



UNIVERSIDAD ESAN
FACULTAD DE INGENIERÍA
INGENIERÍA DE TECNOLOGÍAS DE INFORMACIÓN Y SISTEMAS

**Desarrollo de un Algoritmo de aprendizaje profundo para detectar rutas seguras de las
calles de los Ángeles**

Trabajo de investigación para el curso de Trabajo de Tesis I

Nombre Guillermo Díaz Beltrán

Asesor: Marks Calderón

Lima, 30 de junio de 2024

Índice general

Índice de Figuras	5
Índice de Tablas	8
1. PLANTEAMIENTO DEL PROBLEMA	9
1.1. Descripción de la Realidad Problemática	9
1.2. Formulación del Problema	11
1.2.1. Problema General	12
1.2.2. Problemas Específicos	12
1.3. Objetivos de la Investigación	12
1.3.1. Objetivo General	12
1.3.2. Objetivos Específicos	12
1.4. Justificación de la Investigación	13
1.4.1. Teórica	13
1.4.2. Práctica	13
1.4.3. Metodológica	14
1.5. Delimitación del Estudio	14
1.5.1. Espacial	14
1.5.2. Temporal	14
1.5.3. Conceptual	14

1.6. Hipótesis	15
1.6.1. Hipótesis General	15
1.6.2. Hipótesis Específicas	15
1.6.3. Matriz de Consistencia	15
2. MARCO TEÓRICO	16
2.1. Antecedentes de la investigación	16
2.1.1. Real-time deep reinforcement learning based vehicle navigation (Koh, Zhou, Fang, Yang, Yang, Guan & Ji, 2020)	16
2.1.2. SafeRoute: Learning to navigate streets safely in an urban environment (Levy, Xiong, Belding & Wang, 2020)	18
2.1.3. Predicting motor vehicle theft in Santiago de Chile using graph-convolutional LSTM (Esquivel, Nicolis & Márquez, 2020)	21
2.1.4. Algoritmo para calcular la ruta más segura y óptima (Areiza, Cano, Serna & Riascos, 2022)	24
2.1.5. Route-the safe: A robust model for safest route prediction using crime and accidental data (Soni, Gauri Shankar & Sandeep, 2019)	26
2.1.6. Real-time safest route identification: Examining the trade-off between safest and fastest routes (Ghoul, Sayed & Fu, 2023)	30
2.1.7. A reinforcement learning-based routing algorithm for large street networks. (<i>pride</i>)	32
2.1.8. An Enhanced Motif Graph Clustering-Based Deep Learning Approach for Traffic Forecasting (Zhang, Zhang, Yu & Yu, 2020)	34
2.2. Bases Teóricas	37
2.2.1. Inteligencia Artificial	37
2.2.2. Aprendizaje Automático	38
2.2.3. Aprendizaje Profundo	40
2.2.4. Aprendizaje Profundo por Refuerzo (DRL)	41
2.2.5. Fundamentos de grafos	45

2.2.6. Métodos de Redes Neuronales Profundas basadas en Grafos	48
2.3. Marco Conceptual	50
2.3.1. Aprendizaje Profundo	50
2.3.2. Aprendizaje Profundo (RL)	50
2.3.3. Aprendizaje Profundo por Refuerzo (DRL)	50
2.3.4. Redes Neuronales Profundas (DNN)	50
2.3.5. Graph Neural Network (GNN)	50
2.3.6. Zona Urbana	51
2.3.7. Delincuencia	51
2.3.8. Ruta	51
2.3.9. Riesgo	51
3. METODOLOGÍA DE LA INVESTIGACIÓN	52
3.1. Diseño de la investigación	52
3.1.1. Tipo de investigación	52
3.1.2. Enfoque de investigación	52
3.2. Población y muestra	53
3.3. Operacionalización de Variables	53
3.4. Técnicas de recolección de datos	53
3.5. Técnicas para el procesamiento y análisis de la información	55
3.5.1. Metodología de la implementación de la solución	57
3.5.2. Metodología para la medición de resultados	69
3.6. Cronograma de actividades y presupuesto	72
Anexos	74
A. Anexo I: Matriz de Consistencia	75

A.1. Árbol de Problemas	75
A.2. Árbol de Objetivos	77
B. Anexo II: Resumen de Papers investigados	78
BIBLIOGRAFÍA	80

Índice de Figuras

1.1. Índice de criminalidad de los países de América 2023	10
1.2. Días con más robos de celulares registrados en el 2023	11
2.1. Framework propuesto	17
2.2. Comparación de rendimiento de los modelos Fuente: Koh et al., 2020. «Real-time deep reinforcement learning based vehicle navigation». (p. 14)	18
2.3. Análisis de significancia de los modelas con la prueba de Wilcoxon Fuente: 2020. «Real-time deep reinforcement learning based vehicle navigation». (p. 14)	18
2.4. Elaboración propia: Metodología propuesta por los autores. Fuente: Levy et al., 2020. «SafeRoute: Learning to Navigate Streets Safely in an Urban Environment».	19
2.5. Estructura de la red neuronal del modelo propuesto Fuente: 2020. «SafeRoute: Learning to Navigate Streets Safely in an Urban Environment». (p. 5)	20
2.6. Comparación de rendimiento de los modelos Fuente: 2020. «SafeRoute: Learning to Navigate Streets Safely in an Urban Environment». (p. 11)	21
2.7. Elaboración propia: Metodología propuesta por los autores Fuente: (Esquivel, Nicolis & Márquez, 2020). «Predicting Motor Vehicle Theft in Santiago de Chile using Graph-Convolutional LSTM»	22
2.8. Estructura de la red convolucional LSTM basada en grafos Fuente: (2020). «Predicting Motor Vehicle Theft in Santiago de Chile using Graph-Convolutional LSTM». (p. 4)	23
2.9. Riesgo de acoso sexual mediante combinación lineal Fuente: (Areiza, Cano, Serna & Riascos, 2022). «ALGORITMO PARA CALCULAR LA RUTA MÁS SEGURA Y ÓPTIMA.» (p. 3)	25

2.10. Elaboración propia: Metodología propuesta por los autores Fuente: (2022). «ALGORITMO PARA CALCULAR LA RUTA MÁS SEGURA Y ÓPTIMA.»	26
2.11. Metodología Propuesta Fuente: (Soni, Gauri Shankar & Sandeep, 2019). «Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data». (p. 1419)	27
2.12. Flujo de condición para detectar rutas seguras Fuente: (2019). «Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data». (p. 1420)	29
2.13. Resultados obtenidos Fuente: (2019). «Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data». (p. 1426)	30
2.14. Elaboración propia: Metodología propuesta por los autores Fuente: (Ghoul, Sayed & Fu, 2023). «ALGORITMO PARA CALCULAR LA RUTA MÁS SEGURA Y ÓPTIMA.»	31
2.15. Framework Propuesto Fuente: (pr^{de}). pr^{de} . (p. 202)	33
2.16. Función de política del modelo Fuente: (pr^{de}). pr^{de} . (p. 199)	34
2.17. Puntuaciones de evaluación Fuente: (pr^{de}). pr^{de} . (p. 215)	34
2.18. Metodología EMGC-STGCN propuesta por los autores Fuente: (Zhang, Zhang, Yu & Yu, 2020). «An Enhanced Motif Graph Clustering-Based Deep Learning Approach for Traffic Forecasting». (p. 4)	35
2.19. Resultados de modelos propuestos con el conjunto de datos Nav-BJ Fuente: (2020). «An Enhanced Motif Graph Clustering-Based Deep Learning Approach for Traffic Forecasting». (p. 5)	36
2.20. Resultados de modelos propuestos con el conjunto de datos PeMS Fuente: (2020). «An Enhanced Motif Graph Clustering-Based Deep Learning Approach for Traffic Forecasting». (p. 5)	36
2.21. Representación del Aprendizaje por Refuerzo Fuente: (CEUPE, s.f.). «Aprendizaje por refuerzo: Concepto, características y ejemplo»	39
2.22. Proceso de decisión de Markov Fuente: (AcademiaLab, s.f.). «Proceso de decisión de markov»	40

2.23. Comparación entre una Red Neuronal Artificial y una Red Neuronal Profunda Fuente: (Shopping, s.f.). «What Is Deep Learning, Nature Of Machine Learning And Beauty Of Deep Neural Networks ? - New»	41
2.24. Componentes que conforma el Aprendizaje por Refuerzo Fuente: (Instituto de Ingeniería del Conocimiento, s.f.). «Aprendizaje profundo por refuerzo»	42
2.25. Diferencias entre Q-Learning y Deep Q-Learning Fuente: (DAMAVIS, 2022). «Aprendizaje por refuerzo profundo: DQN»	43
2.26. Propiedades u medidas de un grafo Fuente: (Ma & Tang, 2021). <i>Deep Learning on Graphs</i> . (p. 19)	45
2.27. Operación de filtración de grafos Fuente: (2021). <i>Deep Learning on Graphs</i> . (p. 108)	48
2.28. Operación de filtración de grafos Fuente: (2021). <i>Deep Learning on Graphs</i> . (p. 109)	49
3.1. Elaboración propia: Flujo de recolección de datos	55
3.2. Elaboración propia: Metodología propuesta	56
3.3. Zona de la ciudad de Los Ángeles en OpenStreetMap	61
3.4. Estructura simple de un grafo	64
3.5. Estructura de GCN	65
3.6. Experimentación de tasa de aprendizajes	67
3.7. Efectos de la frecuencia de actualización de la red objetivo en la tarea del labe- rinto MNIST	68
3.8. Elaboración propia: Flujo de condición de elección de ruta	70
3.9. Mapa de Tampa, Florida, creado en Folium y OpenStreetMap	71
3.10. Cronograma de actividades de la investigación	72

Índice de Tablas

2.1. Elaboración propia: Tabla de resultados. Fuente: (Areiza, Cano, Serna & Riascos, 2022). «ALGORITMO PARA CALCULAR LA RUTA MÁS SEGURA Y ÓPTIMA.»	26
3.1. Descripción de las columnas del conjunto de datos de LAPD Fuente: (ILAPD, s.f.). «Crime Data from 2020 to Present».	59
3.2. Presupuesto de los costos personales del autor	73
A.1. Matriz de consistencia. Fuente: Elaboración propia	76
B.1. Cuadro Resumen de Papers investigados. Fuente: Elaboración propia	79

Capítulo 1

PLANTEAMIENTO DEL PROBLEMA

1.1. Descripción de la Realidad Problemática

De acuerdo al análisis del crecimiento y expansión urbana del Centro Nacional de Planeamiento Estratégico (CEPLAN) del 2023, menciona que con respecto al censo 2017 del INEI, ha crecido las zonas urbanas a un 1.6% en comparación al censo de 2007, lo que quiere decir que la población urbana ha crecido en aproximadamente 3 millones de habitantes ([CEPLAN, 2023](#)). Este crecimiento de la urbanización también a traído consigo un aumento significativo a las acciones criminales que enfrenta Perú, y también en todo el mundo; como es el caso de California (Estados Unidos), ya que se revela que los crímenes violentos, robos y asaltos agravados ha aumentado en un 5.7% en el año 2023 ([Magnus Lofstrom, Brandon Martin, 2023](#)). Esto refleja una gran preocupación de la seguridad ciudadana en la que enfrenta la población y las autoridades que buscan tomar medidas cruciales para proteger a los residentes de los entornos urbanos.

