



Inteligencia Artificial 2

Examen final

Gastón Ezequiel Cavallo - Legajo: 11621

14 de octubre de 2024

1. Introducción

El estudio de la movilidad humana ha cobrado gran importancia en las últimas décadas debido al impacto que tiene en fenómenos urbanos complejos como la congestión del tráfico, la contaminación del aire y la difusión de epidemias Ghaffarpasand et al. [2024]. La capacidad de modelar y predecir los flujos de movilidad dentro de las ciudades es esencial para diseñar políticas públicas más efectivas, mejorar el transporte urbano y responder ante emergencias sanitarias o desastres naturales Olivera et al. [2022], Cavallo et al. [2023].

Tradicionalmente, los flujos de movilidad se han modelado utilizando enfoques como el modelo de Gravedad, que asume que el flujo entre dos ubicaciones está determinado por la población y la distancia entre ellas, o el modelo de Radiación, que utiliza información demográfica y oportunidades intermedias entre los puntos de origen y destino Hsu et al. [2021]. Si bien estos modelos han sido útiles para aproximar patrones de movilidad, presentan limitaciones, especialmente en escenarios donde los datos son escasos o los flujos no siguen patrones predefinidos Celes et al. [2023].

Recientemente, el uso de Redes Generativas Adversarias (GANs) ha demostrado un potencial significativo para generar redes de movilidad realistas a partir de datos históricos Huang et al. [2023], Lin et al. [2023]. En particular, el modelo MoGAN (Model based on Generative Adversarial Networks) ha sido propuesto como una alternativa basada en aprendizaje profundo que supera las limitaciones de los modelos tradicionales al generar redes de movilidad más realistas y versátiles Mauro et al. [2022]. MoGAN se basa en una arquitectura de Deep Convolutional GAN (DCGAN), lo que le permite aprender patrones de movilidad a partir de matrices de adyacencia que representan los flujos de personas entre ubicaciones en una ciudad.

El objetivo de este trabajo final para la asignatura Inteligencia Artificial 2¹ es implementar el MoGAN para utilizarlo sobre los datos de demanda del sistema de bicicletas compartidas (Bike Sharing System, BSS) de la Ciudad Autónoma de Buenos Aires (CABA) y comparar sus resultados con los modelos de Gravedad y Radiación Simini et al. [2021], Kang et al. [2015], Li et al. [2023] y la implementación de una mejora del modelo aplicando la función de perdida de Wasserstein y la política de penalización del gradiente Gulrajani et al. [2017]. Para ello, se entrenará MoGAN sobre el dataset disponible y se implementarán los modelos tradicionales para realizar una comparación exhaustiva de la capacidad de cada modelo para generar redes de movilidad realistas.

1.1. Planteamiento del Problema

Dado que las ciudades están en constante cambio, tanto en su estructura como en los hábitos de movilidad de sus habitantes, es fundamental contar con

¹Licenciatura en Ciencias de la Computación, Facultad de Ingeniería, Universidad Nacional de Cuyo, Mendoza, Argentina

modelos que no solo puedan replicar patrones pasados, sino que también sean capaces de generar redes de movilidad que se ajusten a diversas condiciones. Este estudio busca responder a la pregunta: ¿Qué tan efectivos son los modelos generativos basados en GANs en comparación con los enfoques tradicionales para generar redes de movilidad urbana? Para esta pregunta general podemos plantear los objetivos que a continuación aparecen.

1.2. Objetivos

1. Implementar y entrenar el modelo MoGAN utilizando un conjunto de datos de movilidad construido a partir de los datos del sistema de bicicletas compartidas de la Ciudad Autónoma de Buenos Aires.
2. Replicar los modelos tradicionales de Gravedad y Radiación para compararlos con MoGAN Mauro et al. [2022].
3. Implementar la mejora del sistema
4. Evaluar y analizar la calidad de las redes de movilidad generadas por cada modelo utilizando las métricas comúnmente utilizadas, Liu et al. [2018], Luca et al. [2021], Simini et al. [2021].
5. Continuar con trabajos futuros propuestos por los creadores de la red MoGAN y aplicar técnicas innovadoras como la función de perdida de Wasserstein y la política de penalización del gradiente.

1.3. Justificación del tema propuesto

Este trabajo es relevante porque permitirá explorar nuevas metodologías basadas en inteligencia artificial para la generación de redes de movilidad urbana. El uso de GANs podría ofrecer mejores resultados que los enfoques tradicionales, particularmente en situaciones donde los datos son incompletos o poseen ruido. Además, este estudio proporcionará una base sólida para futuras investigaciones en simulación y predicción de movilidad, facilitando la planificación urbana, el análisis de políticas de transporte y la gestión de crisis. Además, el alumno se encuentra realizando su tesis final de carrera en temas afines, lo que permite una sinergia entre los tópicos desarrollados en este proyecto y los temas de su tesis final de grado.

2. Metodología

El presente proyecto se centra en la generación y evaluación de redes de movilidad urbana utilizando, tanto modelos tradicionales, como técnicas de inteligencia artificial, específicamente redes generativas adversarias (GANs). El proyecto se dividirá en varias etapas que incluyen la recopilación y tratamiento de datos, la implementación de modelos, el entrenamiento de redes neuronales

y la comparación de resultados. A grandes rasgos se espera de este sistema que, a partir de una serie de matrices origen destino real, pueda generar datos sintéticos (matrices origen destino) con una distribución lo más cercana posible a la real.



Figura 1: Descripción de etapas a lo largo del proyecto.

2.1. Recopilación y Preprocesamiento de Datos

2.1.1. Datos de movilidad:

Se emplearon conjuntos de datos reales que representan la movilidad dentro de una ciudad. En particular, los datos de demanda del sistema de bicicletas compartidas de la Ciudad Autónoma de Buenos Aires ², figuras 2 y 3. Estos datos son descargados en formato *CSV* y contienen la fecha, longitud y latitud del origen y destino del viaje y debieron ser analizados y limpiados para su utilización, por ej. borrando viajes que tenían como origen o destino la estación destinada al taller, o valores nulos en algún campo de utilidad.

