

ALGORITMOS BASADOS EN MACHINE LEARNING APLICADOS A LA PREDICCIÓN DE FLUJO DE TRÁFICO EN CIUDADES INTELIGENTES

WILLIAM YESID AGUIRRE CARPETA

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
MAESTRÍA EN CIENCIAS DE LA INFORMACIÓN Y LAS COMUNICACIONES
ÉNFASIS EN INGENIERÍA DE SOFTWARE
BOGOTÁ, COLOMBIA
2026

ALGORITMOS BASADOS EN MACHINE LEARNING APLICADOS A LA PREDICCIÓN DE FLUJO DE TRÁFICO EN CIUDADES INTELIGENTES

WILLIAM YESID AGUIRRE CARPETA

Tesis presentada como requisito parcial para optar el título de:

Magíster en Ciencias de la Información y las Comunicaciones con énfasis en Ingeniería de Software

Director del Proyecto de grado

Ing. Roberto Ferro Escobar, PhD

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
MAESTRÍA EN CIENCIAS DE LA INFORMACIÓN Y LAS COMUNICACIONES
ÉNFASIS EN INGENIERÍA DE SOFTWARE
BOGOTÁ, COLOMBIA
2026

AGRADECIMIENTOS

A mis padres, cuyo esfuerzo y amor incondicional sembraron en mí la disciplina y la curiosidad que este trabajo exigió.

A los amigos y colegas que compartieron este camino, aportaron una palabra precisa en el momento oportuno o simplemente estuvieron presentes.

TABLA DE CONTENIDO

Lista de figuras 8
Lista de tablas 10
RESUMEN 11
1. INTRODUCCIÓN 13
2. PLANTEAMIENTO DEL PROBLEMA 15
2.1 Descripción del problema 15
2.2 Formulación del problema 17
2.3 Sistematización del problema 17
3. OBJETIVOS 18
3.1 Objetivo general 18
3.2 Objetivos específicos 18
4. JUSTIFICACIÓN 19
5. MARCO REFERENCIAL 20
5.1 Marco teórico 21
5.1.1 Ciudades Inteligentes 21
5.1.2 Sistemas de Transporte Inteligente (ITS) 22
5.1.3 Aprendizaje automático aplicado al tráfico 22
5.1.4 Predicción de flujo vehicular a corto plazo 24
5.1.5 Métricas de evaluación de modelos predictivos 25
5.2 Estado del arte 26
5.2.1 Enfoques clásicos y de transición 26
5.2.2 Modelos de ensamble basados en árboles 27
5.2.3 Arquitecturas de Deep learning 27
5.2.4 Redes neuronales de grafos 28
5.2.5 Estudios comparativos y el debate entre modelos tabulares vs. Deep learning 29
5.2.6 Contexto latinoamericano y colombiano 30
5.3 Síntesis del marco referencial 32
6. HIPÓTESIS 32
7. METODOLOGÍA 33

7.1 Enfoque y tipo de investigación 33
7.2 Área de estudio 34
7.3 Fuente de datos y sistema de detección 39
7.4 Preprocesamiento de datos 40
7.5 Selección del período de análisis 41
7.6 Ingeniería de características 42
7.7 Partición temporal de los datos 44
7.8 Modelos evaluados 46
7.8.1 Regresión Ridge (línea base) 46
7.8.2 Random Forest 46
7.8.3 XGBoost 46
7.8.4 Red Neuronal Artificial (ANN) 47
7.8.5 LSTM (Long Short-Term Memory) 47
7.9 Protocolo de evaluación 48
7.10 Herramientas tecnológicas 49
8. RESULTADOS 50
8.1 Análisis exploratorio de datos 50
8.1.1 Patrones por hora y día de la semana 50
8.1.2 Perfil diario: días hábiles versus fines de semana 51
8.1.3 Heterogeneidad del corredor 52
8.2 Desempeño global de los modelos 54
8.3 Análisis de dispersión: predicción versus realidad 56
8.3.1 Regresión Ridge 56
8.3.2 Random Forest 57
8.3.3 XGBoost 59
8.3.4 Red Neuronal Artificial (ANN) 60
8.3.5 LSTM 61
8.4 Importancia de características 62
8.5 Análisis por intersección 63
8.6 Comportamiento de convergencia 64
8.7 Síntesis de resultados 65

9. ANÁLISIS Y DISCUSIÓN 66
9.1 Dominancia de los ciclos diario y semanal 66
9.2 El bajo desempeño del LSTM: datos limitados y convergencia prematura 67
9.3 Eficiencia computacional 69
9.4 Intersecciones problemáticas y calidad de los datos 69
9.5 Comparación con la literatura 70
9.6 Implicaciones para la optimización del tráfico 71
9.7 Limitaciones del estudio 72
10. CONCLUSIONES Y TRABAJO FUTURO 74
10.1 Conclusiones 74
10.2 Trabajo futuro 77
REFERENCIAS 80
ANEXO A 86
ANEXO B 90

Lista de figuras

Figura 1 *Polígono de detección. Desde la calle 134 hasta la calle 170, entre la carrera 7.ª y carrera 9.ª.* 34

Figura 2 37

Figura 3 39

Figura 4 *Detección vehicular media por día de la semana y hora del día en el corredor de estudio. Las celdas más oscuras indican mayor flujo medio.* 51

Figura 5 *Perfil diario de tráfico: días hábiles (azul) versus fines de semana (rojo).* 52

Figura 6 *Series temporales de ejemplo para tres puntos de acceso de niveles de flujo contrastantes (una semana representativa).* 54

Figura 7 *Comparación de métricas del conjunto de prueba entre los cinco modelos.* . 56

Figura 8 *Regresión Ridge) Detección predicha versus real en el conjunto de prueba. R^2 = 0,916; RMSE = 162 veh/h. Mejor corrida.* 57

Figura 9 *Random Forest) Detección predicha versus real en el conjunto de prueba. R^2 = 0,955; RMSE = 130 veh/h. Mejor corrida.* 58

Figura 10 *XGBoost) Detección predicha versus real en el conjunto de prueba. R^2 = 0,970; RMSE = 120 veh/h. Mejor corrida.* 59

Figura 11 *Red Neuronal Artificial (ANN): Detección predicha versus real en el conjunto de prueba. R^2 = 0,939; RMSE = 132 veh/h. Mejor corrida.* 60

Figura 12 *LSTM: Detección predicha versus real en el conjunto de prueba. R^2 = 0,909; RMSE = 133 veh/h.* 61

Figura 13 *Importancia de características (ganancia) del modelo XGBoost. Los rezagos de 24 h y 168 h representan conjuntamente más del 67% de la ganancia total.* 62

Figura 14 *RMSE del conjunto de prueba por intersección para los cinco modelos.* 63

Figura 15 *Curva de aprendizaje de XGBoost (RMSE por ronda de boosting). La línea punteada vertical indica la mejor iteración (497).* 64

Figura 16 *Curva de aprendizaje del LSTM (MSE por época en la escala log(1 + y)). La pérdida se estabiliza después de la primera época.* 65

Lista de tablas

Tabla 1 *Intersecciones semaforizadas incluidas en el área de estudio con desglose de puntos de detección por intersección* 35

Tabla 2 *Correspondencia entre accesos, zonas de detección y sentidos de circulación para la intersección EXT 1018 (Carrera 7 × Calle 162)* 38

Tabla 3 *Estructura del conjunto de datos proporcionado por la SDM* 40

Tabla 4 *Características construidas para la predicción del flujo vehicular (21 variables).* 43

Tabla 5 *Partición temporal del conjunto de datos* 45

Tabla 6 *Resumen de hiperparámetros principales por modelo* 47

Tabla 7 *Tiempos de entrenamiento aproximados (30 corridas completas)* 49

Tabla 8 *Comparación de desempeño en el conjunto de prueba. Valores expresados como media ± desviación estándar sobre 30 corridas independientes.* 55

Tabla 9 *Comparación contextualizada con estudios previos de predicción de tráfico* .. 71

RESUMEN

Bogotá ha figurado durante años entre las capitales latinoamericanas con peor desempeño en movilidad urbana debido a múltiples factores que afectan directamente la calidad de vida de los ciudadanos. El presente trabajo de grado reporta un estudio empírico que utiliza los datos de los detectores de video Siemens C900 recopilados por la Secretaría Distrital de Movilidad (SDM) durante el período julio–octubre de 2025, comprendiendo 105.867 registros distribuidos en 39 puntos de detección de 13 intersecciones semaforizadas a lo largo de las Carreras 7 y 9, entre la Calle 134 y la Calle 170.

Se entrenaron y evaluaron cinco modelos de ML (Regresión Ridge, Random Forest, XGBoost, Red Neuronal Artificial (ANN) y LSTM) sobre 21 características construidas a partir de rezagos autorregresivos, estadísticos móviles y codificaciones temporales cíclicas. Cada modelo fue ejecutado 30 veces bajo diferentes condiciones de aleatoriedad. XGBoost alcanzó el mejor desempeño (R^2 = 0,947; RMSE = 126 veh/h), seguido por Random Forest (R^2 = 0,942). El LSTM obtuvo el peor resultado medio (R^2 = 0,822) con una varianza entre corridas 23 veces mayor que la de XGBoost. Sin embargo, la mejor corrida individual del LSTM alcanzó un R^2 de 0,941, lo que indica que la arquitectura recurrente posee capacidad representacional suficiente pero no converge de manera estable, constituyendo un riesgo para un despliegue operativo. El análisis de importancia de características reveló que dos predictores (la detección en t−24 h y en t−168 h) concentran conjuntamente el 78% de la ganancia predictiva de XGBoost, confirmando que los ciclos diarios y semanales de desplazamiento son los principales determinantes del flujo vehicular en el corredor. Para las autoridades de transporte estos resultados ofrecen una recomendación clara: los modelos de gradient boosting tabular son más precisos y significativamente más confiables que las arquitecturas recurrentes profundas, y su estabilidad los hace aptos para un despliegue en producción.

Palabras clave: Predicción de flujo vehicular, aprendizaje automático, XGBoost, LSTM, movilidad inteligente, Bogotá.

Abstract

Bogotá has ranked among the worst-performing Latin American capitals in urban mobility for years, yet only recently has its northern corridor gained real-time traffic instrumentation. This thesis reports an empirical study using the Siemens C900 detector data collected by the Secretaría Distrital de Movilidad (SDM) from July through October 2025, comprising 105,867 records across 39 detection points at 13 signalized intersections along Carreras 7 and 9 between Calle 134 and Calle 170.

Five machine learning models (Ridge Regression, Random Forest, XGBoost, a feedforward ANN, and an LSTM) were trained and tested on 21 features built from autoregressive lags, rolling statistics, and cyclical temporal encodings. To ensure statistical robustness, each model was re-run 30 times under different randomness conditions. XGBoost achieved the best performance (R^2 = 0.947, RMSE = 126 veh/h), with Random Forest as a close second (R^2 = 0.942). The LSTM finished last on average (R^2 = 0.822) and showed run-to-run variance 23 times larger than XGBoost. However, the LSTM's single best run reached R^2 = 0.941, revealing that the recurrent architecture has sufficient representational capacity but fails to converge reliably (a critical shortcoming for operational deployment. Feature importance analysis revealed that two predictors alone (detection counts at t−24 h and t−168 h) accounted for 78% of XGBoost's predictive gain, confirming that daily and weekly commuting cycles are the main drivers of flow in this corridor. For transport authorities operating sensor networks, these results provide a clear recommendation: gradient-boosted tabular models are more accurate and far more reliable than recurrent Deep architectures, and their stability makes them suitable for production deployment.

Keywords: Traffic flow forecasting, machine learning, XGBoost, LSTM, smart mobility, Bogotá.

1. INTRODUCCIÓN

La movilidad urbana constituye uno de los desafíos más críticos que enfrentan las ciudades de economías emergentes en el siglo XXI. Bogotá, capital de Colombia con más de ocho millones de habitantes, ha experimentado un deterioro sostenido en sus indicadores de movilidad y transporte: según el índice Cities in Motion del IESE Business School, la ciudad descendió del puesto 117 en movilidad y transporte en 2019 al puesto 174 en 2025 (Índice IESE Cities in Motion, 2025). Esta tendencia descendente refleja un problema estructural que impacta directamente la calidad de vida, la productividad económica y la sostenibilidad ambiental de la ciudad.

En el paradigma de las Ciudades Inteligentes (Smart Cities), la gestión eficiente del tráfico se apoya crecientemente en las Tecnologías de la Información y las Comunicaciones (TIC) para adquirir, procesar y actuar sobre datos provenientes de la infraestructura de sensorización urbana (Elassy et al., 2024). La predicción del flujo vehicular a corto plazo (anticipar el número de vehículos que circularán por un punto en los próximos minutos u horas) es un componente fundamental de los Sistemas de Transporte Inteligente (ITS), ya que alimenta algoritmos de control semafórico adaptativo, sistemas de información al viajero y herramientas de planificación estratégica (Afandizadeh et al., 2024) (Liu et al., 2026).

Durante la última década, los métodos de aprendizaje automático (Machine Learning, ML) y aprendizaje profundo (Deep Learning, DL) han desplazado progresivamente los enfoques estadísticos clásicos en la predicción del tráfico, mostrando ganancias de desempeño en benchmarks internacionales como PeMS-04, PeMS-08 y METR-LA (Rahmani et al., 2023) (Jiang & Luo, 2022). Sin embargo, la mayoría de esta literatura se fundamenta en conjuntos de datos que cubren meses o años de observaciones continuas a intervalos de cinco minutos, con cientos de sensores distribuidos en redes viales maduras. Las ciudades de economías emergentes operan típicamente bajo un régimen muy diferente: la cobertura de sensores es reciente o limitada, los registros históricos son cortos y los problemas de calidad de datos son

frecuentes debido a múltiples factores como fallas eléctricas, interrupciones en el suministro de conectividad de red, vandalismo, entre otros.

Bajo estas condiciones, el ranking relativo de modelos de ML y DL observado en los grandes benchmarks podría no sostenerse, y una autoridad de tráfico con pocas semanas de datos necesita elegir un modelo sin el beneficio de una sintonización extensiva. Esta brecha motiva la pregunta central del presente trabajo: ¿qué familia de modelos de aprendizaje automático ofrece el mejor desempeño predictivo para el flujo vehicular en un corredor semaforizado de Bogotá cuando los datos disponibles son limitados?

Para responder esta pregunta, se realizó un estudio empírico en el corredor norte de Bogotá, utilizando datos reales de los detectores de video Siemens C900 operados por la Secretaría Distrital de Movilidad (SDM) en 13 intersecciones semaforizadas a lo largo de la Carrera 7 y la Carrera 9, entre la Calle 134 y la Calle 170. El conjunto de datos comprende 105.867 registros horarios distribuidos en 39 puntos de detección durante el período julio–octubre de 2025. Se entrenaron y evaluaron cinco modelos representativos del espectro desde líneas base lineales hasta arquitecturas secuenciales profundas: Regresión Ridge, Random Forest, XGBoost, una red neuronal artificial (ANN) y una red LSTM. Para garantizar la robustez estadística de los resultados, cada modelo fue ejecutado 30 veces bajo diferentes condiciones de aleatoriedad: para Ridge, mediante bootstrap del conjunto de entrenamiento dado que el modelo es determinista en sí mismo; para Random Forest y XGBoost, mediante semillas aleatorias distintas en la construcción de los ensambles; y para la ANN y el LSTM, mediante diferentes inicializaciones aleatorias de los pesos de la red. El desempeño se evaluó mediante cuatro métricas: MAE, RMSE, MAPE y R^2.

Los resultados revelan que XGBoost alcanza el mejor desempeño (R^2 = 0,947; RMSE = 126 veh/h), seguido por Random Forest (R^2 = 0,942) y la ANN (R^2 = 0,924). El LSTM que la mayoría de profesionales esperaría que sobresaliera en una tarea secuencial) obtiene el peor resultado en promedio (R^2 = 0,822), con una varianza entre

corridas 23 veces mayor que la de XGBoost. Si bien la mejor corrida individual del LSTM alcanza un R^2 de 0,941, su desempeño medio es arrastrado por la inestabilidad en la inicialización de los pesos, lo cual indica que la arquitectura recurrente no converge de manera confiable incluso con cuatro meses de datos. Adicionalmente, el análisis de importancia de características demuestra que dos rezagos autorregresivos el de 24 horas y el de 168 horas) concentran conjuntamente el 78% de la ganancia predictiva, confirmando que los ciclos diarios y semanales de desplazamiento son los principales determinantes del flujo vehicular en el corredor.

Es importante señalar que la optimización del flujo de tráfico en intersecciones semaforizadas es un problema de dos etapas: la primera etapa consiste en predecir con precisión el flujo vehicular futuro a partir de datos históricos, y la segunda etapa consiste en utilizar dichas predicciones para ajustar adaptativamente los ciclos semafóricos. Sin una predicción confiable, cualquier estrategia de optimización opera a ciegas. El presente trabajo aborda la primera etapa (la predicción) como componente fundamental e indispensable de un sistema de optimización del tráfico, determinando cuál modelo de aprendizaje automático ofrece la base predictiva más precisa y estable para alimentar futuros algoritmos de control adaptativo.

2. PLANTEAMIENTO DEL PROBLEMA

2.1 Descripción del problema

Los sistemas de transporte urbano de Bogotá enfrentan un desafío de complejidad creciente derivado de múltiples factores que interactúan simultáneamente: el crecimiento sostenido del parque automotor, la expansión urbana hacia zonas periféricas cada vez más alejadas de los centros de actividad económica, las limitaciones de la infraestructura vial existente y la complejidad inherente a la gestión de intersecciones donde confluyen vehículos particulares, transporte público, ciclistas y peatones.

La congestión vehicular resultante tiene consecuencias directas y medibles sobre la calidad de vida de los habitantes: incrementa los tiempos de desplazamiento diarios, aumenta las emisiones contaminantes, eleva los índices de accidentalidad y genera pérdidas económicas significativas por improductividad. Según el índice (Índice IESE Cities in Motion, 2025), Bogotá ocupa el puesto 174 en movilidad y transporte entre las ciudades evaluadas a nivel mundial, un descenso notable desde el puesto 117 que ocupaba en 2019, lo cual evidencia un deterioro progresivo en la gestión de la movilidad urbana.

Un aspecto central de la gestión del tráfico es la optimización de los tiempos de transición de los controladores semafóricos. Actualmente, la Secretaría Distrital de Movilidad (SDM) opera una red de sensores de video-detección Siemens C900 en intersecciones clave de la ciudad, los cuales generan datos continuos sobre conteos vehiculares y porcentajes de ocupación. Sin embargo, estos datos no están siendo aprovechados de manera sistemática para alimentar modelos predictivos que anticipen el comportamiento del tráfico y permitan ajustar proactivamente los ciclos semafóricos.

El problema se agudiza por la escasez de estudios empíricos que evalúen modelos de aprendizaje automático sobre datos reales de semaforización en ciudades latinoamericanas. Mientras que la literatura internacional ofrece comparaciones exhaustivas de modelos sobre data sets como PeMS y METR-LA, estos benchmarks corresponden a autopistas con cientos de sensores y meses de datos a intervalos de cinco minutos, un régimen muy diferente al que enfrenta una ciudad como Bogotá, donde la red de sensores no se encuentra implementada en todas las intersecciones donde hay semáforos, por lo que el historial de datos disponible es de apenas unos meses, nulo o inconstante.

En consecuencia, existe una necesidad concreta de determinar qué familia de modelos de aprendizaje automático es la más adecuada para predecir el flujo vehicular en intersecciones semaforizadas de Bogotá bajo las condiciones reales de datos

limitados, calidad variable de sensores y heterogeneidad de patrones de tráfico que caracterizan el corredor norte de la ciudad.

2.2 Formulación del problema

¿Cuál de los modelos de aprendizaje automático evaluados Regresión Ridge, Random Forest, XGBoost, red neuronal artificial (ANN) y LSTM ofrece el mejor desempeño predictivo para el flujo vehicular a corto plazo en intersecciones semaforizadas del corredor norte de Bogotá, considerando tanto la precisión como la estabilidad del modelo bajo condiciones de datos limitados?

2.3 Sistematización del problema

La pregunta central de investigación se descompone en las siguientes preguntas específicas:

- ¿Cuáles son los patrones temporales dominantes del flujo vehicular en el corredor norte de Bogotá y cómo influyen en la capacidad predictiva de los modelos?

- ¿Qué modelo de aprendizaje automático ofrece el mejor equilibrio entre precisión predictiva y estabilidad de ejecución cuando el volumen de datos de entrenamiento es limitado a un mes de registros horarios?

- ¿Cómo varía el desempeño de los modelos entre las diferentes intersecciones del corredor, y qué factores explican las diferencias observadas?

- ¿Cuáles son las características (features) más determinantes para la predicción del flujo vehicular en el corredor, y qué implicaciones tiene su dominancia para el diseño de modelos futuros?

3. OBJETIVOS

3.1 Objetivo general

Implementar algoritmos Machine Learning para la recolección, análisis y depuración de información del sistema de semaforización de Bogotá que permita identificar y establecer estrategias para la predicción del tráfico en un tramo de la ciudad.

3.2 Objetivos específicos

1. Identificar diferentes métodos de Machine Learning, con el fin de determinar aproximaciones algorítmicas en la optimización del tráfico en el tramo de vía seleccionado.

2. Validar, mediante herramientas informáticas, las aproximaciones algorítmicas mediante la evaluación del índice de error estadístico.

3. Evaluar el rendimiento computacional de los métodos de Machine Learning empleados en la predicción del flujo de tráfico.

4. JUSTIFICACIÓN

La movilidad urbana en Bogotá representa uno de los problemas estructurales más críticos para la calidad de vida de sus habitantes y la sostenibilidad económica de la ciudad. Con más de ocho millones de residentes y un parque automotor en crecimiento sostenido, el sistema vial enfrenta niveles de congestión que se han agravado progresivamente: según el índice Cities in Motion del IESE Business School (2025), Bogotá descendió del puesto 117 al 174 en movilidad y transporte entre 2019 y 2025, posicionándose entre las capitales latinoamericanas con peor desempeño en este indicador.

Ante este panorama, la Secretaría Distrital de Movilidad (SDM) ha desplegado en los últimos años una red de sensores de video detección Siemens C900 en intersecciones semaforizadas clave de la ciudad. Estos dispositivos generan de manera continua datos de conteo vehicular y ocupación que, hasta el momento, no han sido aprovechados sistemáticamente para alimentar modelos predictivos de corto plazo que orienten la gestión adaptativa del tráfico. Esta brecha entre la disponibilidad de infraestructura de sensores y su aprovechamiento analítico constituye la motivación directa del presente trabajo.

Desde la dimensión científica, la literatura internacional sobre predicción de flujo vehicular mediante aprendizaje automático se concentra predominantemente en conjuntos de datos de países desarrollados con características estructuralmente diferentes a las de Bogotá: meses o años de observaciones, cientos de sensores disponibles y granularidad temporal de cinco minutos o menos. Las ciudades de economías emergentes operan bajo un régimen diferente (historiales cortos, calidad de datos variable, infraestructura de sensores recientemente desplegada) que invalida la transferibilidad directa de los rankings de modelos documentados en benchmarks internacionales como PeMS o METR-LA. Esta brecha empírica justifica la necesidad de estudios comparativos contextualizados en el entorno latinoamericano.

