provenientes de diversos sensores a bordo del vehículo o externos.

5. *El ángulo de deslizamiento lateral (slip angle) es un indicador clave para distinguir entre un estilo de conducción técnico/abrupto de uno seguro/cauto.*

6.1.3. Conclusiones Generales

En esta sección se mencionan las deducciones que son aplicables a ambas perspectivas, tanto para Estrategias de Manejo de Energía (EMSs) como para Sistemas Avanzados de Asistencia a la Conducción (ADASs). A continuación se mencionan dichas conclusiones:

1. Los Modelos Vehículo-Conductor tienen la desventaja de que no son generalizables para todos los conductores, es decir, son personalizados; sin embargo, el grado de exactitud o desempeño de los mismos supera la anterior limitación.
2. Las Técnicas de Inteligencia Artificial, o las Técnicas ML, funcionan como una *Caja Negra*. Como se menciona en [93], “el problema no sólo puede encontrarse en la interpretación sino también en el proceso de selección, en la documentación vinculada en el desarrollo, a la selección de features...” Los modelos ML están basados en fundamento matemático, pero no se sabe qué pase dentro. Eso está en continua investigación, y se espera en un futuro se puedan explicar muchas cosas que ahorita no se entienden de los algoritmos. Al punto que se va es que, mientras hay autores que mencionan que cierto modelo ML funcionó mejor que otro para su aplicación, para otra práctica se pueden invertir los resultados. Y no es que un modelo ML sea mejor que otro, sino que para *la aplicación en específico* fue mejor que aquél. Esto deriva en que las técnicas basadas en modelos se sigan utilizando, ya que requieren menos datos y porque tienen un comportamiento más estable (un ejemplo es el filtro kalman). La ventaja de los modelos ML es que pueden manejar una infinidad de datos y modelar relaciones no lineales, pero su capacidad de generalización es muy dependiente de los datos que se utilizaron durante el entrenamiento. Recientemente investigadores ya han adoptado mecanismos híbridos basados en la fusión de ambas técnicas, lo que potencia el rendimiento de las propuestas.
3. Esta tesis se enfocó en los algoritmos ML con miras a dos enfoques (EMSs y ADASs), pero no hacia la implementación en tiempo real / en línea o hacia las estrategias de control. Sólo se limitó a la predicción de los estados de la dinámica vehicular.

6.2. Resumen de Resultados

Se considera muy pertinente la finalización de esta tesis con la exposición de los mejores modelos ML que representaron de la mejor manera, según los criterios de desempeño, los Modelos Vehículo-Conductor para ambos escenarios y para todos los sujetos.

6.2.1. Estrategias de Manejo de Energía

La Tabla 6.1 muestra los modelos ML que mejor representaron los Modelos Vehículo-Conductor para cada sujeto, así como el respectivo caso. Esta tabla condensa todo lo trabajado durante esta tesis para el ámbito de Estrategias de Manejo de Energía.

Tabla 6.1: Estrategias de Manejo de Energía. Resumen de Resultados.

Sujeto	Técnica de Aprendizaje Automático más representativa del Modelo Vehículo-Conductor	Caso	R2
1	LSTM Vainilla	2	0.993
2	LSTM Vainilla	1	0.995
3	LSTM Vainilla	1	0.990
4	LSTM Vainilla	2	0.994

6.2.2. Sistemas Avanzados de Asistencia a la Conducción

La Tabla 6.2 muestra los modelos ML que mejor representaron los Modelos Vehículo-Conductor para cada sujeto, así como el respectivo caso, para el escenario de Sistemas Avanzados de Asistencia a la Conducción. Esta tabla condensa todo lo trabajado durante esta tesis para este rubro.

Tabla 6.2: Sistemas Avanzados de Asistencia a la Conducción. Resumen de Resultados.

Sujeto	Técnica de Aprendizaje Automático más representativa del Modelo Vehículo-Conductor	Caso	Norma L2 R2
1	GRU	14	1.563
2	GRU	11	1.799
3	LSTM Encoder-Decoder	12	1.708
4	LSTM Vainilla	11	1.597

6.3. Prospectivas

Para las prospectivas se dividirá en dos rubros según el escenario tratado: Estrategias de Manejo de Energía y Sistemas Avanzados de Asistencia a la Conducción.

6.3.1. Estrategias de Manejo de Energía

- Esta línea de investigación tiene un futuro muy prometedor, la cuestión aquí es trabajar con horizontes de predicción amplios.
- El usar algoritmos como Modelos Ocultos de Markov puede ser un buen elemento combinatorio para los algoritmos ML trabajados en esta tesis; ello con el motivo de aumentar el poder de predicción.
- El empleo de otros modelos ML puede ser una tarea muy interesante de abordar en un futuro. Hay una variedad de técnicas o combinaciones de éstas que pudieran aportar mejores / peores / semejantes resultados a los presentados en esta tesis. Lo importante es probar nuevos métodos.
- Con respecto a la experimentación valdría la pena recolectar datos de un vehículo de una manera más natural o en entornos más reales que como se trabajó en esta tesis, que fue un circuito cerrado.
- Complementando el punto anterior, realizar experimentación con un vehículo híbrido sería una gran prospectiva.
- Probar con otras variables de los Modelos Vehículos-Conductor también es otra idea que puede resultar interesante de aplicar.
- El uso de un ordenador más potente o de GPUs más poderosas que las empleadas, o la paralelización de la información mediante múltiples GPUs, también valdría la pena intentar.
- La presente tesis se limitó a un área de Estrategias de Manejo de Energía, la predicción de variables vehiculares a través de modelos ML. No obstante, resultaría muy interesante el poder abarcar más áreas como la implementación del control, la planificación de trayectorias y del perfil de velocidad en el vehículo.

6.3.2. Sistemas Avanzados de Asistencia a la Conducción

Para este escenario resultaría interesante implementar en un futuro alguna o varias de estas ideas para mejorar los resultados, en caso de que se quisiera seguir en esta línea de investigación:

- Probar con otro tipo de modelos ML, además de otras variables que definan el Modelo Vehículo-Conductor para este escenario.
- Realizar experimentación con mayor número de sujetos y mayor número de pruebas cada uno, además de definir mejor los requisitos para participación y las condiciones del experimento.

- Cubrir las deficiencias que se encontraron en la formulación de Modelos Vehículo-Conductor para operadores con conducción técnica o acciones precipitadas. Para tratar de enmendar lo anterior, primero, podrían recabarse más pruebas por conductor o realizarse diferentes experimentos; si ello no funciona, implementar otros modelos ML o de pre-procesamiento de información; y si tampoco eso funciona, irse al caso más radical, proponer otro Modelo Vehículo-Conductor (distintas entradas y salidas) con diferentes variables, provenientes de diversos sensores a bordo del vehículo o externos.
- La presente tesis se limitó a un área de Sistemas Avanzados de Asistencia a la Conducción, la predicción de los estados de la dinámica vehicular dadas ciertas entradas del conductor a través de modelos ML. No obstante, resultaría muy interesante el poder abarcar más áreas como la evaluación del riesgo de accidentes, la implementación de estrategias de control para mitigarlos, entre otras.

Anexo A

Resultados. Estrategias de Manejo de Energía

Tabla A.1: Resultados Técnicas ML. Estrategias de Manejo de Energía. Todos los Sujetos.

Sujeto	Técnica Aprendizaje Automático (Técnica ML)	RMSE				R2			
		C1	C2	C3	C4	C1	C2	C3	C4
Sujeto 1	GRU	0.040	0.043	0.046	0.057	0.989	0.987	0.985	0.978
	LSTM Vainilla	0.038	0.033	0.037	0.038	0.990	0.993	0.991	0.990
	Transformer	0.045	0.052	0.043	0.072	0.986	0.981	0.987	0.965
Sujeto 2	GRU	0.044	0.050	0.049	0.064	0.988	0.984	0.985	0.973
	LSTM Vainilla	0.028	0.031	0.035	0.039	0.995	0.994	0.992	0.991
	Transformer	0.059	0.045	0.049	0.059	0.978	0.987	0.985	0.978
Sujeto 3	GRU	0.053	0.056	0.056	0.066	0.986	0.984	0.984	0.978
	LSTM Vainilla	0.045	0.049	0.051	0.053	0.990	0.988	0.987	0.986
	Transformer	0.048	0.071	0.066	0.091	0.988	0.975	0.978	0.958
Sujeto 4	GRU	0.034	0.036	0.039	0.074	0.986	0.985	0.982	0.936
	LSTM Vainilla	0.027	0.023	0.031	0.032	0.992	0.994	0.989	0.988
	Transformer	0.032	0.049	0.043	0.065	0.988	0.972	0.979	0.951

Anexo B

Resultados. Sistemas Avanzados de Asistencia a la Conducción

Tabla B.1: Sujeto 1. Comparación de las Técnicas de Aprendizaje Automático según los dos Criterios de Desempeño: RMSE y R2.

