

LSTM (Long short-term memory) neural network para la predicción a corto plazo del flujo de pasajeros en la red metro de Medellín.

Roy Sandoval Viera. 1, Laura Sofía Jiménez Moreno. 2*, Sofía Mejía Rivas. 3**

1. Estudiante de Ingeniería en Ciencia de datos, Universidad Pontificia Bolivariana, Campus Laureles, Medellín, Colombia

() E-mail: roy.sandoval@upb.edu.co*

2. Estudiante de Ingeniería en Ciencia de datos, Universidad Pontificia Bolivariana, Campus Laureles, Medellín, Colombia

() E-mail: laura.jimenezm@upb.edu.co*

3. Estudiante de Ingeniería en Ciencia de datos, Universidad Pontificia Bolivariana, Campus Laureles, Medellín, Colombia

() E-mail: sofia.mejia@upb.edu.co*

RESUMEN: Este artículo explora el uso de redes neuronales de Memoria a Corto y Largo Plazo (LSTM) para la predicción del flujo de pasajeros en el metro de Medellín. Dada la importancia de predecir la afluencia de pasajeros en sistemas de transporte para optimizar recursos y mejorar la experiencia del usuario, este estudio aplica LSTMs para abordar la variabilidad en la demanda de transporte influida por eventos locales, cambios estacionales y tendencias socioeconómicas.

El enfoque del estudio se centra en adaptar arquitecturas de redes neuronales eficaces previamente validadas, como las de Izudheena et al. (2020), a las especificidades del metro de Medellín. Utilizando datos históricos, se implementan modelos LSTM para generar predicciones precisas que ayuden en la planificación y gestión operativa del sistema de transporte.

El artículo describe las capacidades técnicas de las LSTM, especialmente su habilidad para aprender dependencias a largo plazo y manejar series temporales complejas. Esto se contrasta con las limitaciones de las Redes Neuronales Recurrentes (RNN) estándar, que luchan con dependencias a largo plazo debido a problemas como la desaparición o explosión del gradiente. Las LSTMs superan estas limitaciones con una estructura de "puertas" que regulan el flujo de información, lo que permite retener o descartar información de manera selectiva y mantener el estado a través de largos intervalos de tiempo.

La metodología incluye la utilización de datos abiertos del sistema Metro del Valle de Aburrá, abarcando un periodo desde enero de 2020 hasta enero de 2024. Se detallan los procesos de preprocesamiento de datos, como la identificación de días festivos y el manejo de valores faltantes, y la configuración de la red LSTM, que incluye capas de atención para resaltar la relevancia de ciertos pasos temporales en la secuencia de entrada, dicha red neuronal ofrece una herramienta robusta y eficiente para los administradores del sistema de metro. Esto facilita decisiones informadas para la gestión de recursos y la planificación de horarios, y contribuye a una mejor experiencia del usuario al ajustar el servicio a la demanda real.

Palabras clave: LSTM , red neuronal, Metro de Medellín.

Abstract

This paper investigates the use of Long Short-Term Memory (LSTM) neural networks to predict passenger flow in Medellín's metro system. Given the importance of forecasting passenger demand in transportation systems to optimize resources and enhance user experience, the study leverages LSTMs to address fluctuations influenced by local events, seasonal changes, and socioeconomic trends. The model adapts neural network architectures validated in prior research, such as those by Izudheena et al. (2020), tailoring them to the specific conditions of Medellín's metro.

Our methodology integrates open data from Metro del Valle de Aburrá, covering January 2020 to January 2024, with data preprocessing to account for holiday effects, manage missing values, and configure an LSTM model with attention layers, highlighting the importance of specific temporal steps within sequences. Despite the success of the LSTM model in learning long-term dependencies and handling complex temporal patterns, the results indicate that a single LSTM model is insufficient for accurately capturing the entirety of passenger flow dynamics.

To improve predictive accuracy, alternative models, including ARIMA and SARIMA, are explored, demonstrating complementary strengths in capturing seasonal and trend components not fully addressed by LSTM alone. The inclusion of additional data, such as weather conditions and technical disruptions in the metro, is found to be crucial, as these factors display significant relationships with passenger flow and improve the overall model performance. Our findings emphasize the need for a multi-model approach and a comprehensive data set to effectively support decision-making in transportation planning and operations

Introducción

En la actualidad, la capacidad de predecir con precisión la afluencia de pasajeros en sistemas de transporte masivo es fundamental para optimizar recursos, mejorar la experiencia del usuario y asegurar la sostenibilidad operativa. El sistema de metro de Medellín, una pieza clave en la infraestructura de transporte de la ciudad, enfrenta desafíos constantes debido a la variabilidad en la demanda de sus servicios, impulsada por factores como eventos locales, cambios estacionales y tendencias socioeconómicas. En este contexto, las redes neuronales de Memoria a Corto y Largo Plazo (LSTM, por sus siglas en inglés) emergen como una herramienta prometedora para la modelación y predicción de series temporales complejas, como lo es la afluencia de pasajeros.