Desde la dimensión práctica, una autoridad de tráfico que acaba de desplegar una red de sensores necesita una guía basada en evidencia sobre qué modelo adoptar cuando dispone de pocas semanas de datos. La elección incorrecta tiene consecuencias concretas: invertir en infraestructura computacional de GPU para modelos de Deep learning que resulten inestables, o asumir que los modelos recurrentes son superiores por defecto cuando los datos no son suficientes para estimar sus parámetros de manera confiable. Los resultados de este trabajo (que demuestran que XGBoost supera al LSTM en precisión, estabilidad y eficiencia computacional bajo condiciones de datos limitados) ofrecen una recomendación directamente accionable para la SDM y entidades análogas en otras ciudades de la región.

Desde la dimensión social, la mejora en la predicción del flujo vehicular tiene el potencial de contribuir a la reducción de los tiempos de desplazamiento de los millones de bogotanos que utilizan diariamente el corredor norte de la ciudad, disminuir las emisiones de gases contaminantes asociadas a la congestión y mejorar los índices de seguridad vial, todos ellos factores que impactan directamente el bienestar urbano.

5. MARCO REFERENCIAL

El presente capítulo establece las bases teóricas y conceptuales que fundamentan la investigación, y revisa el estado del arte en predicción de flujo de tráfico mediante aprendizaje automático. La primera sección desarrolla el marco teórico, abordando los conceptos de Ciudades Inteligentes, Sistemas de Transporte Inteligente, aprendizaje automático aplicado al tráfico y métricas de evaluación. La segunda sección presenta una revisión crítica de la literatura reciente, con énfasis en estudios comparativos, enfoques de Deep learning y el contexto latinoamericano.

5.1 Marco teórico

5.1.1 Ciudades Inteligentes

El concepto de Ciudad Inteligente (Smart City) se refiere a un entorno urbano que, a través de la integración de Tecnologías de la Información y las Comunicaciones (TIC), busca mejorar la calidad de los servicios urbanos, reducir el consumo de recursos y minimizar el impacto ambiental (Caragliu et al., 2009). Las ciudades inteligentes, en aras de proteger y mejorar la calidad de vida de sus ciudadanos, utilizan tecnologías para conectar, en áreas clave como transporte, energía, salud, agua y gestión de residuos.

Las TIC constituyen el pilar sobre el que se construyen las ciudades inteligentes, facilitando la recolección, procesamiento e interpretación de datos a gran escala para una gestión más eficiente de los recursos urbanos. En el contexto del tráfico y la movilidad, las TIC permiten recopilar datos en tiempo real de vehículos y sistemas de transporte que pueden ser utilizados para optimizar rutas, reducir congestiones y mejorar la seguridad vial. Los Sistemas de Información Geográfica (GIS) e Internet de las Cosas (IoT) son ejemplos de tecnologías empleadas en ciudades inteligentes que permiten una recolección y análisis detallados de datos geoespaciales y de sensores, fundamentales para el manejo eficaz del tráfico (Thakuriah et al., 2017).

En la gestión de tráfico, las ciudades inteligentes implementan modelos predictivos y adaptativos que utilizan algoritmos de aprendizaje automático para ajustar los patrones de semaforización en respuesta a las condiciones cambiantes del tráfico. (Elassy et al., 2024) presentan una visión integral de cómo los ITS contribuyen al desarrollo de ciudades inteligentes sostenibles, destacando que la predicción del flujo vehicular es uno de los servicios habilitantes más importantes para la toma de decisiones en tiempo real.

5.1.2 Sistemas de Transporte Inteligente (ITS)

Los Sistemas de Transporte Inteligente (ITS, por sus siglas en inglés) integran tecnologías avanzadas de información y comunicación en la infraestructura de transporte y los vehículos para garantizar una movilidad urbana más segura, eficiente y económica. Son elementos clave en un ecosistema de ciudad inteligente (Papageorgiou et al., 2003), ya que facilitan el control en tiempo real del flujo de tráfico, la optimización de rutas de viaje, la minimización de emisiones contaminantes y la mejora de la experiencia del usuario.

Los componentes básicos de los ITS incluyen sistemas de control de tráfico que utilizan cámaras, sensores y algoritmos de aprendizaje automático para monitorear y gestionar el flujo vehicular; plataformas de comunicación en tiempo real basadas en redes V2X (vehicle-to-everything) que proporcionan interacción entre vehículos, infraestructura y usuarios (Lu et al., 2014); sistemas de gestión de incidentes que habilitan respuestas rápidas a accidentes o problemas en la red; y sistemas de pago electrónico que agilizan las transacciones en peajes y transporte público.

Los algoritmos de aprendizaje automático se han integrado progresivamente en los ITS durante las últimas décadas permitiendo identificar patrones complejos, prever con precisión el flujo de tráfico y tomar decisiones de control basadas en resultados (Lv et al., 2015). Entre las ventajas documentadas de los ITS se encuentran la reducción de la congestión vehicular mediante el ajuste dinámico de semáforos (Vlahogianni et al., 2014), la mejora de la seguridad vial, la eficiencia energética y la mejor planificación urbana a partir de los datos generados.

5.1.3 Aprendizaje automático aplicado al tráfico

El aprendizaje automático (Machine Learning, ML) comprende un conjunto de técnicas computacionales que permiten a los sistemas aprender patrones a partir de datos sin ser programados explícitamente para cada tarea. A diferencia de los modelos

estadísticos clásicos, que requieren supuestos rígidos sobre la distribución de los datos o la linealidad de las relaciones, los algoritmos de ML aprenden directamente de los datos, se adaptan a datos dinámicos y de alta dimensión, son capaces de detectar patrones complejos y relaciones no lineales, y pueden operar en entornos con datos incompletos o ruidosos (Witten et al., 2016).

En el contexto de la predicción del tráfico vehicular, los algoritmos de ML se aplican en diversas tareas dentro de los ITS. La predicción del flujo vehicular utiliza técnicas como las redes neuronales recurrentes y sus variantes (LSTM, GRU) para anticipar congestiones con minutos u horas de antelación (Ma et al., 2015). La clasificación de incidentes emplea algoritmos como Support Vector Machines, Random Forest o k-NN para detectar eventos atípicos a partir de datos de sensores (Sun et al., 2006). El control adaptativo de semáforos ajusta en tiempo real la duración de los ciclos semafóricos para optimizar el flujo en intersecciones congestionadas (Arel et al., 2010).

Los modelos de ML más utilizados en el análisis del tráfico vehicular incluyen los siguientes:

- La **Regresión Ridge** (Hoerl & Kennard, 1970) aplica regularización L2 sobre los coeficientes de una regresión lineal, penalizando valores extremos para reducir el sobreajuste; sirve como línea base contra la cual se evalúan modelos más complejos.
- **Random Forest** (Breiman, 2001) es un ensamble de árboles de decisión entrenados sobre subconjuntos aleatorios de datos y características, reconocido por su robustez ante ruido y su interpretabilidad.
- **XGBoost** (Chen & Guestrin, 2016) es un algoritmo de gradient boosting que construye árboles de decisión de forma secuencial, donde cada árbol corrige los errores residuales de los anteriores, y ha demostrado desempeño superior en múltiples competencias de ciencia de datos.
- Las **Redes Neuronales Artificiales (ANN)** son modelos inspirados en la estructura del cerebro biológico que permiten capturar relaciones no lineales

complejas entre múltiples variables del tráfico. Las redes de tipo perceptrón multicapa (MLP) con funciones de activación no lineales como ReLU, técnicas de regularización como dropout y normalización por lote (BatchNorm) constituyen la configuración estándar para tareas de regresión.

- Las redes **LSTM** (Long Short-Term Memory) (Hochreiter & Schmidhuber, 1997) son una variante de las redes neuronales recurrentes diseñadas específicamente para capturar dependencias temporales de largo plazo mediante puertas de entrada, olvido y salida que regulan el flujo de información a través del tiempo.

5.1.4 Predicción de flujo vehicular a corto plazo

La predicción de flujo vehicular a corto plazo consiste en estimar el número de vehículos que circularán por un punto de detección en un horizonte temporal de minutos a pocas horas. Es un componente fundamental de los ITS modernos, ya que alimenta directamente los algoritmos de control semafórico adaptativo, los sistemas de información al viajero y las herramientas de planificación estratégica (Afandizadeh et al., 2024).

Formalmente, sea $y_{s,t}$ el conteo de detección vehicular por hora en el punto de acceso *s* en la hora *t*. La tarea de predicción consiste en estimar $y_{s,t+1}$ dado un vector de características $x_{s,t}$ derivado de las observaciones hasta la hora *t* inclusive. El modelo $f\theta$ con parámetros θ estimados a partir de los datos de entrenamiento produce la predicción $\hat{y}_{s,t+1} = f\theta(x_{s,t})$. Un aspecto clave del diseño es la construcción del vector de características, que típicamente incluye rezagos autorregresivos de la variable objetivo, agregados estadísticos móviles, codificaciones temporales cíclicas y, cuando están disponibles, variables exógenas como datos meteorológicos.

La literatura reciente distingue tres generaciones de métodos para esta tarea. La primera generación comprende los enfoques estadísticos clásicos como ARIMA, SARIMA y filtros de Kalman (Doroshenko & Savchuk, 2025), que funcionan bien con

series estacionarias, pero tienen dificultades con las no-linealidades de los regímenes de congestión. La segunda generación incluye los modelos de aprendizaje automático tabular como Random Forest y gradient boosting, que manejan no-linealidades de forma natural. La tercera generación comprende las arquitecturas de Deep learning, incluyendo redes LSTM, redes convolucionales y, más recientemente, redes neuronales de grafos que modelan conjuntamente las correlaciones espaciales y temporales.

5.1.5 Métricas de evaluación de modelos predictivos

La evaluación rigurosa de modelos predictivos es esencial en el contexto de los ITS, donde los errores del modelo pueden derivar en decisiones operativas inadecuadas que generen más congestión, mayores emisiones o riesgos de seguridad. (Shmueli & Koppius, 2011) argumentan que la evaluación de modelos predictivos en sistemas de información debe centrarse en la capacidad de generalización sobre datos no vistos, medida a través de métricas de error cuantitativas, en lugar de la significancia estadística de los coeficientes.

Las métricas estándar utilizadas en problemas de predicción de tráfico incluyen el Error Absoluto Medio (MAE), que representa el promedio de las diferencias absolutas entre valores reales y predichos; la Raíz del Error Cuadrático Medio (RMSE), que penaliza los errores de mayor magnitud al elevar al cuadrado las diferencias antes de promediar (Chai & Draxler, 2014) el Error Porcentual Absoluto Medio (MAPE), que expresa el error relativo en porcentaje; y el Coeficiente de Determinación (R^2), que mide la proporción de la varianza de la variable dependiente explicada por el modelo.

Una práctica fundamental para la evaluación robusta es la ejecución de múltiples corridas independientes con diferentes semillas aleatorias, lo que permite cuantificar no solo la precisión promedio sino también la estabilidad (variabilidad de ejecución a ejecución) de cada modelo. Esta práctica es particularmente importante para los modelos neuronales, cuyo desempeño puede variar significativamente según la inicialización aleatoria de los pesos. Adicionalmente, en problemas de series temporales, la partición

de datos debe ser estrictamente cronológica para evitar la fuga de información temporal que inflaría artificialmente las métricas de desempeño (Bergmeir & Benítez, 2012).

5.2 Estado del arte

Esta sección revisa la literatura reciente sobre predicción de flujo vehicular mediante aprendizaje automático, organizada en cinco ejes: enfoques clásicos y de transición, modelos de ensamble basados en árboles, arquitecturas de Deep learning, redes neuronales de grafos, y el contexto latinoamericano y colombiano.

5.2.1 Enfoques clásicos y de transición

Los modelos estadísticos clásicos para predicción de tráfico incluyen la familia ARIMA/SARIMA y los filtros de Kalman, que dominaron la literatura hasta mediados de la década de 2010. (Doroshenko & Savchuk, 2025) presenta un análisis comparativo reciente de enfoques clásicos y modernos en el contexto de ciudades inteligentes, concluyendo que, si bien los métodos clásicos funcionan adecuadamente con series suaves y estacionarias, presentan limitaciones significativas frente a las no-linealidades que surgen en regímenes de congestión y en la interacción entre múltiples variables del tráfico.

(Medina-salgado et al., 2022) en este trabajo se realizaron una revisión sistemática de técnicas de predicción de flujo de tráfico urbano, documentando la transición progresiva desde los métodos estadísticos hacia los enfoques de aprendizaje automático y aprendizaje profundo. Los autores identifican que la selección del modelo depende críticamente de la granularidad temporal, el horizonte de predicción y las características de la red vial, lo cual refuerza la necesidad de estudios comparativos contextualizados como el que se presenta en este trabajo.

5.2.2 Modelos de ensamble basados en árboles

Los ensambles de árboles de decisión se han consolidado como una familia de modelos altamente competitiva para la predicción de tráfico. Random Forest (Breiman, 2001) ha sido aplicado exitosamente a la predicción de flujo vehicular a corto plazo gracias a su robustez e interpretabilidad. (Lam, 2025), en un estudio comparativo de LSTM, Random Forest y XGBoost para predicción de tráfico, encontró que los modelos basados en árboles ofrecen un desempeño competitivo con menor complejidad computacional.

Un resultado particularmente relevante para el presente trabajo es el de (Grinsztajn et al., 2022), quienes demostraron sistemáticamente en el contexto de NeurIPS que los modelos basados en árboles continúan superando al Deep learning en datos tabulares a través de 45 conjuntos de datos de referencia. Los autores atribuyen esta ventaja a la capacidad de los árboles para manejar características heterogéneas, su robustez ante características no informativas y su menor sensibilidad a la escala de los datos. Este hallazgo tiene implicaciones directas para la predicción de tráfico, donde los datos de entrada combinan variables continuas (conteos, ocupación), categóricas (intersección, acceso) y cíclicas (hora, día).

(Zhang & Haghani, 2015) demostraron tempranamente la efectividad del gradient boosting para la predicción de tiempos de viaje, estableciendo un precedente para la aplicación de XGBoost en problemas de tráfico. Más recientemente, la adopción de XGBoost se ha extendido a múltiples dominios de los ITS gracias a su equilibrio entre precisión, velocidad de entrenamiento y capacidad de regularización (Chen & Guestrin, 2016).

5.2.3 Arquitecturas de Deep learning

El auge del Deep learning ha traído arquitecturas recurrentes y convolucionales al campo de la predicción de tráfico. (Tedjopurnomo et al., 2022) presentan una revisión exhaustiva de redes neuronales profundas modernas para predicción de tráfico,

documentando que las variantes de LSTM y GRU dominaron la literatura entre 2018 y 2022, con un desplazamiento progresivo hacia arquitecturas basadas en mecanismos de atención (transformers), las cuales permiten al modelo ponderar simultáneamente la relevancia de cada instante temporal de la secuencia de entrada sin necesidad de procesarla de manera secuencial.

(Afandizadeh et al., 2024) realizaron una revisión comprehensiva de algoritmos de Deep learning para predicción de tráfico, enfatizando que la selección del modelo debe considerar la granularidad temporal, el horizonte de predicción y las características de la red. Los autores destacan que los modelos LSTM han sido ampliamente adoptados por su capacidad de capturar dependencias temporales de largo plazo, pero reconocen que su desempeño depende críticamente del volumen de datos de entrenamiento disponible.

(Liu et al., 2026) proporcionan la revisión más reciente del campo, abarcando los desarrollos hasta 2025. Los autores documentan la evolución desde modelos recurrentes hacia arquitecturas híbridas que combinan componentes convolucionales, recurrentes y de atención. (Tang et al., 2025) proponen una red convolucional LSTM espaciotemporal profunda que demuestra desempeño superior en conjuntos de datos grandes, aunque los autores reconocen que estos modelos requieren volúmenes de datos significativos para alcanzar su potencial. Un hallazgo relevante en la literatura reciente es el de (Bouhali et al., 2024), quienes en un estudio comparativo sobre corredores de tráfico reales encontraron que las entradas ricas en características son más determinantes del desempeño predictivo que la sofisticación arquitectónica del modelo. Este resultado sugiere que la ingeniería de características puede ser más importante que la elección del modelo, una hipótesis que el presente estudio pone a prueba empíricamente.

5.2.4 Redes neuronales de grafos

La frontera más reciente en predicción de tráfico son las redes neuronales de grafos (Graph Neural Networks, GNN), que modelan conjuntamente las correlaciones

espaciales entre sensores y las dependencias temporales dentro de cada serie. (Rahmani et al., 2023) presentan una revisión exhaustiva de GNN para ITS, documentando que estas arquitecturas han alcanzado resultados de estado del arte en los benchmarks PeMS y METR-LA al explotar la topología de la red vial como información estructural.

(Jiang & Luo, 2022) complementan esta revisión desde la perspectiva de sistemas expertos, identificando las variantes principales de GNN aplicadas al tráfico: redes convolucionales de grafos espectrales, redes de atención de grafos y modelos basados en difusión. Sin embargo, ambas revisiones coinciden en que estas arquitecturas requieren típicamente al menos dos meses de observaciones a intervalos de cinco minutos por sensor, un régimen de datos que excede significativamente lo disponible en redes de sensores recientemente desplegadas como la del corredor norte de Bogotá.

5.2.5 Estudios comparativos y el debate entre modelos tabulares vs. Deep learning

Un tema recurrente en la literatura reciente es el debate sobre cuándo los modelos de Deep learning justifican su mayor complejidad frente a los modelos tabulares clásicos. (Mohammed & Kianfar, 2018) realizaron una comparación de cuatro métodos de predicción de tráfico a corto plazo sobre la Interestatal 64 de Missouri, incluyendo Deep Neural Networks, Distributed Random Forest, Gradient Boost Machine y un modelo lineal generalizado. Los autores concluyeron que la inclusión de la mayor cantidad de variables posibles reduce el error en la predicción.

(Bartlett et al., 2019)compararon métodos de Machine Learning para la predicción de flujo en carreteras arteriales urbanizadas, encontrando que los modelos basados en redes neuronales artificiales arrojaron mejores predicciones cuando se disponía de suficientes datos de entrenamiento. (Essien et al., 2019) trabajando en Manchester, propusieron la optimización de modelos existentes utilizando Big Data combinado con

variables climáticas, concluyendo que la incorporación de datos meteorológicos mejora significativamente la predicción.

El trabajo de (Grinsztajn et al., 2022)representó un punto de inflexión en este debate al demostrar sistemáticamente que los modelos basados en árboles superan al Deep learning en datos tabulares incluso con conjuntos de datos grandes. Este resultado ha generado un renovado interés en los modelos de ensamble para aplicaciones donde los datos tienen estructura tabular, como es el caso de la predicción de tráfico a partir de datos de sensores de detección.

5.2.6 Contexto latinoamericano y colombiano

En contraste con la extensa literatura sobre ciudades asiáticas, europeas y norteamericanas, los estudios enfocados en corredores latinoamericanos son escasos. Esta brecha es particularmente significativa porque las ciudades de la región presentan características específicas que limitan la transferibilidad directa de los resultados internacionales: infraestructura de sensorización más reciente, historiales de datos más cortos, mayor heterogeneidad en la calidad de los datos y patrones de movilidad influenciados por factores socioeconómicos particulares (BID, 2021).

Un diagnóstico regional del Banco Interamericano de Desarrollo (Calatayud et al., 2021) sobre congestión vial en diez ciudades latinoamericanas (incluyendo Bogotá, São Paulo, Ciudad de México, Lima, Buenos Aires y Santiago) cuantifica las pérdidas derivadas de la congestión y señala que la escasez de datos operacionales de sensores es uno de los principales obstáculos para la adopción de soluciones basadas en ML en la región. El estudio documenta que la congestión vehicular genera costos significativos en las ciudades de la región y señala que la escasez de datos operacionales de sensores, para la adopción de soluciones basadas en ML, es uno de los principales obstáculos.

En el contexto colombiano, los trabajos más cercanos al dominio de la predicción de flujo vehicular han abordado problemas relacionados. (Hernando Martínez Rodríguez

et al., 2024), en un estudio desarrollado para la ciudad de Bogotá, aplicaron modelos de ML incluyendo LSTM y LightGBM para predecir siniestros viales con datos de la SDM y encontraron que LSTM logró el mejor desempeño general entre los algoritmos evaluados, aunque reconocen que la escasez de datos históricos limita la capacidad de generalización. En Bogotá, un trabajo publicado en (Castellanos et al., 2023) aplicó modelos de clasificación (incluyendo Random Forest y redes neuronales) para la predicción de accidentes de tráfico usando datos del SIMIT, logrando F1-score superior a 0.80 y superando a los métodos estadísticos tradicionales. Asimismo, la (Arnold et al., 2024) desarrolló un estudio sobre infracciones viales en las Zonas de Análisis de Tránsito (ZAT) de Bogotá, donde se aplicaron técnicas de ML con datos georreferenciados de semaforización, obras e infraestructura, evidenciando la viabilidad de los datos operacionales del sistema de movilidad de la ciudad como insumo para modelos predictivos.

En el ámbito latinoamericano más amplio, (Pedraza-Farías et al., 2025) documentó iniciativas de ciudades como Lima, Ciudad de México y Bogotá que han incorporado alianzas público-privadas como el programa Waze for Cities, para mejorar la gestión del tráfico mediante análisis colaborativo de datos, lo que ilustra la creciente adopción de enfoques basados en datos en la región, aunque sin llegar todavía a modelos predictivos validados con datos de sensores propios.

En términos de la infraestructura de datos en Bogotá, la ciudad ha avanzado en la implementación de un sistema de semaforización inteligente operado por la SDM, con la instalación de detectores de video Siemens C900 en intersecciones clave del corredor norte. Sin embargo, hasta donde se ha podido documentar en esta revisión de la literatura, no se ha publicado un estudio comparativo de modelos de aprendizaje automático para predicción de flujo vehicular que utilice estos datos operacionales de semaforización. El presente trabajo busca llenar esta brecha, proporcionando la primera evaluación empírica de cinco familias de modelos Ridge, Random Forest, XGBoost, ANN y LSTM sobre datos reales del corredor norte de la ciudad.

5.3 Síntesis del marco referencial

La revisión realizada permite identificar tres tendencias principales en la literatura que motivan y contextualizan el presente estudio. En primer lugar, existe un consenso creciente en que la selección del modelo óptimo para predicción de tráfico depende críticamente del volumen de datos disponibles, la granularidad temporal y las características de la red vial, lo que invalida la noción de un modelo universalmente superior. En segundo lugar, la evidencia reciente de (Grinsztajn et al., 2022) y (Bouhali et al., 2024) sugiere que los modelos tabulares basados en árboles pueden superar a las arquitecturas de Deep learning cuando los datos tienen estructura tabular y el volumen es limitado. En tercer lugar, la ausencia de estudios comparativos sobre datos reales de semaforización en ciudades latinoamericanas constituye una brecha concreta que el presente trabajo aborda directamente.