CASO	TÉCNICAS DE APRENDIZAJE AUTOMÁTICO (TÉCNICAS ML)	RMSE						Norma L2	R2					
		Ángulo de Viraje (Ángulo Yaw) / Ángulo de Deslizamiento Lateral (Slip Angle)	Ángulo de Balanceo (Ángulo Roll)	Tasa de Cambio del Ángulo de Viraje (o Yaw Rate)	Tasa de Cambio del Ángulo de Cabeceo (o Pitch Rate)	Tasa de Cambio del Ángulo de Balanceo (o Roll Rate)	Ángulo de Viraje (Ángulo Yaw) / Ángulo de Deslizamiento Lateral (Slip Angle)		Ángulo de Balanceo (Ángulo Roll)	Tasa de Cambio del Ángulo de Viraje (o Yaw Rate)	Tasa de Cambio del Ángulo de Cabeceo (o Pitch Rate)	Tasa de Cambio del Ángulo de Balanceo (o Roll Rate)	Norma L2	
C1	GRU	0.054	0.071	0.074	0.067	0.062	0.147	0.595	0.959	0.942	0.389	0.820	1.728	
	LSTM Vainilla	0.060	0.072	0.079	0.073	0.062	0.156	0.491	0.958	0.934	0.281	0.819	1.667	
	LSTM Encoder-Decoder	0.071	0.065	0.069	0.071	0.057	0.149	0.292	0.966	0.949	0.316	0.846	1.654	
C2	GRU	0.049	0.070	0.058	0.069	0.057	0.137	0.655	0.960	0.964	0.347	0.848	1.766	
	LSTM Vainilla	0.066	0.064	0.071	0.070	0.057	0.147	0.386	0.967	0.947	0.329	0.848	1.675	
	LSTM Encoder-Decoder	0.070	0.068	0.063	0.073	0.055	0.148	0.306	0.962	0.958	0.275	0.857	1.658	
C3	GRU	0.051	0.076	0.065	0.067	0.058	0.143	0.631	0.953	0.955	0.388	0.844	1.755	
	LSTM Vainilla	0.063	0.066	0.068	0.072	0.059	0.147	0.444	0.964	0.951	0.296	0.836	1.678	
	LSTM Encoder-Decoder	0.070	0.074	0.074	0.075	0.059	0.158	0.319	0.955	0.943	0.232	0.839	1.631	
C4	GRU	0.050	0.070	0.056	0.062	0.052	0.131	0.651	0.960	0.967	0.472	0.875	1.808	
	LSTM Vainilla	0.061	0.059	0.066	0.067	0.054	0.138	0.474	0.972	0.954	0.394	0.862	1.726	
	LSTM Encoder-Decoder	0.068	0.067	0.062	0.073	0.054	0.146	0.344	0.964	0.959	0.278	0.863	1.670	
C5	GRU	0.055	0.060	0.057	0.061	0.050	0.127	0.574	0.970	0.966	0.497	0.883	1.798	
	LSTM Vainilla	0.059	0.068	0.063	0.068	0.054	0.140	0.515	0.963	0.959	0.382	0.863	1.732	
	LSTM Encoder-Decoder	0.068	0.062	0.064	0.074	0.054	0.145	0.351	0.968	0.957	0.252	0.862	1.668	
C6	GRU	0.059	0.083	0.111	0.074	0.069	0.181	0.511	0.944	0.871	0.265	0.774	1.606	
	LSTM Vainilla	0.065	0.079	0.105	0.082	0.068	0.181	0.399	0.949	0.885	0.078	0.783	1.569	
	LSTM Encoder-Decoder	0.079	0.077	0.104	0.079	0.065	0.183	0.135	0.952	0.886	0.143	0.799	1.539	
C7	GRU	0.063	0.085	0.104	0.069	0.067	0.177	0.437	0.941	0.886	0.348	0.789	1.614	
	LSTM Vainilla	0.065	0.079	0.091	0.080	0.063	0.170	0.408	0.949	0.913	0.139	0.815	1.608	
	LSTM Encoder-Decoder	0.076	0.084	0.105	0.078	0.070	0.187	0.180	0.943	0.883	0.162	0.770	1.523	
C8	GRU	0.061	0.084	0.100	0.068	0.066	0.173	0.474	0.943	0.895	0.380	0.795	1.641	
	LSTM Vainilla	0.072	0.085	0.103	0.076	0.067	0.182	0.281	0.941	0.890	0.213	0.790	1.557	
	LSTM Encoder-Decoder	0.075	0.082	0.110	0.083	0.068	0.190	0.207	0.946	0.873	0.062	0.782	1.522	
C9	GRU	0.057	0.082	0.088	0.067	0.061	0.161	0.544	0.945	0.920	0.386	0.827	1.694	
	LSTM Vainilla	0.068	0.085	0.115	0.075	0.072	0.190	0.352	0.941	0.862	0.242	0.758	1.544	
	LSTM Encoder-Decoder	0.075	0.078	0.102	0.082	0.064	0.181	0.216	0.951	0.893	0.089	0.808	1.552	
C10	GRU	0.063	0.083	0.082	0.063	0.059	0.158	0.458	0.944	0.930	0.468	0.841	1.701	
	LSTM Vainilla	0.073	0.091	0.129	0.073	0.078	0.205	0.256	0.933	0.828	0.287	0.717	1.489	
	LSTM Encoder-Decoder	0.085	0.089	0.129	0.082	0.077	0.211	-0.003	0.936	0.827	0.097	0.726	1.448	
C11	GRU	0.068	0.106	0.156	0.072	0.087	0.231	0.349	0.908	0.744	0.291	0.645	1.414	
	LSTM Vainilla	0.077	0.088	0.157	0.081	0.086	0.229	0.164	0.936	0.743	0.106	0.656	1.377	
	LSTM Encoder-Decoder	0.073	0.117	0.190	0.082	0.101	0.268	0.246	0.889	0.623	0.089	0.527	1.235	
C12	GRU	0.074	0.098	0.180	0.069	0.096	0.248	0.238	0.923	0.661	0.352	0.566	1.338	
	LSTM Vainilla	0.073	0.090	0.164	0.079	0.088	0.233	0.260	0.934	0.719	0.148	0.637	1.373	
	LSTM Encoder-Decoder	0.075	0.097	0.171	0.081	0.093	0.244	0.207	0.923	0.692	0.105	0.597	1.319	
C13	GRU	0.066	0.095	0.135	0.067	0.077	0.205	0.389	0.927	0.810	0.401	0.722	1.532	
	LSTM Vainilla	0.074	0.089	0.138	0.080	0.078	0.212	0.244	0.936	0.802	0.138	0.720	1.455	
	LSTM Encoder-Decoder	0.078	0.096	0.116	0.082	0.071	0.201	0.151	0.926	0.860	0.092	0.768	1.489	
C14	GRU	0.067	0.095	0.116	0.071	0.071	0.193	0.385	0.928	0.861	0.326	0.766	1.563	
	LSTM Vainilla	0.077	0.092	0.141	0.072	0.079	0.214	0.187	0.931	0.794	0.312	0.715	1.464	
	LSTM Encoder-Decoder	0.083	0.102	0.153	0.075	0.085	0.232	0.048	0.917	0.758	0.237	0.667	1.385	
C15	GRU	0.069	0.093	0.131	0.067	0.077	0.203	0.344	0.930	0.826	0.399	0.727	1.534	
	LSTM Vainilla	0.073	0.099	0.136	0.079	0.079	0.214	0.280	0.922	0.812	0.176	0.718	1.461	
	LSTM Encoder-Decoder	0.082	0.088	0.130	0.085	0.073	0.209	0.084	0.938	0.829	0.029	0.754	1.464	

Tabla B.2: Sujeto 2. Comparación de las Técnicas de Aprendizaje Automático según los dos Criterios de Desempeño: RMSE y R2.