El índice de criminalidad en Lima representa un 70.9 %, que se sitúa en el puesto número 17, por debajo de ciudades de Latinoamérica como Caracas (Venezuela), San Pedro Sula (Honduras) y Río de Janeiro (Brasil) que también enfrentan varios desafíos para reducir las acciones delictivas y mejorar la seguridad ciudadana ([Numbeo, 2023](#)). A pesar de eso, el índice de representatividad de crímenes en Lima no resta importancia a la clara necesidad de encontrar alternativas efectivas que permita reducir dicho índice de criminalidad en otras áreas urbanas del Perú.

Según datos recopilados en la plataforma de Statista entre Julio y diciembre del 2023, por cada 100 habitantes en las zonas urbanas de Perú, el 11.9 % de esta población ha sido víctimas por el robo de pertenencias personales, como dinero, carteras o celulares, siendo esta, la

Puesto	Ciudad	Índice de Criminalidad	Índice de Seguridad
1	Caracas, Venezuela	83,59	16,41
2	San Pedro Sula, Honduras	80,55	19,45
3	Río de Janeiro, Brasil	77,63	22,37
4	Fortaleza, Brasil	77,34	22,66
5	Natal, Brasil	77,18	22,82
6	Salvador de Bahía, Brasil	76,65	23,35
7	Recife, Brasil	76,36	23,64
8	Puerto España, Trinidad y Tobago	76,21	23,79
9	Baltimore, Maryland, Estados Unidos	75,49	24,51
10	Rosario, Argentina	75,18	24,82
11	Memphis, Tennessee, Estados Unidos	74,79	25,21
12	Detroit, Michigan, Estados Unidos	74,12	25,88
13	Porto Alegre, Brasil	72,39	27,61
14	Guayaquil, Ecuador	71,58	28,42
15	Albuquerque, Nuevo México, Estados Unidos	71,50	28,50
16	Tijuana, México	71,44	28,56
17	Lima, Perú	70,90	29,10
18	Kingston, Jamaica	70,90	29,10
19	Saint Louis, Missouri, Estados Unidos	70,64	29,36
20	Sao Paulo, Brasil	70,49	29,51
21	Cali, Colombia	70,23	29,77
22	Ciudad de México, México	68,49	31,51

Figura 1.1: Índice de criminalidad de los países de América 2023.Fuente: ([2023](#)). *América: Índice de Criminalidad 2023*.

más representativa y la que refleja la parte vulnerable de los ciudadanos al no hacer frente a los robos que se cometan en las calles. Si se toma en cuenta el tipo de acto delictivo que puede atentar a la vida humana en las calles, se tiene que, el robo de vehículos representa un 2 %, mientras que el secuestro y extorsión el 0.2 %; que, si bien son menos frecuentes, representan una amenaza al bienestar y seguridad de los ciudadanos de las zonas urbanas ([Statista Research Department, 2023](#)). El último informe del Organismo Supervisor de Inversión Privada en Telecomunicaciones (Osiptel) de 2023, se registraron un total de 1,706,643 casos de robo de celulares en el año 2023, el cual el mayor número de robo ha sido los días lunes ([Melo, 2024](#)). Asimismo, se informó que, entre el periodo de enero y mayo del año 2023, por cada hora, se roban en promedio, 200 celulares a nivel nacional ([Ramírez, 2023](#)). Estos resultados no solo representan una pérdida material y económica por parte de los afectados, genera desconfianza y preocupación sobre su seguridad personal y la posible vulneración en su privacidad de información que se contenía en el dispositivo del agraviado.

Con lo que respecta al robo de vehículos, que también atenta a la vida y seguridad de la población de las zonas urbanas, según la Asociación Automotriz del Perú (APP) en el año 2023, la venta de vehículos ha representado un aumento del 2.4 % (181,000 autos vendidos) con respecto al año 2022 ([Perú, 2024](#)). Este incremento en la venta de autos, también lleva

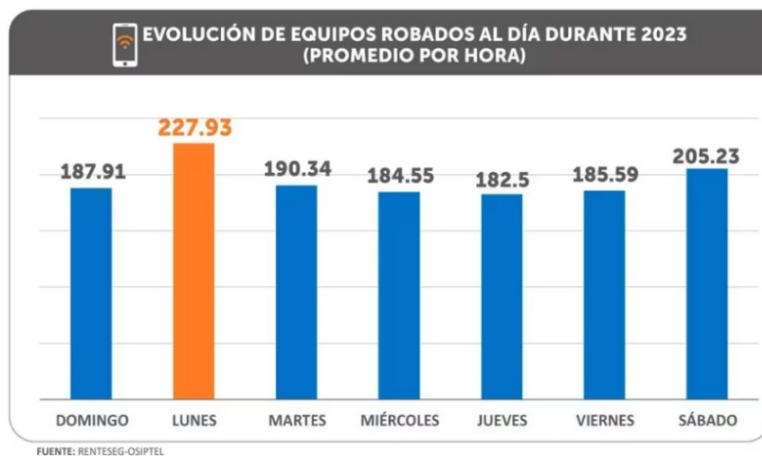


Figura 1.2: Días con más robos de celulares registrados en el 2023.

Fuente: ([2024](#)). *Este es el día que más robos de celulares se registraron en el 2023, según reporte de Osiptel.*

consigo el aumento a los robos de vehículos, que es un problema que se está volviendo cada vez más persistente en Perú, ya que, según la Superintendencia Nacional de Registros Públicos, en promedio se roban 46 autos al día y que en el primer semestre del año 2022 se ha registrado un total de 9,758 autos robados en el Perú ([Castillo, 2023](#)).

Estas cifras demuestran la necesidad de desarrollar herramientas y/o estrategias que permitan mitigar los riesgos relacionados con los actos delictivos en entornos rurales, donde se destaca el robo de celulares y vehículos, que, a su vez, pueden estar relacionadas entre sí. Las aplicaciones móviles para ayudar al usuario a elegir rutas más cortas para llegar a su destino en el menor tiempo posible mediante un motor de crowdsourcing (recopilación de tráficos), son cada vez más comunes y fundamentales de utilizar; ya que, en 2022, más de 150 millones de personas utilizan Waze; mientras que Google maps ha sido instalada por más de 10 mil millones de usuarios ([Hernández, 2022](#)). Sin embargo, estas herramientas de ayuda no tienen una funcionalidad que les permita detectar entre qué calles son seguras o no tomando en cuenta información o denuncias de delitos cometidos en esas zonas, ocasionando que los usuarios queden expuestos a situaciones peligrosas como el robo de su celular, de su auto o ambas.

1.2. Formulación del Problema

formulación de los problemas de la presente investigación, se elaboró un «árbol de problemas» (véase Anexo A.1).

1.2.1. Problema General

¿Es posible utilizar un algoritmo de aprendizaje profundo para detectar rutas seguras en las calles de los Ángeles?

1.2.2. Problemas Específicos

- ¿Qué algoritmo de aprendizaje profundo es el más representativos de los antecedentes?
- ¿Qué alternativas de procesamiento de datos se deben aplicar a la base de datos para entrenar el modelo?
- ¿Qué métricas propuestas en los antecedentes existen para ayudar en la eficiencia y eficacia del modelo propuesto?
- ¿El modelo propuesto evita las zonas con alto nivel de delincuencia?

1.3. Objetivos de la Investigación

Para la formulación de los objetivos de la presente investigación se elaboró un «árbol de objetivos» (véase Anexo A.2).

1.3.1. Objetivo General

Desarrollar un algoritmo de aprendizaje profundo para detectar rutas seguras en las calles de los Ángeles.

1.3.2. Objetivos Específicos

- Determinar qué algoritmo de aprendizaje profundo es el más representativo de cada antecedente para la detección de rutas seguras.
- Seleccionar una alternativa de procesamiento de datos que permita entrenar un modelo de detección de rutas seguras.
- Evaluar la factibilidad del modelo con las métricas propuestas en los antecedentes para mejorar el tiempo de procesamiento con los menores recursos.
- Demostrar que el modelo propuesto evita las zonas con alto nivel de delincuencia.

1.4. Justificación de la Investigación

1.4.1. Teórica

Esta investigación se realiza con el propósito de desarrollar un algoritmo que pueda realizar el procesamiento de direcciones y el cálculo de riesgo en los diferentes nodos que existen en un grafo a fin de poder generar métodos de compensación para llegar a un destino de manera segura y rápida. Además, dicho algoritmo será realizado bajo Aprendizaje profundo, para crear un modelo robusto de detección con menos tiempo de procesamiento y recursos para detectar rutas seguras que eviten zonas de alto nivel de crímenes registrados en una ciudad; asimismo, de ser un modelo escalable que no dependa de volver a pasar por el proceso de entrenamiento para cuando se quiera detectar una ruta distinta a otra ciudad.

Actualmente, existen investigaciones que utilizan modelos tradicionales como el algoritmo de Dijkstra, pero que carecen de escalabilidad y su complejidad computacional aumenta a medida que el tamaño de los datos y nuevos escenarios aumenten (Diya Lia y e.al, 2023). Por último, en el Perú, estos tipos de proyectos nos son muy comunes, por lo que el modelo propuesto puede aportar a las nuevas investigaciones en el uso de algoritmos de Aprendizaje Profundo utilizando grafos.