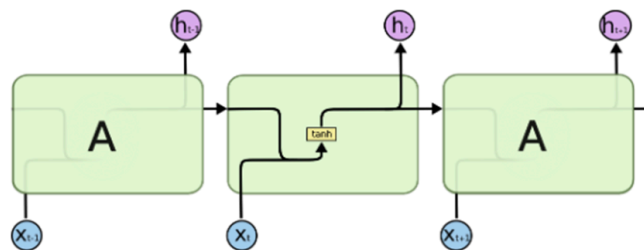
Este artículo se enfoca en explorar el potencial de las redes LSTM para mejorar la predicción y el forecasting de la afluencia de pasajeros en el sistema de metro de Medellín. Aprovechando la capacidad de las LSTMs para aprender dependencias a largo plazo, este estudio busca no solo entender patrones pasados, sino también prever futuras variaciones en la demanda de transporte.

Nuestra investigación toma como base el enfoque metodológico y los hallazgos de arquitecturas de redes neuronales eficaces tales como las entrenadas por Izudheena et al (2020) , adaptándolas a las particularidades del sistema metro de Medellín. Mediante el análisis de datos históricos y la implementación de modelos LSTM, se busca ofrecer predicciones precisas y útiles que puedan ser aplicadas en la planificación y gestión operativa del metro, contribuyendo así a un servicio más eficiente y adaptado a las necesidades cambiantes de la población.

A lo largo de este artículo, detallaremos la estructura de los modelos LSTM utilizados, los datos empleados, y las técnicas de entrenamiento y validación. Finalmente, discutiremos las implicaciones de nuestros hallazgos para la gestión del sistema metro de Medellín.

Aspectos teóricos

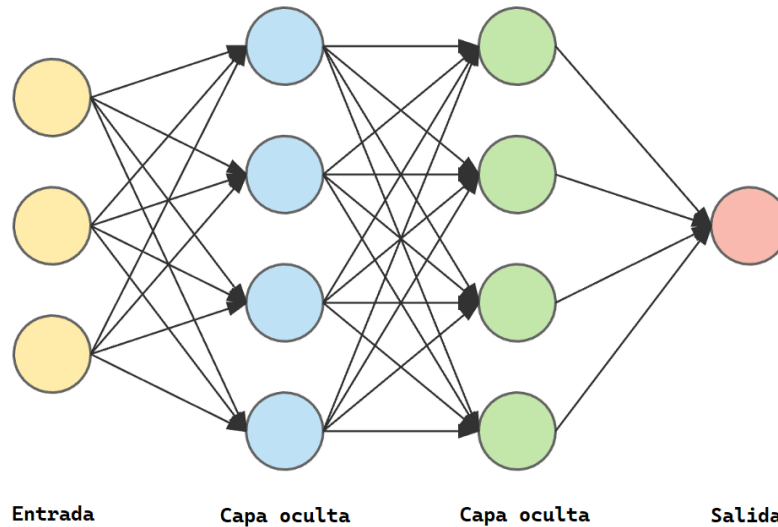
Las RNN (Redes Neuronales Recurrentes) son una clase especial de redes neuronales diseñadas para manejar secuencias de datos. A diferencia de las redes neuronales tradicionales, las RNN tienen la capacidad de mantener un estado o memoria que influye en las entradas futuras basadas en las entradas anteriores. Esto las hace adecuadas para tareas donde es esencial comprender el contexto temporal, como podría ser analizar una película y predecir eventos futuros basándose en los anteriores.



[Figura 1: Módulo de repetición en una RNN convencional. Disponible en: [Understanding LSTM Networks -- colah's blog](#)]

Sin embargo, las RNN estándar a menudo luchan con dependencias a largo plazo debido a problemas técnicos como la desaparición o la explosión del gradiente. Esto limita su capacidad para aprender conexiones en secuencias de datos largas, donde la información relevante puede estar separada por intervalos considerables. Para superar estas limitaciones, se introdujo una variante especializada de RNN conocida como Long Short-Term Memory (LSTM).