CASO	TÉCNICAS DE APRENDIZAJE AUTOMÁTICO (TÉCNICAS ML)	RMSE					Norma L2	R2					Norma L2
		Ángulo de Viraje (Ángulo Yaw) / Ángulo de Deslizamiento Lateral (Slip Angle)	Ángulo de Balanceo (Ángulo Roll)	Tasa de Cambio del Ángulo de Viraje (o Yaw Rate)	Tasa de Cambio del Ángulo de Cabeceo (o Pitch Rate)	Tasa de Cambio del Ángulo de Balanceo (o Roll Rate)		Ángulo de Viraje (Ángulo Yaw) / Ángulo de Deslizamiento Lateral (Slip Angle)	Ángulo de Balanceo (Ángulo Roll)	Tasa de Cambio del Ángulo de Viraje (o Yaw Rate)	Tasa de Cambio del Ángulo de Cabeceo (o Pitch Rate)	Tasa de Cambio del Ángulo de Balanceo (o Roll Rate)	
C1	GRU	0.030	0.044	0.045	0.060	0.074	0.118	0.932	0.985	0.958	0.196	0.884	1.891
	LSTM Vainilla	0.034	0.043	0.049	0.062	0.073	0.121	0.913	0.985	0.951	0.155	0.884	1.874
	LSTM Encoder-Decoder	0.033	0.046	0.048	0.064	0.074	0.122	0.919	0.983	0.953	0.100	0.882	1.874
C2	GRU	0.027	0.047	0.044	0.059	0.070	0.116	0.945	0.982	0.960	0.217	0.894	1.904
	LSTM Vainilla	0.029	0.049	0.050	0.060	0.078	0.125	0.937	0.981	0.949	0.193	0.869	1.879
	LSTM Encoder-Decoder	0.031	0.048	0.048	0.062	0.076	0.124	0.927	0.982	0.952	0.137	0.875	1.874
C3	GRU	0.032	0.043	0.039	0.060	0.065	0.111	0.923	0.985	0.969	0.193	0.908	1.904
	LSTM Vainilla	0.031	0.044	0.049	0.060	0.074	0.120	0.930	0.984	0.951	0.197	0.883	1.886
	LSTM Encoder-Decoder	0.029	0.045	0.044	0.064	0.070	0.118	0.935	0.984	0.961	0.080	0.895	1.890
C4	GRU	0.029	0.039	0.041	0.058	0.065	0.108	0.937	0.988	0.966	0.262	0.909	1.919
	LSTM Vainilla	0.030	0.047	0.055	0.062	0.083	0.130	0.932	0.982	0.937	0.134	0.854	1.860
	LSTM Encoder-Decoder	0.030	0.045	0.043	0.065	0.068	0.116	0.932	0.984	0.963	0.073	0.900	1.892
C5	GRU	0.030	0.041	0.038	0.055	0.065	0.106	0.932	0.987	0.970	0.335	0.909	1.930
	LSTM Vainilla	0.029	0.044	0.052	0.064	0.077	0.125	0.936	0.984	0.944	0.087	0.872	1.873
	LSTM Encoder-Decoder	0.030	0.043	0.045	0.063	0.070	0.117	0.932	0.986	0.959	0.114	0.895	1.891
C6	GRU	0.047	0.062	0.060	0.065	0.078	0.141	0.833	0.969	0.927	0.071	0.870	1.804
	LSTM Vainilla	0.052	0.065	0.067	0.064	0.085	0.151	0.800	0.966	0.909	0.077	0.846	1.767
	LSTM Encoder-Decoder	0.048	0.063	0.063	0.067	0.081	0.146	0.830	0.969	0.919	0.008	0.857	1.791
C7	GRU	0.054	0.064	0.059	0.065	0.081	0.145	0.786	0.968	0.929	0.076	0.860	1.778
	LSTM Vainilla	0.059	0.070	0.056	0.067	0.080	0.150	0.744	0.961	0.936	0.008	0.862	1.760
	LSTM Encoder-Decoder	0.050	0.065	0.055	0.068	0.077	0.143	0.815	0.966	0.938	-0.026	0.874	1.800
C8	GRU	0.044	0.059	0.056	0.063	0.078	0.137	0.852	0.972	0.936	0.113	0.870	1.822
	LSTM Vainilla	0.052	0.065	0.067	0.068	0.087	0.154	0.795	0.966	0.909	-0.037	0.837	1.759
	LSTM Encoder-Decoder	0.036	0.056	0.055	0.069	0.075	0.134	0.902	0.975	0.938	-0.045	0.878	1.849
C9	GRU	0.041	0.060	0.058	0.058	0.080	0.136	0.874	0.971	0.931	0.248	0.863	1.839
	LSTM Vainilla	0.047	0.067	0.060	0.064	0.082	0.145	0.836	0.965	0.928	0.083	0.857	1.798
	LSTM Encoder-Decoder	0.040	0.058	0.055	0.068	0.074	0.135	0.883	0.973	0.938	-0.021	0.882	1.840
C10	GRU	0.040	0.055	0.054	0.059	0.077	0.130	0.880	0.976	0.941	0.224	0.874	1.851
	LSTM Vainilla	0.044	0.065	0.061	0.063	0.087	0.146	0.856	0.966	0.924	0.133	0.840	1.801
	LSTM Encoder-Decoder	0.040	0.065	0.053	0.068	0.076	0.138	0.882	0.966	0.943	-0.023	0.877	1.836
C11	GRU	0.046	0.070	0.061	0.065	0.081	0.146	0.845	0.961	0.925	0.063	0.860	1.799
	LSTM Vainilla	0.057	0.083	0.087	0.067	0.101	0.180	0.755	0.946	0.846	0.011	0.782	1.671
	LSTM Encoder-Decoder	0.046	0.085	0.063	0.069	0.084	0.159	0.844	0.943	0.918	-0.066	0.848	1.780
C12	GRU	0.060	0.074	0.082	0.063	0.098	0.171	0.729	0.957	0.864	0.115	0.794	1.684
	LSTM Vainilla	0.053	0.077	0.072	0.067	0.089	0.162	0.790	0.953	0.894	0.016	0.829	1.738
	LSTM Encoder-Decoder	0.051	0.075	0.075	0.067	0.090	0.163	0.803	0.955	0.885	-0.008	0.825	1.738
C13	GRU	0.059	0.068	0.078	0.060	0.092	0.162	0.744	0.963	0.877	0.203	0.818	1.721
	LSTM Vainilla	0.063	0.078	0.087	0.068	0.103	0.181	0.703	0.951	0.848	-0.012	0.776	1.649
	LSTM Encoder-Decoder	0.058	0.078	0.075	0.067	0.091	0.167	0.746	0.952	0.886	0.001	0.824	1.711
C14	GRU	0.056	0.072	0.066	0.057	0.083	0.151	0.770	0.959	0.911	0.280	0.853	1.774
	LSTM Vainilla	0.055	0.078	0.068	0.066	0.086	0.160	0.777	0.952	0.907	0.034	0.842	1.744
	LSTM Encoder-Decoder	0.051	0.082	0.074	0.067	0.090	0.166	0.805	0.947	0.889	0.014	0.828	1.738
C15	GRU	0.062	0.078	0.067	0.059	0.084	0.158	0.717	0.952	0.910	0.221	0.850	1.738
	LSTM Vainilla	0.059	0.070	0.073	0.064	0.092	0.162	0.742	0.962	0.893	0.096	0.824	1.721
	LSTM Encoder-Decoder	0.050	0.071	0.071	0.066	0.089	0.157	0.818	0.960	0.900	0.034	0.835	1.760

Tabla B.3: Sujeto 3. Comparación de las Técnicas de Aprendizaje Automático según los dos Criterios de Desempeño: RMSE y R2.