1.4.2. Práctica

Al culminar la investigación, se podrá utilizar el algoritmo propuesto, el cuál puede recibir por parte del usuario la dirección a la cuál quiere ir y la dirección en la que se encuentra, para mostrarle en un tiempo de ejecución menor la ruta más segura a la cuál puede ir. Además, podrá tener la capacidad de identificar rutas seguras a partir nuevos datos de crímenes registrados en distintas ciudades.

La presente investigación demostrará que un modelo de aprendizaje profundo puede detectar rutas seguras en grafos cada vez más complejos y ser escalables en comparación a los métodos tradicionales, mejorando de esta manera la calidad de vida de las personas, disminuir el índice de criminalidad, informar a los usuarios que priorizan su seguridad o que no tienen conocimiento de las calles de una ciudad a la que visitan a decidir el camino que debe tomar para sentir más seguro.

1.4.3. Metodológica

. La implementación de este algoritmo puede ayudar a las personas a conocer qué rutas son seguras para transportarse y evitar ser víctima de algún crimen, ya que, si bien se cuenta con aplicaciones de enrutamiento en tiempo real como Waze o Google Maps, estos no cuentan con la capacidad de detectar qué rutas son las más seguras, debido a que no trabajan con datos de registros de delitos. Asimismo, esta investigación puede aportar a la disminución del índice de criminalidad de alguna ciudad e informar a las personas qué rutas no deben tomar.

Para ello, se utilizaron técnicas de Aprendizaje Profundo aplicados en grafos que son entrenados con un conjunto de datos de crímenes registrados en las calles de una ciudad.

1.5. Delimitación del Estudio

1.5.1. Espacial

Para la presente investigación se realizará con los registros de actos delictivos recolectados y almacenados en la web oficial del Departamento de Policía de los Ángeles (APD) para uso público, la cual consiste en denuncias o actos registrados por personas, junto con el tipo de delincuencia y la dirección dónde ocurrió el hecho. Estos datos están registrados en formato “csv”. Además, se utilizará librerías que permitan cartografiar las calles de la ciudad de los Ángeles y poder utilizarlos como grafos.

1.5.2. Temporal

La presente investigación analizará un conjunto de datos de registros delictivos de la ya mencionada base de la web oficial del APD, que empieza desde el año 2020 hasta los últimos registros descargados a la fecha que se inicia este trabajo.

1.5.3. Conceptual

Esta investigación se centrará en el desarrollo de un algoritmo que logre detectar qué rutas de la ciudad de los Ángeles serán seguras para evitar algún caso de crimen basado en un indicador de riesgo. Para lograrlo, se necesitará herramientas de Aprendizaje Profundo y métodos de procesamiento de datos para desarrollar un modelo que se adapte a los diferentes estilos de grafos de forma óptima. El modelo desarrollado busca poder dar una contribución

a la comunidad con el fin de evitar situaciones peligrosas y apoyar a los distintos métodos de reducción en la tasa de criminalidad que se presenta en las zonas urbanas; asimismo aumentar la participación en técnicas de Aprendizaje Profundo utilizando grafos.

1.6. Hipótesis

1.6.1. Hipótesis General

Mediante el desarrollo de un algoritmo de aprendizaje profundo se logra detectar rutas seguras en las calles de los Ángeles.

1.6.2. Hipótesis Específicas

- La selección del algoritmo de aprendizaje profundo más representativos de cada antecedente mejora significativamente la precisión y eficiencia al detectar rutas seguras.
- La identificación de una alternativa de procesamiento de datos influye positivamente para poder entrenar un modelo de detección de rutas seguras.
- La utilización de métricas propuestas en los antecedentes es factible para optimizar el tiempo de procesamiento del modelo.
- Se demuestra que el modelo propuesto evita las zonas con alto nivel de delincuencia.

1.6.3. Matriz de Consistencia

A continuación se presenta la matriz de consistencia elaborada para la presente investigación (véase Anexo A.1).

Capítulo 2

MARCO TEÓRICO

2.1. Antecedentes de la investigación

En esta sección se presentarán diversos artículos de investigación o tesis las cuales abordarán diversas técnicas y enfoques que se emplearon para afrontar problemas similares al de esta tesis. Asimismo, a continuación se presenta un cuadro resumen (véase Anexo B.1) de lo que se presenta en esta sección.

2.1.1. Real-time deep reinforcement learning based vehicle navigation ([Koh, Zhou, Fang, Yang, Yang, Guan & Ji, 2020](#))

[Koh et al.](#) realizaron un artículo de investigación el cual fue publicado en la revista «Applied Soft Computing» en el año 2020. Este fue titulado «[Real-time deep reinforcement learning based vehicle navigation](#)» la cual traducida al español significa «Navegación de vehículos basada en aprendizaje de refuerzo profundo en tiempo real».

2.1.1.1. Planteamiento del Problema y objetivo

Los autores del siguiente artículo mencionan la gravedad del problema de la congestión vehicular en las ciudades contemporáneas, teniendo como consecuencia el alto consumo innecesario de energía, la contaminación y el tiempo invertido en viaje. Además, menciona la dificultad que tienen los algoritmos existentes para la optimización de las carreteras con el fin de afrontar dicho problema, siendo esta la complejidad que tienen los vehículos al interactuar en un entorno dinámico en tiempo real. Por lo que el objetivo de los autores es implementar

un modelo de aprendizaje por refuerzo profundo (DRL) con el fin de construir un sistema de navegación en tiempo real a través de una secuencia de decisiones.

2.1.1.2. Metodología empleada por los autores

Este artículo propone un framework el cual primero construye una simulación en SUMO, que contiene una interfaz de Control de Tráfico (TraCI) y que utiliza mapas urbanos reales en OpenStreetMap mediante el método NETCONVERT. Luego en la misma simulación de SUMO, se puede extraer y pre procesar los datos a través del método TraCI que recupera el número de vehículos, el tiempo de viaje esperado, y la posición actual y el destino.

La segunda parte del framework propuesto es el middleware, que conecta el entorno de SUMO con las redes neuronales DRL bajo un método mejorado de Deep Q-Learning Network (DQN) para mantener y actualizar las políticas de navegación y proporciona comandos para la navegación. El modelo propuesto contiene un esquema de exploración de dos etapas que buscan mejorar la convergencia de la red así como su velocidad. La primera etapa se utiliza la política convencional y la segunda etapa es un método basado en la distancia para reemplazar la selección de una ruta. La estructura de la red neuronal profunda contiene 5 capas conectadas, la primera capa de entrada contiene 150 neuronas, las dos capas ocultas tienen 100 neuronas bajo la función de activación RELU, una estructura de duelo y su capa de salida.

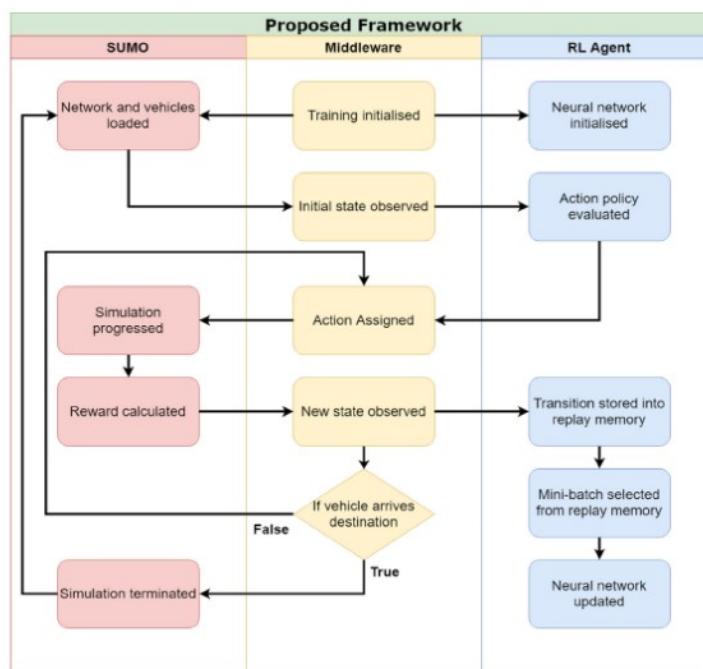


Figura 2.1: Framework propuesto

Fuente: 2020. «Real-time deep reinforcement learning based vehicle navigation». (p. 3)

2.1.1.3. Resultados obtenidos

Para demostrar la eficiencia del modelo propuesto, se compara con diferentes modelos por ciudad, los resultados de una ciudad demuestran que el modelo reduce como máximo el tiempo de viaje en 5.3 %, 5.1 % y 16.4 % dependiendo de la demanda del tráfico. Otro indicador que se utiliza para medir su eficacia es la prueba de Wilcoxon, demostrando que el modelo propuesto es superior a los demás modelos.

Methods	Map 1			Map 2			Map 3		
	20 vehicles	30 vehicles	50 vehicles	30 vehicles	50 vehicles	80 vehicles	50 vehicles	80 vehicles	120 vehicles
GDUE-Dijkstra	81.43 (±3.31)	81.78 (±3.61)	92.66 (±14.47)	109.48 (±5.74)	118.23 (±11.59)	131.26 (±19.03)	162.94 (±12.75)	180.04 (±16.72)	206.82 (±25.37)
GDUE-A*	82.48 (±4.34)	82.96 (±3.87)	95.54 (±13.82)	110.78 (±8.62)	119.14 (±11.76)	131.77 (±19.48)	163.86 (±13.06)	181.70 (±17.44)	208.51 (±28.26)
Dynamic-Dijkstra	79.26 (±4.94)	79.70 (±2.98)	90.84 (±11.99)	105.74 (±4.68)	110.96 (±8.82)	126.42 (±9.46)	148.98 (±11.50)	161.34 (±11.07)	194.32 (±9.80)
Dynamic-A*	81.25 (±6.40)	80.26 (±4.01)	91.18 (±13.36)	107.98 (±5.90)	111.36 (±8.62)	127.76 (±9.82)	149.16 (±11.65)	162.12 (±9.91)	195.96 (±8.92)
Ant Colony	79.56 (±4.79)	79.84 (±4.72)	90.98 (±12.64)	106.90 (±4.63)	113.36 (±11.82)	127.98 (±12.44)	151.08 (±9.07)	176.80 (±18.00)	205.12 (±30.97)
RL Agent	78.10 (±2.66)	78.72 (±4.19)	84.06 (±5.29)	103.44 (±2.48)	107.66 (±4.43)	117.00 (±6.85)	143.64 (±4.84)	147.82 (±5.60)	159.02 (±8.73)