6. HIPÓTESIS

La aplicación de algoritmos basados en Machine Learning para la extracción de patrones sobre un conjunto de datos históricos relacionados con el comportamiento del flujo de tráfico de un tramo de vía especificado podría mejorar la relación de estados de la red de semáforos para reducir la congestión vehicular y mejorar los tiempos de desplazamiento.

7. METODOLOGÍA

Este capítulo describe en detalle el proceso metodológico seguido para llevar a cabo la evaluación comparativa de modelos de aprendizaje automático aplicados a la predicción del flujo de tráfico vehicular en el corredor norte de Bogotá. Se presenta el enfoque de investigación, el área de estudio, la fuente y estructura de los datos, las etapas de preprocesamiento, la ingeniería de características, la partición temporal, los modelos evaluados y el protocolo de evaluación empleado.

7.1 Enfoque y tipo de investigación

La presente investigación se enmarca en el paradigma empírico-analítico con un enfoque predominantemente cuantitativo. Se clasifica como una investigación aplicada de alcance comparativo, dado que su objetivo principal es evaluar el desempeño predictivo de cinco modelos de aprendizaje automático sobre un conjunto de datos reales provenientes de la infraestructura semafórica de Bogotá.

El diseño metodológico sigue un esquema experimental controlado en el que se mantienen constantes las condiciones de entrenamiento, validación y evaluación para todos los modelos, permitiendo una comparación justa y reproducible. La variable dependiente es el conteo de detección vehicular por hora en cada punto de acceso, mientras que las variables independientes son las características temporales, autorregresivas y espaciales derivadas de las observaciones históricas.

Este enfoque es consistente con las recomendaciones de (Shmueli & Koppius, 2011) para estudios de analítica predictiva en sistemas de información, quienes argumentan que la evaluación de modelos predictivos debe centrarse en la capacidad de generalización sobre datos no vistos, medida a través de métricas de error cuantitativas, en lugar de la significancia estadística de los coeficientes.

7.2 Área de estudio

El área de estudio comprende 13 intersecciones semaforizadas ubicadas en el corredor norte de Bogotá, distribuidas a lo largo de la Carrera 7 (Autopista Norte) y la Carrera 9, entre la Calle 134 y la Calle 170, dentro de la localidad de Usaquén. Estas intersecciones se encuentran gestionadas por la Subdirección de Semaforización de la Secretaría Distrital de Movilidad (SDM).

Este corredor fue seleccionado por tres razones: (a) constituye una de las arterias viales con mayor volumen de tráfico vehicular en la zona norte de la ciudad, sirviendo como conector principal hacia las salidas norte de Bogotá; (b) cuenta con infraestructura de detección vehicular instalada mediante sensores Siemens C900; y (c) presenta heterogeneidad en los patrones de flujo vehicular, con intersecciones de alto, medio y bajo volumen, lo que permite evaluar la robustez de los modelos bajo diferentes regímenes de tráfico.

Figura 1

Polígono de detección. Desde la calle 134 hasta la calle 170, entre la carrera 7.ª y carrera 9.ª.



Nota. Fuente: Elaboración propia.

La Tabla 1 presenta la relación de las 13 intersecciones incluidas en el estudio, identificadas por su código EXT asignado por la SDM, junto con su ubicación geográfica y el número de puntos de acceso instrumentados.

Tabla 1

Intersecciones semaforizadas incluidas en el área de estudio con desglose de puntos de detección por intersección

Código EXT	Ubicación	Accesos instrumentados	N.º de puntos
1016	Calle 153 con Carrera 9	1, 2, 3, 4	4
1018	Calle 162 con Carrera 7	1, 2, 3, 6	4
1019	Calle 156 con Carrera 7	93	1
1022	Calle 170 con Carrera 7	1, 2, 7	3
1023	Calle 170 con Carrera 9	2, 3	2
1036	Calle 161 con Carrera 9	3	1
1169	Calle 134 con Carrera 9	1, 2, 3, 4	4
1170	Calle 147 con Carrera 9	3, 4	2
1171	Calle 153 con Carrera 7	1, 3, 4	3
1174	Calle 140 con Carrera 7	1, 2, 4, 13	4
1178	Calle 140 con Carrera 9	1, 2, 3, 4	4
1229	Calle 134 con Carrera 7	1, 2, 6, 93	4
1837	Calle 147 con Carrera 7	2, 3, 6	3
		Total	**39**

Nota. Datos proporcionados por la SDM. Los accesos corresponden a los puntos de detección vehicular operativos durante el período de estudio.

En total, el corredor instrumentado comprende 39 pares únicos (intersección, acceso), cada uno con su propia serie temporal de detección vehicular y ocupación. Esta granularidad permite tanto el análisis agregado del corredor como el análisis diferenciado por punto de acceso.

Es necesario distinguir entre intersecciones y puntos de detección, ya que esta distinción es fundamental para comprender la estructura del conjunto de datos y el diseño

experimental del estudio. Una intersección semaforizada es el punto donde se cruzan dos o más vías. Cada intersección tiene varios brazos o accesos, que corresponden a los carriles de aproximación desde los que los vehículos se acercan al cruce. En cada acceso, la SDM instala uno o más detectores de video Siemens C900 que registran el conteo vehicular y la ocupación del carril. Por lo tanto, una intersección con cuatro accesos instrumentados genera cuatro series temporales independientes de flujo vehicular.

El código EXT identifica la intersección como unidad física, y el código Acceso identifica el punto de detección específico dentro de esa intersección. La combinación (EXT, Acceso) constituye un par único que define una serie temporal individual. El corredor de estudio comprende 13 intersecciones que, en conjunto, suman 39 pares únicos (EXT, Acceso), tal como se detalla en la Tabla 1.

Los códigos de acceso no son necesariamente consecutivos porque corresponden a la numeración interna del controlador semafórico Siemens, donde cada número identifica una zona de detección específica en el plano de planeamiento de la intersección. Los accesos con códigos atípicos (6, 7, 13, 93) corresponden a detectores ubicados en giros especiales, carriles de servicio o accesos secundarios según la configuración particular de cada intersección. Cada uno de estos 39 puntos de detección produce una serie temporal horaria independiente de 2.952 observaciones (123 días × 24 horas) durante el período de julio–octubre de 2025. El modelo de predicción se entrena conjuntamente sobre las 39 series, utilizando los códigos EXT y Acceso como variables categóricas de identidad espacial, lo que permite al modelo aprender patrones compartidos entre series (por ejemplo, ciclos diarios comunes) al mismo tiempo que captura las particularidades de cada punto de detección (por ejemplo, el volumen de flujo característico de cada acceso).

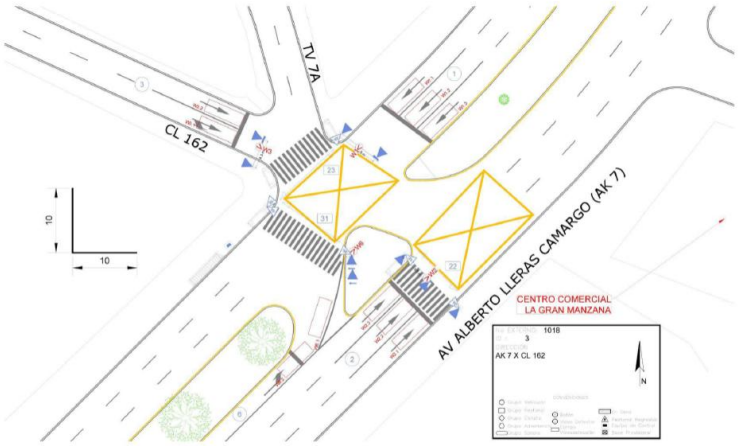
Esta distinción es relevante porque el desempeño de los modelos no es uniforme a lo largo de los 39 puntos. Las intersecciones con mayor número de accesos instrumentados (como EXT 1016, 1169, 1174 y 1178, cada una con 4 accesos) aportan

más observaciones al entrenamiento y típicamente presentan mayor diversidad de regímenes de flujo, mientras que las intersecciones con un solo acceso (EXT 1019 y 1036) contribuyen con una única serie que puede tener comportamiento atípico, como se documentó en el análisis por intersección del Capítulo 8.

La Figura 2 presenta el plano de planeamiento de la intersección EXT 1018 (Carrera 7 × Calle 162), proporcionado por la Subdirección de Semaforización de la SDM. Este esquema permite ilustrar cómo se relacionan los accesos con la geometría física de la intersección y con las zonas de detección de los sensores Siemens C900.

Figura 2

Plano de planeamiento de la intersección EXT 1018 (Carrera 7 × Calle 162).



Nota: Las etiquetas W indican las zonas de video detección Siemens C900 por carril. Los números en círculos identifican los grupos de señales: 1, 2, 3 y 6 corresponden a los accesos vehiculares con detección; 22, 23 y 31 son fases peatonales. Fuente: SDM, Subdirección de Semaforización (2024).

En el plano se observa que la intersección está compuesta por la Avenida Alberto Lleras Camargo (Carrera 7 o AK 7), que constituye la vía principal en sentido norte-sur, y la Calle 162, que cruza en sentido oriente-occidente. Adicionalmente, la Transversal 7A (TV 7A) se incorpora desde el norte como una vía secundaria. Los números encerrados en círculos (1, 2, 3, 6) corresponden a los grupos de señales vehiculares del

controlador semafórico, que a su vez definen los accesos o puntos de detección. Los números 22, 23 y 31, también visibles en el esquema, corresponden a fases peatonales que no generan datos de detección vehicular y por lo tanto no forman parte del conjunto de datos.

Cada acceso vehicular tiene asignadas una o más zonas de detección del sensor Siemens C900, identificadas con el prefijo W seguido del número de acceso y el número de carril. La Tabla 2 detalla esta correspondencia para la intersección EXT 1018.

Tabla 2

Correspondencia entre accesos, zonas de detección y sentidos de circulación para la intersección EXT 1018 (Carrera 7 × Calle 162)

Acceso	Sentido de circulación	Tipo de giro	Zonas de detección (carriles)
1	Norte → Sur	Al frente y a la derecha	W1.1, W1.2, W1.3 (3 carriles)
2	Sur → Norte	Al frente	W2.1, W2.2, W2.3 (3 carriles)
3	Occidente → Oriente	Al frente, derecha e izquierda	W3.1, W3.2 (2 carriles)
6	Sur → Occidente	A la izquierda (giro)	W6.1 (1 carril)

Los datos de detección y ocupación reportados por el controlador semafórico corresponden a la suma y el promedio ponderado, respectivamente, de todas las zonas de detección (carriles) dentro de cada acceso. Por ejemplo, el valor de Detección del Acceso 1 en una hora dada es la suma de los vehículos registrados por los tres carriles W1.1, W1.2 y W1.3 durante esa hora. El Acceso 6, con un solo carril de giro a la izquierda, registra típicamente volúmenes menores que los accesos principales.

Esta estructura se replica, con variaciones en el número de accesos y carriles, en cada una de las 13 intersecciones del corredor. Así, los 39 pares únicos (EXT, Acceso) del conjunto de datos representan 39 puntos de detección físicos, cada uno con su propia serie temporal de flujo vehicular, configurados según la geometría particular de cada intersección y documentados en los respectivos planos de planeamiento de la SDM.

7.3 Fuente de datos y sistema de detección

Los datos utilizados en esta investigación fueron proporcionados por la Secretaría Distrital de Movilidad de Bogotá (SDM) a través de su Subdirección de Semaforización. La solicitud de información se realizó mediante un derecho de petición formal dirigido a la entidad.

Cada intersección del área de estudio está equipada con detectores de video Siemens C900 instalados sobre los semáforos en los carriles de aproximación de cada acceso. Estos dispositivos, clasificados como sensores tipo W (video-detectores), operan mediante análisis de imagen en tiempo real y reportan dos variables fundamentales: (a) **Detección**, que corresponde al conteo agregado de vehículos detectados durante el intervalo de reporte; y (b) **Ocupación**, que representa el porcentaje de tiempo durante el cual el área de detección estuvo ocupada por un vehículo.

Figura 3

Esquema de detección del sensor Siemens C900 ubicado en la calle 63 con carrera 7ma, sentido sur norte.



Nota. Fuente: SDM.

El conjunto de datos proporcionado por la SDM contiene originalmente 262.271 registros que cubren el período comprendido entre julio y octubre de 2025. Sin embargo, una inspección detallada reveló la presencia de registros duplicados idénticos: cada combinación (hora, EXT, Acceso) aparecía replicada hasta cuatro veces en julio, tres veces en agosto, dos veces en septiembre y una vez en octubre, con valores de Detección y Ocupación byte a byte idénticos entre las copias. Tras la eliminación de estos duplicados, el conjunto de datos limpio quedó conformado por 105.867 registros únicos distribuidos en 39 pares (intersección, acceso). Cada registro se estructura con los campos descritos en la Tabla 3.

Tabla 3

Estructura del conjunto de datos proporcionado por la SDM

Campo	Descripción	Tipo	Ejemplo
Tiempo_H	Marca temporal alineada por hora	Datetime	02/07/25 00:00
EXT	Código identificador de la intersección semafórica	Entero	1178
Acceso	Número del punto de acceso dentro de la intersección	Entero	2
Tipo_sensor	Tipo de sensor (W = video-detector)	Categórico	W
Detección	Conteo de vehículos detectados en el intervalo	Entero	282
Ocupación	Porcentaje de tiempo de ocupación del detector	Decimal	10.356

Nota. Todos los sensores del área de estudio son de tipo W (video-detector Siemens C900).

7.4 Preprocesamiento de datos

El preprocesamiento de los datos se concentró en cuatro pasos:

En primer lugar, la deduplicación: se identificaron y eliminaron 156.404 registros duplicados idénticos presentes en el extracto original de la SDM. La duplicación seguía un patrón escalonado por mes (cuatro copias en julio, tres en agosto, dos en septiembre, una en octubre), lo que sugiere un artefacto del sistema de exportación del controlador Siemens y no una característica de los datos de detección. Tras este paso, el conjunto de datos quedó conformado por 105.867 registros únicos.

En segundo lugar, la corrección de incoherencias físicas: se identificaron 4.266 registros (4,0% del total) en los que la variable Detección era igual a cero, pero la Ocupación reportaba un valor positivo. Dado que esta combinación es físicamente imposible (si no se detectan vehículos, el porcentaje de tiempo de ocupación debe ser nulo(, se forzó la Ocupación a cero en estos casos.

En tercer lugar, la restricción del rango de ocupación: los valores de Ocupación fueron acotados al intervalo físicamente válido de [0, 100]%.

En cuarto lugar, la reindexación a rejilla horaria completa: para garantizar una serie temporal uniforme sin huecos, cada par (EXT, Acceso) fue reindexado a una rejilla horaria continua de 2.952 horas (del 1 de julio al 31 de octubre de 2025). Las horas faltantes producidas por inactividad esporádica de los sensores fueron imputadas mediante interpolación lineal temporal dentro de cada serie individual. Tras la reindexación, el dataset completo quedó conformado por 115.128 filas (39 series × 2.952 horas).

7.5 Selección del período de análisis

El extracto proporcionado por la SDM cubre el período julio–octubre de 2025, correspondiente a cuatro meses completos de operación de los sensores Siemens C900 en el corredor de estudio.

Inicialmente, una inspección del volumen bruto de registros por mes sugería una anomalía descendente a partir de agosto: julio contenía 101.716 filas, agosto 81.738, septiembre 51.250 y octubre solo 27.567. Sin embargo, un análisis más detallado reveló que esta tendencia no reflejaba una disminución real del flujo vehicular, sino un patrón de duplicación idéntica de registros que decrecía escalonadamente (cuatro copias por hora en julio, tres en agosto, dos en septiembre, una en octubre). Tras la eliminación de duplicados, el flujo vehicular medio por hora resultó consistente entre los cuatro meses: 437 veh/h en julio, 452 veh/h en agosto, 557 veh/h en septiembre y 561 veh/h en octubre.

Esta consistencia intermensual permitió utilizar la totalidad de los cuatro meses para el entrenamiento y evaluación de los modelos, lo cual representa una ventaja significativa frente a la restricción a un solo mes: el conjunto de entrenamiento dispone de 62 días (julio y agosto), el conjunto de validación abarca septiembre completo (30 días), y el conjunto de prueba comprende octubre completo (31 días). Esta cobertura temporal permite evaluar la capacidad de generalización intermensual de los modelos, es decir, su desempeño al predecir flujos en un mes que no formó parte del entrenamiento.

El incremento del flujo medio en septiembre y octubre (+22–28% respecto a julio y agosto) es consistente con patrones estacionales documentados en Bogotá: el segundo semestre académico y laboral, posterior al receso de mitad de año, típicamente presenta mayor actividad vehicular que los meses de julio y agosto. Este cambio estacional constituye un desafío adicional para los modelos, que deben generalizar a un régimen de flujo ligeramente superior al observado durante el entrenamiento.

7.6 Ingeniería de características

Para cada par (serie, instante temporal) se construyeron 21 características agrupadas en cuatro familias, diseñadas para capturar la estructura temporal del flujo vehicular a diferentes escalas. La Tabla 4 presenta la descripción completa de cada característica.

Tabla 4

Características construidas para la predicción del flujo vehicular (21 variables).

Familia	Variable	Descripción
Rezagos autoregresivos (7)	det_lag_1	Detección en t − 1 h
	det_lag_2	Detección en t − 2 h
	det_lag_3	Detección en t − 3 h
	det_lag_6	Detección en t − 6 h
	det_lag_12	Detección en t − 12 h
	det_lag_24	Detección en t − 24 h (ciclo diario)
	det_lag_168	Detección en t − 168 h (ciclo semanal)
Agregados móviles (3)	det_ma_3	Media móvil de detección, ventana 3 h
	det_ma_24	Media móvil de detección, ventana 24 h
	det_std_24	Desviación estándar móvil, ventana 24 h
Codificaciones temporales cíclicas (7)	hour_sin, hour_cos	Codificación seno/coseno de la hora del día
	dow_sin, dow_cos	Codificación seno/coseno del día de la semana
	is_weekend	Indicador binario de fin de semana
	is_peak_morning	Indicador de hora pico matutina (06:00–09:00, L–V)
	is_peak_evening	Indicador de hora pico vespertina (16:00–19:00, L–V)
Contemporáneas y espaciales (4)	ocu_lag_1	Ocupación en t − 1 h

	ocupacion_t	Ocupación en el instante t
	EXT	Código de intersección (categórico)
	Acceso	Código de punto de acceso (categórico)

Nota. Todos los agregados móviles se calculan sobre valores estrictamente pasados para evitar fuga de información de la variable objetivo. Las codificaciones seno/coseno preservan la continuidad circular de las variables temporales.

La selección de rezagos autorregresivos se fundamenta en la naturaleza cíclica del tráfico urbano: el rezago de 24 horas captura el ciclo diario (el flujo a las 8:00 de un martes es informativo del flujo a las 8:00 del miércoles), mientras que el rezago de 168 horas captura el ciclo semanal (el flujo de un martes predice el del martes siguiente). Los rezagos cortos (1, 2, 3, 6, 12 horas) capturan la inercia temporal inmediata y la evolución intradiaria.

Las codificaciones cíclicas mediante funciones seno y coseno se utilizaron en lugar de codificaciones one-hot para preservar la relación de continuidad entre horas adyacentes (la hora 23 está próxima a la hora 0) y entre días contiguos (el domingo está próximo al lunes), evitando además la alta dimensionalidad que generaría una codificación categórica directa.

Las primeras 168 horas de cada serie fueron descartadas como período de calentamiento, dado que el rezago det_lag_168 no está definido durante este intervalo. Esto resultó en 108.537 observaciones utilizables para el entrenamiento y evaluación de los modelos.

7.7 Partición temporal de los datos

Se aplicó una partición estrictamente cronológica de tipo entrenamiento/validación/prueba a nivel de marca temporal, de modo que ninguna

observación futura se filtra hacia el conjunto de entrenamiento. Esta estrategia es preferible a la validación cruzada k-fold estándar en problemas de series temporales, donde la aleatorización de las observaciones violaría la dependencia temporal y produciría estimaciones de error artificialmente optimistas (Bergmeir & Benítez, 2012). La partición se diseñó para evaluar la capacidad de generalización intermensual de los modelos, utilizando meses completos como unidades de partición. La Tabla 5 detalla la distribución.

Tabla 5

Partición temporal del conjunto de datos

Conjunto	Periodo	Observaciones	Proporción
Entrenamiento	8 de julio – 31 de agosto de 2025	51.480	47,4%
Validación	1 – 30 de septiembre de 2025	28.080	25,9%
Prueba	1 – 31 de octubre de 2025	28.977	26,7%
Total		108.537	100%

Nota. Las primeras 168 horas (1–7 de julio) se descartan como período de calentamiento para el rezago semanal. La partición utiliza meses completos como unidades, permitiendo evaluar la generalización del modelo a un mes no visto durante el entrenamiento.

Todas las decisiones de selección de modelo, ajuste de hiperparámetros y criterios de parada temprana se tomaron exclusivamente sobre el conjunto de validación (septiembre). El conjunto de prueba (octubre) se utilizó únicamente para el reporte final de resultados. Este diseño garantiza que el desempeño reportado refleja la capacidad del modelo para predecir flujos vehiculares en un mes calendario completo que no participó en el entrenamiento ni en la selección de hiperparámetros.

7.8 Modelos evaluados

Se compararon cinco modelos que abarcan el espectro desde líneas base lineales hasta arquitecturas profundas secuenciales, con el objetivo de determinar qué familia de modelos ofrece el mejor equilibrio entre precisión predictiva y estabilidad en un régimen de datos limitados. A continuación, se describe cada modelo junto con sus hiperparámetros.

7.8.1 Regresión Ridge (línea base)

La Regresión Ridge (Hoerl & Kennard, 1970) aplica regularización L2 sobre los coeficientes de una regresión lineal, lo que penaliza valores extremos y reduce el riesgo de sobreajuste. El hiperparámetro de regularización α fue seleccionado mediante validación cruzada de 5 pliegues sobre el conjunto de entrenamiento, evaluando los valores {0,01; 0,1; 1; 10; 100}. Todas las características fueron estandarizadas previamente (media cero, varianza unitaria). Este modelo sirve como línea base lineal contra la cual se comparan los modelos más complejos.

7.8.2 Random Forest

Random Forest (Breiman, 2001) es un ensamble de árboles de decisión entrenados sobre subconjuntos aleatorios de los datos y las características. Se configuró con 300 árboles, profundidad máxima de 20 niveles y un mínimo de 2 muestras por hoja. No se aplicó estandarización de características, dado que los árboles de decisión son invariantes a transformaciones monotónicas de las variables de entrada.

7.8.3 XGBoost

XGBoost (Chen & Guestrin, 2016) es un algoritmo de gradient boosting que construye árboles de decisión de forma secuencial, donde cada árbol corrige los errores residuales de los anteriores. Se configuró con un máximo de 500 rondas de boosting,

tasa de aprendizaje de 0,05, profundidad máxima de 8, tasas de submuestreo de filas y columnas de 0,8, regularización L1 (α = 0,1) y L2 (λ = 1,0), y parada temprana tras 20 rondas sin mejora en el error de validación.