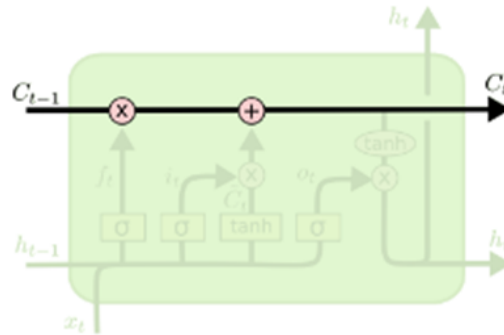
En el contexto de sistemas de transporte como la red de metro de Medellín, Antioquia, anticipar con precisión el flujo de pasajeros es esencial para optimizar la operación y elevar la experiencia del usuario. Al emplear redes neuronales LSTM (Long Short-Term Memory) para predecir estas fluctuaciones, resulta fundamental comprender primero el concepto de redes recurrentes. Siendo estas es un tipo de red neuronal artificial que utiliza datos secuenciales o datos de series de tiempo. Cabe mencionar que están incorporados en aplicaciones populares como Siri, búsqueda por voz y Google Translate. Al igual que las redes neuronales feedforward y convolucionales (CNN), las redes neuronales recurrentes utilizan datos de entrenamiento para aprender. Se distinguen por su "memoria", ya que toman información de entradas anteriores para utilizarse en los datos de entrada y en los resultados.



[Figura 2 : Red Neuronal Convolucional. Disponible en: [Qué son las redes neuronales y sus aplicaciones | OpenWebinars](#)]

Las LSTM son capaces de aprender dependencias a largo plazo de manera más efectiva. Su diseño incluye una estructura de "puertas" que regulan el flujo de información, permitiendo que la red retenga o descarte información de manera selectiva. Esto les permite mantener el estado a través de largos intervalos de tiempo sin el riesgo de perder información importante debido a problemas técnicos.

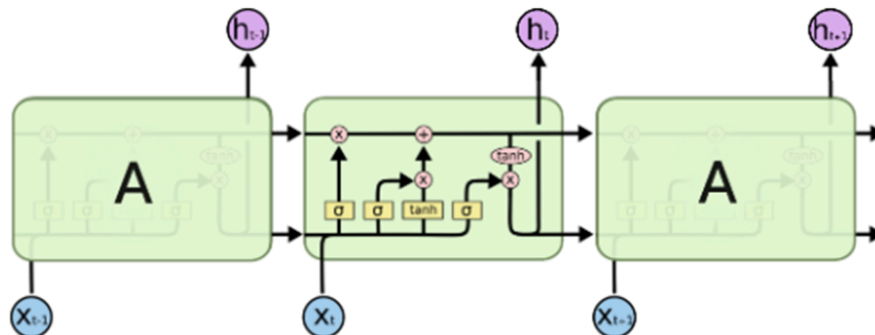
El fundamento de las LSTM es el estado de la celda, una línea que corre a través de la red, transportando información a lo largo de la secuencia completa sin alteraciones significativas, a menos que las puertas lo permitan. Esto se complementa con puertas de olvido, entrada y salida, que deciden qué información se mantiene, se actualiza o se descarta.



[Figura 3: Estado de la celda. Disponible en: [Understanding LSTM Networks -- colah's blog](https://colah.github.io/posts/Understanding-LSTMs/)]

Así, la celda de memoria está compuesta por varios componentes que interactúan para regular el flujo de información:

1. **Puerta de Olvido (Forget Gate):** La primera etapa en una LSTM es decidir qué información se debe descartar del estado de la celda. Esto se realiza mediante una capa sigmoideal que revisa tanto el estado oculto anterior y devuelve un número entre 0 y 1 para cada número en el estado de la celda. Un 1 significa "conservar completamente este dato", mientras que un 0 significa "descartar completamente este dato".
2. **Puerta de Entrada (Input Gate):** Esta puerta decide qué nueva información se añadirá al estado de la celda. Funciona en dos partes: una capa sigmoideal decide qué valores se actualizarán, y una capa con activación tanh crea un vector de nuevos valores candidatos, que podrían ser añadidos al estado.
3. **Actualización del Estado de la Celda:** Una vez que las puertas han decidido qué conservar y qué añadir, el estado de la celda se actualiza a un nuevo estado. El estado anterior se multiplica por la salida de la puerta de olvido, descartando los datos que se han decidido olvidar. A continuación, se suma el producto de la salida de la puerta de entrada y el vector de candidatos generados por la capa tanh.
4. **Puerta de Salida (Output Gate):** La última parte de la celda LSTM decide qué se va a enviar como salida. La puerta de salida mira el estado de la celda y decide qué partes del estado de la celda pasar a través de una función tanh para obtener la próxima salida oculta que, a su vez, será utilizada para la entrada en el siguiente paso de tiempo, junto con la próxima entrada real de la secuencia.



[Figura 4: Módulo de repetición en LSTM con tres capas interactuantes. Disponible en: [Understanding LSTM Networks -- colah's blog](https://colah.github.io/posts/Understanding-LSTMs/)]

Cómo consiguiente, utilizaremos el mecanismo de atención, el cual es introducido principalmente para resolver los problemas asociados con la compresión de información en un vector de longitud fija en arquitecturas de tipo encoder-decoder. En las RNN y LSTM tradicionales, toda la información de una secuencia debe ser codificada en un solo vector de estado, lo cual puede llevar a la pérdida de información, especialmente en secuencias largas. El mecanismo de atención permite al modelo "atender" diferentes partes de la secuencia de entrada de manera dinámica, mejorando la capacidad del modelo para manejar secuencias largas y complejas sin perder detalles importantes.