Figura 2.2: Comparación de rendimiento de los modelos

Fuente: 2020. «Real-time deep reinforcement learning based vehicle navigation». (p. 14)

RL agent vs previous works	Map 1			Map 2			Map 3		
	20 vehicles	30 vehicles	50 vehicles	30 vehicles	50 vehicles	80 vehicles	50 vehicles	80 vehicles	120 vehicles
GDUE-Dijkstra	7.11e ⁻³	4.55e ⁻⁷	3.34e ⁻⁹	2.48e ⁻⁷	1.56e ⁻⁸	3.27e ⁻¹⁰	1.26e ⁻⁹	3.73e ⁻¹¹	7.58e ⁻¹¹
GDUE-A*	3.33e ⁻⁷	5.12e ⁻⁸	6.27e ⁻¹¹	4.65e ⁻⁷	6.77e ⁻⁸	3.55e ⁻¹⁰	2.31e ⁻⁹	4.12e ⁻¹¹	7.79e ⁻¹¹
Dynamic-Dijkstra	1.90e ⁻³	2.86e ⁻²	4.70e ⁻⁷	2.52e ⁻⁴	9.71e ⁻⁵	8.20e ⁻⁷	2.15e ⁻⁵	3.25e ⁻⁹	5.98e ⁻¹⁰
Dynamic-A*	7.21e ⁻⁷	3.90e ⁻⁷	2.63e ⁻⁸	1.23e ⁻⁶	5.74e ⁻⁸	1.02e ⁻⁸	2.21e ⁻⁵	7.06e ⁻¹⁰	6.86e ⁻¹⁰
Ant-Colony	1.87e ⁻³	2.30e ⁻³	3.24e ⁻⁷	2.71e ⁻⁵	4.53e ⁻⁸	3.24e ⁻⁸	2.01e ⁻⁷	8.24e ⁻¹⁰	6.99e ⁻¹¹

Figura 2.3: Análisis de significancia de los modelos con la prueba de Wilcoxon

Fuente: 2020. «Real-time deep reinforcement learning based vehicle navigation». (p. 14)

2.1.2. SafeRoute: Learning to navigate streets safely in an urban environment (Levy, Xiong, Belding & Wang, 2020)

Levy et al. realizaron un artículo de investigación el cual fue publicado en la revista «ACM Transactions on Intelligent Systems and Technology» en el año 2020. Este fue titulado «SafeRoute: Learning to Navigate Streets Safely in an Urban Environment» la cual traducida al español significa «SafeRoute: aprender a navegar por las calles de forma segura en un entorno urbano».

2.1.2.1. Planteamiento del Problema y objetivo

Ante los resultados de estudios realizados en la Universidad de Cornell y Hollaback en 2014, se muestra que el 85% de mujeres tomaron rutas diferentes para ir a su destino con el fin de evitar ser víctimas de acoso o agresión. También se menciona la necesidad de las personas, como los turistas, de contar con una aplicación que pueda detectar rutas seguras. Con esto en mente, se propone SafeRoute, que es una solución de estos problemas utilizando algoritmos de aprendizaje por refuerzo profundo, que tiene ventaja bajo los algoritmos clásicos como Dijkstra, ya que es una opción a problemas que requieren tomar decisiones incrementales por cada intersección que se presenta entre el origen y el destino, además que tiene la ventaja de adaptarse a nuevos datos sin requerir a ser entrenado nuevamente.

2.1.2.2. Metodología empleada por los autores

Se emplea información pública de delitos de la ciudad de Nueva York, San Francisco y Boston, y se filtra los tipos de delitos relacionados con acoso callejero y asalto, los cuales son tiroteo, asalto y robo. Asimismo, para la obtención cartográfica de dichas ciudades, se utilizó OpenStreetMap (OSM).

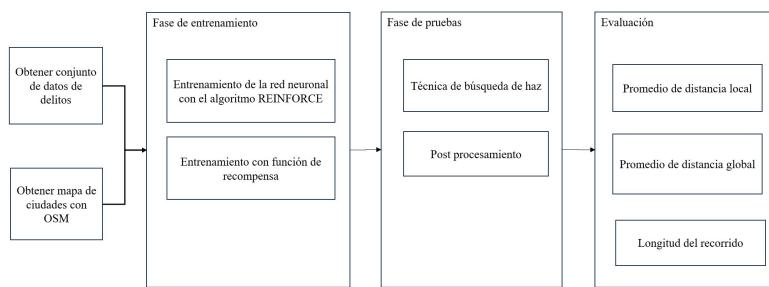


Figura 2.4: Elaboración propia: Metodología propuesta por los autores.

Fuente: [2020. «SafeRoute: Learning to Navigate Streets Safely in an Urban Environment»](#).

De acuerdo a la metodología propuesta por los autores, la arquitectura se divide en dos partes, la primera es donde interviene la red neuronal de aprendizaje por refuerzo, y el segundo es la red de políticas definidas para que la red neuronal utilice al momento de tomar decisiones. Este entorno se presenta como proceso de decisión de Markov con tuplas que contienen los estados continuos del mapa, las acciones disponibles para el algoritmo y la probabilidad de pasar de un estado a otro.

Se establece una función de recompensa para optimizar las múltiples preferencias del modelo, en este caso es evitar las áreas de crimen al crear una ruta. La definición de esta función

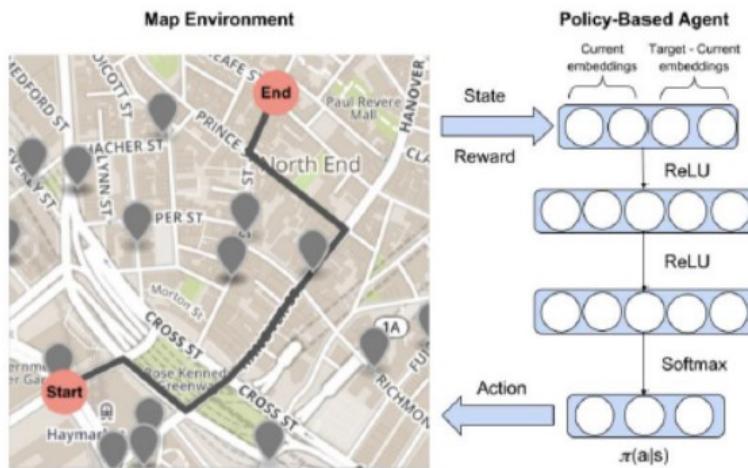


Figura 2.5: Estructura de la red neuronal del modelo propuesto

Fuente: 2020. «SafeRoute: Learning to Navigate Streets Safely in an Urban Environment». (p. 5)

se establece de la siguiente manera:

$$r_{CRIME} = \begin{cases} \frac{\sum_{i=1}^n \sum_{j=1}^m distance(x_i, c_j)}{\sum_{i=1}^n number(c_i)}, & \text{if } c \neq \emptyset \\ k, & \text{otherwise} \end{cases} \quad (\text{Ecuación 2.1})$$

Donde n es el número de bordes de cada camino, m es el número de delitos de cada nodo, x es la lista de puntos de cada camino, c es la lista de delitos de cada radio, p es el recorrido y k es un hiper parámetro.

En la fase de entrenamiento, consta de dos partes, la primera es el pre entrenamiento de la red neuronal para detectar rutas más cortas, en este proceso se utilizan técnicas de aprendizaje por refuerzo, específicamente el algoritmo Monte Carlo Policy Gradient (REINFORCE) para que actualice los parámetros de la red neuronal. La segunda es el entrenamiento con recompensas, donde se entrena nuevamente a la red neuronal considerando la seguridad en cada ruta, premiandolo por evitar áreas con alto índice de crimen, el cuál es la función de recompensa, y de esta manera actualizando la política de la red neuronal.

En la fase de pruebas, se utiliza una técnica de búsqueda de haz para encontrar varias rutas potenciales, luego se realiza un post procesamiento que evite bucles externos que se hayan creado durante la navegación, con el fin de que la ruta generada sea coherente y eficiente. Para la evaluación del modelo propuesto con respecto a otros modelos, se aplicaron métricas como el promedio de la distancia local, el promedio de la distancia global y la longitud del recorrido.

2.1.2.3. Resultados obtenidos

El modelo propuesto se comparó con el algoritmo de Dijkstra y SafePath, el cual crea produce caminos no dominados tomando en cuenta la seguridad. Se obtuvieron los mejores resultados con una tasa de aprendizaje de 0.0005 y realizando 5 saltos para un primer entrenamiento y luego 10 saltos.

City	Model	5-hops			10-hops		
		Local	Global	Length	Local	Global	Length
Boston	Dijkstra	0.0554	0.8620	0.2023	0.0484	0.8479	0.4200
	SafePath(Median)	0.0566	0.8648	0.2044	0.0498	0.8639	0.4372
	SafePath(Safest)	0.0568	0.8651	0.2050	0.04817	0.8768	0.4619
	SafeRoute	0.0630	0.8627	0.2361	0.0571	0.8510	0.4903
San Francisco	Dijkstra	0.0882	0.9268	0.4704	0.0964	0.8966	1.0121
	SafePath(Median)	0.0934	0.9423	0.4821	0.1046	0.9344	1.1063
	SafePath(Safest)	0.0937	0.9489	0.5001	0.1084	0.9650	1.2402
	SafeRoute	0.0990	0.9880	0.5468	0.1036	0.9611	1.1659
New York	Dijkstra	0.1344	1.1978	0.4222	0.1016	1.1842	0.7907
	SafePath(Median)	0.1341	1.2125	0.4515	0.0987	1.2267	0.8748
	SafePath(Safest)	0.1344	1.2242	0.4935	0.0978	1.2537	0.9731
	SafeRoute	0.1454	1.2004	0.47267	0.1179	1.1992	0.8915

Figura 2.6: Comparación de rendimiento de los modelos

Fuente: [2020. «SafeRoute: Learning to Navigate Streets Safely in an Urban Environment»](#). (p. 11)

Se observa que las distancias entre un ruta a la zona de crimen, el que presenta un valor más bajo es Boston, debido a su densidad de criminalidad, mientras que las rutas generadas en Nueva York son las que más se alejan. Sin embargo, se demuestra que tanto en la época 5 como la época 10, el modelo propuesto SafeRoute supera al modelo Dijkstra y los modelos de SafePath, a excepción de la ciudad de San Francisco, el cuál el número que más se aleja de los riesgos es el modelo más seguro de SafePath en la época 10.