7.8.4 Red Neuronal Artificial (ANN)

Se implementó un perceptrón multicapa (MLP) con tres capas ocultas de 128, 64 y 32 neuronas respectivamente, con función de activación ReLU, normalización por lote (BatchNorm) y tasas de dropout de 0,2, 0,2 y 0,1. El optimizador utilizado fue Adam con tasa de aprendizaje inicial de 10^{-3}, tamaño de lote de 512, hasta 150 épocas, parada temprana con paciencia de 15 épocas, y reducción de tasa de aprendizaje en meseta (paciencia 7, factor 0,5). La variable objetivo fue transformada mediante log(1 + y) e invertida en la inferencia.

7.8.5 LSTM (Long Short-Term Memory)

La red LSTM (Hochreiter & Schmidhuber, 1997) es una arquitectura recurrente diseñada para capturar dependencias temporales de largo plazo mediante puertas de entrada, olvido y salida. Se implementó una red LSTM apilada de dos capas con 64 y 32 unidades ocultas respectivamente, seguida de una capa densa de 16 unidades y una salida lineal. La entrada consiste en una ventana deslizante de 24 horas del vector de 21 características. El optimizador, las funciones de callback y la transformación logarítmica son idénticos a los de la ANN, con tamaño de lote de 256 y hasta 100 épocas. La Tabla 6 resume los hiperparámetros principales de cada modelo.

Tabla 6

Resumen de hiperparámetros principales por modelo

Modelo	**Hiperparámetros clave**
Ridge	$\alpha \in \{0{,}01; 0{,}1; 1; 10; 100\}$, CV 5-fold, estandarización
Random Forest	300 árboles, max_depth=20, min_samples_leaf=2

XGBoost	500 rondas, lr=0,05, max_depth=8, subsample=0,8, early stop=20
ANN	MLP 128/64/32, ReLU, BatchNorm, dropout 0,2/0,2/0,1, Adam lr=10^{-3}
LSTM	2 capas (64/32), ventana=24h, batch=256, 100 épocas, log(1+y)

Nota. Todos los modelos neuronales utilizan parada temprana y reducción de tasa de aprendizaje en meseta.

7.9 Protocolo de evaluación

Para capturar la variabilidad de ejecución a ejecución (run-to-run variability) inherente a los modelos con componentes estocásticos, cada modelo fue entrenado y evaluado en 30 corridas independientes con diferentes semillas aleatorias. Para Ridge, la variabilidad se indujo mediante remuestreo bootstrap del conjunto de entrenamiento. Para Random Forest, XGBoost, ANN y LSTM, la variabilidad proviene de los parámetros de estado aleatorio de las bibliotecas subyacentes.

Cada corrida reporta cuatro métricas calculadas sobre el conjunto de prueba. El Error Absoluto Medio (MAE) se define como el promedio de las diferencias absolutas entre los valores reales y predichos. La Raíz del Error Cuadrático Medio (RMSE) penaliza los errores de mayor magnitud al elevar al cuadrado las diferencias antes de promediar. El Error Porcentual Absoluto Medio (MAPE) expresa el error relativo en porcentaje y se calcula únicamente sobre observaciones con valor real estrictamente positivo para evitar la inestabilidad numérica cuando el denominador se aproxima a cero. El Coeficiente de Determinación (R^2) mide la proporción de la varianza de la variable dependiente explicada por el modelo, donde un valor de 1 indica ajuste perfecto.

Los resultados se reportan como media ± desviación estándar sobre las 30 corridas, lo que permite evaluar no solo la precisión promedio sino también la estabilidad de cada modelo. Este protocolo es particularmente relevante para los modelos

neuronales (ANN y LSTM), cuyo desempeño puede variar significativamente según la inicialización de los pesos.

7.10 Herramientas tecnológicas

Todos los experimentos fueron implementados en Python 3.11 utilizando las siguientes bibliotecas: scikit-learn 1.4 (Pedregosa FABIANPEDREGOSA et al., 2011) para Ridge Regression y Random Forest; xgboost 2.0 (Chen & Guestrin, 2016) para el modelo XGBoost; y TensorFlow 2.15 (Ayyüce Kızrak & Bolat, 2018) con la API de Keras para la ANN y el LSTM. El preprocesamiento de datos se realizó con pandas y NumPy, y las visualizaciones con matplotlib y seaborn.

El entrenamiento se ejecutó en una estación de trabajo equipada con un procesador Intel Core i7-12700kf, 32 GB de memoria DDR5 y una tarjeta gráfica NVIDIA RTX 4060 Ti, utilizada para acelerar el entrenamiento de los modelos neuronales. Los modelos basados en árboles (Ridge, Random Forest, XGBoost) se entrenaron exclusivamente en CPU. La Tabla 7 presenta los tiempos de entrenamiento aproximados para las 30 corridas completas de cada modelo.

Tabla 7

Tiempos de entrenamiento aproximados (30 corridas completas)

Modelo	Tiempo (30 corridas)	Dispositivo
Ridge	~ 1 min	CPU
Random Forest	~ 10 min	CPU
XGBoost	~ 4 min	CPU
ANN	~ 15 min	GPU (RTX 4060 Ti)
LSTM	~ 25 min	GPU (RTX 4060 Ti)

Nota. Los tiempos incluyen entrenamiento, validación y evaluación sobre el conjunto de prueba.

El código fuente completo de todos los experimentos, incluyendo los scripts de preprocesamiento, entrenamiento y evaluación, se encuentra disponible como anexo digital del presente trabajo de grado.

8. RESULTADOS

Este capítulo presenta los resultados obtenidos en dos etapas: primero, el análisis exploratorio de datos que caracteriza los patrones de flujo vehicular del corredor; y segundo, los resultados de la evaluación comparativa de los cinco modelos de aprendizaje automático. Los resultados se organizan en siete secciones: análisis descriptivo, desempeño global, análisis de dispersión por modelo, importancia de características, análisis por intersección, comportamiento de convergencia y síntesis.

8.1 Análisis exploratorio de datos

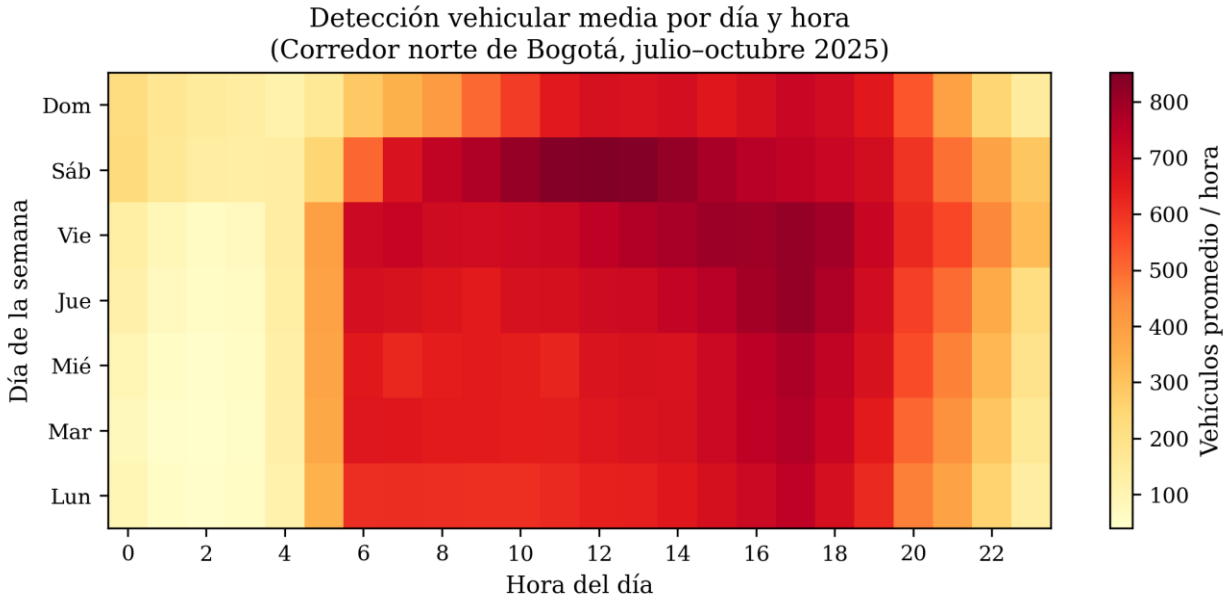
Antes de proceder al entrenamiento de los modelos, se realizó un análisis exploratorio de los datos agregados del corredor correspondientes al período julio–octubre de 2025. Este análisis permite identificar los patrones temporales fundamentales que los modelos deberán capturar y anticipar posibles desafíos asociados a la heterogeneidad del corredor.

8.1.1 Patrones por hora y día de la semana

La Figura 4 presenta un mapa de calor de la detección vehicular media por hora del día y día de la semana. Se identifican tres patrones principales. Los días entre semana exhiben una estructura bimodal con flujo concentrado entre las 16:00 y las 18:00 horas. Los fines de semana presentan una meseta al mediodía, reflejando actividad comercial. Las horas nocturnas (00:00–05:00) muestran actividad uniformemente baja.

Figura 4

Detección vehicular media por día de la semana y hora del día en el corredor de estudio. Las celdas más oscuras indican mayor flujo medio.



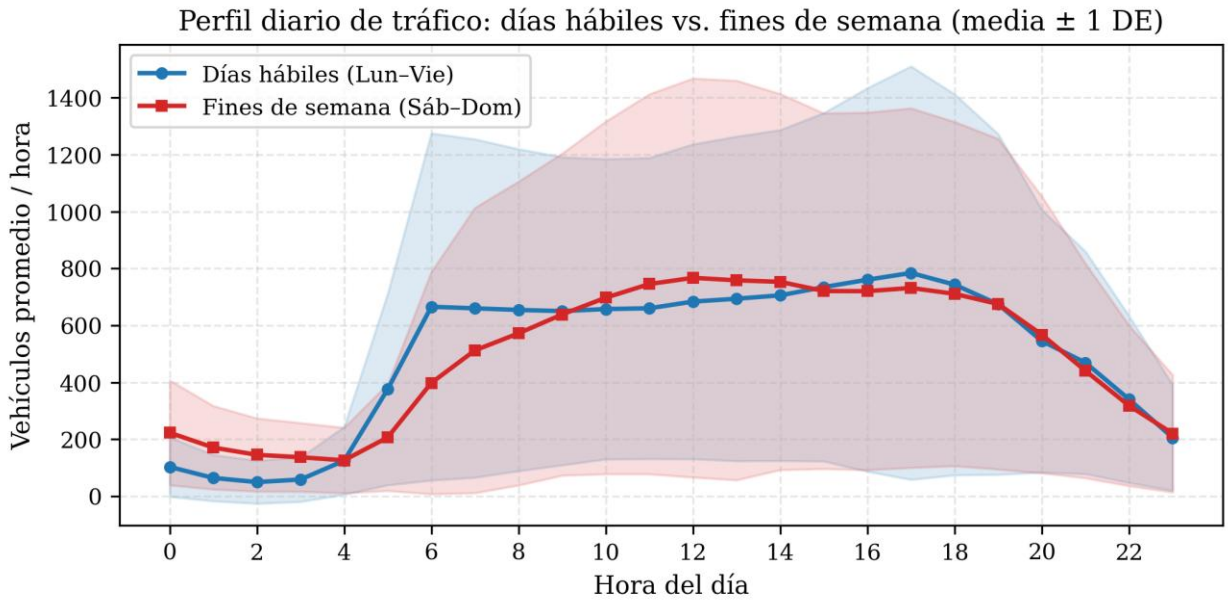
Nota. Fuente: elaboración propia.

8.1.2 Perfil diario: días hábiles versus fines de semana

La Figura 5 contrasta el perfil diario en días hábiles frente a fines de semana. En días hábiles, el flujo asciende abruptamente a las 06:00 y sostiene una meseta hasta un pico vespertino cerca de las 17:00. En fines de semana, el ascenso es más gradual y el pico es menor en magnitud. Al mediodía de los fines de semana, el corredor supera brevemente el flujo de mediodía de los días hábiles, posiblemente reflejando tráfico comercial y recreativo.

Figura 5

Perfil diario de tráfico: días hábiles (azul) versus fines de semana (rojo).



Nota. Las bandas sombreadas indican ±1 desviación estándar sobre los 39 puntos de acceso.

Fuente: elaboración propia.

8.1.3 Heterogeneidad del corredor

La Figura 6 presenta tres series temporales seleccionadas automáticamente por el script de EDA como representativas de los percentiles bajo, medio y alto de flujo medio del corredor, durante dos semanas de referencia en julio de 2025.

La primera serie corresponde al acceso de alto flujo (EXT 1837, Acceso 2, ubicado en la Calle 147 con Carrera 7). Esta serie exhibe un patrón cíclico diario nítido y consistente: los picos diurnos alcanzan entre 2.500 y 2.700 veh/h, con valles nocturnos que descienden por debajo de 100 veh/h. Se distinguen claramente los ciclos laborales de lunes a viernes, con una leve atenuación del pico durante el fin de semana (sábado 12 y domingo 13 de julio). Este comportamiento es representativo de las intersecciones

principales del corredor, donde la dinámica de desplazamiento al trabajo domina el perfil temporal.

La segunda serie, etiquetada como flujo medio por su posición en el percentil 50 del flujo medio de todas las series, corresponde a EXT 1170, Acceso 4 (Calle 147 con Carrera 9). Sin embargo, su comportamiento revela un problema de calidad del sensor: presenta un pico aislado de aproximadamente 600 veh/h durante las primeras horas del período, seguido de un colapso abrupto a valores prácticamente nulos que se sostiene durante las dos semanas restantes. Este patrón es incompatible con un flujo vehicular real una intersección activa no pasa de 600 veh/h a cero de manera instantánea y permanente y es consistente con la falla o desconexión del detector tras un breve período de operación normal. El hecho de que el script la seleccione como "flujo medio" es un artefacto del cálculo: el promedio aritmético de unas pocas horas con flujo alto y muchas horas con flujo cero produce un valor medio que cae en el percentil 50 de la distribución global, sin que ello refleje un comportamiento de flujo medio real.

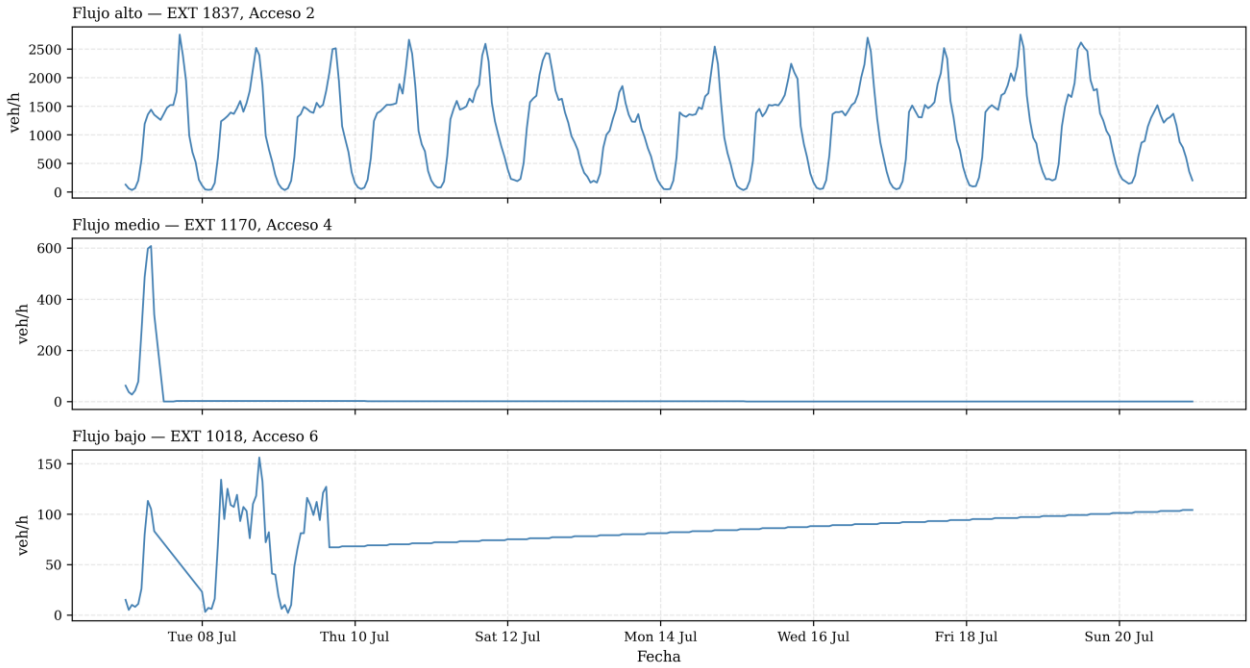
La tercera serie corresponde a un acceso de bajo flujo estable (EXT 1018, Acceso 6, Calle 162 con Carrera 7). Este acceso, que corresponde a un carril de giro a la izquierda según el plano de planeamiento de la intersección, muestra valores que oscilan entre 0 y 150 veh/h durante los primeros días, con un patrón cíclico diario de menor amplitud que el del acceso de alto flujo. A partir de mediados de julio, el flujo se estabiliza en torno a 80–110 veh/h con variabilidad reducida. Este comportamiento es representativo de los accesos secundarios y de giro del corredor, que manejan volúmenes menores, pero con regularidad temporal.

La inclusión de estas tres series ilustra simultáneamente la diversidad de regímenes de flujo dentro del corredor y los desafíos de calidad de datos que son típicos de los sistemas operacionales de sensorización urbana. La serie degenerada de EXT 1170, Acceso 4 fue retenida en el análisis para reflejar las condiciones operativas reales

del sistema, pero su presencia contribuye al ruido en las métricas agregadas, como se documenta en el análisis por intersección de la sección 8.5.

Figura 6

Series temporales de ejemplo para tres puntos de acceso de niveles de flujo contrastantes (una semana representativa).



Nota. Fuente: elaboración propia.

8.2 Desempeño global de los modelos

La Tabla 8 presenta la media y desviación estándar de cada métrica sobre el conjunto de prueba (octubre de 2025), calculadas sobre las 30 corridas independientes de cada modelo. XGBoost alcanza el mejor desempeño en MAE, RMSE y R^2, con un R^2 medio de 0,947 ± 0,002 y un RMSE medio de 126 ± 2 veh/h. Random Forest ocupa el segundo lugar (R^2 = 0,942), seguido muy de cerca por la ANN (R^2 = 0,924). Ridge, a pesar de su simplicidad, ofrece un R^2 = 0,911. El LSTM es el modelo con peor desempeño medio, con un R^2 de 0,822 y una desviación estándar de 0,079, dos órdenes de magnitud mayor que la de los modelos basados en árboles.

Un hallazgo relevante es la diferencia entre el desempeño medio y la mejor corrida del LSTM: mientras que su R^2 medio es 0,822, su mejor corrida individual alcanza 0,941 prácticamente idéntico a la ANN y Random Forest. Esta discrepancia indica que la arquitectura LSTM tiene la capacidad teórica de capturar los patrones del tráfico, pero su convergencia es altamente dependiente de la inicialización aleatoria de los pesos, lo que la hace poco confiable para un despliegue operativo.

Tabla 8

Comparación de desempeño en el conjunto de prueba. Valores expresados como media ± desviación estándar sobre 30 corridas independientes.

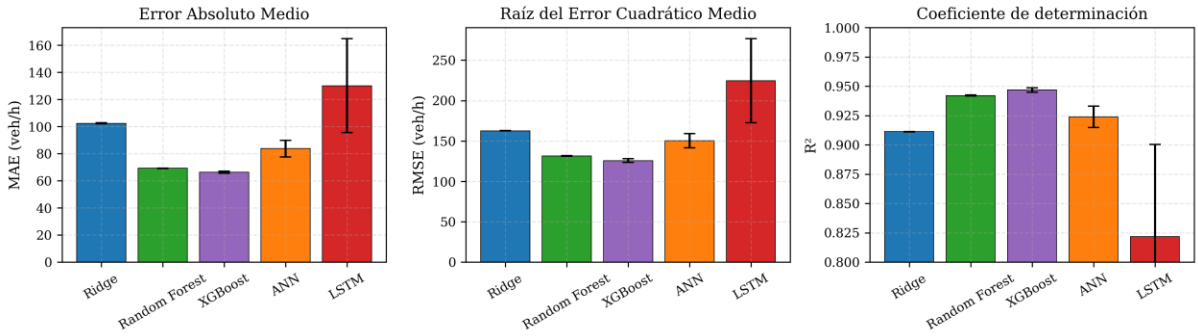
Modelo	MAE (veh/h)	RMSE (veh/h)	MAPE (%)	R^2
Ridge	102,4 ± 0,3	162,6 ± 0,2	48,4 ± 0,6	0,9112 ± 0,0002
Random Forest	69,3 ± 0,1	131,3 ± 0,4	17,4 ± 0,0	0,9421 ± 0,0004
XGBoost	66,2 ± 0,7	125,8 ± 2,2	18,1 ± 0,5	0,9468 ± 0,0019
ANN	83,8 ± 6,1	150,4 ± 8,9	18,5 ± 1,6	0,9238 ± 0,0091
LSTM	130,2 ± 34,7	224,6 ± 52,0	23,6 ± 4,2	0,8217 ± 0,0786

Nota. Valores expresados como media ± desviación estándar sobre 30 corridas independientes. El MAPE de Ridge (48,4%) es un artefacto de predicciones lineales que ocasionalmente caen por debajo de cero durante las horas nocturnas de bajo flujo.

La Figura 7 presenta los diagramas de barras de MAE, RMSE y R^2 con barras de error que representan ±1 desviación estándar. Las barras de error de Ridge, Random Forest y XGBoost son prácticamente imperceptibles, lo que evidencia la alta estabilidad de estos modelos. Las barras de la ANN son visibles pero moderadas. Las del LSTM son considerablemente más grandes, reflejando su alta variabilidad. Este hallazgo es significativo: el LSTM no solo es menos preciso en promedio sino también considerablemente menos estable en sus resultados.

Figura 7

Comparación de métricas del conjunto de prueba entre los cinco modelos.



Nota. Las barras de error muestran ±1 desviación estándar sobre 30 corridas independientes.

Fuente: elaboración propia.