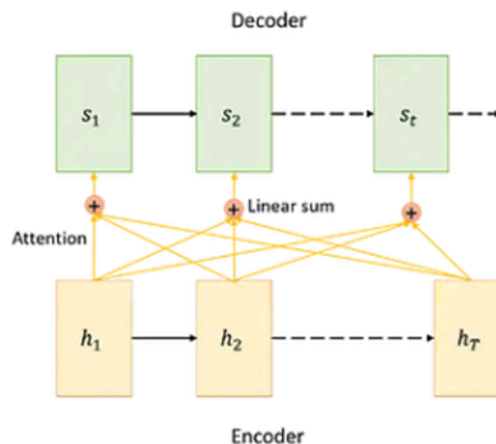
El funcionamiento del mecanismo de atención se puede describir en los siguientes pasos clave:

1. Codificación: Se procesa la entrada mediante una red RNN o LSTM para obtener una serie de estados ocultos que representan la secuencia desde diferentes puntos en el tiempo.

2. Cálculo de pesos de atención: Para cada paso en el decodificador, el modelo calcula un conjunto de pesos de atención que determinan cuánta importancia se debe dar a cada estado oculto del codificador. Esto se hace utilizando una red neuronal que evalúa el estado actual del decodificador y cada uno de los estados ocultos del codificador.

3. Creación del vector de contexto: Utilizando los pesos de atención, se crea un vector de contexto como una combinación lineal de los estados ocultos del codificador, ponderada por estos pesos. Este vector de contexto contiene la información relevante seleccionada de toda la secuencia para realizar predicciones en el paso actual.

4. Decodificación / Traducción: Finalmente, el vector de contexto se utiliza junto con el estado anterior del decodificador para generar el siguiente estado y la salida correspondiente.



[Figura 5 : Arquitectura del modelo de atención basado en encoder-decoder. Disponible en: [A simple overview of RNN, LSTM and Attention Mechanism | by Manu | The Startup | Medium](#)]

Para nuestro caso de estudio, el mecanismo de atención nos permitirá:

- **Captura de eventos relevantes:** Al permitir que el modelo atienda a eventos específicos en la secuencia de datos (como días festivos, eventos especiales o cambios en el clima) que tienen un impacto significativo en los patrones de afluencia, el modelo puede ajustar sus predicciones de manera más efectiva.

- **Manejo de secuencias temporales de entrada:** Dado que las series temporales de datos de afluencia pueden ser largas y complejas, el mecanismo de atención ayuda al modelo a enfocarse en las partes más informativas de la secuencia, mejorando la precisión de las predicciones.
- **Adaptabilidad:** La atención permite al modelo ser más adaptable a diferentes patrones y cambios en los datos sin necesidad de reajustar completamente el modelo, lo cual es crucial para manejar variaciones estacionales y de otro tipo en los datos de afluencia de pasajeros.

En resumen, el mecanismo de atención potencia las capacidades de las LSTM al permitirles procesar y utilizar la información de manera más selectiva y efectiva.

Metodología

Los datos históricos de afluencia de pasajeros del metro de la ciudad, que abarcan desde enero de 2020 hasta enero de 2024, proporcionados por el sistema Metro del Valle de Aburrá como datos abiertos. Este conjunto de datos es invaluable para el análisis y la planificación del transporte público en la ciudad. A continuación, se describen las columnas y la estructura de los datos:

- **Columna "Dia":** Esta columna registra la fecha específica de cada día, permitiendo identificar patrones diarios, semanales, mensuales y anuales en la afluencia de pasajeros.
- **Columnas de Horas de Operación:** Estas columnas registran la afluencia de pasajeros en intervalos horarios desde las 4AM hasta las 11PM. Cada columna representa la cantidad de pasajeros que utilizaron el metro en esa hora específica. Este desglose horario permite un análisis detallado de los picos y valles en la demanda del servicio de metro a lo largo del día.
- **Columna "Total General":** Esta columna proporciona el total de pasajeros que utilizaron el metro durante todo el día, sumando los datos de las columnas horarias. Es útil para evaluar la demanda diaria total y realizar comparaciones entre diferentes días.
- **Columna "Línea de Servicio":** Esta columna identifica la línea específica del metro a la que corresponden los datos de afluencia. Dado que el sistema metro tiene múltiples líneas de metro, esta columna es crucial para analizar la distribución de pasajeros entre las diferentes rutas del sistema.