CASO	TÉCNICAS DE APRENDIZAJE AUTOMÁTICO (TÉCNICAS ML)	RMSE					Norma L2	R2					Norma L2
		Ángulo de Viraje (Ángulo Yaw) / Ángulo de Deslizamiento Lateral (Slip Angle)	Ángulo de Balanceo (Ángulo Roll)	Tasa de Cambio del Ángulo de Viraje (o Yaw Rate)	Tasa de Cambio del Ángulo de Cabeceo (o Pitch Rate)	Tasa de Cambio del Ángulo de Balanceo (o Roll Rate)		Ángulo de Viraje (Ángulo Yaw) / Ángulo de Deslizamiento Lateral (Slip Angle)	Ángulo de Balanceo (Ángulo Roll)	Tasa de Cambio del Ángulo de Viraje (o Yaw Rate)	Tasa de Cambio del Ángulo de Cabeceo (o Pitch Rate)	Tasa de Cambio del Ángulo de Balanceo (o Roll Rate)	
C1	GRU	0.021	0.053	0.064	0.116	0.105	0.178	0.912	0.979	0.957	0.120	0.866	1.863
	LSTM Vainilla	0.022	0.064	0.065	0.126	0.111	0.192	0.903	0.968	0.956	-0.036	0.851	1.842
	LSTM Encoder-Decoder	0.022	0.064	0.063	0.122	0.103	0.185	0.900	0.968	0.959	0.021	0.871	1.851
C2	GRU	0.020	0.054	0.065	0.118	0.104	0.180	0.919	0.977	0.957	0.091	0.869	1.865
	LSTM Vainilla	0.022	0.051	0.072	0.122	0.108	0.187	0.896	0.980	0.947	0.023	0.859	1.843
	LSTM Encoder-Decoder	0.023	0.070	0.071	0.119	0.108	0.191	0.889	0.962	0.947	0.077	0.859	1.832
C3	GRU	0.021	0.055	0.061	0.119	0.098	0.176	0.908	0.976	0.961	0.077	0.884	1.868
	LSTM Vainilla	0.022	0.056	0.077	0.123	0.111	0.192	0.900	0.976	0.939	0.017	0.850	1.835
	LSTM Encoder-Decoder	0.022	0.073	0.071	0.127	0.105	0.195	0.902	0.959	0.947	-0.048	0.865	1.839
C4	GRU	0.021	0.058	0.062	0.116	0.098	0.175	0.907	0.974	0.960	0.121	0.884	1.868
	LSTM Vainilla	0.021	0.075	0.081	0.126	0.116	0.205	0.909	0.957	0.932	-0.042	0.836	1.819
	LSTM Encoder-Decoder	0.023	0.072	0.073	0.123	0.106	0.193	0.892	0.960	0.945	0.010	0.865	1.832
C5	GRU	0.019	0.053	0.061	0.114	0.096	0.171	0.923	0.979	0.962	0.145	0.889	1.883
	LSTM Vainilla	0.021	0.059	0.072	0.127	0.107	0.191	0.909	0.973	0.947	-0.051	0.863	1.848
	LSTM Encoder-Decoder	0.019	0.062	0.070	0.122	0.103	0.186	0.924	0.970	0.950	0.031	0.871	1.859
C6	GRU	0.026	0.070	0.099	0.128	0.123	0.217	0.859	0.962	0.898	-0.077	0.817	1.773
	LSTM Vainilla	0.030	0.098	0.129	0.130	0.150	0.258	0.813	0.925	0.827	-0.115	0.726	1.656
	LSTM Encoder-Decoder	0.029	0.084	0.100	0.126	0.125	0.222	0.827	0.946	0.897	-0.039	0.810	1.744
C7	GRU	0.024	0.076	0.093	0.126	0.121	0.214	0.884	0.956	0.910	-0.040	0.823	1.789
	LSTM Vainilla	0.028	0.095	0.103	0.131	0.130	0.233	0.832	0.930	0.891	-0.115	0.797	1.732
	LSTM Encoder-Decoder	0.027	0.088	0.101	0.126	0.128	0.225	0.850	0.940	0.894	-0.029	0.804	1.747
C8	GRU	0.025	0.071	0.095	0.125	0.114	0.208	0.867	0.961	0.907	-0.018	0.842	1.791
	LSTM Vainilla	0.029	0.097	0.136	0.134	0.153	0.265	0.821	0.928	0.808	-0.171	0.719	1.654
	LSTM Encoder-Decoder	0.029	0.073	0.106	0.129	0.126	0.223	0.831	0.959	0.884	-0.090	0.808	1.748
C9	GRU	0.028	0.079	0.107	0.120	0.126	0.221	0.834	0.952	0.883	0.060	0.810	1.744
	LSTM Vainilla	0.026	0.092	0.121	0.131	0.143	0.247	0.865	0.935	0.851	-0.118	0.755	1.712
	LSTM Encoder-Decoder	0.031	0.095	0.131	0.128	0.149	0.256	0.808	0.931	0.823	-0.063	0.734	1.655
C10	GRU	0.030	0.074	0.125	0.121	0.138	0.236	0.817	0.958	0.840	0.046	0.772	1.700
	LSTM Vainilla	0.029	0.096	0.145	0.134	0.162	0.274	0.823	0.928	0.784	-0.157	0.687	1.628
	LSTM Encoder-Decoder	0.028	0.093	0.124	0.130	0.137	0.246	0.836	0.934	0.844	-0.093	0.775	1.700
C11	GRU	0.031	0.080	0.123	0.134	0.143	0.247	0.801	0.950	0.843	-0.178	0.755	1.690
	LSTM Vainilla	0.044	0.094	0.150	0.135	0.164	0.280	0.601	0.931	0.767	-0.190	0.676	1.520
	LSTM Encoder-Decoder	0.031	0.089	0.132	0.134	0.146	0.256	0.796	0.939	0.820	-0.170	0.744	1.665
C12	GRU	0.028	0.087	0.129	0.131	0.147	0.252	0.836	0.942	0.829	-0.121	0.740	1.683
	LSTM Vainilla	0.034	0.113	0.142	0.128	0.159	0.275	0.756	0.901	0.793	-0.066	0.694	1.580
	LSTM Encoder-Decoder	0.025	0.110	0.116	0.128	0.138	0.248	0.869	0.907	0.862	-0.069	0.769	1.708
C13	GRU	0.029	0.087	0.118	0.127	0.134	0.238	0.825	0.942	0.856	-0.058	0.784	1.708
	LSTM Vainilla	0.031	0.102	0.132	0.127	0.150	0.260	0.796	0.920	0.820	-0.060	0.728	1.639
	LSTM Encoder-Decoder	0.032	0.093	0.137	0.130	0.152	0.262	0.789	0.933	0.807	-0.110	0.720	1.636
C14	GRU	0.034	0.089	0.133	0.129	0.148	0.255	0.760	0.939	0.819	-0.077	0.738	1.637
	LSTM Vainilla	0.035	0.120	0.173	0.128	0.181	0.308	0.746	0.890	0.692	-0.061	0.607	1.483
	LSTM Encoder-Decoder	0.033	0.109	0.177	0.129	0.185	0.308	0.779	0.909	0.680	-0.079	0.589	1.499
C15	GRU	0.032	0.088	0.145	0.123	0.155	0.262	0.795	0.940	0.786	0.027	0.717	1.627
	LSTM Vainilla	0.030	0.110	0.145	0.131	0.160	0.277	0.813	0.907	0.786	-0.110	0.698	1.613
	LSTM Encoder-Decoder	0.036	0.109	0.202	0.131	0.201	0.334	0.733	0.909	0.585	-0.107	0.518	1.409

Tabla B.4: Sujeto 4. Comparación de las Técnicas de Aprendizaje Automático según los dos Criterios de Desempeño: RMSE y R2.