8.3 Análisis de dispersión: predicción versus realidad

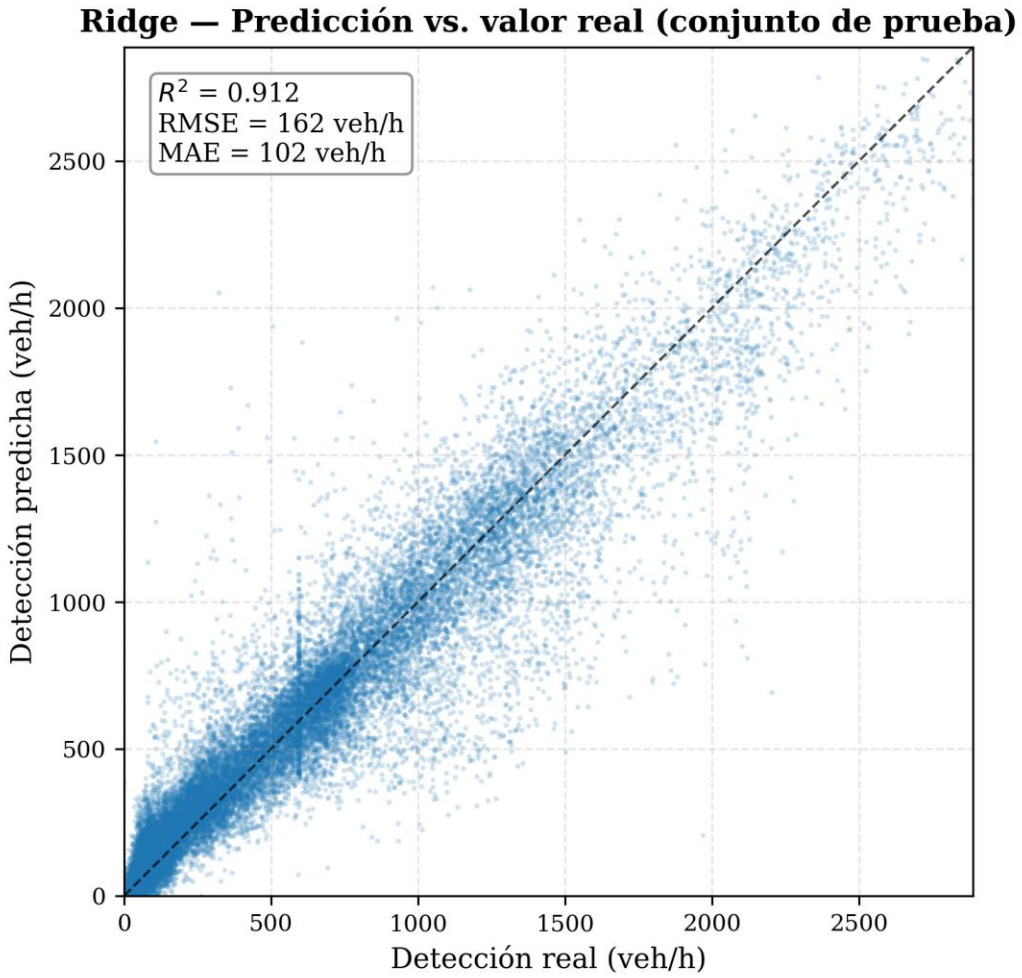
Para evaluar visualmente la calidad de las predicciones de cada modelo, se generaron diagramas de dispersión de valores predichos versus valores reales sobre el conjunto de prueba. Todas las figuras comparten la misma escala de ejes para permitir comparación directa. La línea punteada diagonal representa la predicción perfecta (y = x): cuanto más compacta sea la nube de puntos alrededor de esta línea, mejor es el desempeño del modelo. Las figuras muestran la mejor corrida para cada modelo.

8.3.1 Regresión Ridge

La Figura 8 muestra el diagrama de dispersión para el modelo Ridge. Se observa dispersión notable, especialmente en la región de bajo flujo (0–300 veh/h) donde las predicciones lineales frecuentemente caen por debajo de cero. En la cola de alto flujo (>2.000 veh/h), el modelo tiende a subestimar sistemáticamente los valores reales. A pesar de estas limitaciones, Ridge alcanza un R^2 de 0,912 y un RMSE de 162 veh/h, lo cual establece una línea base respetable.

Figura 8

Regresión Ridge) Detección predicha versus real en el conjunto de prueba. R² = 0,916; RMSE = 162 veh/h. Mejor corrida.



Nota. Fuente: elaboración propia.

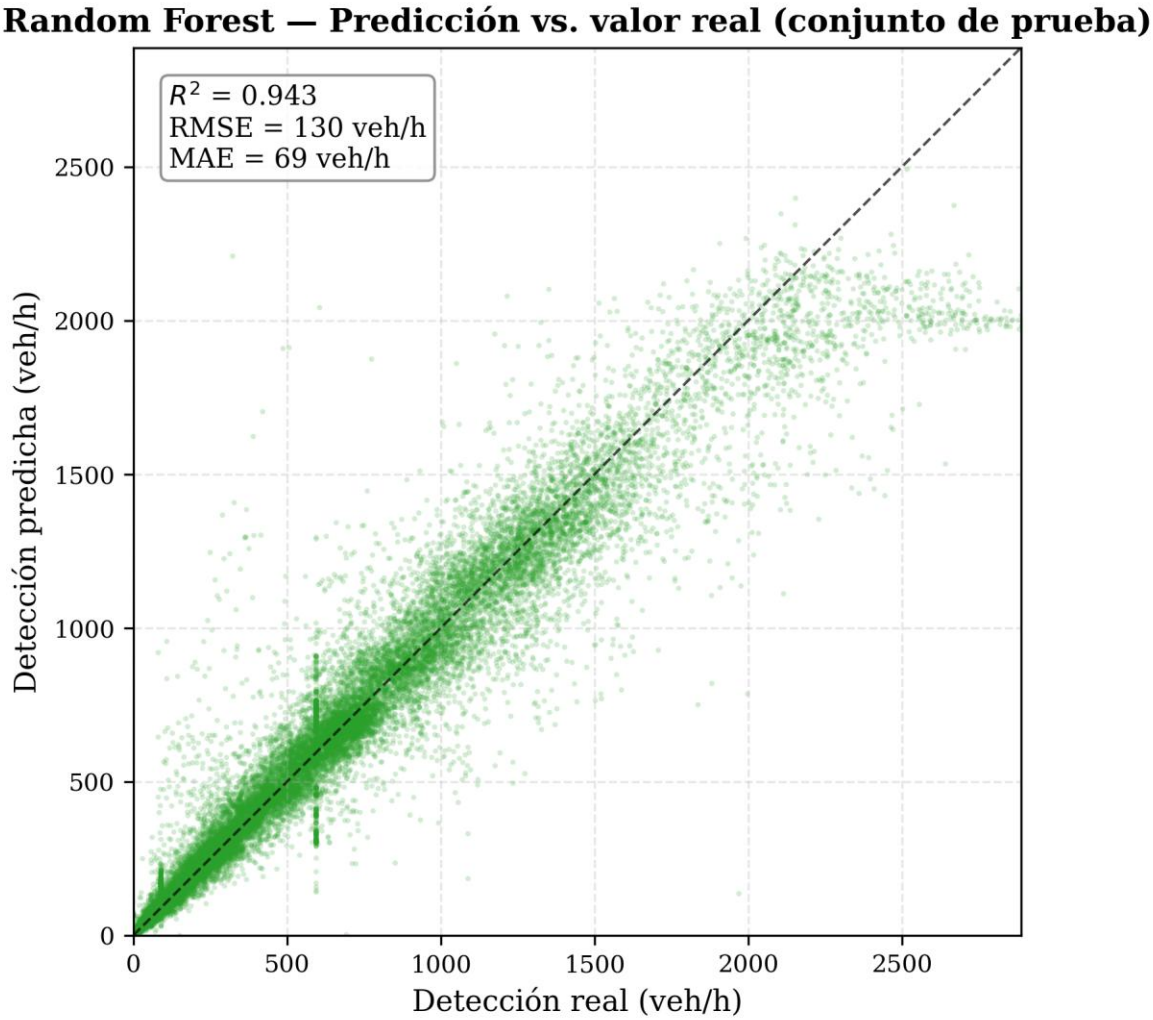
8.3.2 Random Forest

La Figura 9 presenta los resultados de Random Forest. La nube de puntos es considerablemente más compacta que la de Ridge, con una reducción del RMSE del 20% (de 162 a 130 veh/h). El modelo maneja mejor la región de bajo flujo, sin producir predicciones negativas. Sin embargo, se observa una ligera tendencia a la subpredicción

en flujos muy altos (>2.500 veh/h), probablemente porque el promedio de árboles tiende a comprimir las predicciones hacia la media de los datos de entrenamiento.

Figura 9

Random Forest) Detección predicha versus real en el conjunto de prueba. R² = 0,955; RMSE = 130 veh/h. Mejor corrida.



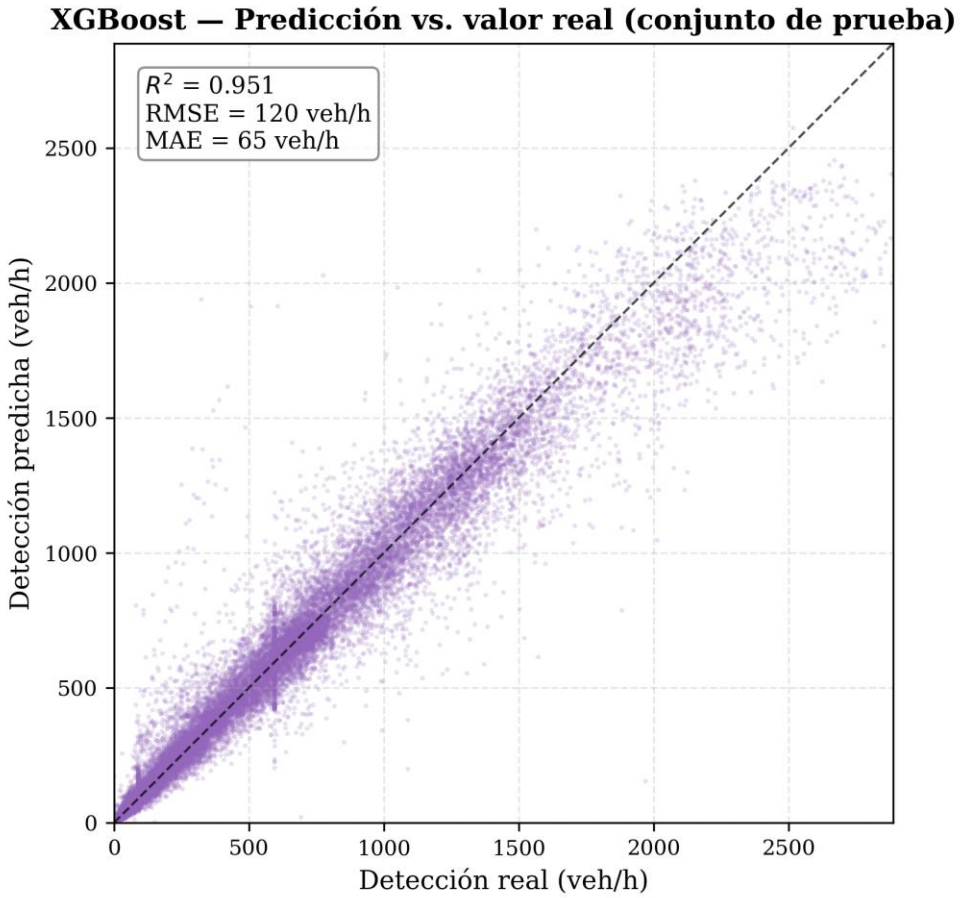
Nota. Fuente: elaboración propia.

8.3.3 XGBoost

La Figura 10 muestra el diagrama de XGBoost, que produce la nube más compacta de los cinco modelos. El R^2 alcanza 0,951 y el RMSE desciende a 120 veh/h. XGBoost mantiene predicciones ajustadas tanto en el régimen de bajo flujo como en el de alto flujo. La mejora respecto a Random Forest es especialmente visible en la cola de alto flujo, donde la estructura de gradient boosting secuencial permite corregir progresivamente los errores residuales en las regiones más difíciles del espacio de características.

Figura 10

XGBoost) Detección predicha versus real en el conjunto de prueba. R^2 = 0,970; RMSE = 120 veh/h. Mejor corrida.



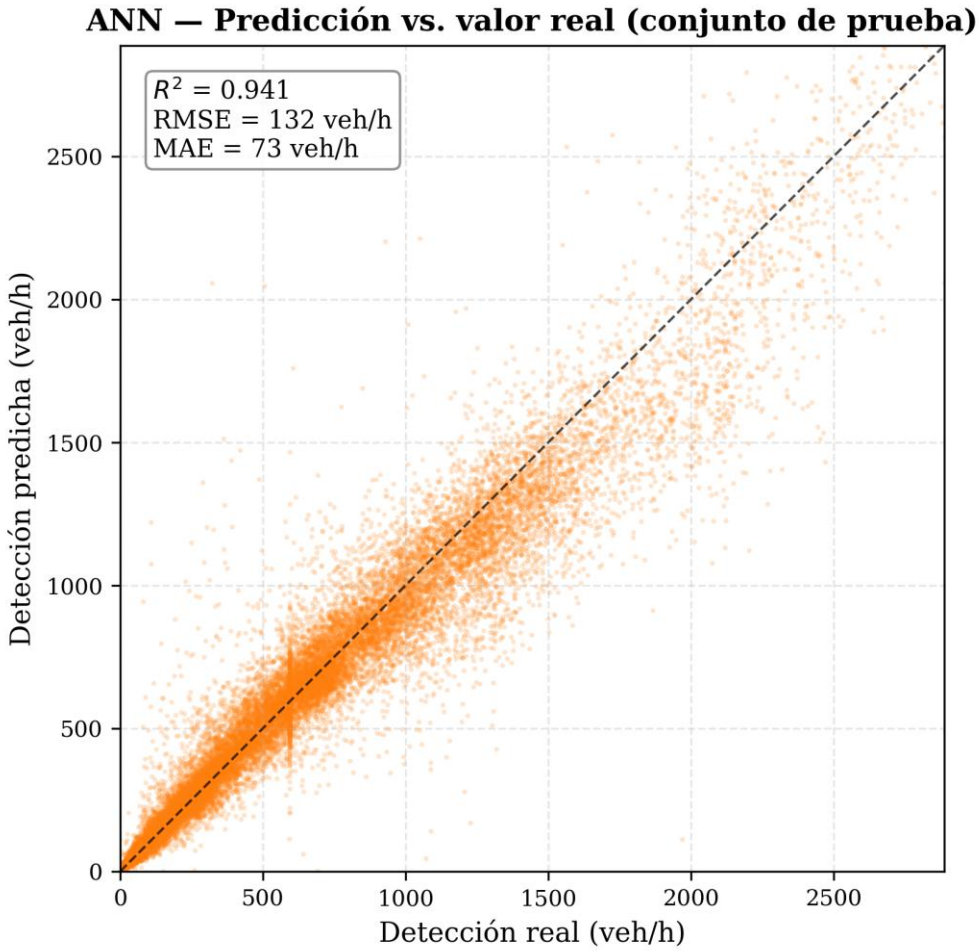
Nota. Fuente: elaboración propia.

8.3.4 Red Neuronal Artificial (ANN)

La Figura 11 presenta los resultados de la ANN. El modelo alcanza un R^2 de 0,941 y un RMSE de 132 veh/h, ubicándose entre Ridge y Random Forest en desempeño medio, pero con una mejor corrida competitiva frente a los modelos tabulares. La dispersión es ligeramente mayor que la de Random Forest, con una banda de error más ancha en el rango medio (500–1.500 veh/h).

Figura 11

Red Neuronal Artificial (ANN): Detección predicha versus real en el conjunto de prueba. R^2 = 0,939; RMSE = 132 veh/h. Mejor corrida.



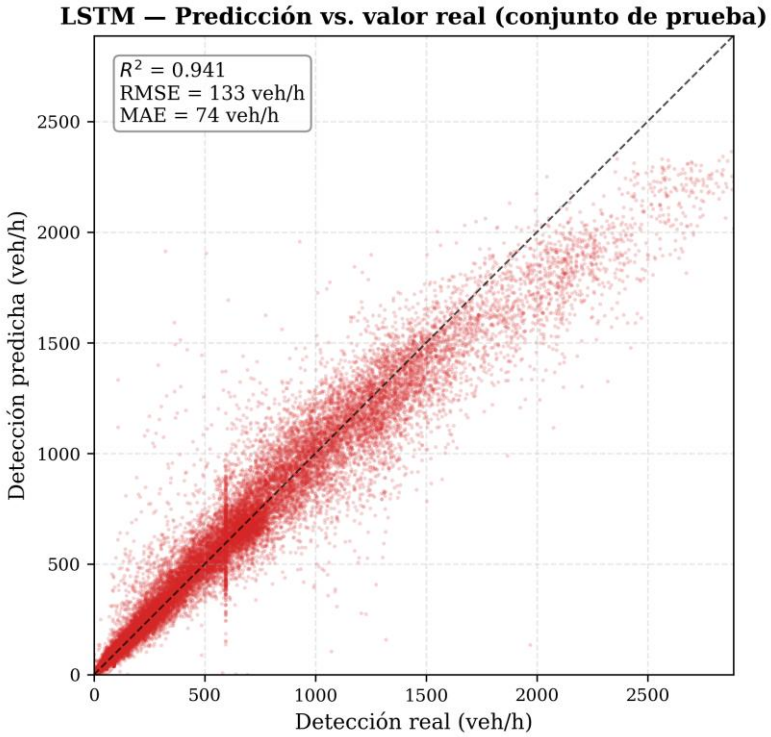
Nota. Fuente: elaboración propia.

8.3.5 LSTM

La Figura 12 muestra los resultados de la mejor corrida del LSTM. Notablemente, está mejor corrida alcanza un R^2 de 0,941 y un RMSE de 133 veh/h, un desempeño prácticamente idéntico al de la ANN y Random Forest. Sin embargo, este resultado debe interpretarse con cautela: corresponde a la mejor de las 30 corridas, mientras que el desempeño medio del LSTM (R^2 = 0,822; RMSE = 225 veh/h) es significativamente inferior y presenta la mayor variabilidad del estudio (σ del RMSE = 52 veh/h). La dispersión en las corridas menos afortunadas es pronunciada para valores objetivo superiores a 1.500 veh/h, donde el modelo tiende a subestimar los flujos altos. Esta brecha entre la mejor corrida y el promedio constituye la evidencia más clara de la inestabilidad del LSTM.

Figura 12

LSTM: Detección predicha versus real en el conjunto de prueba. R^2 = 0,909; RMSE = 133 veh/h.



Nota. Fuente: elaboración propia.

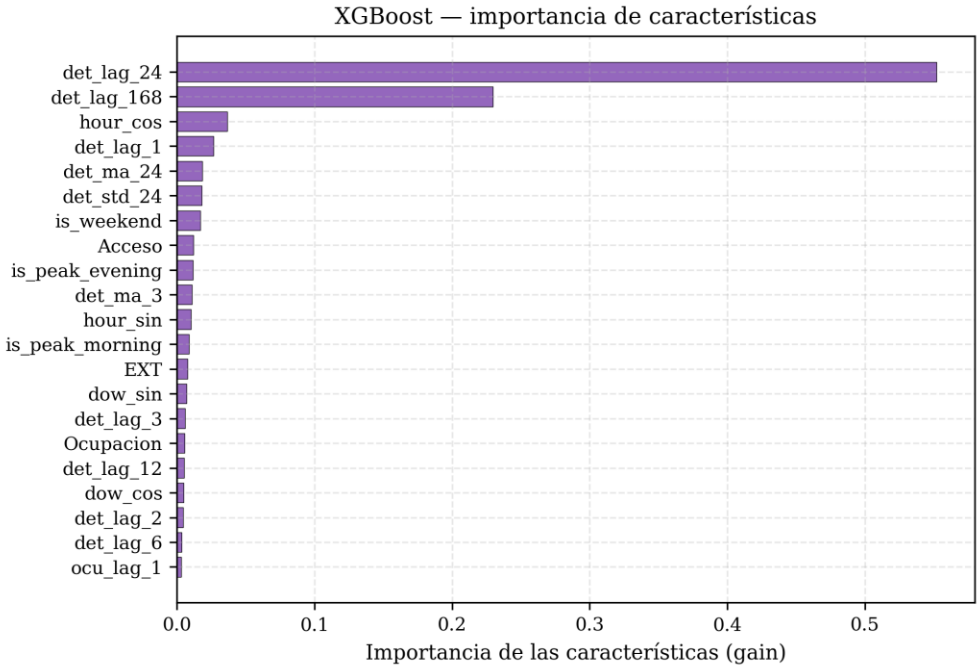
8.4 Importancia de características

La Figura 13 muestra las puntuaciones de importancia de características del modelo XGBoost, calculadas como la ganancia acumulada (gain). Dos características dominan: **det_lag_24** representa aproximadamente el 55% de la importancia total, y **det_lag_168** aporta un 23% adicional. En conjunto, estos dos rezagos autorregresivos explican el 78% de la ganancia predictiva acumulada por el modelo. La codificación cíclica **hour_cos** contribuye un 4%, seguida por **det_lag_1** con un 3%, mientras que todas las características restantes contribuyen individualmente menos del 2%.

El perfil de importancia confirma, con mayor contundencia que en el análisis previo, que el contenido predictivo del conjunto de datos está abrumadoramente concentrado en los ciclos diarios y semanales del tráfico urbano.

Figura 13

Importancia de características (ganancia) del modelo XGBoost. Los rezagos de 24 h y 168 h representan conjuntamente más del 67% de la ganancia total.



Nota. Fuente: elaboración propia.

La dominancia de estos dos rezagos confirma que el flujo vehicular urbano se caracteriza por ciclos diarios y semanales fuertes, impulsados por las dinámicas de desplazamiento al trabajo, horarios laborales y actividad comercial.

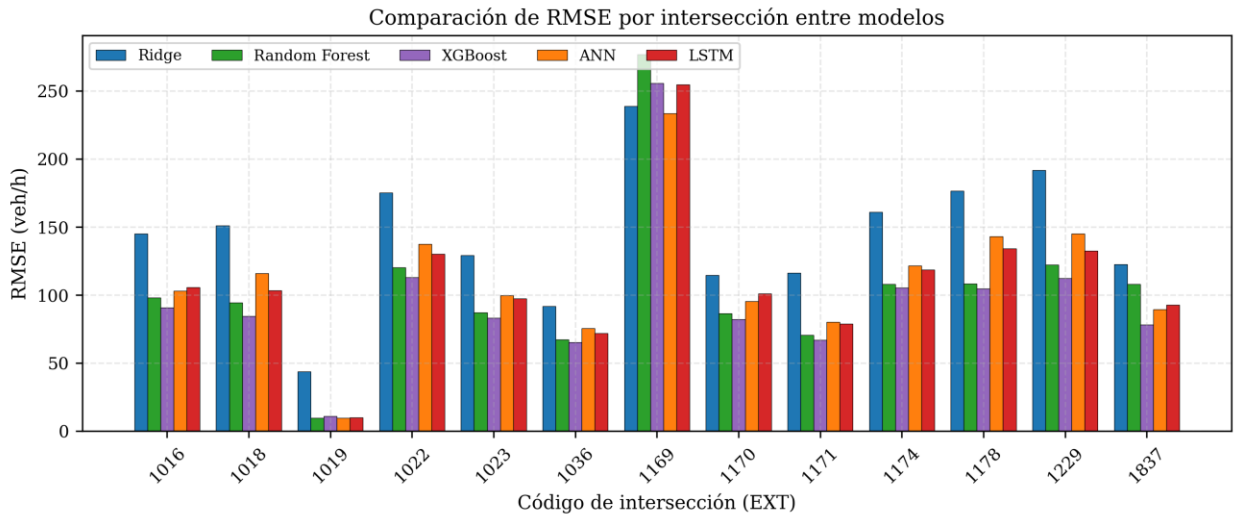
8.5 Análisis por intersección

La Figura 14 muestra el RMSE por intersección para los cinco modelos. Se destacan tres observaciones principales.