2.1.3. Predicting motor vehicle theft in Santiago de Chile using graph-convolutional LSTM ([Esquivel, Nicolis & Márquez, 2020](#))

[Esquivel et al.](#) realizaron un artículo de investigación el cual fue presentado en la «39th International Conference of the Chilean Computer Science Society» en el año 2020. Este fue titulado «[Predicting Motor Vehicle Theft in Santiago de Chile using Graph-Convolutional LSTM](#)» la cual traducida al español significa «[Predicción del robo de vehículos de motor en Santiago de Chile utilizando LSTM gráfico convolucional](#)».

2.1.3.1. Planteamiento del Problema y objetivo

Ante el problema del incremento de autos robados en Chile en el año 2017, se menciona que este tipo de delitos, que, si bien es de bajo riesgo, puede desencadenar otros delitos más graves como el contrabando de armas o drogas, o crímenes internacionales. Es por esto que a través de un modelo de predicción de robo de vehículos puede contribuir a la mitigación y mayor seguridad por parte de la fuerza policial en las zonas con mayor probabilidad de robos. En este trabajo de investigación busca crear una nueva metodología para la predicción de robos de vehículos en la ciudad de Santiago de Chile desarrollando una red neuronal convolucional LSTM basada en grafos (GCLSTM) utilizando la técnica de preprocesamiento de datos, la cuál es la regresión LOESS.

2.1.3.2. Metodología empleada por los autores

Los autores proponen la siguiente metodología:

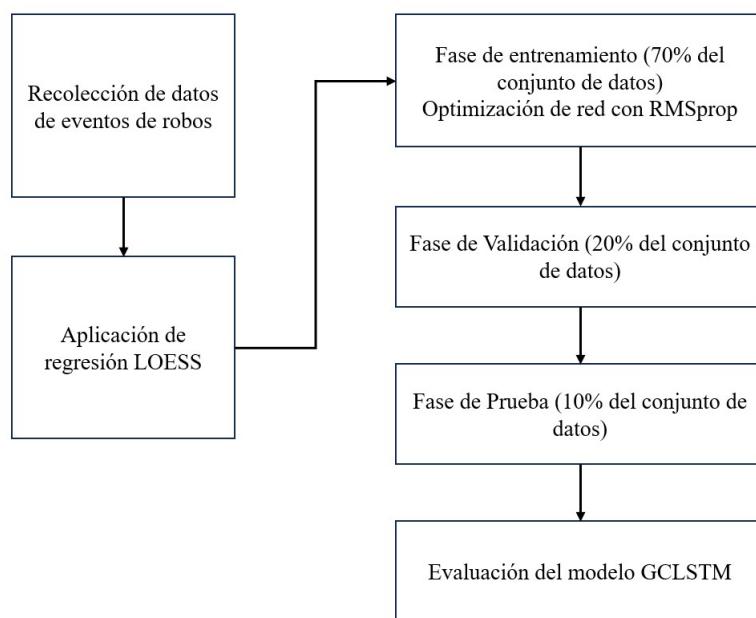


Figura 2.7: Elaboración propia: Metodología propuesta por los autores

Fuente: (2020). «Predicting Motor Vehicle Theft in Santiago de Chile using Graph-Convolutional LSTM»

El primer paso de esta metodología empieza con la recolección de los datos, que en este caso son los eventos de robos de vehículos en las comunas de la Región de Chile del año 2015 al 2019, que contiene un total de 129.127 registros. Luego se procedió a realizar la suma de los eventos de robo por día de cada comuna para la creación de una serie de tiempo y posterior a

esto, aplicar la regresión LOESS, el cual es una técnica no paramétrica que utiliza los mínimos cuadrados ponderados para asignar un mayor peso a los más cercanos y menos a los que están más lejos. La aplicación de dicha técnica permite la reducción de ruido de la serie de tiempo. De este paso, se obtuvo una matriz suavizada de número de eventos de robos, el cuál sirve como input para la red GCLSTM.

Dicha red suma las matrices de adyacencia e identidad para realizar la convolución con parámetros K , luego se le asigna a cada vecino un peso por iteración y estas van a las compuertas LSTM para la extracción de la última celda para el desarrollo de una capa de salida lineal con la función tanh. Para la optimización de la red GCLSTM, se aplicó el algoritmo de RMSprop para entrenar el modelo hasta que haya un sobreajuste en los datos.

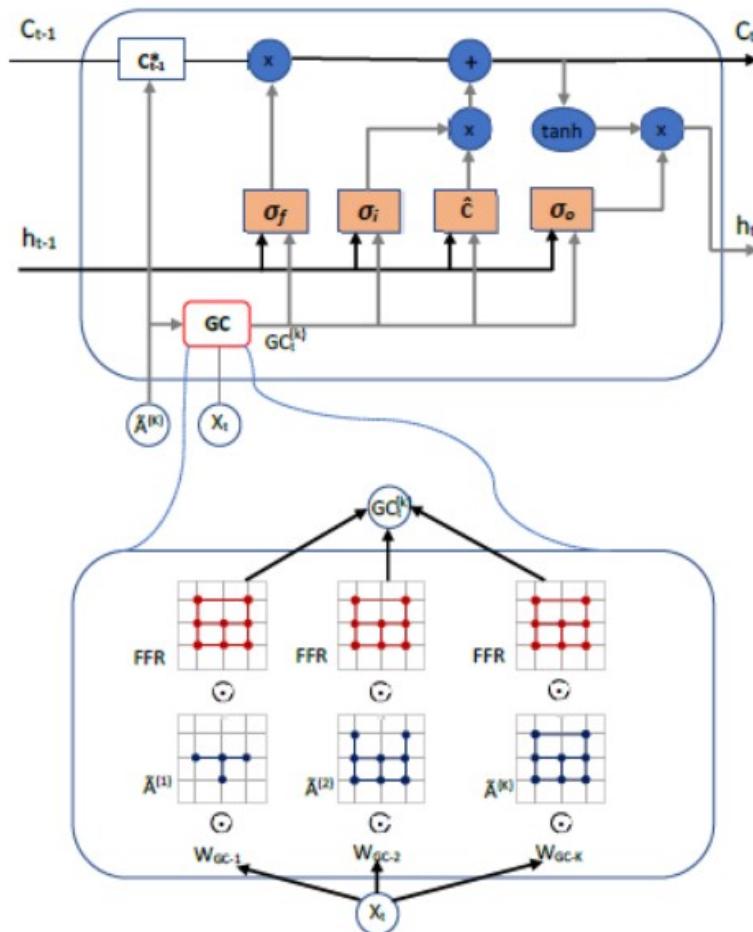


Figura 2.8: Estructura de la red convolucional LSTM basada en grafos

Fuente: (2020). «Predicting Motor Vehicle Theft in Santiago de Chile using Graph-Convolutional LSTM». (p. 4)

Para la fase de entrenamiento, se utilizó el 70 % del conjunto de datos, 20 % de validación y 10 % para la fase de prueba.

2.1.3.3. Resultados obtenidos

Como parte de los resultados, en la fase de entrenamiento, se entrenó en el modelo tradicional LSTM y el modelo GCLSTM, dando como resultado que a partir de la iteración 31, la curva *MSE* del modelo GCLSTM desciende más que el tradicional. Mientras que, en la fase de pruebas, se demuestra, con la métrica R^2 , que el modelo propuesto obtuvo 0.86, y el modelo LSTM tuvo 0.70, teniendo una mayor similitud la red GCLSTM entre lo esperado con lo observado. Por otra parte, el *MSE* del modelo propuesto fue 0.014 y el modelo LSTM es de 0.031; demostrando que el modelo puede obtener patrones temporales y espaciales para la predicción de crimen en distintas zonas de una ciudad.

2.1.4. Algoritmo para calcular la ruta más segura y óptima ([Areiza, Cano, Serna & Riascos, 2022](#))

[Areiza et al.](#) realizaron un artículo de investigación el cual fue publicado en el año 2022. Este fue titulado «[ALGORITMO PARA CALCULAR LA RUTA MÁS SEGURA Y ÓPTIMA.](#)»

2.1.4.1. Planteamiento del Problema y objetivo

La creciente cifra de casos de acoso callejero en las calles de Medellín, así como casos de asaltos y violación, ha tenido como consecuencia que el miedo de las mujeres aumente y por ende, se sientan inseguras por cuál ruta tomar. A pesar de los distintos métodos de preventión, se menciona que no siempre son efectivos para evitar ser víctima de acoso en espacios públicos. Además, aplicaciones de generación de rutas para el transporte, como Google Maps o Waze, no cuentan con la capacidad de detectar rutas seguras, ya que es propio sistema de estas sólo calculan sus rutas en base a la distancia más corta. Este trabajo de investigación busca solucionar este problema implementando el algoritmo de Dijkstra, para calcular tres caminos diferentes tomando en cuenta la distancia y el índice de riesgo.

2.1.4.2. Metodología empleada por los autores

Para la obtención del mapa de Medellín, se empleó Open Street Maps (OSM) en Python, el cual contiene en cada segmento su longitud en metros, en qué sentidos va, y sus representaciones binarias proporcionadas por la propia librería OSM. Para el cálculo del riesgo, se empleó un conjunto de datos de la encuesta de calidad de vida de Medellín del año 2017; luego se pro-

cedió a realizar una combinación lineal (*CL*) mediante el análisis de componentes principales, el cual es la máxima varianza de la proporción de hogares que se sienten inseguros entre la proporción de hogares que tienen ingresos inferiores al salario mínimo; después se normaliza la combinación lineal y la proporción de esta es considerada como el riesgo de acoso.