²<https://data.buenosaires.gob.ar/dataset/bicicletas-publicas>

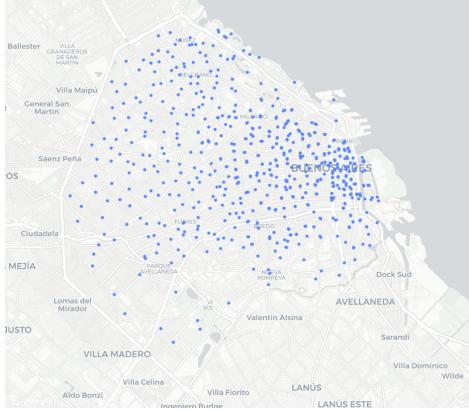


Figura 2: Estaciones de bicicletas compartidas en la CABA.

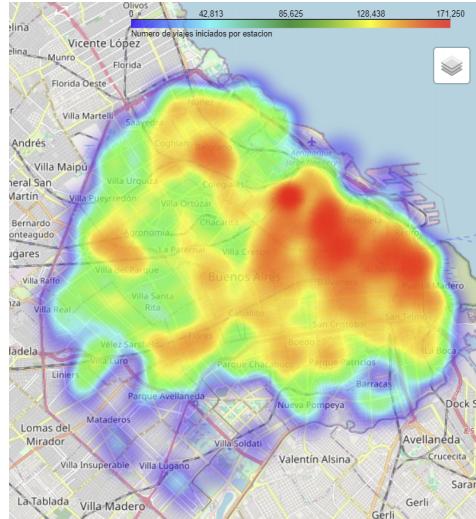


Figura 3: Mapa de calor de los viajes en CABA.

2.1.2. Agrupamiento de datos:

Debido a que el dataset contiene datos para los años 2018 a 2023, se ha propuesto agruparlo por semana, y por día, de esta forma podemos abarcar dos enfoques distintos, con una distribución más uniforme de los datos (agrupado semanalmente) y con una granularidad más baja (agrupado diariamente). En la figura 4 podemos observar una vista previa del dataset agrupado por semanas.

	week_year	origin	destination	flow
0	00-2018	31	31	91
1	00-2018	31	32	19
2	00-2018	31	38	6
3	00-2018	31	39	30
4	00-2018	31	40	134
...
223711	52-2022	1	1	14
223712	52-2022	1	10	4
223713	52-2022	1	11	1
223714	52-2022	1	14	1
223715	52-2022	1	17	1
1253 rows × 4 columns				

Figura 4: Vista previa del dataset agrupado por semanas.

2.1.3. Teselación espacial:

Para generar redes de movilidad, se dividirá el área geográfica en mosaicos (tiles) uniformes, fig. 5. Cada mosaico representará un nodo en la red, y los flujos de personas entre nodos se representarán como bordes ponderados Gra-
ser [2017], en la figura 6 se puede ver el resultado de las redes de movilidad generada a partir de los datos de viajes del dataset y los mosaicos definidos mediante la teselación.

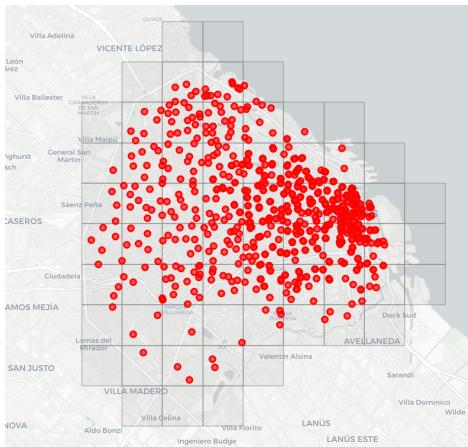


Figura 5: Teselación espacial aplicada a la superficie de CABA, en rojo la ubicación de las estaciones.

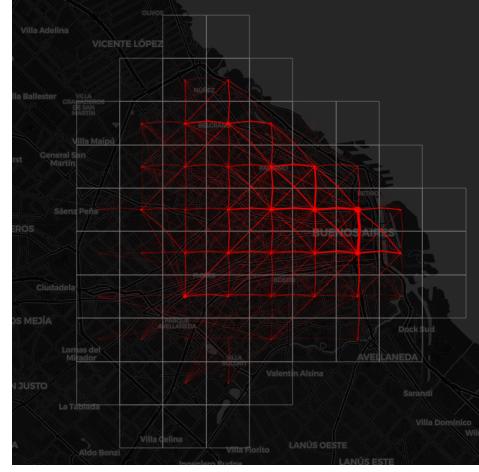


Figura 6: Gráfico de las trayectorias en la Teselación espacial aplicada a la superficie de CABA para la 38 ava semana de 2019.

2.1.4. Generación de matrices de origen destino:

A partir de los datos pre procesados, se generarán matrices origen destino (matrices OD) entre los mosaicos, donde cada celda (i, j) representará el flujo de personas entre la ubicación i (mosaico i) y la ubicación j (mosaico j). Las figuras 7 y 8 muestran los resultados de una matriz OD en particular.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	155	1	11	0	0	0	0	3	0	1	0	4	33	0	6	0	0	0	1	0	0
2	0	0	236	1	0	0	0	0	0	0	0	0	1	0	0	3	10	1	0	0	0	0
3	0	25	0	73	0	0	0	0	0	0	0	0	3	30	0	0	9	1	0	0	0	0
4	0	0	0	0	382	0	0	0	0	29	300	30	3	0	0	0	0	36	48	27	37	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	72	0	1	2	1	3	0	0	0	2	4	1	7
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	20	0	0	0	0	5	0	112	86	9	7	0	0	14	8	181	50	11
12	0	0	2	0	277	0	0	0	0	3	0	54	1831	172	15	0	0	2	0	343	439	191
13	0	1	0	4	25	0	0	0	0	4	0	4	164	302	116	0	0	7	1	12	22	44
14	0	27	0	46	13	0	0	0	0	2	0	2	13	98	881	0	0	16	2	156	5	3
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	8	3	6	0	0	0	0	5	0	14	7	7	11	0	0	72	4	21	10	0	2
18	0	0	11	1	0	0	0	0	1	0	6	3	1	0	0	4	175	30	0	0	4	0
19	0	4	2	0	36	0	0	0	1	0	141	253	22	159	0	0	6	27	3532	506	68	35

Figura 7: Vista previa de la matriz de adyacencia.