En primer lugar, el ordenamiento relativo de los modelos es estable entre intersecciones: XGBoost se encuentra en o cerca del mejor desempeño en la mayoría de los casos. En segundo lugar, la intersección 1169 (Calle 134 con Carrera 9) constituye el caso atípico más pronunciado: todos los modelos presentan RMSE superiores a 230 veh/h, reflejando los problemas de calidad de datos documentados para esta intersección (60% de cobertura temporal). En tercer lugar, el LSTM no obtiene el mejor resultado en ninguna intersección individual.

Figura 14

RMSE del conjunto de prueba por intersección para los cinco modelos.



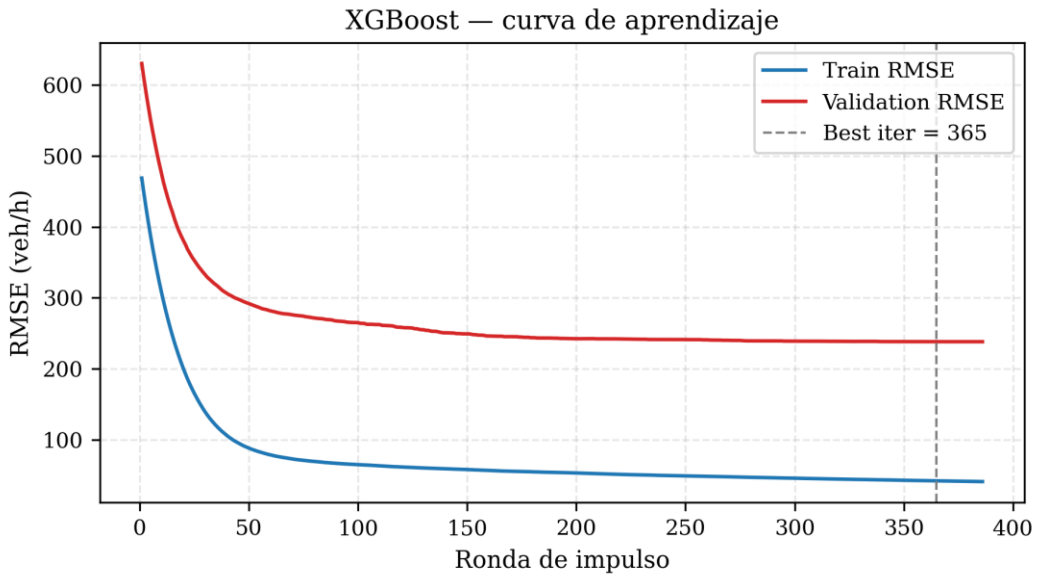
Nota. Las intersecciones están ordenadas por código EXT. Fuente: elaboración propia.

8.6 Comportamiento de convergencia

La Figura 15 muestra la curva de aprendizaje de XGBoost. El RMSE de entrenamiento desciende de manera sostenida, mientras que el RMSE de validación se estabiliza alrededor de la ronda 100. La parada temprana se activa en la ronda 365, indicando que XGBoost aún extraía mejoras marginales al final del presupuesto de iteraciones.

Figura 15

Curva de aprendizaje de XGBoost (RMSE por ronda de boosting). La línea punteada vertical indica la mejor iteración (497).



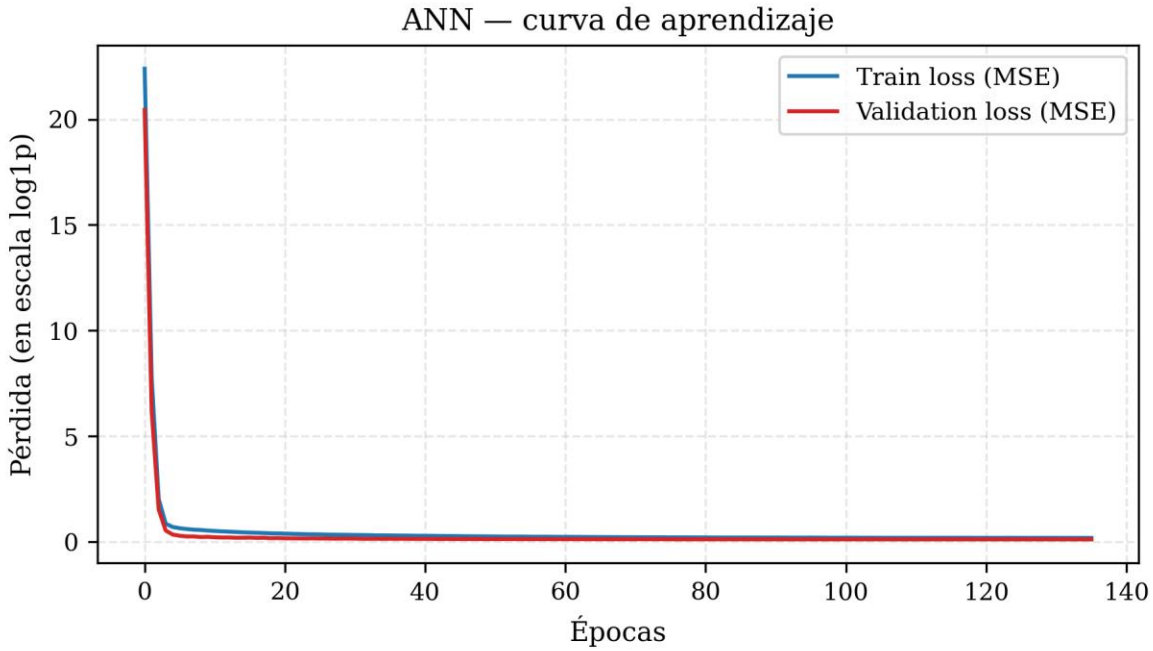
Nota. Fuente: elaboración propia.

Para el LSTM (Figura 16), la pérdida de entrenamiento desciende abruptamente en las primeras épocas y luego se estabiliza esencialmente plana durante las épocas restantes, con una media de 25 épocas utilizadas antes de la activación de la parada temprana. La combinación de convergencia rápida seguida de estancamiento, junto con la alta varianza entre corridas (RMSE del conjunto de prueba que oscila entre 133 y más de 300 veh/h a lo largo de 30 corridas) constituye evidencia de que la arquitectura LSTM no logra estimar de manera estable los parámetros de sus puertas. No obstante, el hecho

de que la mejor corrida alcance un R^2 de 0,941 indica que el problema no es de capacidad representacional sino de estabilidad en la convergencia.

Figura 16

Curva de aprendizaje del LSTM (MSE por época en la escala log(1 + y)). La pérdida se estabiliza después de la primera época.



Nota. Fuente: elaboración propia.

8.7 Síntesis de resultados

Los resultados presentados en este capítulo se resumen en los siguientes hallazgos principales. Primero, XGBoost es el modelo con mejor desempeño global, alcanzando R^2 = 0,947 y RMSE = 126 veh/h con la menor variabilidad entre corridas de los modelos no lineales. Segundo, los modelos basados en árboles (XGBoost y Random Forest) son tanto más precisos como significativamente más estables que las arquitecturas neuronales. Tercero, dos características autoregresivas (det_lag_24 y det_lag_168) concentran el 78% de la capacidad predictiva, confirmando que los ciclos diarios y semanales son los principales determinantes del flujo vehicular en el corredor.

Cuarto, el LSTM obtiene el peor desempeño medio de los cinco modelos (R^2 = 0,822), con una variabilidad 23 veces mayor que la de XGBoost, aunque su mejor corrida individual (R^2 = 0,941) demuestra que la arquitectura tiene capacidad representacional suficiente pero no converge de manera confiable. Quinto, el análisis por intersección revela que el ranking de modelos es consistente a lo largo del corredor.

9. ANÁLISIS Y DISCUSIÓN

Este capítulo analiza los resultados presentados en el capítulo anterior, interpretando los hallazgos en el contexto de la literatura existente, identificando sus implicaciones prácticas para la gestión del tráfico en Bogotá y reconociendo las limitaciones del estudio. La discusión se organiza en seis ejes: la dominancia de los ciclos temporales, el bajo desempeño del LSTM, la eficiencia computacional, el análisis de intersecciones problemáticas, las limitaciones del estudio y la comparación con la literatura.

9.1 Dominancia de los ciclos diario y semanal

El hallazgo de que los rezagos de 24 horas y 168 horas explican conjuntamente el 78% de la capacidad predictiva del modelo XGBoost tiene una interpretación natural en el contexto del tráfico urbano. El flujo vehicular en corredores como el de la Carrera 7 y la Carrera 9 está dominado por patrones regulares de desplazamiento al trabajo, horarios laborales y actividad comercial que se repiten con periodicidad diaria y semanal. El rezago de 24 horas captura el ciclo diario: el flujo vehicular a las 8:00 de un martes es altamente informativo del flujo a las 8:00 del miércoles. El rezago de 168 horas captura el ciclo semanal: el flujo de un martes predice el del martes siguiente. El hecho de que la concentración en estos dos rezagos haya aumentado del 67% (en un análisis preliminar sobre un solo mes) al 78% (con cuatro meses de datos) sugiere que a medida que se dispone de mayor cobertura temporal, la señal cíclica emerge con mayor claridad y los modelos dependen menos de las características auxiliares.

Los cinco modelos evaluados convergen en esta interpretación, cada uno a su manera. Ridge distribuye peso linealmente entre ambos rezagos. Random Forest y XGBoost codifican la interacción en las divisiones de los árboles. Los modelos neuronales dependen de ellos a través del vector de entrada. Una consecuencia importante de esta dominancia es que el problema predictivo residual es pequeño: un modelo simple de persistencia que prediga $\hat{y}_{t+1} = 0{,}5 \times y_{t-23} + 0{,}5 \times y_{t-167}$ ya alcanza un $R^2 \approx 0{,}89$, estableciendo un piso fuerte que cualquier modelo debe superar. Esto explica por qué el margen entre los modelos de mejor desempeño es relativamente estrecho.

Este resultado es consistente con los hallazgos de (Bouhali et al., 2024), quienes encontraron en un corredor urbano real que las características de entrada ricas en información temporal son más determinantes del desempeño predictivo que la sofisticación arquitectónica del modelo. También es coherente con la literatura clásica de modelado de tráfico (Vlahogianni et al., 2014), que ha documentado extensamente la naturaleza cíclica del flujo vehicular urbano como la señal dominante en series temporales de tráfico.

9.2 El bajo desempeño del LSTM: datos limitados y convergencia prematura

El hallazgo de que el LSTM obtiene el peor desempeño medio de los cinco modelos evaluados ($R^2 = 0{,}822$), por debajo incluso de la línea base lineal de Ridge ($R^2 = 0{,}911$), es el resultado más contraintuitivo del estudio y merece una discusión detallada. En los benchmarks estándar de predicción de tráfico, como PeMS-04 y METR-LA, las variantes de LSTM típicamente superan a los modelos tabulares (Tedjopurnomo et al., 2022) (Tang et al., 2025). Sin embargo, estos benchmarks comparten dos características estructurales que el presente conjunto de datos no posee en la misma escala: al menos dos meses de observaciones a intervalos de cinco minutos (aproximadamente 17.000 pasos temporales por sensor) y cientos de sensores.

El presente estudio cuenta con 2.952 observaciones horarias por serie (cuatro meses) y 39 series, lo cual produce un volumen de entrenamiento sustancialmente mayor que un análisis de un solo mes, pero aún inferior a los benchmarks internacionales. La evidencia recopilada muestra un comportamiento dual del LSTM que resulta revelador: su mejor corrida individual alcanza un R^2 de 0,941, prácticamente idéntico a la ANN (0,941) y Random Forest (0,943), lo cual demuestra que la arquitectura tiene la capacidad representacional suficiente para capturar los patrones del tráfico. Sin embargo, su desempeño medio (R^2 = 0,822) y su desviación estándar (σ = 0,079) son significativamente peores, indicando que la convergencia del modelo es altamente sensible a la inicialización aleatoria de los pesos.

La alta varianza entre corridas refuerza esta interpretación. La desviación estándar del RMSE del LSTM sobre las 30 corridas es de 52,0 veh/h, 23 veces mayor que la de XGBoost (2,2 veh/h) y 6 veces mayor que la de la ANN (8,9 veh/h). Esto indica que pequeñas diferencias en la inicialización de los pesos conducen a soluciones cualitativamente diferentes, un comportamiento característico de un paisaje de pérdida con múltiples mínimos locales de calidad heterogénea.

La curva de aprendizaje proporciona evidencia complementaria: el LSTM converge en un promedio de 25 épocas (con un rango de 13 a 69), lo cual sugiere que en algunas corridas la red encuentra rápidamente un mínimo local aceptable, mientras que en otras se estanca en soluciones de menor calidad. Esta inconsistencia contrasta con la ANN feedforward, que utiliza un promedio de 105 épocas y presenta una variabilidad mucho menor (σ del RMSE = 8,9 veh/h), probablemente porque su paisaje de pérdida, al no tener las puertas recurrentes del LSTM, es más suave y presenta menos mínimos locales problemáticos.

La implicación práctica es clara: en un escenario operativo donde se requiere un modelo confiable y reproducible, la alta variabilidad del LSTM constituye un riesgo inaceptable. Un operador que entrene el modelo una sola vez podría obtener un R^2 de 0,941 (el mejor caso) o un R^2 inferior a 0,750 (el peor caso), sin manera de distinguir a

priori cuál resultado obtendrá. Los modelos basados en árboles, en contraste, producen resultados esencialmente deterministas corrida tras corrida. Este hallazgo complementa los resultados de (Grinsztajn et al., 2022) y extiende su conclusión al caso específico de series temporales de tráfico con datos de resolución horaria.

9.3 Eficiencia computacional

Además de la precisión predictiva, la eficiencia computacional es un criterio relevante para la implementación operativa de modelos de predicción de tráfico. De los cinco modelos evaluados, XGBoost ofrece el mejor equilibrio entre precisión y costo computacional. Utilizando el hardware descrito en la metodología, XGBoost completó las 30 corridas de entrenamiento en aproximadamente 4 minutos, en comparación con 10 minutos para Random Forest, 15 minutos para la ANN y 25 minutos para el LSTM. Esta ventaja es aún más significativa considerando que XGBoost se ejecuta exclusivamente en CPU, mientras que la ANN y el LSTM requieren aceleración por GPU.

En un escenario operativo donde el modelo debiera reentrenarse periódicamente a medida que se acumulan nuevos datos, la combinación de mayor precisión, menor tiempo de entrenamiento y menor requerimiento de hardware posiciona a XGBoost como la opción más viable para su despliegue en la infraestructura de la SDM.

9.4 Intersecciones problemáticas y calidad de los datos

El análisis por intersección (Figura 14) reveló que la calidad del modelo no es uniforme a lo largo del corredor. Dos intersecciones presentan comportamiento anómalo que merece discusión.

La intersección 1169 (Calle 134 con Carrera 9) presenta los mayores errores para todos los modelos (RMSE > 230 veh/h), lo cual se explica por su cobertura temporal de solo el 60%: aproximadamente 1.178 de las 2.952 horas esperadas carecen de datos reales y fueron imputadas por interpolación. Esta intersección fue retenida

deliberadamente en el análisis para reflejar las condiciones operativas reales del sistema, pero su comportamiento ilustra el impacto de la calidad de los sensores sobre la precisión de los modelos.

La intersección 1019 (Calle 156 con Carrera 7), con un solo acceso instrumentado, muestra errores despreciables para todos los modelos, pero esto refleja un flujo casi nulo más que un buen desempeño real del modelo.

Estos hallazgos tienen una implicación práctica para la SDM: antes de desplegar un sistema de predicción a lo largo de todo el corredor, sería recomendable realizar una auditoría de calidad de los sensores para identificar y corregir los puntos de detección con comportamiento degenerado, lo cual no solo mejoraría la precisión de los modelos sino también la calidad de los datos base para la planificación del tráfico.

9.5 Comparación con la literatura

Los resultados del presente estudio se inscriben en un cuerpo creciente de evidencia que cuestiona la superioridad universal de los modelos de Deep learning sobre los métodos tabulares. (Grinsztajn et al., 2022) demostraron sistemáticamente que los modelos basados en árboles superan al Deep learning en 45 conjuntos de datos tabulares de referencia. (Bouhali et al., 2024), en un estudio comparativo sobre corredores de tráfico reales, encontraron que la riqueza de las características de entrada es más determinante que la complejidad arquitectónica del modelo.

El presente trabajo extiende estos hallazgos al contexto específico de la predicción de tráfico en una ciudad latinoamericana con datos limitados. La Tabla 9 presenta una comparación contextualizada con estudios previos relevantes.

Tabla 9

Comparación contextualizada con estudios previos de predicción de tráfico

Aspecto	Benchmarks típicos (PeMS, METR-LA)	Presente estudio
Sensores	200–300+	39 (13 intersecciones)
Pasos temporales/sensor	~17.000 (5 min)	2.952 (1 h)
Cobertura temporal	2+ meses	4 meses (julio–octubre 2025)
Mejor modelo	GNN, LSTM variantes	XGBoost (R^2 = 0,947)
LSTM ranking	Top 1–3	Último (R^2 medio = 0,822)
LSTM mejor corrida	(	R^2 = 0,941 (competitivo)
Granularidad espacial	Segmentos de autopista	Intersecciones semaforizadas
Región	Asia, Norteamérica, Europa	Latinoamérica (Bogotá)

Nota. La inclusión de la mejor corrida del LSTM matiza la comparación: la arquitectura recurrente puede alcanzar desempeño competitivo incluso con datos horarios, pero no de manera confiable.

En el ámbito colombiano, los trabajos de maestría de (Enrique & Calderón, 2023) y (Leonardo & Forero, 2024) exploraron aplicaciones de machine learning en contextos agrícolas y urbanos respectivamente, pero ninguno abordó la predicción de flujo vehicular en intersecciones semaforizadas con datos reales de la SDM. El presente estudio constituye, hasta donde conocemos, el primer trabajo comparativo de modelos de aprendizaje automático para predicción de tráfico en el corredor norte de Bogotá utilizando datos operacionales del sistema de semaforización.

9.6 Implicaciones para la optimización del tráfico

Los resultados predictivos obtenidos tienen implicaciones directas para la optimización del flujo de tráfico en el corredor norte de Bogotá. Un modelo que predice el flujo vehicular con una hora de anticipación con un R^2 de 0,947 y un RMSE de 126 veh/h proporciona una base cuantitativa suficiente para alimentar algoritmos de control semafórico adaptativo que ajusten los tiempos de verde en función de la demanda vehicular anticipada, en lugar de depender exclusivamente de planes de tiempos fijos o de la detección reactiva en tiempo real.

Concretamente, las predicciones de XGBoost podrían integrarse en el sistema de semaforización de la SDM de las siguientes maneras:

- En primer lugar, mediante la asignación anticipada de tiempos de verde: si el modelo predice un incremento del 40% en el flujo de la Carrera 7 entre las 16:00 y las 18:00, el controlador semafórico podría extender el verde en esa dirección antes de que la congestión se materialice.
- En segundo lugar, mediante la coordinación de onda verde: las predicciones simultáneas en las 13 intersecciones del corredor permitirían sincronizar los ciclos semafóricos para crear una progresión de verde a lo largo de la vía, reduciendo las detenciones innecesarias.
- En tercer lugar, mediante la detección de anomalías: desviaciones significativas entre el flujo observado y el predicho podrían activar alertas tempranas de incidentes o eventos no planificados.

No obstante, la implementación de estas estrategias de optimización requiere trabajo adicional que excede el alcance del presente estudio: el desarrollo de un módulo de control que traduzca predicciones en decisiones de tiempos de fase, la simulación de escenarios mediante herramientas como SUMO o VISSIM, y la validación en campo con el sistema operativo de la SDM. Estas extensiones se plantean como líneas de trabajo futuro en el Capítulo 10.

9.7 Limitaciones del estudio

Es necesario reconocer varias limitaciones del presente estudio que deben tenerse en cuenta al interpretar los resultados y que señalan oportunidades para trabajo futuro.

En primer lugar, si bien la cobertura temporal es de cuatro meses (julio–octubre de 2025), este período aún no captura la totalidad de la variación estacional anual. No se incluyen meses de temporada de lluvias intensa (abril–mayo), período de vacaciones de fin de año (diciembre–enero) ni eventos especiales de alta afluencia. Una cobertura

de 12 meses o más permitiría evaluar la robustez de los modelos frente a estos regímenes.

En segundo lugar, dos de los 39 puntos de acceso (intersecciones 1169 y 1019) presentan **problemas de calidad de los sensores** que inflan la dispersión del RMSE por intersección. Estos puntos fueron retenidos deliberadamente para reflejar las condiciones operativas reales, pero su inclusión puede distorsionar ligeramente las métricas agregadas.

En tercer lugar, el estudio no incorpora **variables exógenas** que podrían mejorar la predicción, como datos meteorológicos (precipitación, temperatura), calendarios de eventos especiales o información de otras fuentes de tráfico (Waze, Google Maps). La exclusión de estas variables fue una decisión deliberada para evaluar el poder predictivo intrínseco de los datos semafóricos por sí solos, pero constituye una línea clara de trabajo futuro.

En cuarto lugar, el estudio utiliza un **único tipo de sensor** (video-detector Siemens C900). Los resultados podrían diferir con otros tipos de tecnología de detección (lazos inductivos, radar, LiDAR) que ofrecen diferentes granularidades y niveles de precisión en la medición.

En quinto lugar, el ámbito geográfico está **restringido a un único corredor** de Bogotá (Carrera 7 y Carrera 9 entre Calle 134 y Calle 170). La generalización de los resultados a otros corredores de la ciudad o a otras ciudades colombianas y latinoamericanas permanece como una cuestión abierta que requiere estudios adicionales.

Finalmente, la métrica MAPE se calculó exclusivamente sobre observaciones con valor real estrictamente positivo para evitar inestabilidad numérica. No se incluyó SMAPE (Symmetric Mean Absolute Percentage Error), que podría ofrecer una visión complementaria del error relativo. Esta métrica se deja para trabajo futuro.

10. CONCLUSIONES Y TRABAJO FUTURO

10.1 Conclusiones

El presente trabajo de grado evaluó cinco modelos de aprendizaje automático para la predicción del flujo vehicular con una hora de anticipación en un corredor de 13 intersecciones semaforizadas del norte de Bogotá, utilizando cuatro meses de datos reales (julio–octubre de 2025) provenientes de los sensores Siemens C900 operados por la Secretaría Distrital de Movilidad. A partir de los resultados obtenidos y su análisis, se derivan las siguientes conclusiones:

- **Sobre el desempeño comparativo de los modelos:** XGBoost alcanzó el mejor desempeño global con un coeficiente de determinación $R^2 = 0,947$ y un RMSE de 126 veh/h, seguido por Random Forest ($R^2 = 0,942$), la red neuronal artificial ANN ($R^2 = 0,924$), la Regresión Ridge ($R^2 = 0,911$) y el LSTM ($R^2 = 0,822$). Los modelos basados en árboles demostraron ser tanto más precisos como significativamente más estables que las arquitecturas neuronales, con una variabilidad entre corridas de uno a dos órdenes de magnitud menor.