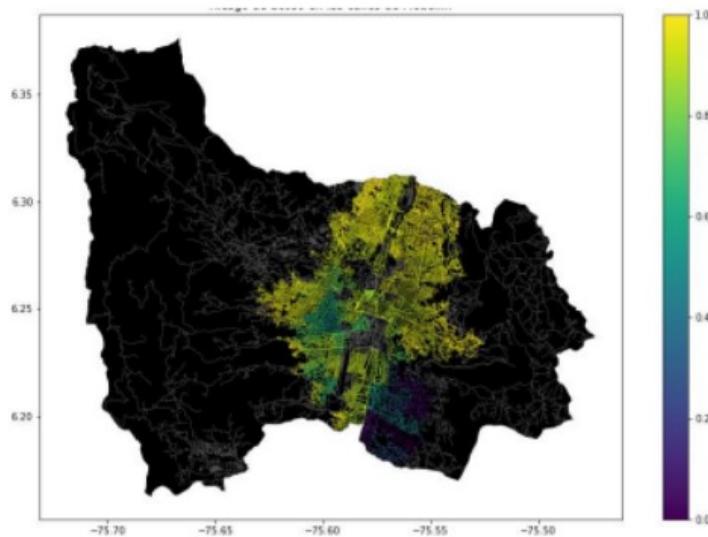


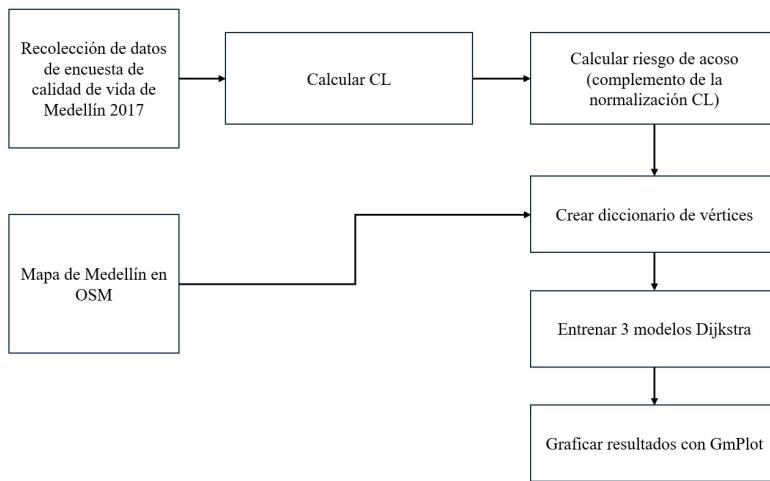
Figura 2.9: Riesgo de acoso sexual mediante combinación lineal

Fuente: (2022). «[ALGORITMO PARA CALCULAR LA RUTA MÁS SEGURA Y ÓPTIMA.](#)» (p. 3)

Se crea un diccionario que almacena el peso de cada vértice mediante el producto entre la longitud y riesgo de acoso de cada relación entre origen y destino. Se implementa el algoritmo de Dijkstra que requiere de tres parámetros los cuales son los grafos creados, y los vértices de origen y destino. Este algoritmo calcula cada vecino del vértice actual a través de sus pesos que están almacenados en el diccionario para calcular el camino con menos peso. Además del cálculo tradicional que emplea, el cuál es $r * d$, donde r es el riesgo de acoso, y d es la distancia en metros, se calculó dos caminos más. El primero es la suma entre la distancia y el riesgo, mientras que el segundo eleva la distancia al riesgo de acoso. Para graficar estos tres resultados, se empleó la librería GmPlot, une el camino del origen y destino mediante las coordenadas ubicadas en Google maps, siendo más fácil de visualizar para el usuario.

2.1.4.3. Resultados obtenidos

Se presenta los resultados que son obtenidos por el algoritmo, estableciendo el origen en la Universidad EAFIT hasta la Universidad Nacional, el cual se puede visualizar en la tabla 2.1.

**Figura 2.10:** Elaboración propia: Metodología propuesta por los autoresFuente: (2022). «[ALGORITMO PARA CALCULAR LA RUTA MÁS SEGURA Y ÓPTIMA.](#)»

Camino	Distancia	Riesgo	Tiempo de ejecución
r^*d	9061.75m	0.58	1.01033s
$r+d$	8574m	0.69	1.01030s
d^r	16642m	0.35	1.00640s

Tabla 2.1: Elaboración propia: Tabla de resultados.Fuente: (2022). «[ALGORITMO PARA CALCULAR LA RUTA MÁS SEGURA Y ÓPTIMA.](#)»

Se observa que en términos de evitar el mayor riesgo, la mejor opción es el tercer camino d^r ; sin embargo, la distancia que puede recorrer una persona es amplia a comparación de los otros caminos; asimismo, la opción más balanceada en cuanto al índice de riesgo y la distancia recorrida es el primer camino ($r * d$). También se demuestra que el tiempo de ejecución es bastante bajo, por lo que abre la oportunidad de ser empleado en una aplicación móvil para el fácil uso de los usuarios que lo emplearán en su vida cotidiana.

2.1.5. Route-the safe: A robust model for safest route prediction using crime and accidental data ([Soni, Gauri Shankar & Sandeep, 2019](#))

[Soni et al.](#) realizaron un artículo de investigación el cual fue publicado la entrevista en el año 2019. Este fue titulado «[Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data](#)» la cual traducida al español significa «Ruta segura: un modelo sólido para la predicción de rutas más seguras utilizando datos sobre delitos y accidentes».

2.1.5.1. Planteamiento del Problema y objetivo

La falta de conocimiento de la zona de una ciudad para un turista, así como invertir su tiempo en investigar sobre esta, puede crear una dependencia a sus conductores o pueden verse expuestas a ciertos peligros, como el robo de sus pertenencias. Asimismo, también se menciona la falta de capacidad que tienen las aplicaciones de rutas al momento de elegir un camino seguro. Con el objetivo de ser una herramienta de ayuda a la hora de elegir la ruta más segura para viajar, este proyecto utiliza algoritmos de aprendizaje automático para la predicción de rutas seguras mediante el cálculo de puntuación de riesgo de cada ruta de las ciudad de Nueva York utilizando datos sobre delitos y accidentes.

2.1.5.2. Metodología empleada por los autores

Este trabajo de investigación presenta la siguiente metodología utilizada para la implementación de su modelo:

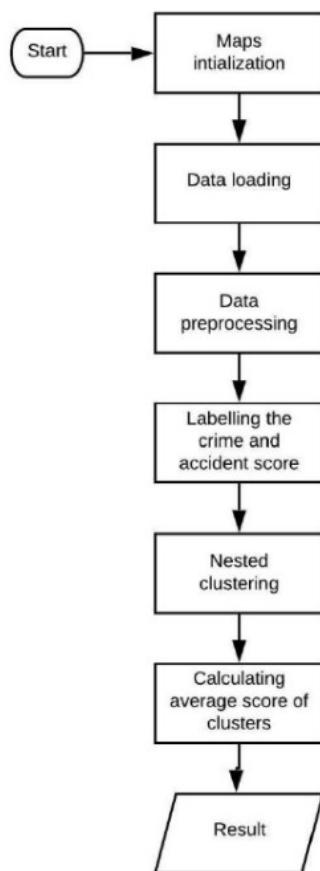


Figura 2.11: Metodología Propuesta

Fuente: (2019). «Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data». (p. 1419)

Como primer paso de la metodología propuesta, se inicializa el mapa con el API de Google Maps, para luego proceder a configurar su función gmaps. Luego se cargan los datos, los cuales contienen datos de delitos y accidentes desde el sitio web de datos abiertos de Nueva York, que brindan información del tipo de delito o accidente, así como su latitud y longitud. El siguiente paso se aplica al preprocesamiento de los datos para limpiar los datos atípicos o faltantes. Una vez realizado el preprocesamiento de los datos, se procede a etiquetar la puntuación de accidentes de un punto (AS) y la puntuación de delitos de un punto (CS). Para la puntuación de delitos se asigna ponderaciones que van desde 1 a 15 dependiendo del delito y tipo de castigo que se le impone al sospechoso. Mientras que la puntuación de accidentes se calcula de la siguiente manera:

$$AS = PK * 2 + CK * 2 + MK * 2 + PI + CI + MI \quad (\text{Ecuación 2.2})$$

Donde:

PK : Recuento de peatones fallecidos

CK : Recuento de ciclistas fallecidos

MK : Recuento de automovilistas fallecidos

PI : Recuento de peatones heridos

CI : Recuento de ciclistas heridos

MI : Recuento de automovilistas heridos

El siguiente paso es la formación de los grupos utilizando K Means, que es un algoritmo de agrupación de aprendizaje automático no supervisado, basado en la latitud y longitud, así como los conjuntos de datos se concatenan para agruparlos y determinar las regiones de riesgo. Específicamente, primero se agrupa en función a la longitud y latitud donde ocurrieron los hechos, para dividir el mapa de Nueva York en regiones más pequeñas. Asimismo, para generar las regiones de riesgo, se agrupa bajo los grupos ya formados en un distrito, como por ejemplo, el distrito de Manhattan.

Después de hacer las agrupaciones de los grupos formados de un distrito, se realiza la puntuación promedio de cada agrupación, los cuales se calculan de la siguiente manera:

$$C = \sum_{i=1}^N \frac{C_S}{N} \quad (\text{Ecuación 2.3})$$

$$A = \sum_{i=1}^N \frac{A_S}{N} \quad (\text{Ecuación 2.4})$$

Donde N es el número de puntos de un grupo. Después se procede a calcular la puntuación del riesgo (RS) que toma como entrada el origen y destino de la ruta a calcular gracias al API de google maps llamado Direction Service, que proporciona una serie de pasos al usuario para seguir la ruta. Luego se utiliza el modelo KNN Regresor para hallar el vecino más cercano según el grado de riesgo de cada punto. La puntuación de riesgo (RS) se calcula de la siguiente manera:

$$RS = \sum_{i=1}^W C_S + \sum_{i=1}^W A_S \quad (\text{Ecuación 2.5})$$

Donde W es el número de puntos de rutas que hay entre el origen y el destino.

Como último punto, el modelo debe mostrar la ruta más segura bajo las siguientes dos condiciones: el primero es que la ruta debe presentar la puntuación de riesgo más baja, y la segunda es que si hay más de una ruta segura, se debe elegir la distancia más corta.

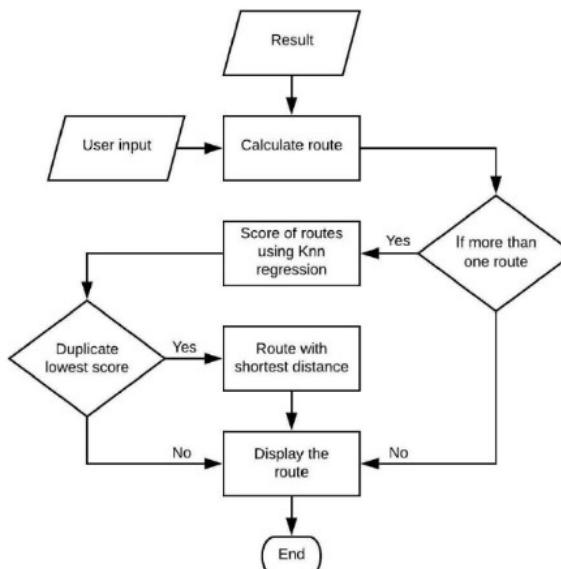


Figura 2.12: Flujo de condición para detectar rutas seguras

Fuente: (2019). «Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data». (p. 1420)

2.1.5.3. Resultados obtenidos

Se establece un punto de origen A y un punto de destino B en el distrito de Manhattan, dando como resultado que bajo el modelo de KNN Regressor, el R cuadrado de la puntuación de accidentes sea de 0.91 y la puntuación de criminalidad sea de 0.974; concluyendo que el modelo se ajusta mejor a sus datos.