El desarrollo y entrenamiento del modelo se realizaron utilizando Google Colab, que proporciona un entorno de Jupyter notebook accesible y colaborativo. Se utilizó una GPU T

40 proporcionada por Colab para el entrenamiento del modelo, lo que permitió una reducción significativa en el tiempo de entrenamiento comparado con el uso de una CPU.

Para realizar el preprocesamiento de los datos, se importó la función `'get_colombia_holidays_by_year'` del módulo `'holidays_co'`. Esto se hizo con el fin de identificar y marcar los días festivos de Colombia en el conjunto de datos, lo cual resultó ser un paso crucial para garantizar que los análisis posteriores considerarán las variaciones en la afluencia de pasajeros debido a estos días especiales.

Una vez importadas todas las librerías necesarias, que incluyen pandas, numpy y otras herramientas de procesamiento y análisis de datos, se procedió a cargar los conjuntos de datos que contenían los registros históricos de la afluencia de pasajeros del metro. Inicialmente, estas bases de datos presentaban columnas con nombres genéricos como 'Unnamed: ', las cuales fueron renombradas para reflejar los horarios específicos correspondientes, lo que facilitó la interpretación y manipulación de los datos. Por ejemplo, las columnas fueron etiquetadas para indicar las horas desde las 4AM hasta las 11PM.

Además de los renombramientos, se agregó una nueva columna al conjunto de datos para indicar si un día era festivo, basándose en la información obtenida de `'get_colombia_holidays_by_year'`. Esta columna es esencial para realizar análisis diferenciados entre días laborables y festivos, lo que permite una mejor comprensión de las tendencias de uso del metro en diferentes tipos de días.

Con el objetivo de asegurar la integridad de los datos, se reemplazaron los valores 'None' por nulos utilizando la funcionalidad proporcionada por la librería numpy. Esto permitió la utilización de técnicas estándar de manejo de datos faltantes. Posteriormente, estos valores faltantes fueron imputados con la media de cada columna correspondiente, asegurando así que los análisis posteriores no se vieran afectados por la ausencia de datos.

Como siguiente paso, se llevó a cabo un análisis exhaustivo para identificar los valores atípicos (outliers) en cada una de las columnas de horarios. Estos valores pueden distorsionar los resultados del análisis, por lo que fueron imputados utilizando la media de cada columna. Esta técnica de imputación permitió suavizar las anomalías y mantener la consistencia de los datos, asegurando que el modelo de predicción de la afluencia de pasajeros del metro fuera robusto y preciso.

Previo a la creación del modelo, se definió una capa de atención llamada YearAttention. Esta capa se emplea dentro de un modelo de red neuronal para otorgar más relevancia a ciertos años dentro de una secuencia de datos. Los pesos de atención se calculan para cada paso de tiempo basándose en los años proporcionados. En este caso, se asignó más relevancia a los años 2019 y menos relevancia a los años 2020. Estos pesos de atención se utilizan para calcular un contexto ponderado, que se devuelve junto con los pesos de atención calculados.

Posteriormente, se prepararon los datos para el entrenamiento y la evaluación. Esto incluyó la división del conjunto de datos en conjuntos de entrenamiento y prueba, la selección de características y objetivo, la conversión de variables categóricas a su representación

adecuada, y el manejo de valores NaN, rellenandolos con ceros o valores apropiados. Además, se escalan las características y el objetivo utilizando MinMaxScaler de scikit-learn.

Finalmente, se definió un conjunto de datos personalizado llamado TimeSeriesDataset para manejar los datos de series temporales, y se crearon los cargadores de datos correspondientes para los conjuntos de entrenamiento y prueba.

Creación LSTM

Para nuestro modelo con mejores resultados se establece la arquitectura del modelo LSTM. En primer lugar, cuenta con **1 capa LSTM** configuradas para trabajar en serie, donde cada capa tiene un tamaño de estado oculto de 128. Además, se ha especificado un total de **cuatro capas de atención**, que permiten resaltar la relevancia de ciertos pasos de tiempo en la secuencia de entrada. Después de la capa de atención, encontramos **tres capas totalmente conectadas**, cada una seguida de una función de activación ReLU. La primera capa totalmente conectada tiene una salida de 256 unidades, seguida de una capa con 128 unidades y, finalmente, una capa de salida con el tamaño de salida deseado. El modelo se entrena con un método de aprendizaje supervisado, donde la función *forward* define cómo fluye la información a través de la red. En este método, se inicializan los estados ocultos y de celda de las capas LSTM, y luego la entrada se pasa a través de estas capas. Posteriormente, la salida se dirige a las capas de atención y, finalmente, a las capas totalmente conectadas para generar la salida final.

Parámetros del modelo:

Para la configuración de la red neuronal LSTM utilizada en la predicción del flujo de pasajeros en el Metro de Medellín, se definieron los siguientes hiperparámetros clave, que influyen directamente en la estructura y el rendimiento del modelo:

Capas LSTM: El modelo incluye cuatro capas LSTM (Long Short-Term Memory), cada una con 128 unidades. Esta configuración de múltiples capas permite al modelo procesar y aprender de dependencias temporales complejas, lo que es crucial para la predicción precisa en series temporales de tráfico de pasajeros.