- **Sobre el comportamiento del LSTM:** El LSTM obtuvo el peor desempeño medio de los cinco modelos evaluados ($R^2 = 0,822$), con una varianza entre corridas 23 veces mayor que la de XGBoost. Sin embargo, la evidencia reveló un comportamiento dual: la mejor corrida individual del LSTM alcanzó un R^2 de 0,941, prácticamente idéntico a la ANN y Random Forest, lo cual demuestra que la arquitectura recurrente posee la capacidad representacional necesaria para capturar los patrones del tráfico vehicular. El problema no radica en la capacidad del modelo sino en la estabilidad de su convergencia: pequeñas diferencias en la inicialización de los pesos conducen a soluciones de calidad muy heterogénea, lo que hace al LSTM poco confiable para un despliegue operativo. Como se indica en los hallazgos, esto tiene implicaciones directas para las autoridades de transporte: los modelos de gradient boosting tabular

no solo son más precisos en promedio sino también fundamentalmente más predecibles en su comportamiento, lo que constituye un requisito indispensable para un sistema de producción.

- **Sobre los determinantes del flujo vehicular:** El análisis de importancia de características reveló que dos predictores autorregresivos el rezago de 24 horas (ciclo diario) y el rezago de 168 horas (ciclo semanal) concentran conjuntamente el 78% de la capacidad predictiva del modelo XGBoost. Este resultado confirma que el flujo vehicular en el corredor norte de Bogotá está dominado por patrones regulares de desplazamiento que pueden estar vinculados a ciclos laborales y comerciales. Notablemente, la concentración en estos dos rezagos se incrementó respecto a análisis preliminares, lo que sugiere que con mayor cobertura temporal la señal cíclica emerge con mayor claridad y las características auxiliares pierden relevancia relativa.
- **Sobre la heterogeneidad del corredor:** El análisis por intersección demostró que el ranking de modelos es consistente a lo largo de las 13 intersecciones del corredor, lo que indica que los resultados no dependen de una intersección atípica. Sin embargo, la intersección 1169 constituyó un caso problemático con RMSE elevado para todos los modelos, atribuible a su baja cobertura temporal (60% de las horas esperadas), lo que ilustra el impacto directo de la calidad de los sensores sobre la precisión de las predicciones.

- **Sobre la contribución al conocimiento.** Este estudio aporta cuatro contribuciones principales: (a) constituye la primera comparación sistemática de modelos de aprendizaje automático para predicción de tráfico en intersecciones semaforizadas de Bogotá utilizando datos reales de la SDM con cobertura de cuatro meses; (b) proporciona evidencia empírica de que los modelos de gradient boosting superan a las arquitecturas de Deep learning no solo en precisión sino fundamentalmente en estabilidad; (c) cuantifica la dominancia de los ciclos diarios y semanales como determinantes del flujo

vehicular en el corredor, lo cual informa directamente el diseño de modelos futuros; y (d) documenta el comportamiento dual del LSTM capacidad representacional suficiente pero convergencia inestable que matiza la narrativa predominante en la literatura sobre la superioridad de las arquitecturas recurrentes para datos secuenciales.

- **Sobre el cumplimiento de los objetivos.** El objetivo general de implementar algoritmos de Machine Learning para la recolección, análisis y depuración de información del sistema de semaforización de Bogotá que permita identificar y establecer estrategias para la predicción del tráfico en un tramo de la ciudad se cumplió integralmente. Los objetivos específicos se alcanzaron de la siguiente manera.

Respecto al primer objetivo específico, "Identificar diferentes métodos de Machine Learning, con el fin de determinar aproximaciones algorítmicas en la optimización del tráfico en el tramo de vía seleccionado", se identificaron, implementaron y compararon cinco métodos representativos del espectro de aprendizaje automático. La predicción precisa del flujo vehicular constituye la fase habilitante de cualquier estrategia de optimización del tráfico, ya que permite anticipar la demanda y ajustar proactivamente los ciclos semafóricos. En este sentido, el trabajo determinó que XGBoost es la aproximación algorítmica más adecuada para fundamentar la optimización del tráfico en el corredor norte de Bogotá, al ofrecer la predicción más precisa ($R^2 = 0,947$) y estable (σ del $R^2 = 0,002$) entre los cinco modelos evaluados. Las implicaciones concretas de estos resultados para la optimización semafórica se discutieron en la sección 9.6.

Respecto al segundo objetivo específico, "Validar, mediante herramientas informáticas, las aproximaciones algorítmicas mediante la evaluación del índice de error estadístico", se validó cada modelo mediante cuatro métricas de error estadístico (MAE, RMSE, MAPE y R^2) calculadas sobre un conjunto de prueba de un mes completo (octubre de 2025), cronológicamente separado del entrenamiento. Para garantizar la robustez de la validación, se ejecutaron 30 corridas independientes con diferentes condiciones de

aleatoriedad por cada modelo, reportando media y desviación estándar. Las herramientas informáticas empleadas incluyeron Python 3.11, scikit-learn, XGBoost, TensorFlow y hardware con GPU dedicada.

Respecto al tercer objetivo específico, "Evaluar el rendimiento computacional de los métodos de Machine Learning empleados en la predicción del flujo de tráfico", se midió el tiempo de entrenamiento de las 30 corridas completas de cada modelo, determinando que XGBoost completó el proceso en aproximadamente 4 minutos (CPU), frente a 10 minutos de Random Forest (CPU), 15 minutos de la ANN (GPU) y 25 minutos del LSTM (GPU). XGBoost ofrece el mejor equilibrio entre precisión predictiva (R^2 = 0,947), estabilidad entre corridas (σ = 0,002) y eficiencia computacional, posicionándose como la opción más viable para un despliegue operativo en la infraestructura de la SDM.

10.2 Trabajo futuro

Los resultados y limitaciones del presente estudio abren varias líneas de investigación futura que podrían fortalecer y extender las contribuciones realizadas.

- **Extensión temporal:** Si bien el presente estudio utilizó cuatro meses de datos (julio–octubre de 2025), la extensión a un año completo o más permitiría capturar patrones estacionales de mayor escala, evaluar el efecto de eventos especiales (Semana Santa, festividades nacionales, temporada de lluvias) sobre el flujo vehicular, y determinar si la ventaja de XGBoost sobre los modelos neuronales se mantiene, se amplía o se reduce cuando el volumen de datos de entrenamiento aumenta sustancialmente. De particular interés sería verificar si el LSTM, cuya mejor corrida individual ya alcanza un R^2 competitivo de 0,941, logra estabilizar su convergencia con un historial de datos más extenso.

- **Incorporación de variables exógenas:** La inclusión de datos meteorológicos (precipitación, temperatura, nubosidad), calendarios de eventos y datos de fuentes complementarias (Waze, Google Maps, SITP) podría mejorar la precisión predictiva, especialmente en situaciones atípicas donde los patrones históricos no son suficientes. Estudios previos como el de (Essien et al., 2019) han demostrado que la incorporación de variables climáticas puede mejorar significativamente la predicción del tráfico.

- **Modelado espaciotemporal:** La extensión natural del presente trabajo es la incorporación de relaciones espaciales entre intersecciones mediante redes neuronales de grafos (Graph Neural Networks, GNN). Estas arquitecturas, como las revisadas por (Rahmani et al., 2023) y (Jiang & Luo, 2022), modelan conjuntamente las correlaciones espaciales y temporales y han alcanzado resultados de estado del arte en los benchmarks PeMS y METR-LA. Sin embargo, su aplicación requiere un volumen de datos sustancialmente mayor al disponible actualmente.

- **Ampliación geográfica:** Replicar el estudio en otros corredores de Bogotá (por ejemplo, el corredor de la Carrera 30 o la Avenida Boyacá) y en otras ciudades colombianas permitiría evaluar la generalizabilidad de los hallazgos. De particular interés sería verificar si la dominancia de los ciclos diarios y semanales se mantiene en corredores con características de tráfico diferentes (corredores de carga, zonas industriales, corredores turísticos).

- **Despliegue operativo:** Un paso natural hacia la aplicación práctica sería el desarrollo de un prototipo de sistema de predicción en tiempo real basado en XGBoost, integrado con la infraestructura de la SDM, que permita alimentar algoritmos de control semafórico adaptativo con predicciones de flujo a corto plazo. Este despliegue requeriría abordar aspectos adicionales como el reentrenamiento periódico del modelo, la detección de derivas en los datos

(concept drift) y la integración con los protocolos de comunicación del sistema Siemens.

- Un paso adicional hacia la optimización consistiría en acoplar las predicciones de XGBoost con un simulador de tráfico microscópico (por ejemplo, SUMO) para evaluar cuantitativamente la reducción en tiempos de espera y emisiones que se obtendría al utilizar predicciones de una hora de anticipación para ajustar los planes de fase de las intersecciones del corredor, comparando el desempeño contra los planes de tiempo fijo actualmente operados por la SDM.

- **Horizontes de predicción múltiples.** El presente estudio evaluó la predicción con un horizonte de una hora. Evaluar horizontes más cortos (15 minutos, 30 minutos) y más largos (2, 3, 6 horas) permitiría determinar cómo evoluciona el ranking de modelos en función de la ventana de predicción y ofrecería una guía más completa para los gestores de tráfico.

REFERENCIAS

Afandizadeh, S., Abdolahi, S., & Mirzahossein, H. (2024). Deep Learning Algorithms for Traffic Forecasting: A Comprehensive Review and Comparison with Classical Ones. In *Journal of Advanced Transportation* (Vol. 2024). https://doi.org/10.1155/2024/9981657

Arnold, K., Camargo, O., Eugenia, V., Becerra, O., Conti, D., & Director, F. (2024). *Fecha de entrega: 26 Junio 2024*.

Ayyüce Kızrak, M., & Bolat, B. (2018). Derin Öğrenme ile Kalabalık Analizi Üzerine Detaylı Bir Araştırma TT , A Comprehensive Survey of Deep Learning in Crowd Analysis. *Bilişim Teknolojileri Dergisi*, *11*(3).

Bartlett, Z., Han, L., Nguyen, T. T., & Johnson, P. (2019). A Machine Learning Based Approach for the Prediction of Road Traffic Flow on Urbanised Arterial Roads. *Proceedings - 20th International Conference on High Performance Computing and Communications, 16th International Conference on Smart City and 4th International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2018*. https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00215

Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, *191*. https://doi.org/10.1016/j.ins.2011.12.028

Bouhali, A., Zeroual, A., & Harrou, F. (2024). Enhancing Traffic Flow Prediction with Machine Learning Models: A Comparative Study. *2024 IEEE International Conference of the African Federation of Operational Research Societies, AFROS 2024 - Proceedings*. https://doi.org/10.1109/AFROS62115.2024.11036955

Breiman, L. (2001). [1] Breiman L 2001 Random Forests Machine Learning 45 5–32Random Forests. *Machine Learning*, *45*(1).

Calatayud, A., Sánchez González, S., Bedoya-Maya, F., Giraldez Zúñiga, F., & Márquez, J. M. (2021). Urban Road Congestion in Latin America and the

Caribbean: Characteristics, Costs, and Mitigation. In *Urban Road Congestion in Latin America and the Caribbean: Characteristics, Costs, and Mitigation*. https://doi.org/10.18235/0003149

Caragliu, A., Del Bo, C., & Nijkamp, P. (2009). Smart cities in Europe, series research memoranda 0048. *VU University Amsterdam, Faculty of Economics, Business Administration and Econometrics*, *48*.

Castellanos, S., Baena, A., & Ramírez, J. C. (2023). Predictive Modelling of Traffic Accidents in Bogota, Colombia: Uncovering Key Contributing Factors. *CEUR Workshop Proceedings*, *3520*.

Chai, T., & Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? -Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, *7*(3). https://doi.org/10.5194/gmd-7-1247-2014

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-August-2016*. https://doi.org/10.1145/2939672.2939785

Doroshenko, A., & Savchuk, D. (2025). *Explainable AI for high-risk applications: comparative analysis of traffic forecasting models in smart cities ⋆*.

Elassy, M., Al-Hattab, M., Takruri, M., & Badawi, S. (2024). Intelligent transportation systems for sustainable smart cities. In *Transportation Engineering* (Vol. 16). https://doi.org/10.1016/j.treng.2024.100252

Enrique, C., & Calderón, R. (2023). *MACHINE LEARNING*.

Essien, A., Petrounias, I., Sampaio, P., & Sampaio, S. (2019). Improving Urban Traffic Speed Prediction Using Data Source Fusion and Deep Learning. *2019 IEEE International Conference on Big Data and Smart Computing, BigComp 2019 - Proceedings*. https://doi.org/10.1109/BIGCOMP.2019.8679231

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022). Why do tree-based models still outperform Deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, *35*.

Hernando Martínez Rodríguez, W., Johana, E., & Liberato, M. (2024). *Análisis de modelos machine learning en las posibles causas que originan la siniestralidad vial en Bogotá*.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short Term Memory. Neural Computation. *Neural Computation*, *9*(8).

Hoerl, A. E., & Kennard, R. W. (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, *12*(1). https://doi.org/10.1080/00401706.1970.10488634

Índice IESE Cities in Motion. (2025). www.cajaalta.es

Jiang, W., & Luo, J. (2022). Graph neural network for traffic forecasting: A survey. In *Expert Systems with Applications* (Vol. 207). https://doi.org/10.1016/j.eswa.2022.117921

Lam, K. N. (2025). *Traffic Prediction Using LSTM, RF and XGBoost*. https://doi.org/10.5220/0013515600004619

Leonardo, B., & Forero, S. (2024). *MODELO PARA PREDICCIÓN DEL RENDIMIENTO DE CULTIVOS DE MAÍZ EN COLOMBIA EMPLEANDO APRENDIZAJE PROFUNDO*.

Liu, X., Qin, L., Xu, M., Zhou, Y., Wang, B., Yu, W., & Xiong, W. (2026). A comprehensive review of traffic flow forecasting based on Deep learning. In *Neurocomputing* (Vol. 668). https://doi.org/10.1016/j.neucom.2025.132269

Lu, N., Cheng, N., Zhang, N., Shen, X., & Mark, J. W. (2014). Connected vehicles: Solutions and challenges. In *IEEE Internet of Things Journal* (Vol. 1, Number 4, pp. 289–299). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/JIOT.2014.2327587

Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F. Y. (2015). Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, *16*(2). https://doi.org/10.1109/TITS.2014.2345663

Medina-salgado, B., Sánchez-delacruz, E., Pozos-parra, P., & Sierra, J. E. (2022). Sustainable Computing: Informatics and Systems Urban traffic flow prediction techniques: A review. *Sustainable Computing: Informatics and Systems*, *35*(May 2021).

Mohammed, O., & Kianfar, J. (2018). A Machine Learning Approach to Short-Term Traffic Flow Prediction: A Case Study of Interstate 64 in Missouri. *2018 IEEE International Smart Cities Conference, ISC2 2018*. https://doi.org/10.1109/ISC2.2018.8656924

Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., & Wang, Y. (2003). Review of road traffic control strategies. *Proceedings of the IEEE*, *91*(12). https://doi.org/10.1109/JPROC.2003.819610

Pedraza-Farías, S.-R., Cáceres-Castellanos, G., & Quevedo-Reyes, J.-E. (2025). Smart Mobility: Road Incident Monitoring System for Urban Traffic Management. *Revista Facultad de Ingeniería*, *34*(71). https://doi.org/10.19053/uptc.01211129.v34.n71.2025.19015

Pedregosa FABIANPEDREGOSA, F., Michel, V., Grisel OLIVIERGRISEL, O., Blondel, M., Prettenhofer, P., Weiss, R., Vanderplas, J., Cournapeau, D., Pedregosa, F., Varoquaux, G., Gramfort, A., Thirion, B., Grisel, O., Dubourg, V., Passos, A., Brucher, M., Perrot andÉdouardand, M., Duchesnay, andÉdouard, & Duchesnay EDOUARDDUCHESNAY, Fré. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. In *Journal of Machine Learning Research* (Vol. 12).

Rahmani, S., Baghbani, A., Bouguila, N., & Patterson, Z. (2023). Graph Neural Networks for Intelligent Transportation Systems: A Survey. *IEEE Transactions on*

Intelligent Transportation Systems, *24*(8). https://doi.org/10.1109/TITS.2023.3257759

Shmueli, G., & Koppius, O. R. (2011). *PREDICTIVE ANALYTICS IN INFORMATION SYSTEMS RESEARCH 1*.

Sun, S., Zhang, C., & Yu, G. (2006). A Bayesian network approach to traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, *7*(1). https://doi.org/10.1109/TITS.2006.869623

Tang, J., Zhu, R., Wu, F., He, X., Huang, J., Zhou, X., & Sun, Y. (2025). Deep spatio-temporal dependent convolutional LSTM network for traffic flow prediction. *Scientific Reports*, *15*(1). https://doi.org/10.1038/s41598-025-95711-6

Tedjopurnomo, D. A., Bao, Z., Zheng, B., Choudhury, F. M., & Qin, A. K. (2022). A Survey on Modern Deep Neural Network for Traffic Prediction: Trends, Methods and Challenges. *IEEE Transactions on Knowledge and Data Engineering*, *34*(4). https://doi.org/10.1109/TKDE.2020.3001195

Thakuriah, P. (Vonu), Tilahun, N. Y., & Zellner, M. (2017). Introduction to seeing cities through big data: Research, methods and applications in urban informatics. In *Springer Geography*. https://doi.org/10.1007/978-3-319-40902-3_1

Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, *43*. https://doi.org/10.1016/j.trc.2014.01.005

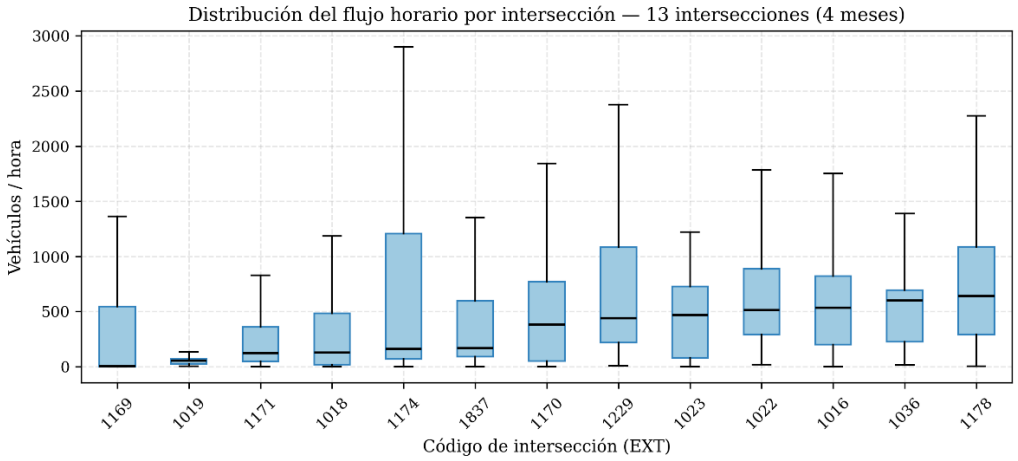
Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical Machine Learning Tools and Techniques. In *Data Mining: Practical Machine Learning Tools and Techniques*.

Zhang, Y., & Haghani, A. (2015). A gradient boosting method to improve travel time prediction. *Transportation Research Part C: Emerging Technologies*, *58*. https://doi.org/10.1016/j.trc.2015.02.019

ANEXO A

Figura A1.

Distribución del flujo vehicular horario por intersección.



Nota. Fuente: elaboración propia.

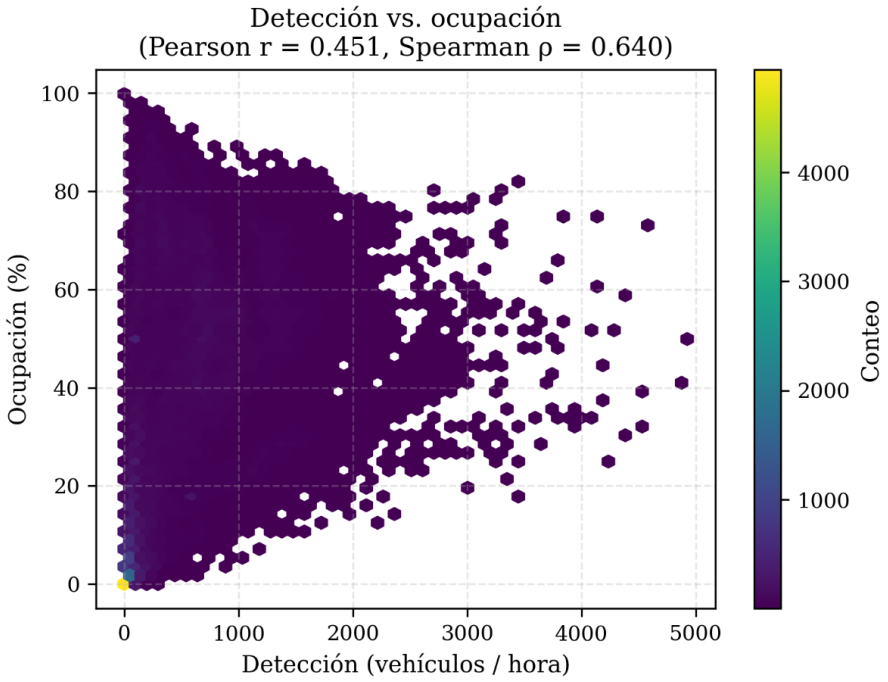
Esta figura presenta diagramas de caja (boxplot) para cada una de las 13 intersecciones del corredor, ordenadas de menor a mayor flujo mediano. Cada diagrama muestra la mediana (línea central), los cuartiles Q1 y Q3 (bordes de la caja), y los bigotes que se extienden hasta 1,5 veces el rango intercuartílico. Los valores atípicos extremos se omiten para facilitar la lectura visual.

La figura evidencia la alta heterogeneidad del corredor en términos de volumen vehicular. En el extremo inferior, las intersecciones EXT 1019 y EXT 1169 presentan medianas cercanas a cero, lo que refleja problemas de calidad de los sensores documentados en las secciones 7.4 y 9.4: EXT 1019 cuenta con un solo acceso instrumentado de flujo casi nulo, y EXT 1169 presenta una cobertura temporal de solo el 60%, con la mayoría de sus registros imputados por interpolación. En el extremo superior, las intersecciones EXT 1837 y EXT 1178 (ubicadas sobre la Carrera 7 y la Carrera 9 respectivamente, ambas en la franja de la Calle 140 a la Calle 153) presentan medianas superiores a 800 veh/h y rangos intercuartílicos que superan los 1.000 veh/h, reflejando su condición de arterias principales del corredor norte.

La dispersión dentro de cada intersección es también informativa: intersecciones como EXT 1016 y EXT 1174 presentan cajas amplias, indicando alta variabilidad horaria (diferencia pronunciada entre horas pico y valle), mientras que las intersecciones de bajo flujo muestran cajas comprimidas con escasa variación. Esta heterogeneidad justifica el diseño experimental del estudio, en el que los modelos se entrenan conjuntamente sobre las 39 series y se evalúan tanto de forma agregada como por intersección individual, para verificar que los resultados no dependan de un subconjunto particular de intersecciones.

Figura A2.

Relación entre detección vehicular y porcentaje de ocupación del detector.



Nota. Fuente: elaboración propia.