CASO	TÉCNICAS DE APRENDIZAJE AUTOMÁTICO (TÉCNICAS ML)	RMSE						R2					
		Ángulo de Viraje (Ángulo Yaw) / Ángulo de Deslizamiento Lateral (Slip Angle)	Ángulo de Balanceo (Ángulo Roll)	Tasa de Cambio del Ángulo de Viraje (o Yaw Rate)	Tasa de Cambio del Ángulo de Cabeceo (o Pitch Rate)	Tasa de Cambio del Ángulo de Balanceo (o Roll Rate)	Norma L2	Ángulo de Viraje (Ángulo Yaw) / Ángulo de Deslizamiento Lateral (Slip Angle)	Ángulo de Balanceo (Ángulo Roll)	Tasa de Cambio del Ángulo de Viraje (o Yaw Rate)	Tasa de Cambio del Ángulo de Cabeceo (o Pitch Rate)	Tasa de Cambio del Ángulo de Balanceo (o Roll Rate)	Norma L2
C1	GRU	0.026	0.053	0.057	0.092	0.084	0.149	0.869	0.983	0.938	0.193	0.822	1.821
	LSTM Vainilla	0.026	0.056	0.062	0.093	0.089	0.155	0.864	0.981	0.927	0.180	0.801	1.800
	LSTM Encoder-Decoder	0.026	0.056	0.055	0.094	0.080	0.149	0.870	0.981	0.942	0.157	0.838	1.826
C2	GRU	0.027	0.056	0.064	0.092	0.091	0.157	0.854	0.980	0.922	0.197	0.791	1.790
	LSTM Vainilla	0.027	0.054	0.062	0.095	0.088	0.156	0.851	0.982	0.926	0.137	0.808	1.794
	LSTM Encoder-Decoder	0.028	0.055	0.063	0.095	0.088	0.156	0.850	0.981	0.924	0.132	0.808	1.792
C3	GRU	0.029	0.052	0.059	0.094	0.085	0.152	0.837	0.983	0.934	0.152	0.819	1.799
	LSTM Vainilla	0.028	0.054	0.061	0.099	0.088	0.158	0.842	0.982	0.929	0.073	0.805	1.786
	LSTM Encoder-Decoder	0.030	0.057	0.065	0.099	0.090	0.162	0.818	0.980	0.920	0.061	0.797	1.765
C4	GRU	0.029	0.050	0.051	0.092	0.078	0.143	0.833	0.985	0.950	0.195	0.849	1.824
	LSTM Vainilla	0.029	0.049	0.056	0.097	0.082	0.150	0.839	0.985	0.939	0.095	0.831	1.805
	LSTM Encoder-Decoder	0.029	0.050	0.063	0.096	0.089	0.156	0.835	0.984	0.924	0.122	0.802	1.783
C5	GRU	0.029	0.048	0.054	0.088	0.078	0.141	0.836	0.986	0.944	0.251	0.847	1.828
	LSTM Vainilla	0.027	0.052	0.059	0.098	0.083	0.153	0.853	0.983	0.934	0.070	0.826	1.804
	LSTM Encoder-Decoder	0.031	0.052	0.062	0.095	0.085	0.154	0.808	0.983	0.928	0.127	0.818	1.779
C6	GRU	0.039	0.076	0.099	0.104	0.108	0.199	0.706	0.964	0.813	-0.028	0.708	1.610
	LSTM Vainilla	0.034	0.073	0.094	0.101	0.103	0.191	0.770	0.967	0.830	0.027	0.733	1.660
	LSTM Encoder-Decoder	0.037	0.073	0.102	0.103	0.112	0.201	0.723	0.967	0.800	-0.017	0.688	1.604
C7	GRU	0.035	0.073	0.086	0.097	0.098	0.182	0.761	0.967	0.858	0.103	0.758	1.684
	LSTM Vainilla	0.036	0.077	0.086	0.099	0.101	0.186	0.743	0.964	0.859	0.057	0.743	1.665
	LSTM Encoder-Decoder	0.039	0.094	0.095	0.100	0.105	0.201	0.693	0.945	0.827	0.046	0.725	1.608
C8	GRU	0.037	0.068	0.087	0.099	0.102	0.184	0.733	0.972	0.854	0.056	0.741	1.663
	LSTM Vainilla	0.037	0.073	0.082	0.103	0.097	0.183	0.733	0.967	0.873	-0.022	0.764	1.679
	LSTM Encoder-Decoder	0.034	0.071	0.084	0.101	0.098	0.182	0.771	0.969	0.866	0.014	0.758	1.690
C9	GRU	0.038	0.071	0.094	0.094	0.107	0.189	0.710	0.969	0.833	0.131	0.715	1.633
	LSTM Vainilla	0.036	0.078	0.091	0.103	0.105	0.193	0.741	0.963	0.842	-0.034	0.723	1.646
	LSTM Encoder-Decoder	0.043	0.071	0.099	0.104	0.109	0.198	0.638	0.969	0.815	-0.052	0.707	1.585
C10	GRU	0.049	0.082	0.110	0.097	0.116	0.210	0.537	0.959	0.774	0.079	0.666	1.502
	LSTM Vainilla	0.037	0.071	0.092	0.098	0.106	0.189	0.729	0.969	0.841	0.052	0.722	1.644
	LSTM Encoder-Decoder	0.039	0.071	0.103	0.099	0.113	0.199	0.705	0.970	0.801	0.035	0.685	1.596
C11	GRU	0.041	0.084	0.104	0.102	0.113	0.207	0.673	0.956	0.793	0.010	0.679	1.568
	LSTM Vainilla	0.039	0.087	0.101	0.100	0.110	0.203	0.704	0.954	0.806	0.052	0.700	1.597
	LSTM Encoder-Decoder	0.037	0.088	0.110	0.103	0.114	0.212	0.727	0.953	0.771	-0.007	0.674	1.577
C12	GRU	0.050	0.087	0.128	0.102	0.128	0.231	0.512	0.953	0.690	0.004	0.590	1.412
	LSTM Vainilla	0.040	0.082	0.108	0.102	0.115	0.208	0.685	0.959	0.781	0.008	0.671	1.565
	LSTM Encoder-Decoder	0.040	0.080	0.104	0.101	0.110	0.203	0.684	0.961	0.794	0.015	0.697	1.584
C13	GRU	0.045	0.084	0.122	0.101	0.126	0.223	0.602	0.956	0.720	0.020	0.607	1.471
	LSTM Vainilla	0.044	0.085	0.124	0.104	0.127	0.226	0.617	0.956	0.708	-0.036	0.600	1.469
	LSTM Encoder-Decoder	0.044	0.090	0.124	0.104	0.128	0.229	0.627	0.950	0.709	-0.038	0.593	1.466
C14	GRU	0.045	0.095	0.142	0.098	0.139	0.245	0.605	0.945	0.623	0.066	0.521	1.387
	LSTM Vainilla	0.046	0.091	0.133	0.101	0.136	0.238	0.593	0.950	0.667	-0.004	0.541	1.411
	LSTM Encoder-Decoder	0.056	0.092	0.153	0.105	0.147	0.260	0.398	0.949	0.559	-0.074	0.467	1.263
C15	GRU	0.045	0.097	0.137	0.096	0.137	0.241	0.601	0.943	0.648	0.086	0.541	1.404
	LSTM Vainilla	0.049	0.090	0.130	0.102	0.130	0.234	0.540	0.951	0.682	-0.041	0.582	1.415
	LSTM Encoder-Decoder	0.043	0.091	0.119	0.102	0.123	0.223	0.642	0.950	0.737	-0.034	0.628	1.501

Bibliografía

- [1] Mohamed Afify. “RWIS BASED ROAD CONDITION PREDICTION USING MACHINE LEARNING ALGORITHMS”. En: (2021).
- [2] A Vijay Alagappan, KV Narasimha Rao y R Krishna Kumar. “A comparison of various algorithms to extract Magic Formula tyre model coefficients for vehicle dynamics simulations”. En: *Vehicle System Dynamics* 53.2 (2015), págs. 154-178.
- [3] Afshine Amidi y Shervine Amidi. *Recurrent Neural Networks cheatsheet* <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks>. Accessed: 2022-04-14. s.f.
- [4] Anónimo. http://www.face.ubiobio.cl/~jparra/Dinamica_Vehiculo.htm. s.f.
- [5] Martin Atzmueller. *Enterprise big data engineering, analytics, and management*. IGI Global, 2016.
- [6] Jihwan Bae y col. “Curriculum Learning for Vehicle Lateral Stability Estimations”. En: *IEEE Access* 9 (2021), págs. 89249-89262.
- [7] Pragati Baheti. *Train, Validation, and Test Sets: How to Split Your Machine Learning Data* <https://www.v7labs.com/blog/train-validation-test-set>. Accessed: 2022-04-14. 2022.
- [8] Bruno Henrique Groenner Barbosa y col. “Lateral force prediction using gaussian process regression for intelligent tire systems”. En: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2021).
- [9] David Berezowsky y Marcos Carrasco. *Status of Electromobility in Mexico* <https://www.jdsupra.com/legalnews/status-of-electromobility-in-mexico-5261956/>. Accessed: 2022-02-20. 2022.
- [10] Rozi Bibi y col. “Edge AI-based automated detection and classification of road anomalies in VANET using deep learning”. En: *Computational intelligence and neuroscience* 2021 (2021).