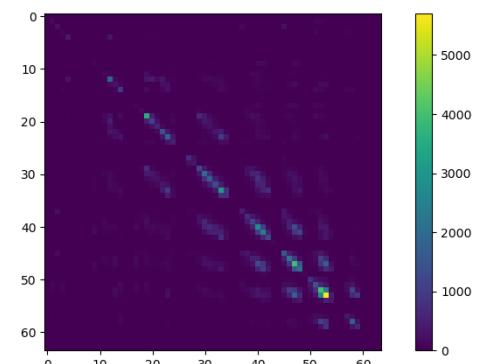


Figura 8: Vista general de la matriz de adyacencia.

2.2. Implementación del Modelo MoGAN

2.2.1. Arquitectura de la GAN:

Se implementará MoGAN utilizando una arquitectura basada en Deep Convolutional GANs (DCGANs) Mauro et al. [2022]. Cuyo esquema general se puede observar en la figura 9. Este modelo toma las matrices origen destino y las interpreta como imágenes, luego intenta recrearlas a partir de un vector de ruido, para ello, en primer lugar tenemos al Generador, que crea un vector de valores aleatorios (ruido) y mediante escalado y técnicas de convolución incrementalmente gradualmente la dimensionalidad de los datos y aplica transformaciones no lineales para construir la matriz OD sintética. Luego el discriminador tiene como objetivo distinguir entre las matrices OD reales y las sintéticas, produciendo como salida una puntuación que demuestra que tan probable es que la matriz generada sea real. En cuanto a la estructura del discriminador, sigue una arquitectura simétrica o invertida en comparación con el generador. Su objetivo es extraer características de las matrices OD y evaluar que tan reales o falsas son, tiene una función sigmoide al final de la red para obtener una salida en el rango $[0, 1]$, interpretada como la probabilidad de que la muestra sea real. Durante el entrenamiento el generador recibe retroalimentación del discriminador sobre que tan realistas son sus salidas y ajusta sus pesos del entrenamiento para mejorar la generación, a través de las convoluciones y activaciones el generador aprende a crear patrones en las matrices que reflejen las propiedades estructurales de las redes de movilidad reales. En general, MoGAN tiene la siguiente estructura:

- **Generador:** Convertirá un vector de ruido aleatorio en una matriz de adyacencia de 64×64 que representará una red de movilidad.
- **Discriminador:** Clasificará las redes de movilidad como reales o generadas, retroalimentando al generador para mejorar la calidad de las redes sintéticas.

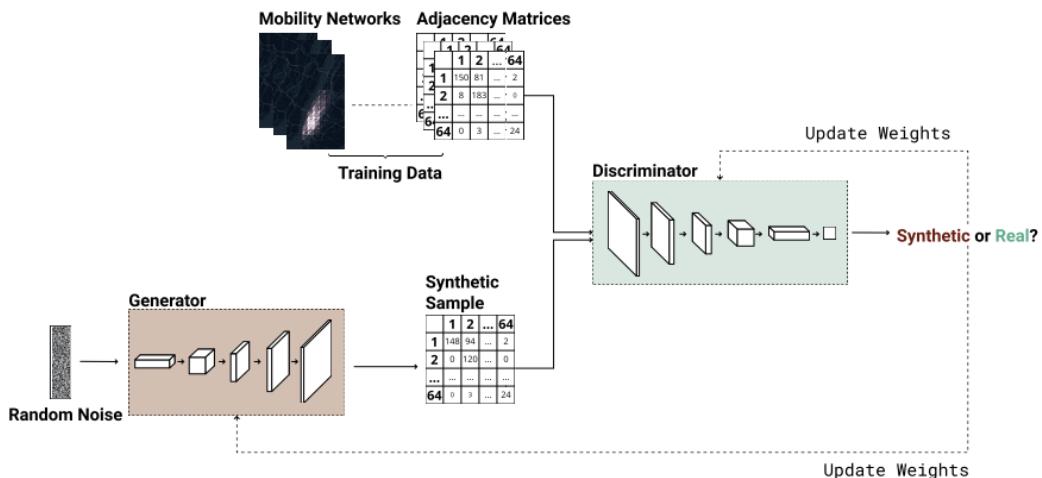


Figura 9: Esquema del modelo MoGAN propuesto por Mauro et al. [2022]

2.2.2. Entrenamiento de MoGAN:

El modelo se entrenó utilizando una fracción del conjunto de datos de bicicletas compartidas para que el generador aprenda a producir redes de movilidad que se asemejen a las redes reales.

- **Librerías:** Para este proyecto se utilizaron, en principio, las siguientes librerías:
 - **PyTorch:** Para la implementación de la GAN.
 - **NumPy y Pandas:** Para el manejo de matrices y manipulación de los datos.
 - **Matplotlib:** Para la visualización de los resultados.

2.3. Mejora del Modelo MoGAN (WMoGAN-GP)

2.3.1. Implementación de la función de perdida de Wasserstein:

Se utilizó la función de perdida de Wasserstein como una de las mejoras debido a que está comprobado que mejora el entrenamiento y la convergencia del modelo Gulrajani et al. [2017]. Para ello se modificó la función de perdida, reemplazando Binary Cross Entropy (BCE) por la función de Wasserstein. A diferencia de los discriminadores en GANs tradicionales, en WMoGAN-GP el discriminador no produce probabilidades mediante una función sigmoide; en su lugar, produce valores escalares reales que estiman la distancia de Wasserstein entre las distribuciones reales y generadas.

2.3.2. Aplicación de la política de penalización del gradiente:

Adicional a la función de pérdida mejorada, se utilizó la política de penalización del gradiente, basado en el mismo trabajo de Gulrajani et al. [2017], mejora considerablemente la convergencia del generador y discriminador produciendo datos sintéticos más precisos y solucionó algunos problemas presentes en el modelo original como la desaparición del gradiente y el colapso del modo.