Tasa de aprendizaje: Se estableció una tasa de aprendizaje de 0.0001, utilizada con el optimizador Adam. Esta tasa de aprendizaje relativamente baja ayuda a evitar que el modelo sobrepase los mínimos durante el entrenamiento, permitiendo un ajuste más fino y gradual de los pesos.

Tamaño de lote (Batch size): El modelo procesa los datos en lotes de 16 ejemplos. Este tamaño de lote es un compromiso entre la eficiencia computacional y la capacidad de generalización, permitiendo un gradiente más estable y un entrenamiento más rápido sin agotar la memoria de la GPU.

Funciones de activación: Las capas intermedias utilizan la función de activación ReLU (Rectified Linear Unit) para introducir no linealidades en el modelo, lo que permite captar

relaciones complejas en los datos de entrada sin caer en los problemas de los gradientes desvanecientes que a menudo afectan a las redes más profundas.

Dimensiones de entrada y salida: La entrada del modelo consiste en un vector que combina múltiples características horarias y categóricas, mientras que la salida es un único valor escalar que predice el número de pasajeros para un intervalo dado.

Métricas de Rendimiento:

Para evaluar el desempeño del modelo LSTM en la tarea de predicción del flujo de pasajeros, se utilizan las siguientes métricas:

Error Absoluto Medio (MAE): Esta métrica cuantifica el promedio de las diferencias absolutas entre los valores predichos y los valores reales. Para esta métrica se obtuvo un MAE de 5365 significa que, en promedio, las predicciones del modelo se desvían en 5365 pasajeros del valor real.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

[Figura 6 : MAE. Disponible en: *Short-Term Passenger Count Prediction for Metro Stations using LSTM*, pág. 4]

Error Porcentual Absoluto Medio (MAPE): El MAPE mide el error como un porcentaje del valor real, proporcionando una comprensión intuitiva de la precisión del modelo en términos relativos. Se obtuvo MAPE de 1.51% indicando que los errores del modelo representan, en promedio, el 1.51% del valor real de los datos. En términos prácticos, un MAPE del 1.51% es bastante bajo, lo que indica que el modelo es bastante preciso y puede ser fiable para la planificación y gestión basada en sus predicciones.

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i} 100\%$$

[Figura 7: MAPE. Disponible en: *Short-Term Passenger Count Prediction for Metro Stations using LSTM*, pág. 4]

Resultados y discusión

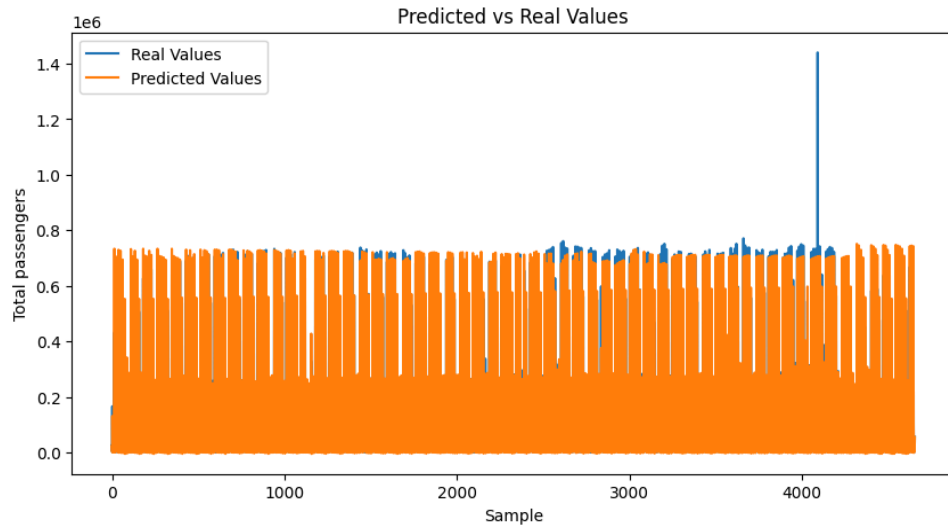
Para la realización del presente artículo, se realizaron 4 intentos de modelo LSTM

Métrica	Modelo 1	Modelo 2	Modelo 3	Modelo 4
MAE	5365	6881	5468	15993
MAPE	1.5160	2.354	2.0360	22.5094