- [11] Sebastian Blume y col. “Neural roll angle estimation in a model predictive control system”. En: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE. 2019, págs. 1625-1630.
- [12] Beatriz L Boada y col. “A robust observer based on H filtering with parameter uncertainties combined with Neural Networks for estimation of vehicle roll angle”. En: *Mechanical Systems and Signal Processing* 99 (2018), págs. 611-623.
- [13] Beatriz L Boada y col. “Sideslip angle estimator based on ANFIS for vehicle handling and stability”. En: *Journal of Mechanical Science and Technology* 29.4 (2015), págs. 1473-1481.
- [14] Maria Jesus L Boada, Beatriz L Boada y Hui Zhang. “Event-triggering H_∞ -based observer combined with NN for simultaneous estimation of vehicle sideslip and roll angles with network-induced delays”. En: *Nonlinear Dynamics* 103.3 (2021), págs. 2733-2752.
- [15] Ali Boyali, Simon Thompson y David Robert Wong. “Identification of Vehicle Dynamics Parameters Using Simulation-based Inference”. En: *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. IEEE. 2021, págs. 306-312.
- [16] Jason Brownlee. *A Gentle Introduction to Long Short-Term Memory Networks by the Experts*. <https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/>. Accessed: 2022-04-14. 2021.
- [17] Jason Brownlee. *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning*. <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/#:~:text=Adam%20combines%20the%20best%20properties,do%20well%20on%20most%20problems..> Accessed: 2022-04-17. 2017.
- [18] Jason Brownlee. *Long short-term memory networks with python: develop sequence prediction models with deep learning*. Machine Learning Mastery, 2017.
- [19] Jason Brownlee. *Train-Test Split for Evaluating Machine Learning Algorithms* <https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/>. Accessed: 2022-04-14. 2020.
- [20] Jason Brownlee. *Why One-Hot Encode Data in Machine Learning?* <https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/>. Accessed: 2022-04-14. 2020.
- [21] Hyunsoo Cha y col. “A Comparative Study between the Parameter-Optimized Pacejka Model and Artificial Neural Network Model for Tire Force Estimation”. En: *Journal of Auto-vehicle Safety Association* 13.4 (2021), págs. 33-38.
- [22] Zheng Chen y col. “A hierarchical energy management strategy for power-split plug-in hybrid electric vehicles considering velocity prediction”. En: *IEEE Access* 6 (2018), págs. 33261-33274.

- [23] Zheng Chen y col. “Energy management strategy for power-split plug-in hybrid electric vehicle based on MPC and double Q-learning”. En: *Energy* (2022), pág. 123182.
- [24] Daniel Chindamo y Marco Gadola. “Estimation of vehicle side-slip angle using an artificial neural network”. En: *MATEC web of conferences*. Vol. 166. EDP Sciences. 2018, pág. 02001.
- [25] Daniel Chindamo, Basilio Lenzo y Marco Gadola. “On the vehicle sideslip angle estimation: a literature review of methods, models, and innovations”. En: *applied sciences* 8.3 (2018), pág. 355.
- [26] Daniel Chindamo y col. “Experimental Comparison of The Two Most Used Vehicle Sideslip Angle Estimation Methods for Model-Based Design Approach”. En: *Journal of Physics: Conference Series*. Vol. 1888. 1. IOP Publishing. 2021, pág. 012006.
- [27] Dongho Choi y col. “Machine learning-based vehicle trajectory prediction using v2v communications and on-board sensors”. En: *Electronics* 10.4 (2021), pág. 420.
- [28] *Cross-validation: evaluating estimator performance*. scikit-learn. 2022. URL: https://scikit-learn.org/stable/modules/cross_validation.html.
- [29] M D’Inverno y col. “A benchmark study on the model-based estimation of the go-kart sideslip angle”. En: *Journal of Physics: Conference Series*. Vol. 2090. 1. IOP Publishing. 2021, pág. 012156.
- [30] Saul Dobilas. *LSTM Recurrent Neural Networks — How to Teach a Network to Remember the Past*. <https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to-remember-the-past-55e54c2ff22e>. Accessed: 2022-04-14. 2022.
- [31] Rian Dolphin. *LSTM Networks | A Detailed Explanation* <https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9#:~:text=LSTMs\%20use\%20a\%20series\%20of,each\%20their\%20own\%20neural\%20network.> Accessed: 2022-04-14. 2020.
- [32] Ronghua Du y col. “Battery aging-and temperature-aware predictive energy management for hybrid electric vehicles”. En: *Journal of Power Sources* 473 (2020), pág. 228568.
- [33] edmunds. *Used 2008 Mazda 3 s Sport Specs Features*. URL: <https://www.edmunds.com/mazda/3/2008/st-100987490/features-specs/>.
- [34] Nevin Fawzy, Hany F Habib y Sohrab Mokhtari. “Performance evaluation of electric vehicle model under skid control technique”. En: *World Electric Vehicle Journal* 12.2 (2021), pág. 83.
- [35] Federal Highway Association. *Human Factors*. Accessed: 2022-04-14. 2021.

- [36] Estefanía Freire y Sarahí Silva. *Redes neuronales* <https://bootcampai.medium.com/redes-neuronales-13349dd1a5bb#:~:text=Funci%C3%B3n%20Lineal&text=Por%20lo%20tanto%2C%20esta%20funci%C3%B3n,de%20un%20n%C3%BAmero%20de%20ventas>. . Accessed: 2022-04-14. 2019.
- [37] Jun Fu y col. “A Real-Time Load Prediction Control for Fuel Cell Hybrid Vehicle”. En: *Energies* 15.10 (2022), pág. 3700.
- [38] Javier Garcia Guzman y col. “Real-time vehicle roll angle estimation based on neural networks in IoT low-cost devices”. En: *Sensors* 18.7 (2018), pág. 2188.
- [39] GeeksforGeeks. *Gated Recurrent Unit Networks*. 2022. URL: <https://www.geeksforgeeks.org/gated-recurrent-unit-networks/g>.
- [40] GeeksforGeeks. *ML | Label Encoding of datasets in Python* <https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/>. Accessed: 2022-04-14. 2022.
- [41] Thomas D Gillespie. *Fundamentals of vehicle dynamics*. Inf. téc. SAE Technical Paper, 1992.
- [42] Ligdi González. *Diferencia entre Machine Learning y la Programación Tradicional*. Accessed: 2022-04-14. 2020.
- [43] Lisardo Prieto González y col. “Simultaneous estimation of vehicle roll and sideslip angles through a deep learning approach”. En: *Sensors* 20.13 (2020), pág. 3679.
- [44] Alison Gordon y col. “Autonomous vehicle interaction control software and smart sustainable urban mobility behaviors in network connectivity systems”. En: *Contemporary Readings in Law and Social Justice* 13.1 (2021), págs. 40-49.
- [45] Torben Gräber y col. “A hybrid approach to side-slip angle estimation with recurrent neural networks and kinematic vehicle models”. En: *IEEE Transactions on Intelligent Vehicles* 4.1 (2018), págs. 39-47.
- [46] Grapheverywhere. <https://www.grapheverywhere.com/machine-learning-que-es-tipos-ejemplos-y-como-implementarlo/>. Accessed: 2022-02-17.
- [47] Maria Eusebia Guerrero-Sánchez y col. “Nonlinear control strategies for a UAV carrying a load with swing attenuation”. En: *Applied Mathematical Modelling* 91 (2021), págs. 709-722.
- [48] Kang Gui y col. “Road surface condition detection utilizing impedance spectroscopy and machine-learning algorithm”. En: *Life-Cycle Civil Engineering: Innovation, Theory and Practice*. CRC Press, 2021, págs. 1447-1451.
- [49] Ningyuan Guo y col. “Real-time predictive energy management of plug-in hybrid electric vehicles for coordination of fuel economy and battery degradation”. En: *Energy* 214 (2021), pág. 119070.

- [50] Jie Han y col. “Predictive energy management for plug-in hybrid electric vehicles considering electric motor thermal dynamics”. En: *Energy Conversion and Management* 251 (2022), pág. 115022.
- [51] Hongwen He y col. “An improved MPC-based energy management strategy for hybrid vehicles using V2V and V2I communications”. En: *Energy* 225 (2021), pág. 120273.
- [52] Tsung-Ming Hsu, Cheng-Hsien Wang y Yu-Rui Chen. “End-to-end deep learning for autonomous longitudinal and lateral control based on vehicle dynamics”. En: *Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality*. 2018, págs. 111-114.
- [53] Zhaoxia Huang. “Application of interval state estimation in vehicle control”. En: *Alexandria Engineering Journal* 61.1 (2022), págs. 911-916.
- [54] ICHI.PRO. <https://ichi.pro/es/que-es-el-conjunto-de-validacion-y-en-que-se-diferencia-del-conjunto-de-prueba-159013053745697>. Accessed: 2021-12-14.
- [55] Instituto Mexicano del Transporte. *Anuario estadístico de colisiones en carreteras federales 2020*. Accessed: 2022-04-14. 2021.
- [56] Instituto Mexicano del Transporte. <https://imt.mx/images/files/GRAL/documentos/Folleto-Pista-Pruebas.pdf>. Accessed: 2021-12-14.
- [57] Instituto Mexicano del Transporte. https://imt.mx/images/files/IVIE/Caracteristicas_Pista.pdf. Accessed: 2021-12-14. 2013.
- [58] Instituto Nacional de Salud Pública. *Consejos para prevenir accidentes viales. Gaceta INSP*. <http://gaceta.insp.mx/?vol=1>. Accessed: 2021-12-20. 2017.
- [59] Woongsun Jeon y col. “Simultaneous state estimation and tire model learning for autonomous vehicle applications”. En: *IEEE/ASME Transactions on Mechatronics* 26.4 (2021), págs. 1941-1950.
- [60] Xianjian Jin, Guodong Yin y Nan Chen. “Advanced estimation techniques for vehicle system dynamic state: A survey”. En: *Sensors* 19.19 (2019), pág. 4289.
- [61] Guo Jinquan y col. “A novel MPC-based adaptive energy management strategy in plug-in hybrid electric vehicles”. En: *Energy* 175 (2019), págs. 378-392.
- [62] Bernardo Murta Junqueira, Alessandro Correa Victorino y José Guilherme Coelho Baêta. “A Model-less Approach for Estimating Vehicles Sideslip Angle by a Neural Network Concept”. En: *2021 IEEE International Conference on Mechatronics (ICM)*. IEEE. 2021, págs. 1-6.
- [63] Seyed Mehran Kazemi y col. “Time2vec: Learning a vector representation of time”. En: *arXiv preprint arXiv:1907.05321* (2019).
- [64] Dongchan Kim y col. “Vehicle sideslip angle estimation using deep ensemble-based adaptive Kalman filter”. En: *Mechanical Systems and Signal Processing* 144 (2020), pág. 106862.