2.3.3. Diferencias clave con respecto al modelo original

- **Función de Activación Final:** Con BCE, el discriminador utiliza una función sigmoide al final, mientras que en WGAN-GP no se utiliza función de activación final, el discriminador produce valores reales sin restricciones
- **Salida Interpretada como Probabilidad:** Con BCE, la salida del discriminador es una probabilidad, en WGAN-GP, la salida es un valor que estima la diferencia entre distribuciones.
- **Problemas Asociados con BCE:**

- Desaparición de Gradientes: Cuando el discriminador es demasiado bueno, el generador recibe gradientes muy pequeños.
- Inestabilidad en el Entrenamiento: Mayor riesgo de oscilaciones y colapso de modo.

2.4. Implementación de Modelos Tradicionales

Se implementaron dos modelos clásicos de generación de redes de movilidad para su comparación con MoGAN. Para la implementación de estos modelos se utilizó la librería scikit-mobility Pappalardo et al. [2022]. esta librería se encuentra implementada en Python y extiende a Pandas para proveer un entorno de simulación de movilidad y cuenta con varios modelos implementados.

2.4.1. Modelo de Gravedad:

Este modelo generará flujos entre ubicaciones basándose en la población y la distancia entre ellas.

- **Ecuación:** El flujo entre dos ubicaciones se modela como proporcional al producto de sus poblaciones e inversamente proporcional a la distancia entre ellas.
- **Librería:** Se utilizó scikit-mobility Pappalardo et al. [2022], que incluye una implementación de este modelo.

2.4.2. Modelo de Radiación:

Este modelo no requiere parámetros, y generará flujos en función de la población y las oportunidades intermedias entre las ubicaciones. Se implementa sobre scikit-mobility.

3. Análisis y Evaluación de los Resultados

Al finalizar la búsqueda de hiperparámetros ideales, se obtuvo una configuración para los modelos de $lr_G = 0,002$, $lr_D = 0,0002$, $\lambda_{gp} = 10$, donde lr_G y lr_D son el learning rate del Generador y Discriminador respectivamente y λ_{gp} es la penalización del gradiente en el modelo WMoGAN.

Para el análisis de resultados se utilizó el dataset anteriormente mencionado separándolo en ventanas de tiempo de una semana (251 redes) y de un día (1726 redes), dividiéndolo en conjuntos de entrenamiento (200 y 1380 redes para la separación entre semana y día respectivamente) y de validación (51 y 346 redes para la separación entre semana y día respectivamente), para ambos modelos se utilizaron 6000 épocas durante el entrenamiento. Se evaluaron los modelos computando la diferencia entre cada red en los conjuntos sintéticos y reales. Si las redes de movilidad generadas son realistas, estas debían diferir de las reales de la misma forma que las reales difieren entre sí. Para llevar más a fondo este

aspecto, creamos un conjunto de redes de movilidad (mixtas), en las cuales la mitad de ellas eran elegidas de manera uniforme aleatoriamente del conjunto de datos reales y la otra mitad fue elegida de la misma manera del conjunto de datos sintéticos. Luego, computamos la diferencia por pares entre cualquier par de redes de movilidad posible del conjunto mixto

A fin de evaluar la potencialidad de MoGAN para el caso de estudio de CABA en contraposición con los modelos, se utilizaron dos enfoques separados, uno de ellos fue computar el error entre dos matrices de adyacencia de las redes. Para esto, utilizamos diferentes métricas: *Cut Distance* (CD) Liu et al. [2018], *Common Part of Commuters* (CPC) Luca et al. [2021] y Normalized Root Mean Square Error (NRMSE) Simini et al. [2021] que son populares en la evaluación de la movilidad urbana

La distancia de corte (CD) está basada en la noción de recorte de peso, ampliamente usada en teoría de grafos, y mide hasta qué punto un grafo es bipartito. La norma de corte $\|A\|_C$ de una matriz real $A = (a_{ij})$, $i \in R$, $j \in S$ con un conjunto de filas indexadas por R y un conjunto de columnas indexadas por S , es el máximo total $I \subset R$, $J \subset S$ del valor $|\sum_{i \in I, j \in J} a_{ij}|$. La CD entre dos matrices de adyacencia A y B es la norma de corte de su diferencia:

$$CD(A,B) = \max_{S \in V} \frac{1}{|V|} |e_A(S, S^C) - e_B(S, S^C)|$$

siendo V el número de nodos (64 en nuestro caso), $e_G(S, T) = \sum_{i \in S, j \in T} w_{ij}$ es el peso de corte de la matriz de adyacencia G con pesos w_{ij} , esto es, la suma de los pesos de las aristas que empiezan en S y terminan en T y $S^C = V \setminus S$. Para calcular CD, he utilizado la implementación de Python disponible en la librería cutnorm.

La parte común de los viajeros (CPC), es una buena métrica para computar la similitud entre matrices generadas y reales, y es definida como

$$CPC(A,B) = \frac{2 \sum_{i,j=1}^n \min(a_{ij}, b_{ij})}{\sum_{i,j=1}^n a_{ij} + \sum_{i,j=1}^n b_{ij}}$$

CPC es una métrica utilizada ampliamente en estudios de movilidad urbana y se mide en rangos de 0 a 1. Un CPC de 1 indica una coincidencia perfecta entre las redes generadas y las reales. Por otro lado, 0 muestra un mal desempeño con nada de coincidencia. En otros términos, CPC puede ser interpretado como una métrica de precisión.