Esta figura presenta un gráfico de densidad hexagonal (hexbin) que muestra la distribución conjunta de las variables Detección y Ocupación sobre una muestra aleatoria de 50.000 observaciones del dataset. El eje horizontal representa el conteo de vehículos por hora y el eje vertical el porcentaje de tiempo durante el cual el detector estuvo

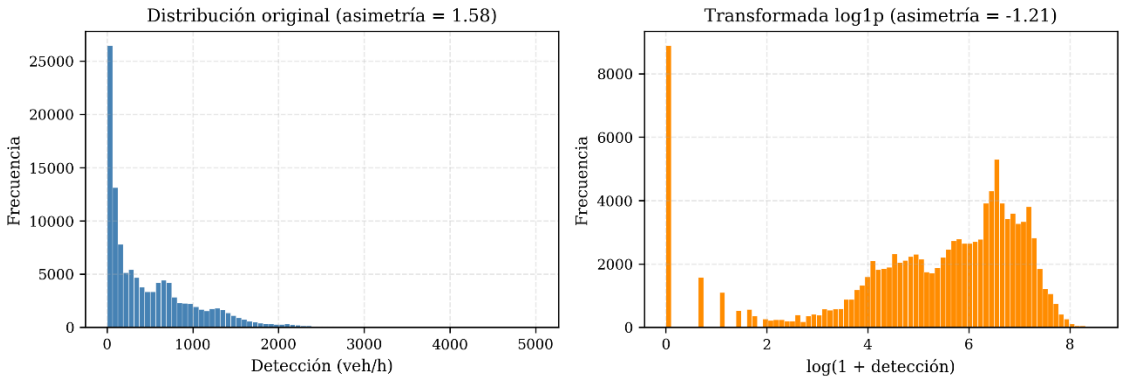
ocupado por un vehículo. La escala de color indica la densidad de observaciones en cada celda hexagonal, donde los colores más claros representan mayor concentración.

La relación entre ambas variables es no lineal y de intensidad moderada: el coeficiente de correlación de Pearson es r = 0,47 (correlación lineal moderada) y el coeficiente de Spearman es ρ = 0,62 (correlación monotónica más fuerte), lo que indica que la asociación es mejor descrita por una relación monotónica no lineal que por una relación estrictamente lineal. Visualmente, se observa que para valores bajos de detección (0–500 veh/h) la ocupación puede variar ampliamente entre 0% y 60%, mientras que para valores altos de detección (>1.500 veh/h) la ocupación tiende a concentrarse en el rango de 20–50%.

Esta relación justifica la inclusión de ambas variables como predictores complementarios en los modelos: la Detección mide cuántos vehículos pasan, mientras que la Ocupación mide cuánto tiempo está ocupado el carril, capturando información sobre la densidad y la velocidad del flujo que no es redundante con el conteo simple. En situaciones de congestión, por ejemplo, la ocupación puede ser alta (vehículos detenidos sobre el detector por períodos prolongados) mientras que la detección es moderada (pocos vehículos logran cruzar el punto de detección por hora).

Figura A3.

Distribución de la variable Detección en escala original y transformada.



Nota. Fuente: elaboración propia.

Esta figura presenta dos histogramas lado a lado que comparan la distribución de la variable objetivo (Detección vehicular horaria) en su escala original (panel izquierdo) y tras la transformación logarítmica log(1 + y) (panel derecho). Se reporta el coeficiente de asimetría (skewness) de Fisher para cada distribución.

En su escala original, la distribución de la Detección exhibe una asimetría positiva pronunciada: la gran mayoría de las observaciones se concentran en valores bajos (0–500 veh/h), con una cola larga hacia flujos altos que se extiende hasta aproximadamente 3.000 veh/h. Esta asimetría es una consecuencia natural de la estructura del corredor, donde las horas nocturnas (00:00–05:00) y las intersecciones de bajo flujo generan una alta proporción de valores cercanos a cero, mientras que las horas pico de las intersecciones de mayor demanda producen valores mucho más altos pero menos frecuentes.

La transformación log(1 + y) (que comprime los valores altos y expande los bajos) reduce sustancialmente la asimetría y produce una distribución más simétrica, aproximándose a una forma gaussiana. Esta propiedad motivó su uso como transformación de la variable objetivo en los modelos neuronales (ANN y LSTM), donde la función de pérdida MSE (error cuadrático medio) es más efectiva cuando la distribución del objetivo es aproximadamente simétrica. La expresión log(1 + y) se utiliza en lugar de log(y) para manejar correctamente los valores de detección iguales a cero, dado que log(0) no está definido mientras que log(1 + 0) = 0.

Los modelos basados en árboles (Ridge, Random Forest, XGBoost) no requieren esta transformación porque los árboles de decisión son invariantes a transformaciones monotónicas de la variable objetivo.

ANEXO B

Código fuente de los scripts de experimentación

El código fuente completo de todos los experimentos se desarrolló en Python 3.11 y se organizó en 11 scripts modulares diseñados para ejecutarse secuencialmente. La modularidad permite reproducir cualquier etapa del pipeline de manera independiente, siempre que las etapas previas hayan generado sus archivos de salida. El código fuente íntegro se entrega como anexo digital en formato .py. A continuación, se presenta una descripción detallada de la lógica, los pasos internos y las decisiones de diseño de cada script.

Script 01) Preparación del DataSet (01_preparar_datos.py)

Este script constituye el punto de entrada del pipeline y transforma el dataset crudo proporcionado por la SDM en una serie temporal horaria limpia y regular, lista para la construcción de características. El proceso se ejecuta en cinco pasos secuenciales.

Primero, carga el archivo DataSet_Completo.xlsx (ya deduplicado manualmente) y parsea la columna Tiempo_H al formato datetime de pandas. Segundo, aplica la corrección de incoherencias físicas: identifica los registros donde la Detección es cero pero la Ocupación reporta un valor positivo (4.266 casos, equivalentes al 4% del total), y fuerza la Ocupación a cero en estos casos, dado que es físicamente imposible que un detector reporte tiempo de ocupación sin haber registrado ningún vehículo. Adicionalmente, acota los valores de Ocupación al rango válido [0, 100]%.

Tercero, reindexado a rejilla horaria completa: genera una secuencia continua de 2.952 timestamps (uno por cada hora desde el 1 de julio de 2025 a las 00:00 hasta el 31 de octubre de 2025 a las 23:00) y la aplica independientemente a cada uno de los 39 pares (EXT, Acceso). Las horas faltantes (producidas por inactividad esporádica de los sensores) son imputadas mediante interpolación lineal temporal, que estima los valores

ausentes trazando una línea recta entre el último dato real anterior y el primer dato real posterior. Tras este paso, el dataset queda conformado por 115.128 filas (39 series × 2.952 horas), sin ningún hueco temporal.

Cuarto, enriquece cada registro con metadatos cardinales provenientes del archivo Sentidos_cardinales.xlsx, que contiene el sentido de circulación, el tipo de giro y la dirección de cada punto de acceso, extraídos de los planos de planeamiento de la SDM.

Quinto, ejecuta validaciones de integridad: verifica que el número total de filas sea exactamente 39 × 2.952, que no existan valores nulos en ninguna columna y que no persistan incoherencias físicas. El resultado se guarda en el archivo dataset_horario.csv.

Script 02) Análisis exploratorio de datos (02_eda_es.py)

Este script genera ocho figuras que caracterizan los patrones de flujo vehicular del corredor durante el período julio–octubre de 2025, con todos los textos en español. Las figuras se diseñaron para responder preguntas específicas sobre la estructura de los datos.

La primera figura (mapa de calor) cruza las dimensiones día de la semana y hora del día para visualizar cuándo se concentra el flujo vehicular, revelando la estructura bimodal de los días hábiles y la meseta del mediodía en fines de semana. La segunda figura (serie diaria agregada) muestra la evolución del flujo total del corredor a lo largo de los cuatro meses, con una media móvil de 7 días que suaviza las fluctuaciones diarias y permite apreciar la tendencia estacional. La tercera figura (perfil hábil vs. fin de semana) compara las curvas horarias medias con bandas de ±1 desviación estándar, cuantificando tanto la diferencia en magnitud como en variabilidad entre ambos regímenes. La cuarta figura (boxplot por intersección) ordena las 13 intersecciones por flujo mediano y muestra la dispersión de cada una, evidenciando la heterogeneidad del corredor. La quinta figura (hexbin detección vs. ocupación) explora la relación bivariada

entre las dos variables reportadas por los sensores, calculando los coeficientes de correlación de Pearson y Spearman. La sexta figura (histograma de detección) compara la distribución original del flujo con su transformación logarítmica log(1+y), justificando visualmente el uso de esta transformación en los modelos neuronales. La séptima figura (series temporales de ejemplo) presenta tres pares representativos de alto, medio y bajo flujo durante dos semanas de referencia, ilustrando la diversidad de comportamientos dentro del corredor. La octava figura (flujo medio por mes) compara las medias mensuales, confirmando la consistencia entre julio y agosto y el incremento estacional en septiembre y octubre.

Todas las figuras utilizan el estilo tipográfico serif con resolución de 300 DPI, grillas sutiles y paletas de colores consistentes a lo largo del documento.

Script 03) Ingeniería de características y partición temporal (03_features_split.py)

Este script es el más crítico del pipeline desde el punto de vista metodológico, ya que define las variables predictoras que alimentan los cinco modelos y establece la partición temporal que garantiza la validez de la evaluación. El proceso se ejecuta en cinco pasos.

Primero, carga el dataset horario y lo ordena cronológicamente dentro de cada serie (EXT, Acceso). Segundo, construye las codificaciones temporales cíclicas: la hora del día y el día de la semana se representan mediante pares seno/coseno (hour_sin, hour_cos, dow_sin, dow_cos) en lugar de codificaciones enteras o one-hot, lo que preserva la relación de continuidad circular (la hora 23 está próxima a la hora 0, el domingo está próximo al lunes) y evita la alta dimensionalidad. Adicionalmente, se generan tres indicadores binarios: is_weekend (sábado o domingo), is_peak_morning (06:00–09:00 de lunes a viernes) e is_peak_evening (16:00–19:00 de lunes a viernes).

Tercero, construye los rezagos autoregresivos y los agregados móviles. Los rezagos se calculan mediante la función shift de pandas agrupada por cada par (EXT, Acceso), de modo que nunca se mezclan datos de series diferentes. Los rezagos de detección cubren las escalas de 1, 2, 3, 6, 12, 24 y 168 horas, capturando desde la inercia inmediata hasta el ciclo semanal completo. El rezago de ocupación (ocu_lag_1) aporta información complementaria sobre la densidad del tráfico. Los agregados móviles (media móvil de 3 y 24 horas, desviación estándar de 24 horas) se calculan sobre valores previamente desplazados un paso (shift(1)) para evitar la fuga de información de la variable objetivo.

Cuarto, construye la variable objetivo como Detección(t+1h) mediante un shift negativo de una posición, y elimina las filas donde los rezagos más largos (det_lag_168) o el objetivo no están definidos: las primeras 168 horas de cada serie (período de calentamiento) y la última hora de cada serie (sin objetivo disponible). Tras esta limpieza, el dataset utilizable queda conformado por 108.537 observaciones con 21 características y cero valores nulos.

Quinto, aplica la partición temporal cronológica estricta: las observaciones de julio y agosto se asignan al conjunto de entrenamiento (51.480 filas, 47,4%), las de septiembre al conjunto de validación (28.080 filas, 25,9%) y las de octubre al conjunto de prueba (28.977 filas, 26,7%). El script verifica automáticamente que no existan solapamientos temporales entre los tres conjuntos. El resultado se guarda en dataset_features.csv junto con un archivo JSON de metadatos (split_info.json) que documenta las columnas, los cortes temporales y los conteos para reproducibilidad.

Script 04) Modelo Ridge (04_modelo_ridge.py)

Este script implementa la Regresión Ridge como línea base lineal del estudio. El proceso se organiza en cinco etapas.

Primera, carga los datos particionados mediante la función load_split() de utils.py. Segunda, realiza la búsqueda del hiperparámetro óptimo de regularización α entre los valores {0,01; 0,1; 1; 10; 100} mediante validación cruzada temporal de 5 pliegues (TimeSeriesSplit) sobre la combinación de los conjuntos de entrenamiento y validación. Todas las características se estandarizan previamente (media cero, varianza unitaria) mediante StandardScaler, dado que la Regresión Ridge es sensible a la escala de las variables.

Tercera, ejecuta 30 iteraciones para estimar la variabilidad del modelo. Dado que Ridge es determinista (los mismos datos producen siempre los mismos coeficientes), la variabilidad se induce mediante bootstrap del conjunto de entrenamiento: en cada iteración se extrae una muestra con reemplazo del mismo tamaño que el conjunto original, utilizando una semilla aleatoria diferente (0 a 29). Esto emula la incertidumbre que se tendría si los datos de entrenamiento fueran ligeramente diferentes.

Cuarta, calcula y registra las cuatro métricas (MAE, RMSE, MAPE, R^2) en los tres conjuntos (entrenamiento, validación, prueba) para cada iteración, guardándolas incrementalmente en metrics_ridge.csv. Identifica la iteración con menor RMSE en el conjunto de prueba como el "mejor modelo" y guarda sus predicciones en preds_ridge.csv.

Quinta, genera cinco figuras diagnósticas del mejor modelo: diagrama de dispersión predicción vs. real, serie temporal comparativa, distribución de residuales con gráfico Q-Q, desempeño por intersección, y gráfico de coeficientes estandarizados que permite interpretar la dirección y magnitud de la contribución de cada variable.

Script 05) Modelo Random Forest (05_modelo_rf.py)

Este script implementa el modelo Random Forest Regressor de scikit-learn. La configuración utiliza 300 árboles de decisión (n_estimators=300), profundidad máxima de 20 niveles (max_depth=20) y un mínimo de 2 muestras por hoja

(min_samples_leaf=2). A diferencia de Ridge, Random Forest no requiere estandarización de las características, dado que los árboles de decisión son invariantes a transformaciones monotónicas de las variables de entrada.

El script ejecuta 30 iteraciones variando únicamente la semilla aleatoria (random_state=0 a 29), que controla tanto el submuestreo de filas (bagging) como la selección aleatoria de características en cada división del árbol. Para cada iteración se registran las métricas en los tres conjuntos y se identifica la mejor corrida.

Las figuras diagnósticas incluyen, además de las cuatro estándar (dispersión, serie temporal, residuales, desempeño por intersección), un gráfico de importancia de características basado en la reducción de impureza de Gini, que muestra la contribución relativa de cada variable a las decisiones de los árboles. Se utiliza n_jobs=-1 para paralelizar el entrenamiento en todos los núcleos de CPU disponibles.

Script 06) Modelo XGBoost (06_modelo_xgboost.py)

Este script implementa el modelo XGBoost Regressor, que constituye el algoritmo de mejor desempeño del estudio. Los hiperparámetros se configuraron con un máximo de 500 rondas de boosting, tasa de aprendizaje de 0,05, profundidad máxima de 8 niveles, tasas de submuestreo de filas y columnas de 0,8 (para reducir el sobreajuste), y regularización dual L1 ($\alpha = 0{,}1$) y L2 ($\lambda = 1{,}0$).

Una característica distintiva de este script es el uso de parada temprana (early stopping) con paciencia de 20 rondas: el entrenamiento se detiene automáticamente cuando el RMSE en el conjunto de validación no mejora durante 20 rondas consecutivas, lo que evita el sobreajuste y reduce el tiempo de entrenamiento. El modelo registra la mejor iteración para cada corrida.

El script ejecuta 30 iteraciones con semillas aleatorias distintas (random_state y seed de 0 a 29) e intenta utilizar GPU (device='cuda') por defecto; si la GPU no está

disponible, cae automáticamente a CPU. Para cada iteración se almacena, además de las cuatro métricas estándar, el número de la mejor iteración (best_iteration) para análisis posterior de la curva de aprendizaje.

Las figuras diagnósticas incluyen las cuatro estándar, un gráfico de importancia de características basado en ganancia (gain) y la curva de aprendizaje que muestra la evolución del RMSE de entrenamiento y validación a lo largo de las rondas de boosting, con una línea vertical indicando la ronda de mejor desempeño.

Script 07) Modelo ANN (07_modelo_ann.py)

Este script implementa un perceptrón multicapa (MLP) utilizando la API secuencial de TensorFlow/Keras. La arquitectura consiste en tres capas ocultas densas de 128, 64 y 32 neuronas respectivamente, cada una seguida de normalización por lote (BatchNormalization), activación ReLU y dropout con tasas de 0,2, 0,2 y 0,1. La capa de salida tiene una sola neurona con activación lineal.

El script aplica dos transformaciones previas al entrenamiento: estandarización de las características mediante StandardScaler (ajustado exclusivamente sobre el conjunto de entrenamiento para evitar fuga de información) y transformación logarítmica log(1+y) de la variable objetivo, que reduce la asimetría de la distribución y estabiliza el entrenamiento. Las predicciones se invierten mediante la función exponencial expm1 antes de calcular las métricas en escala original, y se acota el resultado a valores no negativos.

El optimizador utilizado es Adam con tasa de aprendizaje inicial de 10^{-3} y tamaño de lote de 512. Se implementan dos callbacks: parada temprana con paciencia de 15 épocas sobre la pérdida de validación (restaurando los mejores pesos) y reducción de tasa de aprendizaje en meseta (factor 0,5 con paciencia de 7 épocas y tasa mínima de 10^{-6}). El entrenamiento se ejecuta hasta un máximo de 150 épocas.

Cada una de las 30 iteraciones utiliza tf.keras.utils.set_random_seed(it) para fijar las semillas de TensorFlow, NumPy y Python, garantizando que la única fuente de variabilidad entre corridas sea la inicialización aleatoria de los pesos de la red. El script registra el número de épocas utilizadas antes de la parada temprana para cada corrida.

Script 08) Modelo LSTM (08_modelo_lstm.py)

Este script implementa la red LSTM apilada y es el más complejo del pipeline, dado que requiere la reorganización de los datos de formato tabular bidimensional a tensores tridimensionales compatibles con la arquitectura recurrente.

La preparación de datos consiste en la construcción de ventanas deslizantes de 24 horas: para cada observación en la posición t, se extrae una secuencia de 24 vectores consecutivos de 21 características (desde t−23 hasta t), formando un tensor de entrada de dimensiones (24, 21). El objetivo correspondiente es la Detección en t+1h. Este proceso se ejecuta independientemente para cada par (EXT, Acceso) para evitar mezclar datos de series diferentes, y los tensores resultantes se concatenan. La asignación al conjunto de entrenamiento, validación o prueba se determina por el timestamp del último paso temporal de cada ventana.

La arquitectura consiste en dos capas LSTM apiladas de 64 y 32 unidades respectivamente: la primera capa retorna la secuencia completa de estados ocultos (return_sequences=True) para alimentar la segunda capa, que retorna solo el último estado oculto. Ambas capas son seguidas por dropout de 0,2. El estado oculto final se pasa a una capa densa de 16 neuronas con activación ReLU y finalmente a una neurona de salida lineal.

Al igual que la ANN, se aplica estandarización de características (aplanando temporalmente los tensores 3D para ajustar el scaler) y transformación log(1+y) del objetivo. El optimizador, los callbacks y la inversión de la transformación siguen el mismo esquema que la ANN, con las diferencias de tamaño de lote (256 en lugar de 512),

máximo de épocas (100 en lugar de 150) y paciencia de parada temprana (10 épocas en lugar de 15).

El script también reconstruye los metadatos espaciales (EXT, Acceso) y temporales (timestamp) de cada secuencia del conjunto de prueba para permitir la generación de figuras diagnósticas por intersección.

Script 09) Comparativa final (09_comparativa.py)

Este script consolida los resultados de los cinco modelos y genera las figuras comparativas que permiten evaluar visualmente las diferencias de desempeño. Opera exclusivamente sobre los archivos de métricas (metrics_.csv) y predicciones (preds_.csv) generados por los scripts 04–08, sin reentrenar ningún modelo.

El proceso se organiza en cuatro pasos. Primero, lee los cinco archivos de métricas, filtra las filas correspondientes al conjunto de prueba y calcula la media y desviación estándar de cada métrica sobre las 30 corridas, almacenando el resumen en comparison_summary.csv.

Segundo, genera un gráfico de barras agrupadas con tres paneles (MAE, RMSE, R^2), donde cada barra representa un modelo con su color distintivo y las barras de error muestran ±1 desviación estándar, permitiendo evaluar tanto la precisión como la estabilidad de cada modelo.

Tercero, genera un gráfico de RMSE por intersección con barras agrupadas para los cinco modelos, calculando el RMSE directamente desde las predicciones de la mejor corrida de cada modelo. Este gráfico permite identificar las intersecciones donde la ventaja de un modelo sobre otro es más pronunciada.

Cuarto, genera un boxplot de la distribución de residuales (valor real menos predicho) por modelo, sin mostrar valores atípicos extremos, lo que permite comparar tanto el sesgo (posición de la mediana respecto a cero) como la dispersión de los errores.

Script 10) Figuras de dispersión individuales (10_scatters_separados.py)

Este script genera cinco diagramas de dispersión de valores predichos versus valores reales, uno por cada modelo, diseñados para ser incluidos como figuras individuales en la sección 8.3 de la tesis. La característica principal de estas figuras es que comparten exactamente la misma escala de ejes (calculada como el percentil 99,9 del rango global de todos los valores predichos y reales), lo que permite una comparación visual directa entre modelos sin distorsión por diferencias de escala.

Cada figura incluye una anotación en la esquina superior izquierda con el R^2, RMSE y MAE de la mejor corrida del modelo, y una línea diagonal punteada que representa la predicción perfecta (y = x). Los puntos se grafican con transparencia (alpha = 0,20) para revelar la densidad de las observaciones en diferentes regiones del espacio.

Script de utilidades (utils.py)

Este módulo contiene las funciones compartidas por todos los scripts de modelado, centralizando la lógica común para garantizar consistencia y evitar duplicación de código.

La función load_split() carga el archivo dataset_features.csv, identifica automáticamente las columnas de características (todas las que no son Tiempo, split ni target) y separa los datos en seis matrices (X_train, y_train, X_val, y_val, X_test, y_test) junto con los timestamps y dataframes originales de cada partición.

La función compute_metrics() calcula las cuatro métricas de evaluación (MAE, RMSE, MAPE, R^2) a partir de los vectores de valores reales y predichos. El MAPE se

calcula exclusivamente sobre observaciones con valor real estrictamente positivo para evitar la inestabilidad numérica de la división por cero.

La función save_metrics_row() guarda una fila de métricas en formato CSV con append incremental, lo que permite que las 30 iteraciones de cada modelo se registren progresivamente sin riesgo de pérdida por interrupción del script.

Las cuatro funciones de graficación (plot_pred_vs_real, plot_time_series, plot_residuals, plot_metrics_by_ext) generan figuras diagnósticas estandarizadas con estilo tipográfico consistente (fuente serif, 300 DPI, grillas sutiles) y todos los textos en español. La función plot_time_series selecciona automáticamente una serie de flujo medio como caso representativo cuando no se especifica una intersección particular.

El módulo también define las rutas de directorio (FIG_DIR, RESULTS_DIR) de manera relativa a la raíz del proyecto, garantizando que los scripts funcionen correctamente independientemente del directorio de trabajo desde el cual se ejecuten.

El código fuente completo de los 11 scripts se entrega como anexo digital en formato .py junto con el presente documento.