Figura 6. Ruta sugerida por google maps



Figura 7. Ruta sugerida a través del modelo propuesto

Figura 2.13: Resultados obtenidos

Fuente: (2019). «Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data». (p. 1426)

2.1.6. Real-time safest route identification: Examining the trade-off between safest and fastest routes (Ghoul, Sayed & Fu, 2023)

Ghoul et al. realizaron un artículo de investigación el cual fue publicado en la revista «Analytic methods in accident research» en el año 2023. Este fue titulado «Real-time safest route identification: Examining the trade-off between safest and fastest routes» la cual traducida al español significa «Identificación de la ruta más segura en tiempo real: examen de la relación entre las rutas más seguras y las más rápidas»..

2.1.6.1. Planteamiento del Problema y objetivo

Este trabajo de investigación propone un nuevo enfoque de predicción de rutas seguras debido a la creciente congestión vehicular y accidentes, esto con el fin de mejorar el flujo vehicular de un determinado sector utilizando datos en tiempo real para que se sincronice con el modelo y proporcionar avisos para que el usuario evite regiones congestionadas o que están expuestos a peligros en las carreteras. Tiene como objetivo desarrollar un algoritmo de enruteamiento en tiempo real bajo una métrica propuesta que combina el riesgo de accidentes con el tiempo de viaje a lo largo de diferentes puntos de una ruta.

2.1.6.2. Metodología empleada por los autores

Se utiliza un conjunto de datos de código abierto llamado pNEUMA, de la ciudad de Atenas, Grecia. Este conjunto de datos utiliza drones para recopilar los datos sobre el tráfico en horas pico, asimismo, bajo esta recopilación, se extrae los datos de la posición, velocidad, aceleración y tipo de vehículo. Luego se pre procesaron los datos mediante la comparación por pares de trayectorias. Después se calculó el tiempo de coalición (TTC) de cada ubicación para aplicarlo al modelo propuesto.

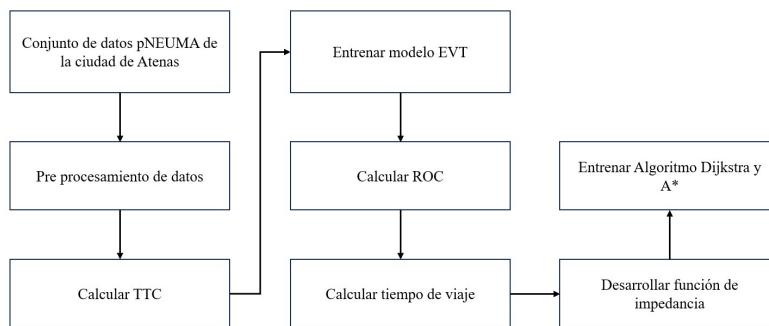


Figura 2.14: Elaboración propia: Metodología propuesta por los autores

Fuente: (2023). «[ALGORITMO PARA CALCULAR LA RUTA MÁS SEGURA Y ÓPTIMA.](#)»

El modelo propuesto es un modelo de teoría de valores extremos (EVT), específicamente el modelo bayesiano jerárquico para estimar el riesgo de accidente de cada ubicación en tiempo real y se incorporan covariables para optimizar el rendimiento del modelo propuesto. Después de desarrollar el modelo bayesiano, se calcula el riesgo de colisión (ROC), que estima el número total de accidentes que se observan en una ubicación, también se calcula una métrica de tiempo de viaje para desarrollar una función de impedancia para realizar comparaciones entre los distintos nodos de una red de carreteras bajo un algoritmo de enrutamiento. Para el cálculo de la función de impedancia (S) es de la siguiente manera:

$$S_t = \sum_{i=1}^n ROC_{i,t} \times T\bar{T}_{i,t} \quad (\text{Ecuación 2.6})$$

Posterior al cálculo de la función de impedancia, se utiliza en el algoritmo Dijkstra o el algoritmo A*. Despues se desarrolla un algoritmo que elige la mejor ruta, el cual identifica distintas rutas según la preferencia del usuario, tomando en cuenta tanto la seguridad como la movilidad.

2.1.6.3. Resultados obtenidos

Bajo la metodología propuesta, se encontró que el 23 % de las rutas más rápidas eran similares a las rutas más seguras. Además, la ruta más segura comparte en un 54 % los mismos enlaces que la ruta más rápida, existiendo un equilibrio entre seguridad y movilidad en distintos escenarios. Este trabajo de investigación aporta con un modelo de recomendación de rutas que presenta una buena eficacia en cuanto a cambios dinámicos en tiempo real en diferentes tipos de condiciones; sin embargo, para el cálculo de riesgo, depende de una percepción de seguridad más que la seguridad real, por lo que se recomienda que se establezca bajo su propia preferencia y prioridades, ya que es difícil cuantificar el riesgo real de un sector.

2.1.7. A reinforcement learning-based routing algorithm for large street networks. (pr·de)

pr·de realizaron un artículo de investigación el cual fue publicado en la « International Journal of Geographical Information Science» en el año 2024. Este fue titulado pr·de la cual traducida al español significa «Un algoritmo de enrutamiento basado en aprendizaje de refuerzo para grandes redes de calles».

2.1.7.1. Planteamiento del Problema y objetivo

Ante la problemática de contar con sistemas de evacuación y rutas de emergencia para minimizar los daños ocasionados durante los desastres naturales, los algoritmos tradicionales de enrutamiento cuentan con problemas de escalavidad cuando estas se enfrentan a evacuaciones a gran escala que involucran a diferentes rutas, asimismo, la complejidad computacional de estos algoritmos puede aumentar a medida que crece y se vuelve más complejo un escenario de evacuación, llevando a tener largos tiempos de cálculos, además que trabajan con pocos parámetros que capturan la complejidad y dinamismo de las emergencias por inundaciones. Ante estas limitaciones y con las contribuciones recientes de algoritmos de aprendizaje por refuerzo profundo (DRL), se desarrolló un algoritmo de aprendizaje por refuerzo para mejorar la eficiencia de enrutamiento en una gran red de carreteras en tiempo real, para lograr dicho objetivo, se entrenó dicho algoritmo en una supercomputadora FASTER, que maneja grandes redes de carreteras de una base de datos espacial.

2.1.7.2. Metodología empleada por los autores

Se recopilan los datos de redes de transporte de la ciudad de Nueva York y Houston, que contienen la longitud y latitud de cada punto de interés, asimismo para los datos de transporte se exporta información cartográfica del área central de Houston y la ciudad de Nueva York.

Para el algoritmo de aprendizaje por refuerzo, el sistema de enrutamiento puede modelarse como una tupla de proceso de decisión de Markov (MDP). Cada tupla contiene el conjunto de estados de cada vértice, conjunto de acciones que considera la red neuronal durante el entrenamiento, la función de recompensa para optimizar la preferencia del usuario, donde se incorpora el factor de seguridad. Además, se establece políticas con el método de gradientes de políticas, el cual se conecta con el algoritmo de ascenso de gradiente estocástico. Para disminuir la varianza del estimado de gradientes del algoritmo, se utiliza la Optimización de Política Proximal (PPO).

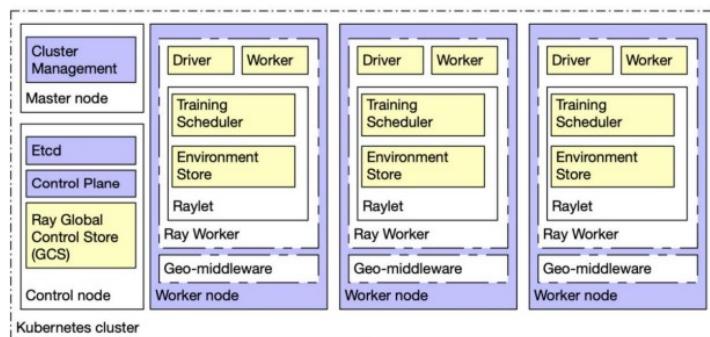


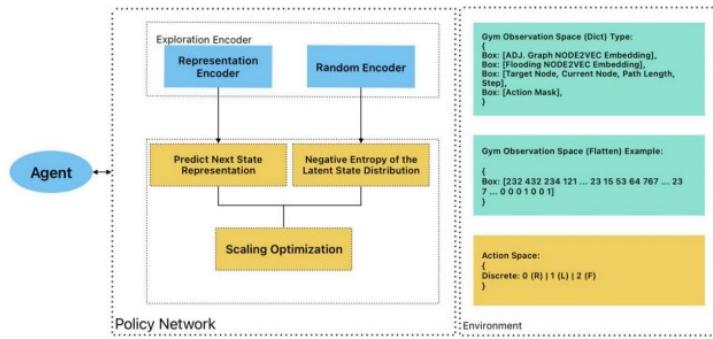
Figura 2.15: Framework Propuesto

Fuente: (pr'de). pr'de. (p. 202)

Para hacer frente a los entornos complejos y dinámicos, se requiere de una red neuronal que memorice los nodos y elija una acción válida a través de Q-learning, este algoritmo evita no seleccionar rutas inválidas con la red de políticas.

2.1.7.3. Resultados obtenidos

Se comparó con el método de Dijkstra y el algoritmo clásico de aprendizaje por refuerzo mediante puntuación por métricas de seguridad y recompensa de cada ciudad, como también su tiempo. Los resultados indican que el mejor algoritmo es el modelo propuesto, ya que tienen un mayor puntaje de seguridad, así como también un mayor indicador de recompensa. Sin embargo, tiene una mayor duración al ejecutar cada enrutamiento o detectar rutas seguras. También se destaca el hecho de que el modelo propuesto no atraviesa los puntos inundados, superando en

**Figura 2.16:** Función de política del modelo

Fuente: (pr'de). pr'de. (p. 199)

términos de seguridad a los algoritmos tradicionales.