El Error cuadrático medio (RMSE) definido como:

$$RMSE(A,B) = \sqrt{\frac{1}{n} \sum_{i,j=1}^n (a_{ij} - b_{ij})^2}$$

donde a_{ij} y b_{ij} son los elementos (flujos) en posición (i, j) en las dos matrices de adyacencia A y B de las redes y n es el número de elementos de las matrices (64 x 64). El error cuadrático medio normalizado (NRMSE) es una normalización minimax del RMSE, definido como:

$$NRMSE = \frac{RMSE(A,B)}{\max(A,B) - \min(A,B)}$$

El otro enfoque utilizado para computar la diferencia entre dos redes de movilidad consiste en comparar sus distribuciones de pesos de aristas (edge weights) y pesos-distancia (weight-distances). Los pesos de aristas indican los valores (flujos) de las matrices de adyacencia que describen las dos redes de movilidad. Mientras que la relación peso-distancia muestra la combinación del peso de una arista (flujo) y la distancia entre dos nodos que componen la arista. Computamos entonces la matriz pesos-distancia de una red de movilidad como $\hat{A} = A/(d+\epsilon)$, donde A es la matriz de pesos de la red, d la matriz de distancias con la misma cardinalidad y orden de A y representa las distancias geográficas entre todos los pares de nodos. Es necesario añadir el término residual $\epsilon = 0,01$ al denominador para evitar la división por cero en los elementos de la diagonal de las matrices de adyacencia. Dadas dos redes de movilidad, mientras más similares son sus distribuciones de pesos de aristas y pesos-distancia, más es la similitud entre ellas. Para medir la similitud entre dos distribuciones utilizamos la divergencia de Jensen-Shannon (JSD) Fuglede and Topsoe [2004].

$$JS(P||Q) = \frac{1}{2}KL(P||M) + \frac{1}{2}KL(Q||M),$$

donde P y Q son dos distribuciones de densidad, $M = \frac{1}{2}(P + Q)$, y KL es la divergencia de Kullback-Leibler (KL) definida como:

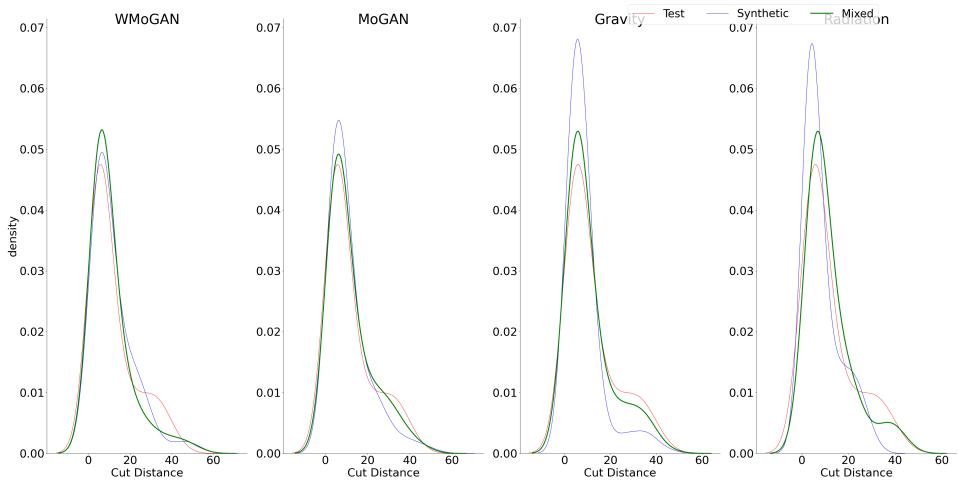
$$KL(P||Q) = \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right),$$

Nótese que para el análisis de datos agrupados por día no hemos sido capaces de probar el modelo de Radiation, esto sucedió por una limitación en la implementación del modelo, sin embargo, es más que suficiente el análisis con el modelo de Gravity y la comparación entre los modelos MoGAN y WMoGAN.

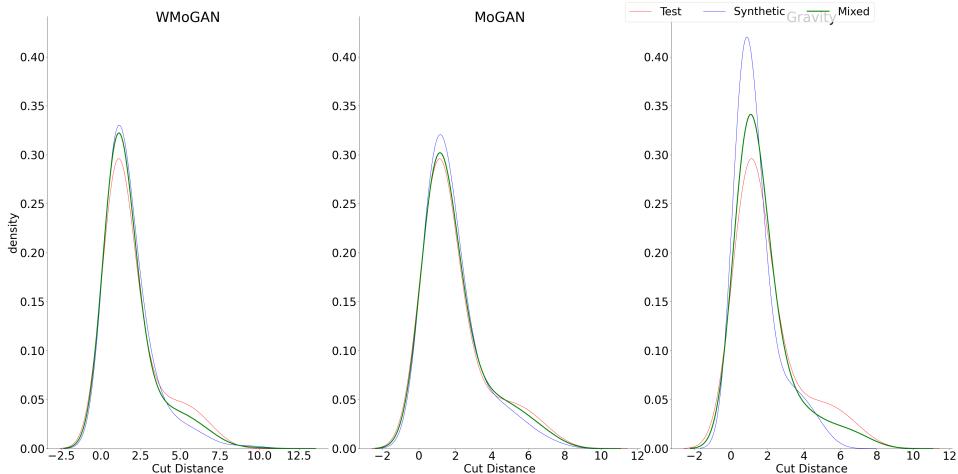
3.1. Evaluación

Las figuras 10a y 10b muestran la distribución de Cut Distance (CD) evaluada en los tres tipos de redes, reales (rojo), sintéticas (azul) y mixtas (verde) para los modelos WMoGAN, MoGAN, Radiation y Gravity.

En el dataset semanal podemos ver que MoGAN presenta una ligera mejora frente a WMoGAN verificando que la distribución se solapa casi en su totalidad, esto significa que ambos modelos generan redes de movilidad que son apenas distinguibles de las reales en comparación con los baselines Gravity y Radiation. Esta misma ventaja se puede apreciar a lo largo de las métricas CPC 11 y NRMSE 12