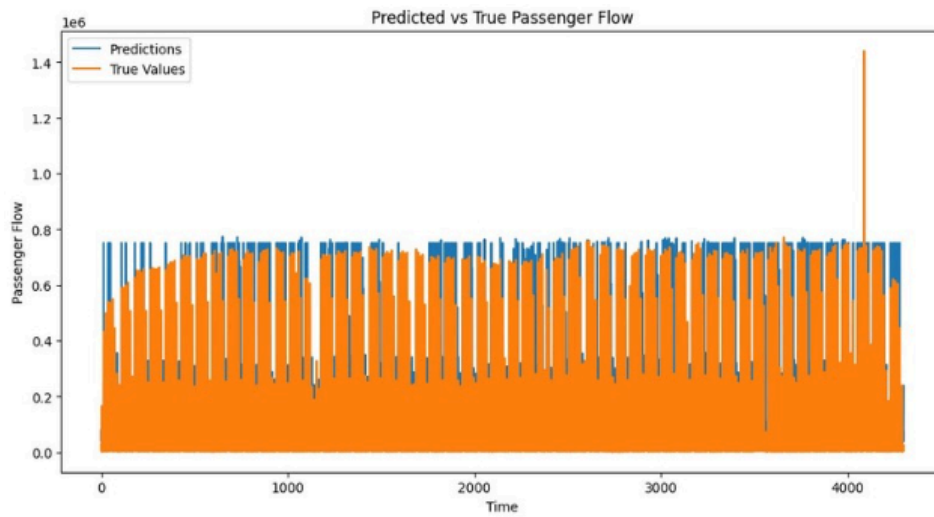
[Tabla 1: Resultados de métricas para cada modelo]

Considerando que los últimos cuatro modelos no incorporaron la atención en su entrenamiento, el modelo 1 sobresalió al utilizar Attention junto con los siguientes parámetros: `hidden_size = 128`, `num_layers = 4`, `output_size = 1`. Se empleó el optimizador Adam y se utilizó el "Mean Squared Error Loss" como función de pérdida. El modelo se entrenó durante 100 épocas.

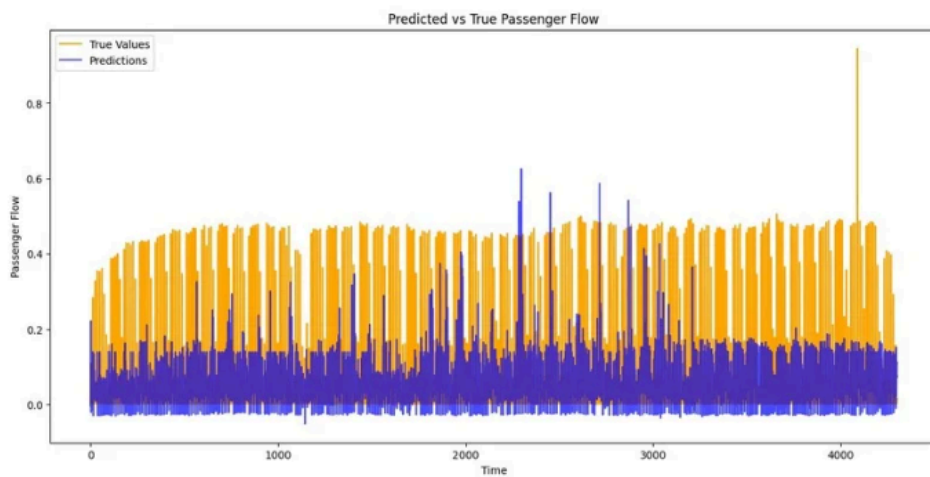
La gráfica resultante muestra la comparación entre los valores reales y los predichos, destacando la capacidad del modelo para capturar patrones y tendencias en los datos de flujo de pasajeros en la red del metro de Medellín.



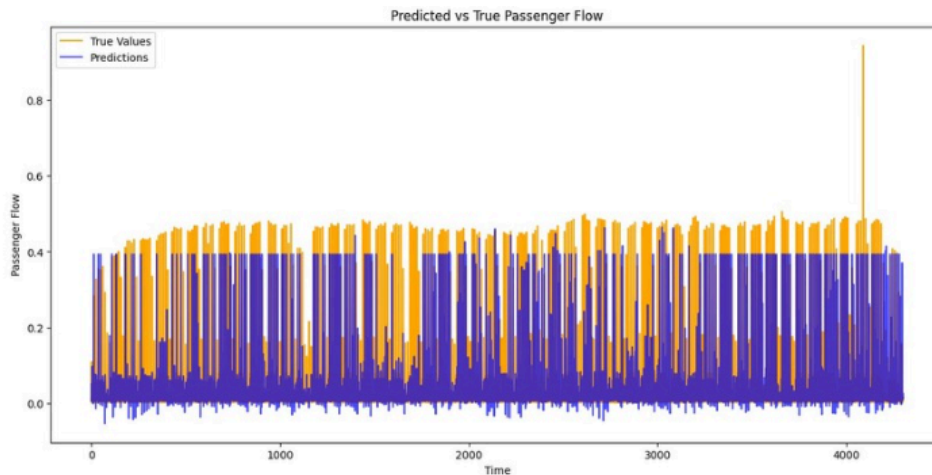
[Figura 8 : Resultados para el modelo LSTM más eficiente]