- [65] Hee-Un Kim y Tae-Suk Bae. “Deep learning-based GNSS network-based real-time kinematic improvement for autonomous ground vehicle navigation”. En: *Journal of Sensors* 2019 (2019).
- [66] Amin Habibnejad Korayem, Amir Khajepour y Baris Fidan. “A review on vehicle-trailer state and parameter estimation”. En: *IEEE Transactions on intelligent transportation systems* (2021).
- [67] Simeon Kostadinov. <https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be>. Accessed: 2021-12-14.
- [68] Dhruva Krishna. <https://towardsdatascience.com/the-components-of-a-neural-network-af6244493b5b>. Accessed: 2021-12-14.
- [69] Hiroaki Kuwahara y Toshiyuki Murakami. “Tracked Vehicle Velocity Estimation by Disturbance Observer and Machine Learning, and its Application to Driving Force Control for Slippage Suppression”. En: *IEEJ Journal of Industry Applications* (2021), pág. 21002955.
- [70] Ji Li y col. “Dual-loop online intelligent programming for driver-oriented predict energy management of plug-in hybrid electric vehicles”. En: *Applied Energy* 253 (2019), pág. 113617.
- [71] Lin Li y col. “Velocity prediction based on vehicle lateral risk assessment and traffic flow: A brief review and application examples”. En: *Energies* 14.12 (2021), pág. 3431.
- [72] Menglin Li y col. “Data-driven predictive energy management and emission optimization for hybrid electric buses considering speed and passengers prediction”. En: *Journal of Cleaner Production* 304 (2021), pág. 127139.
- [73] Yuran Liang y col. “Vehicle side-slip angle estimation with deep neural network and sensor data fusion”. En: *10th International Munich Chassis Symposium 2019*. Springer. 2020, págs. 159-178.
- [74] Martin R Licea e Ilse Cervantes. “Robust indirect-defined envelope control for rollover and lateral skid prevention”. En: *Control Engineering Practice* 61 (2017), págs. 149-162.
- [75] Xinyou Lin, Songrong Zeng y Xuefan Li. “Online correction predictive energy management strategy using the Q-learning based swarm optimization with fuzzy neural network”. En: *Energy* 223 (2021), pág. 120071.
- [76] Haorui Liu y col. “Soft sensor of vehicle state estimation based on the kernel principal component and improved neural network”. En: *Journal of Sensors* 2016 (2016).
- [77] Kuan Liu y col. *Vehicle velocity prediction and energy management strategy part 1: Deterministic and stochastic vehicle velocity prediction using machine learning*. Inf. téc. SAE Technical Paper, 2019.
- [78] Youssef Nait Malek y col. “Multivariate deep learning approach for electric vehicle speed forecasting”. En: *Big Data Mining and Analytics* 4.1 (2021), págs. 56-64.

- [79] Markets and Markets. *Hybrid Vehicle Market by Electric Powertrain (Parallel, Series), Degree of Hybridization (Full, Micro, and Mild), Propulsion (HEV, PHEV, and NGV), Vehicle Type (PC, CV), Component (Battery, Electric Motor, and Transmission), and Region - Global Forecast to 2025* <https://www.marketsandmarkets.com/Market-Reports/hybrid-vehicle-market-159441728.html>. Accessed: 2021-12-20. 2019.
- [80] Bernard Marr. *Deep Learning Vs Neural Networks – What’s The Difference?* <https://bernardmarr.com/deep-learning-vs-neural-networks-whats-the-difference/>. Accessed: 2022-04-14. s.f.
- [81] MathWorks. <https://la.mathworks.com/help/matlab/ref/filter.html>. Accessed: 2021-12-14.
- [82] Gys Albertus Marthinus Meiring y Hermanus Carel Myburgh. “A review of intelligent driving style analysis systems and related artificial intelligence algorithms”. En: *Sensors* 15.12 (2015), págs. 30653-30682.
- [83] S Melzi y E Sabbioni. “On the vehicle sideslip angle estimation through neural networks: Numerical and experimental results”. En: *Mechanical Systems and Signal Processing* 25.6 (2011), págs. 2005-2019.
- [84] Rick Merritt. <https://blogs.nvidia.com/blog/2022/03/25/what-is-a-transformer-model/>. Accessed: 2022-02-17. 2022.
- [85] Minitab. <https://support.minitab.com/es-mx/minitab/18/help-and-how-to/modeling-statistics/regression/supporting-topics/basics/what-are-categorical-discrete-and-continuous-variables/>. Accessed: 2021-12-14.
- [86] Minitab Blog Editor. <https://blog.minitab.com/es/analisis-de-regresion-como-puedo-interpretar-el-r-cuadrado-y-evaluar-la-bondad-de-ajuste>. Accessed: 2022-03-20. 2019.
- [87] M Montani y col. “Vehicle localization combining non-linear state observer with artificial neural network”. En: *IOP Conference Series: Materials Science and Engineering*. Vol. 1214. 1. IOP Publishing. 2022, pág. 012040.
- [88] Mathieu Nguyen-H y Chi Zhou. “Improving GPS/INS integration through neural networks”. En: *arXiv preprint arXiv:1005.5115* (2010).
- [89] Tommaso Novi, Renzo Capitani y Claudio Annicchiarico. “An integrated artificial neural network–unscented Kalman filter vehicle sideslip angle estimation based on inertial measurement unit measurements”. En: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 233.7 (2019), págs. 1864-1878.

- [90] Organización Mundial de la Salud. *10 datos sobre la seguridad vial en el mundo*. <http://www.who.int/features/factfiles/roadsafety/es/>. Accessed: 2021-12-20. 2018.
- [91] Margot P. *Cross-Validation : definición e importancia en Machine Learning*. DataScientest. 2022. URL: <https://datascientest.com/es/cross-validation-definicion-e-importancia>.
- [92] Aishwarya Panday y Hari Om Bansal. “A review of optimal energy management strategies for hybrid electric vehicle”. En: *International Journal of Vehicular Technology* 2014 (2014).
- [93] PROFESORUOC. <https://blogs.uoc.edu/informatica/machine-learning-la-maldicion-de-las-cajas-negras/>. Accessed: 2022-04-20. 2017.
- [94] Professor Ryan. *Long Short Term Memory (LSTM) Networks in 20 minutes*. Youtube. 2019. URL: <https://www.youtube.com/watch?v=o0Iqrp0h5ig>.
- [95] programador clic. <https://programmerclick.com/article/1375926681/>. Accessed: 2021-12-14.
- [96] Luis F Quirama y col. “Driving cycles that reproduce driving patterns, energy consumptions and tailpipe emissions”. En: *Transportation Research Part D: Transport and Environment* 82 (2020), pág. 102294.
- [97] Paloma Recuero de los Santos. <https://empresas.blogthinkbig.com/que-algoritmo-elegir-en-ml-aprendizaje/>. Accessed: 2022-04-17. 2021.
- [98] Enrique Rus Arias. <https://economipedia.com/definiciones/variable-categorica.html>. Accessed: 2021-12-14. 2021.
- [99] Kamal Saadeddin y col. “Performance enhancement of low-cost, high-accuracy, state estimation for vehicle collision prevention system using ANFIS”. En: *Mechanical Systems and Signal Processing* 41.1-2 (2013), págs. 239-253.
- [100] MFM Sabri, Kumeresan A Danapalasingam y Mohd Fuaad Rahmat. “A review on hybrid electric vehicles architecture and energy management strategies”. En: *Renewable and Sustainable Energy Reviews* 53 (2016), págs. 1433-1442.
- [101] Joaquín Salas Rodríguez. Comunicación personal, 07 de junio de 2022.
- [102] Constantine Samaras y Kyle Meisterling. *Life cycle assessment of greenhouse gas emissions from plug-in hybrid vehicles: implications for policy*. 2008.
- [103] SAS. *Aprendizaje automático. Qué es y por qué es importante*. Accessed: 2022-04-14. 2022.
- [104] Shipra Saxena. *Introduction to Gated Recurrent Unit (GRU)*. Analytics Vidhya. 2021. URL: <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-gated-recurrent-unit-gru/>.