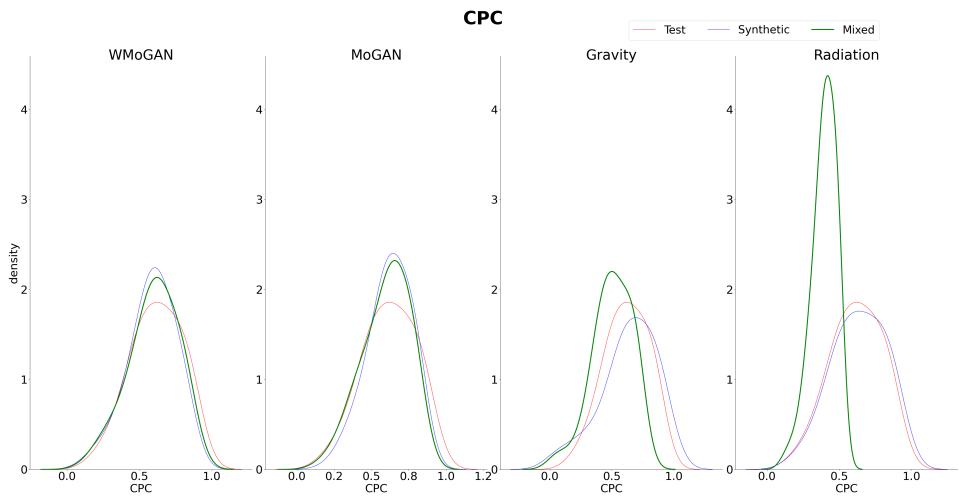


(a) CD por semana

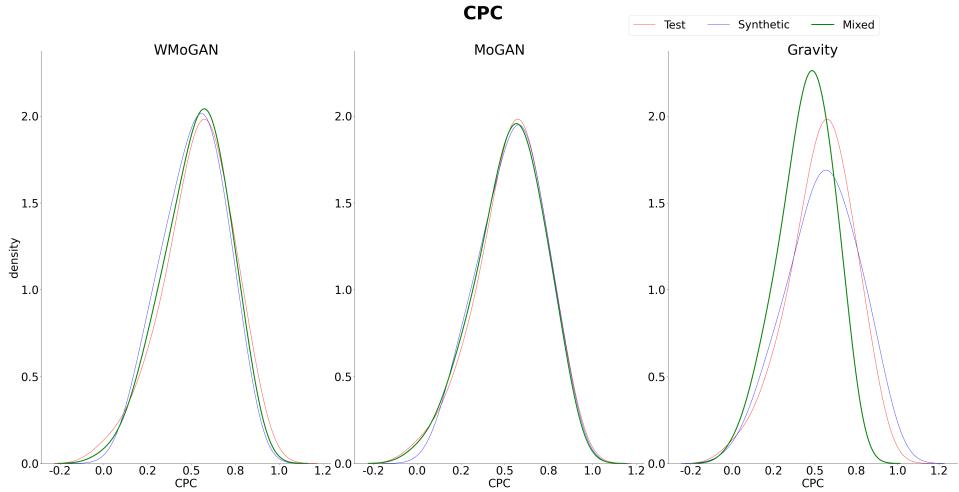


(b) CD por dia

Figura 10: Cut Distance para el dataset dividido en semanas arriba, en días abajo.

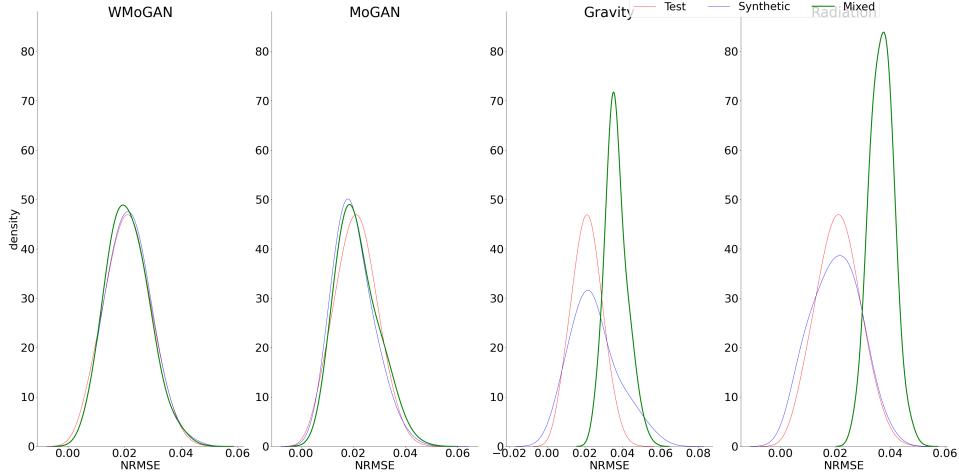


(a) CPC por semana

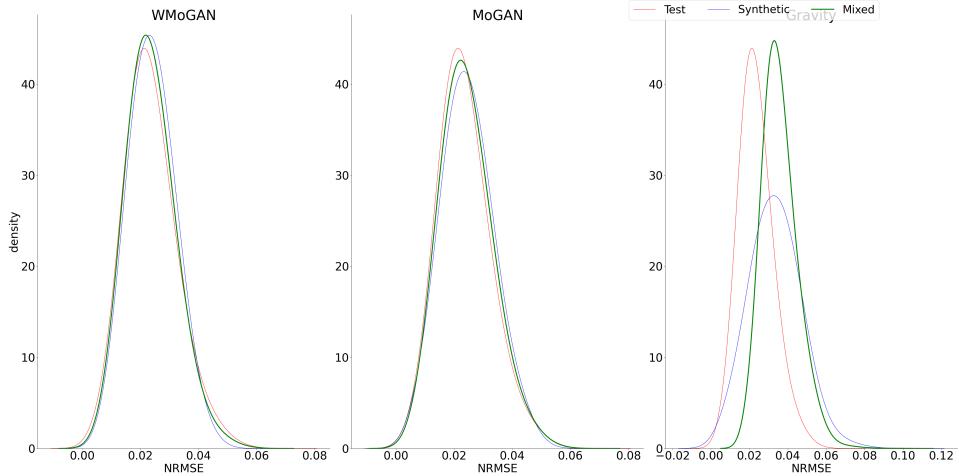


(b) CPC por semana

Figura 11: CPC para el dataset dividido en semanas arriba, en días abajo.



(a) RMSE por semana



(b) RMSE por semana

Figura 12: RMSE para el dataset dividido en semanas arriba, en días abajo.