[Figura 9: Modelo 2]



[Figura 10:Modelo 4]



[Figura 11: Modelo 3]

La configuración de todos los parámetros discutidos previamente condujo a una mejora notable en la precisión de las predicciones y una reducción en las tasas de error globales al emplear el modelo LSTM. En comparación con modelos como KNN, el LSTM presenta ventajas significativas. En primer lugar, el LSTM es capaz de capturar relaciones temporales complejas y dependencias a largo plazo en los datos secuenciales, lo que lo hace más adecuado para la predicción de series temporales como el flujo de pasajeros en el metro. Esta capacidad contrasta con la limitación de KNN para modelar la secuencialidad de los datos, lo que puede dificultar su capacidad para capturar tendencias a largo plazo. Además, el LSTM forma parte de las redes neuronales recurrentes (RNN), lo que le permite aprender representaciones profundas y significativas de los datos, adaptándose mejor a la complejidad inherente de los datos de flujo de pasajeros. Esta profundidad de aprendizaje es una ventaja sobre el enfoque más superficial de KNN. Además, el LSTM puede generalizar mejor a diferentes escenarios y condiciones, lo que lo hace más versátil en la predicción del flujo de pasajeros en el metro. Mientras tanto, KNN puede ser más sensible a la configuración específica de los datos y tener dificultades para adaptarse a nuevos contextos. En resumen, el uso del modelo LSTM ofrece una serie de beneficios que lo hacen más adecuado para abordar la complejidad y la dinámica del flujo de pasajeros en el metro en comparación con enfoques más simples como KNN.

Conclusiones

Las redes LSTM son intrínsecamente adecuadas para modelar series temporales debido a su capacidad para capturar dependencias a largo plazo y su memoria selectiva, que permite retener o descartar información a través del tiempo. Esto es esencial en el contexto del transporte público, donde la demanda de pasajeros no solo sigue patrones diarios y estacionales, sino que también puede ser influida significativamente por eventos inusuales o perturbaciones. La pandemia de coronavirus presentó variaciones drásticas y no estacionarias en la movilidad urbana, haciendo que los patrones previos de afluencia de pasajeros se alteraran sustancialmente. El mecanismo de atención permite al modelo centrarse en los segmentos críticos de la serie temporal, particularmente durante los años afectados por la pandemia, donde los comportamientos de viaje sufrieron cambios significativos. Esta capacidad de "atender" a periodos específicos ayuda a entender mejor y predecir cómo eventos extraordinarios pueden influir en los patrones de movilidad.

Los hallazgos sugieren que futuras investigaciones deberían continuar explorando y refinando modelos de atención dinámica en contextos de grandes perturbaciones sociales o ambientales. Esto podría incluir el análisis de cómo variables externas adicionales (por ejemplo, anuncios gubernamentales importantes o cambios en las preferencias de transporte personal frente al público) pueden integrarse en el modelo para mejorar su precisión y relevancia.

Referencias bibliográficas

1. IBM. ¿Qué son las redes neuronales recurrentes?. (s.f). <https://www.ibm.com/mx-es/topics/recurrent-neural-networks>
2. Izudheena, S., Mulerikkalb, J. P., Johnc, M. J., Kd, M., Joshye, J., & Beveiraf, G. M. (s.f). *Short-Term Passenger Count Prediction for Metro Stations using LSTM Network*.
3. S. Miguel. Septiembre 22, 2021. *Ventajas y limitaciones de las Redes Neuronales Recurrentes y LSTM*. (s/f). CodificandoBits. <https://www.codificandobits.com/curso/fundamentos-deep-learning-python/redes-recurrentes-13-ventajas-limitaciones>
4. Ma, J.; Zeng, X.; Xue, X.; Deng, R. Metro Emergency Passenger Flow Prediction on Transfer Learning and LSTM Model. Appl. Sci. 2022, 12, 1644. <https://doi.org/10.3390/app12031644>
5. F. Fesalbon Daniel. Mayo 6, 2024. *Forecasting ferry Passenger Flow using long-short term memory neural networks*.
6. Z. Jinlei; F. Chen; Z. Cui; Y. Guo; Y. Zhu. (s.f). *Deep learning Architecture for Short-term Passenger Flow Forecasting in Urban Rail Transit*
7. Colah's blog. Agosto 27, 2015. *Understanding LSTM Networks*. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
8. StatQuest. Noviembre 7, 2022. *Long-Short Term Memory (LSTM) Clearly Explained*. <https://www.youtube.com/watch?v=YCzL96nL7j0>