- [105] Jan Schmitz. *Stock predictions with state-of-the-art Transformer and Time Embeddings*. 2020. URL: <https://towardsdatascience.com/stock-predictions-with-state-of-the-art-transformer-and-time-embeddings-3a4485237de6>.
- [106] Scikit-learn. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html?highlight=min%20max%20scaler#sklearn.preprocessing.MinMaxScaler>. Accessed: 2021-12-14.
- [107] Scikit-learn. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html>. Accessed: 2021-12-14.
- [108] Jaewook Shin y MyoungHo Sunwoo. “Vehicle speed prediction using a Markov chain with speed constraints”. En: *IEEE Transactions on Intelligent Transportation Systems* 20.9 (2018), págs. 3201-3211.
- [109] Philipp Maximilian Sieberg y Dieter Schramm. “Ensuring the Reliability of Virtual Sensors Based on Artificial Intelligence within Vehicle Dynamics Control Systems”. En: *Sensors* 22.9 (2022), pág. 3513.
- [110] Philipp M Sieberg y col. “Hybrid State Estimation-A Contribution Towards Reliability Enhancement of Artificial Neural Network Estimators”. En: *IEEE Transactions on Intelligent Transportation Systems* (2021).
- [111] sitio big data. <https://sitiobigdata.com/2019/12/24/funciones-comunes-de-perdida-en-el-aprendizaje-automatico/#:~:text=Funciones%20de%20p%C3%A9rdida%20y%20optimizaciones&text=Es%20un%20m%C3%A9todo%20para%20evaluar,arrojar%20un%20n%C3%BAmero%20muy%20grande..> Accessed: 2022-04-17. 2019.
- [112] Grant Maloy Smith. *What is ADAS (Advanced Driver Assistance Systems)?*. <https://dewesoft.com/daq/what-is-adas>. Accessed: 2021-12-14. 2021.
- [113] Rui Song y Yongchun Fang. “Vehicle state estimation for INS/GPS aided by sensors fusion and SCKF-based algorithm”. En: *Mechanical Systems and Signal Processing* 150 (2021), pág. 107315.
- [114] Miguel Sotaquirá. *La Función de Activación* <https://www.codificandobits.com/blog/funcion-de-activacion/>. Accessed: 2022-04-14. 2018.
- [115] Salvatore Strano, Mario Terzo y Ciro Tordela. “A supervised machine learning framework for smart tires”. En: *2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI)*. IEEE. 2021, págs. 364-369.

- [116] Xiaoqiang Sun y col. “Piecewise affine modeling and hybrid optimal control of intelligent vehicle longitudinal dynamics for velocity regulation”. En: *Mechanical Systems and Signal Processing* 162 (2022), pág. 108089.
- [117] TechData. <https://blog.techdata.com/ts/latam/aprendizaje-automatico-vs-estadisticas>. 10 de Noviembre de 2021.
- [118] José Antonio Tellez y Karime Vázquez. *ELECTRIC VEHICLE REGULATION AND LAW IN MEXICO* <https://cms.law/en/int/expert-guides/cms-expert-guide-to-electric-vehicles/mexico>. Accessed: 2021-12-20. 2018.
- [119] *Validación Cruzada*. WIKIPEDIA. 2022. URL: https://es.wikipedia.org/wiki/Validaci%C3%B3n_cruzada.
- [120] Leandro Vargas-Melendez y col. “Sensor Fusion based on an integrated neural network and probability density function (PDF) dual Kalman filter for on-line estimation of vehicle parameters and states”. En: *Sensors* 17.5 (2017), pág. 987.
- [121] Leandro Vargas-Meléndez y col. “A sensor fusion method based on an integrated neural network and Kalman filter for vehicle roll angle estimation”. En: *Sensors* 16.9 (2016), pág. 1400.
- [122] Ashish Vaswani y col. “Attention is all you need”. En: *Advances in neural information processing systems* 30 (2017).
- [123] VBOX AUTOMOTIVE. <https://www.vboxautomotive.co.uk/index.php/en/products/data-loggers/vb3i>. Accessed: 2021-12-14.
- [124] Susan Wallace y George Lăzăroiu. “Predictive Control Algorithms, Real-World Connected Vehicle Data, and Smart Mobility Technologies in Intelligent Transportation Planning and Engineering”. En: *Contemporary Readings in Law and Social Justice* 13.2 (2021), págs. 79-92.
- [125] Junyi Wang y col. “Road surface real-time detection based on Raspberry Pi and recurrent neural networks”. En: *Transactions of the Institute of Measurement and Control* 43.11 (2021), págs. 2540-2550.
- [126] Weida Wang y col. “A multi-objective optimization energy management strategy for power split HEV based on velocity prediction”. En: *Energy* 238 (2022), pág. 121714.
- [127] Yaqian Wang y Xiaohong Jiao. “Dual Heuristic Dynamic Programming Based Energy Management Control for Hybrid Electric Vehicles”. En: *Energies* 15.9 (2022), pág. 3235.
- [128] Wang Wei y col. “Vehicle sideslip angle estimation based on general regression neural network”. En: *Mathematical Problems in Engineering* 2016 (2016).
- [129] WIKIPEDIA. https://en.wikipedia.org/wiki/Online_and_offline. Accessed: 2022-02-17. 2022.

- [130] Barbara Woodward y Tomas Kliestik. “Intelligent Transportation Applications, Autonomous Vehicle Perception Sensor Data, and Decision-Making Self-Driving Car Control Algorithms in Smart Sustainable Urban Mobility Systems”. En: *Contemporary Readings in Law and Social Justice* 13.2 (2021), págs. 51-64.
- [131] Xin Xia y col. “Advancing estimation accuracy of sideslip angle by fusing vehicle kinematics and dynamics information with fuzzy logic”. En: *IEEE Transactions on Vehicular Technology* 70.7 (2021), págs. 6577-6590.
- [132] Fan Xu y col. “Estimation of Sideslip Angle with Tire-Road Friction Adaptation Using Non-linear Observability Theory”. En: *Proceedings of China SAE Congress 2020: Selected Papers*. Springer. 2022, págs. 627-653.
- [133] Nan Xu y col. “Tire force estimation in intelligent tires using machine learning”. En: *IEEE Transactions on Intelligent Transportation Systems* (2020).
- [134] Fengqi Zhang y col. “Energy management strategies for hybrid electric vehicles: Review, classification, comparison, and outlook”. En: *Energies* 13.13 (2020), pág. 3352.
- [135] Qian Zhang y col. “A Novel PWA Lateral Dynamics Modeling Method and Switched TS Observer Design for Vehicle Sideslip Angle Estimation”. En: *IEEE Transactions on Industrial Electronics* 69.2 (2021), págs. 1847-1857.
- [136] Chenyu Zhou y col. “Robust sideslip angle observer of commercial vehicles based on cornering stiffness estimation using neural network”. En: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* (2022), pág. 09544070211072401.
- [137] Yang Zhou, Alexandre Ravey y Marie-Cécile Péra. “A survey on driving prediction techniques for predictive energy management of plug-in hybrid electric vehicles”. En: *Journal of Power Sources* 412 (2019), págs. 480-495.
- [138] Yang Zhou, Alexandre Ravey y Marie-Cécile Péra. “Multi-objective energy management for fuel cell electric vehicles using online-learning enhanced Markov speed predictor”. En: *Energy Conversion and Management* 213 (2020), pág. 112821.
- [139] Yang Zhou y col. “An integrated predictive energy management for light-duty range-extended plug-in fuel cell electric vehicle”. En: *Journal of Power Sources* 451 (2020), pág. 227780.