Luego, un análisis más profundo es presentado al comparar los valores de las distancias JS entre las distribuciones de los modelos, en el cuadro 1 y en los siguientes, se puede observar para cada distribución el resultado de la divergencia de Jensen-Shannon para cada conjunto de datos (mixto y sintético) comparado con el conjunto de datos real, esto se repite para cada modelo y al final de la tabla se muestra una mejora relativa que presenta el modelo frente a los baselines. A modo de ayuda visual se comparan por separados los modelos WMoGAN y MoGAN.

En las últimas filas de cada tabla encontraremos los resultados de las distribuciones de pesos de aristas y pesos-distancia.

Aquí podemos analizar detenidamente la puntuación que obtuvo cada modelo en cada caso, tanto para los datos semanales como para los diarios. Dejando a un costado la comparación con los baselines podemos comparar numéricamente que tan capaces de replicar los datos fueron los dos modelos computacionales. Por un lado, vemos que en los datos semanales WMoGAN 1

obtuvo mejores puntuaciones (más bajas) de pesos de aristas y pesos-distancia que MoGAN 2, esto significa, que al volver a comparar con las otras métricas un análisis más profundo indica que con un mínimo ajuste o más entrenamiento el modelo puede replicar mejor las redes que MoGAN. Por otro lado, al analizar los resultados de los datos diarios (tablas 3 y 4) podemos ver que esta vez los modelos no tienen gran diferencia entre sí, aunque al no presentar la misma tendencia de mejora WMoGAN como lo hace MoGAN nos da una idea de que no es capaz de generar mejores redes con más datos.

Dist	WMoGAN		Gravity		Radiation		Mejora relativa			
	m	s	m	s	m	s	$\Delta_{m,G}$	$\Delta_{s,G}$	$\Delta_{m,R}$	$\Delta_{s,R}$
CD	0.14	0.17	0.07	0.17	0.18	0.22	-105 %	2 %	26 %	26 %
CPC	0.13	0.16	0.35	0.2	0.58	0.09	62 %	19 %	77 %	-86 %
NMRSE	0.09	0.08	0.66	0.23	0.7	0.12	86 %	66 %	86 %	33 %
JS _{ew}	0.19	0.2	0.21	0.26	0.19	0.32	7 %	24 %	-1 %	39 %
JS _{wd}	0.11	0.12	0.11	0.41	0.19	0.45	7 %	71 %	44 %	74 %

Cuadro 1: Comparación de los modelos WMoGAN, Gravity y Radiation con mejoras relativas en los datos semanales. Cada fila representa la distancia JS que tiene cada conjunto de datos(mixto y sintético) con respecto al real evaluado para alguna distribucion

Dist	MoGAN		Gravity		Radiation		Mejora relativa			
	m	s	m	s	m	s	$\Delta_{m,G}$	$\Delta_{s,G}$	$\Delta_{m,R}$	$\Delta_{s,R}$
CD	0.08	0.13	0.07	0.17	0.18	0.22	-28 %	23 %	54 %	42 %
CPC	0.2	0.15	0.35	0.2	0.58	0.09	43 %	23 %	66 %	-75 %
NMRSE	0.14	0.1	0.66	0.23	0.7	0.12	79 %	56 %	80 %	13 %
JS _{ew}	0.32	0.38	0.21	0.26	0.19	0.32	-51 %	-46 %	-64 %	-17 %
JS _{wd}	0.16	0.15	0.11	0.41	0.19	0.45	-40 %	64 %	16 %	67 %

Cuadro 2: Comparación de modelos MoGAN, Gravity y Radiation con mejoras relativas en los datos semanales. Cada fila representa la distancia JS que tiene cada conjunto de datos(mixto y sintético) con respecto al real evaluado para alguna distribucion

Dist	WMoGAN		Gravity		Mejora relativa	
	<i>m</i>	<i>s</i>	<i>m</i>	<i>s</i>	$\Delta_{m,G}$	$\Delta_{s,G}$
CD	0.05	0.1	0.06	0.19	19 %	45 %
CPC	0.08	0.12	0.26	0.11	70 %	-12 %
NMRSE	0.04	0.08	0.47	0.33	92 %	75 %
JS _{ew}	0.14	0.2	0.23	0.14	39 %	20 %
JS _{wd}	0.05	0.09	0.14	0.39	61 %	78 %

Cuadro 3: Comparación de modelos WMoGAN y Gravity con mejoras relativas en los datos diarios. Cada fila representa la distancia JS que tiene cada conjunto de datos(mixto y sintético) con respecto al real evaluado para alguna distribucion

Dist	MoGAN		Gravity		Mejora relativa	
	<i>m</i>	<i>s</i>	<i>m</i>	<i>s</i>	$\Delta_{m,G}$	$\Delta_{s,G}$
CD	0.03	0.07	0.06	0.19	49 %	61 %
CPC	0.04	0.09	0.26	0.11	83 %	20 %
NMRSE	0.04	0.07	0.47	0.33	92 %	78 %
JS _{ew}	0.12	0.17	0.23	0.14	48 %	31 %
JS _{wd}	0.05	0.08	0.14	0.39	67 %	79 %

Cuadro 4: Comparación de modelos MoGAN y Gravity con mejoras relativas en los datos diarios. Cada fila representa la distancia JS que tiene cada conjunto de datos(mixto y sintético) con respecto al real evaluado para alguna distribucion

4. Resultados Esperados

El objetivo principal de este proyecto es generar redes de movilidad realistas y comparar la capacidad predictiva de diferentes modelos, incluyendo MoGAN y tres enfoques tradicionales: Gravedad, Radiación y Pesos aleatorios. En cuanto a la potencialidad de MoGAN esperamos demostrar la potencialidad de este enfoque para su utilización en movilidad urbana donde la demanda de origen y destino se encuentra bien determinada. También se espera observar patrones de conducta de los ciudadanos. El fin último es a partir de la movilidad, poder a futuro mejorar la calidad de vida de las ciudades. También se realizarán diferentes visualizaciones de las redes para comparar la similitud y diferencias con el flujo real y que pueda reflejar patrones de movilidad no triviales como bucles en una misma zona o bien patrones poblacionales (género, edades, etc.) que los modelos tradicionales no capturan.

Este estudio demuestra que los modelos generativos adversarios, específicamente MoGAN y su variante mejorada WMoGAN, tienen un gran potencial para generar redes de movilidad urbana más realistas en comparación con los

enfoques tradicionales, como los modelos de Gravedad y Radiación. Los resultados obtenidos a lo largo de las diferentes métricas de evaluación (Cut Distance, CPC, NRMSE) revelan que las redes de movilidad generadas por MoGAN y WMoGAN son más cercanas a las reales, mostrando menores diferencias en comparación con los modelos basados en reglas tradicionales.

En particular, MoGAN presenta una mejor capacidad para replicar las redes de movilidad, con un rendimiento notable en el análisis semanal y diario. WMoGAN, por su parte, gracias a la implementación de la función de pérdida de Wasserstein y la política de penalización del gradiente, mostró mejoras en la estabilidad y calidad de las redes generadas, aunque los beneficios fueron más evidentes en los datos semanales que en los diarios.

Finalmente, se concluye que, aunque los modelos tradicionales siguen siendo herramientas útiles, los enfoques basados en GANs, y específicamente MoGAN y WMoGAN, superan las limitaciones de los métodos convencionales al generar redes de movilidad más ajustadas a la realidad, especialmente en escenarios donde los datos son escasos o ruidosos. Esta investigación abre la puerta a futuras mejoras y aplicaciones, como la integración de datos de diversas fuentes y la generación de redes de movilidad con una mayor capacidad de adaptación a contextos urbanos en constante evolución.

Referencias

- Gastón Ezequiel Cavallo, Emmanuel N. Millán, Pablo Javier Vidal, and Ana Carolina Olivera. Análisis visual del impacto del covid-19 sobre el sistema de bicicletas compartidas en la ciudad autónoma de buenos aires. In *Actas del XII Encuentro de Investigadores y Docentes de Ingeniería (EnIDI)*, 2023. to appear.
- Clayson Celes, Azzedine Boukerche, and Antonio A.F. Loureiro. Generating and analyzing mobility traces for bus-based vehicular networks. *IEEE Transactions on Vehicular Technology*, 72:16409–16425, 12 2023. ISSN 19399359. doi:10.1109/TVT.2023.3290154.
- B. Fuglede and F. Topsøe. Jensen-shannon divergence and hilbert space embedding. In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, pages 31–, 2004. doi:10.1109/ISIT.2004.1365067.
- Omíd Ghaffarpasand, Deo Okure, Paul Green, Saba Sayyahi, Priscilla Adong, Richard Sserunjogi, Engineer Bainomugisha, and Francis D. Pope. The impact of urban mobility on air pollution in kampala, an exemplar sub-saharan african city. *Atmospheric Pollution Research*, 15, 4 2024. ISSN 13091042. doi:10.1016/j.apr.2024.102057.
- Anita Graser. Tessellating urban space based on street intersections and barriers to movement. *GI Forum*, 1:114–125, 01 2017. doi:10.1553/giscience2017_01_s114.
- Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017. URL <https://arxiv.org/abs/1704.00028>.
- Chiawei Hsu, Chao Fan, and Ali Mostafavi. Limitations of gravity models in predicting fine-scale spatial-temporal urban mobility networks. *arXiv preprint arXiv:2109.03873*, 2021.
- Tongge Huang, Pranamesh Chakraborty, and Anuj Sharma. Deep convolutional generative adversarial networks for traffic data imputation encoding time series as images. *International Journal of Transportation Science and Technology*, 12: 1–18, 3 2023. ISSN 20460449. doi:10.1016/j.ijtst.2021.10.007.
- Chaogui Kang, Yu Liu, Diansheng Guo, and Kun Qin. A generalized radiation model for human mobility: Spatial scale, searching direction and trip constraint. *PLoS ONE*, 10, 11 2015. ISSN 19326203. doi:10.1371/journal.pone.0143500.
- Zhitao Li, Chen Yuan, Jinjun Tang, Kexin Zhu, and Xiaoyi Pan. Understanding the mobility of public transport systems based on weighted multiplex networks. *Physica A: Statistical Mechanics and its Applications*, 624, 8 2023. ISSN 03784371. doi:10.1016/j.physa.2023.128977.
- Hongyi Lin, Yang Liu, Shen Li, and Xiaobo Qu. How generative adversarial networks promote the development of intelligent transportation systems: A survey. *IEEE/CAA Journal of Automatica Sinica*, 10:1781–1796, 9 2023. ISSN 23299274. doi:10.1109/JAS.2023.123744.

Qun Liu, Zhishan Dong, and En Wang. Cut based method for comparing complex networks. *Scientific Reports*, 8, 12 2018. ISSN 20452322. doi:10.1038/s41598-018-21532-5.

Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. A survey on deep learning for human mobility. *ACM Comput. Surv.*, 55(1), nov 2021. ISSN 0360-0300. doi:10.1145/3485125.

Giovanni Mauro, Massimiliano Luca, Antonio Longa, Bruno Lepri, and Luca Pappalardo. Generating mobility networks with generative adversarial networks. *EPJ Data Science*, 11, 12 2022. ISSN 21931127. doi:10.1140/epjds/s13688-022-00372-4.

Ana Carolina Olivera, Pablo Javier Vidal, Enrique Gabriel Baquela, Marisa Haderne, Mariel Denise Volman Stern, and Gastón Cavallo. Inteligencia computacional aplicada a movilidad urbana. In Universidad Champagnat, editor, *Libro de Actas WICC 2022, XXIV Edición del Workshop de investigadores en Ciencias de la Computación*, pages 51–53, April 2022. ISBN 978-987-48222-3-9.

Luca Pappalardo, Filippo Simini, Gianni Barlacchi, and Roberto Pellungrini. scikit-mobility: A python library for the analysis, generation, and risk assessment of mobility data. *Journal of Statistical Software*, 103(4):1–38, 2022. doi:10.18637/jss.v103.i04. URL <https://www.jstatsoft.org/index.php/jss/article/view/v103i04>.

Filippo Simini, Gianni Barlacchi, Massimiliano Luca, and Luca Pappalardo. A deep gravity model for mobility flows generation. *Nature Communications*, 12, 12 2021. ISSN 20411723. doi:10.1038/s41467-021-26752-4.