Method	Environment	Metric.	Score
Dijkstra	New York City (Large)	Safety	1450.23
ReinforceRouting	New York City (Large)	Safety	1663.01
Dijkstra	Houston (Large)	Safety	1798.06
ReinforceRouting	Houston (Large)	Safety	1963.90
Vanilla-RL	Houston (Small)	Reward	-0.54
RE-RL	Houston (Small)	Reward	0.71
Vanilla-RL	Houston (Small)	Episode Length	20.5
RE-RL	Houston (Small)	Episode Length	87.7
RE-RL	Houston (Large)	Reward	0.12
ReinforceRouting	Houston (Large)	Reward	0.62
Vanilla-RL	Houston (Large)	Episode Length	20.5
ReinforceRouting	Houston (Large)	Episode Length	87.7

Figura 2.17: Puntuaciones de evaluación

Fuente: (pr'de). pr'de. (p. 215)

2.1.8. An Enhanced Motif Graph Clustering-Based Deep Learning Approach for Traffic Forecasting (Zhang, Zhang, Yu & Yu, 2020)

Zhang et al. realizaron un artículo de investigación el cual fue publicado en el « GLO-BECOM 2020 » en el año 2020. Este fue titulado «An Enhanced Motif Graph Clustering-Based Deep Learning Approach for Traffic Forecasting» la cual traducida al español significa «Un enfoque de aprendizaje profundo basado en agrupamiento de gráficos de motivos mejorado para la previsión del tráfico».

2.1.8.1. Planteamiento del Problema y objetivo

Por la gran complejidad que existe para el logro de una predicción precisa, como es el gran tamaño y condiciones de redes de tráfico, los métodos de Aprendizaje Profundo basados en grafos ha sido una alternativa de solución en tendencia. Además, mientras más complejo sea

las redes de tráfico, el rendimiento de los modelos de Graph Convolution Network (GCN) son cada vez más bajos. Para afrontar estos problemas, los autores del presente artículo proponen un enfoque basado en agrupamiento de grafos para aplicar un modelo de Spatio-Temporal Graph Convolutional Networks (STGCN) para la predicción de la velocidad de tráfico en lugares urbanos a gran escala.

2.1.8.2. Metodología empleada por los autores

La metodología propuesta empieza utilizando dos conjuntos de datos, el primer conjunto de datos pertenece a Nav-BJ y contiene datos de 1159 estaciones de sensores desplegadas en las diferentes carreteras de la ciudad de Beijing del mes de marzo de 2019, mientras que el segundo conjunto de datos pertenece a Caltrans Performance Measurement System (PeMS) que contiene 228 estaciones de sensores en el distrito 7 de California de los meses de mayo y junio de 2012. Luego se realiza un preprocesamiento de los datos, se realiza una normalización $Z - Score$ y se aplica una interpolación lineal para la recuperación de datos faltantes.

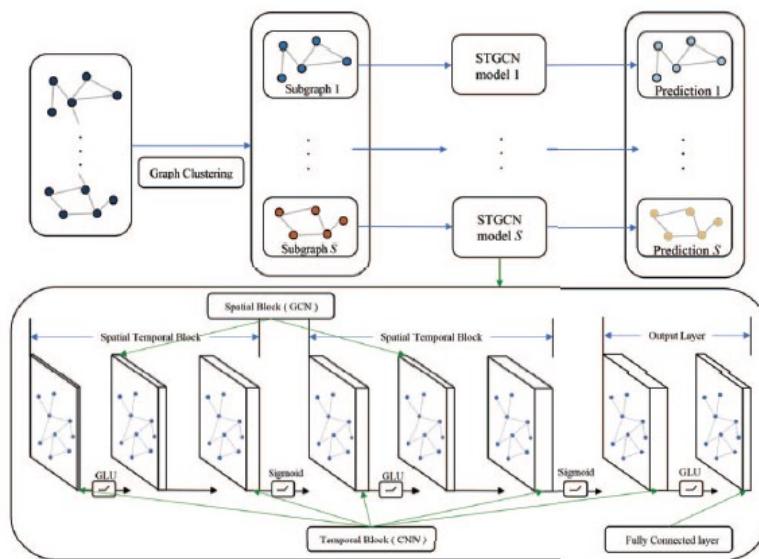


Figura 2.18: Metodología EMGC-STGCN propuesta por los autores

Fuente: (2020). «[An Enhanced Motif Graph Clustering-Based Deep Learning Approach for Traffic Forecasting](#)». (p. 4)

Se utilizan otros tres enfoques del modelo propuesto EMGC-STGCN (agrupación de grafos), el primero es el modelo STGCN sin agrupación de grafos, el segundo es Random-STGCN, el cual utiliza el método de agrupación de grafos aleatorios, mientras que el último enfoque es Metis-STGCN que utiliza el método ingenuo de Metis. Para el entrenamiento del modelo propuesto, se necesita como parámetro las matrices de adyacencia de los dos conjuntos

de datos y que representa el 60% de estas; además, se utiliza el optimizador Adam durante 50 épocas. Luego se realiza el proceso de prueba del 20% del conjunto de datos y por último el proceso de validación que representa el 20%.

Se compara el modelo propuesto de los 3 enfoques distintos con los métodos de Promedio Histórico (HA) y ARIMA, así como el modelo GRU. Para esto, se hace uso de las métricas de Error Absoluto Medio (*MAE*), Error Porcentual Absoluto Medio (*MAPE*) y Raíz del Error Cuadrático Medio (*RMSE*).

2.1.8.3. Resultados obtenidos

Model	15 min			30 min			45 min		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
HA	7.54	4.19	12.28	7.54	4.19	12.28	7.54	4.19	12.28
ARIMA	8.03	4.39	10.88	12.21	5.91	12.66	19.07	8.85	18.56
GRU	7.34	4.02	13.56	7.61	4.23	14.32	7.56	4.32	14.67
STGCN	4.53	3.10	11.13	4.84	3.28	11.84	5.02	3.40	12.30
Random-STGCN	4.23	2.89	9.90	4.48	3.05	10.60	4.68	3.19	11.36
Metis-STGCN	4.20	2.84	9.72	4.41	3.02	10.38	4.56	3.14	10.97
EMGC-STGCN	4.11	2.81	9.52	4.33	2.99	10.26	4.46	3.08	10.68

Figura 2.19: Resultados de modelos propuestos con el conjunto de datos Nav-BJ

Fuente: (2020). «An Enhanced Motif Graph Clustering-Based Deep Learning Approach for Traffic Forecasting». (p. 5)

Mediante la comparación de los distintos módulos y métodos, se evidencia que el modelo propuesto es mejor que sus distintos enfoques, como también los métodos HA, ARIMA y el modelo GRU.

Model	15 min			30 min			45 min		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
HA	7.20	4.01	10.61	7.20	4.01	10.61	7.20	4.01	10.61
ARIMA	9.00	5.55	12.92	9.13	5.86	13.94	9.38	6.27	15.20
GRU	4.15	2.35	7.25	5.36	3.04	9.12	6.19	3.52	10.14
STGCN	3.55	2.02	4.82	4.91	2.85	7.10	5.45	3.14	7.67
Random-STGCN	3.52	2.06	4.74	4.73	2.67	6.51	5.37	3.04	7.52
Metis-STGCN	3.49	1.99	4.67	4.59	2.58	6.19	5.34	3.03	7.45
EMGC-STGCN	3.47	1.98	4.56	4.62	2.58	6.20	5.40	3.03	7.42

Figura 2.20: Resultados de modelos propuestos con el conjunto de datos PeMS

Fuente: (2020). «An Enhanced Motif Graph Clustering-Based Deep Learning Approach for Traffic Forecasting». (p. 5)

2.2. Bases Teóricas

2.2.1. Inteligencia Artificial

La Inteligencia Artificial es el campo de la ciencia informática basada en máquinas que buscan replicar la inteligencia humana, implementando dotes como el razonamiento, aprendizaje y actuar que comúnmente podría hacer un humano ([Tableu, s.f.](#)). La Inteligencia Artificial va más allá que solo la comprensión, sino también el esfuerzo que conlleva construir entidades inteligentes; esto lleva a que su participación en la actualidad esté presente en una gran cantidad de campos, como el aprendizaje y la percepción, como también a campos más específicos como lo es el ajedrez ([Russell & Norvig, 2004](#)).

2.2.1.1. Enfoques de Inteligencia Artificial ([2004](#))

Russell también menciona que la definición de lo que es la Inteligencia Artificial sigue cuatro enfoques, en donde el sistema puede pensar y actuar de forma racional, como también de forma humana:

- **Actuar como humano:** Alan Turing sugiere una prueba llamada la Prueba de Turing el cual intenta demostrar las pocas diferencias que hay entre un agente inteligente y los seres humanos, ya que menciona que el computador necesita cumplir con seis disciplinas, las cuales son el procesamiento del lenguaje natural, la representación del conocimiento, un razonamiento automático que le permita extraer sus propias conclusiones y un aprendizaje automático que le pueda permitir adaptarse y detectar patrones; además, debe estar dotado con visión computacional y robótica.
- **Pensar como humano:** Considerado como el enfoque del modelo cognitivo, sugiere que el programa debe contar con un mecanismo necesario que le permita determinar cómo piensa un humano, por lo que es necesario penetrar el funcionamiento de dicha mente humana. Para lograr dicha penetración es a través de dos formas, la primera es capturando todos los pensamientos a medida que van apareciendo y la segunda es media una experimentación psicológica.
- **Actuar racionalmente:** Se espera que un agente racional actúe acorde al mejor resultado, o si en caso de que haya incertidumbre, elija la opción que proporcione el mejor resultado esperado. La Prueba de Turing puede permitir realizar dichas acciones racionales.
- **Pensar racionalmente:** Trata de construir sistemas inteligentes que sigan una lógica, el

cuál es un esquema de estructuras de argumentación que le permita llegar a una conclusión correcta a partir de premisas correctas.

2.2.2. Aprendizaje Automático

El Aprendizaje Automático, o Machine Learning (ML) en inglés, forma parte de la rama de lo que es la Inteligencia Artificial, centrando su desarrollo en algoritmos y modelos capaces de dotar a las computadoras con habilidades que mejoren el rendimiento de realizar una tarea específica sin requerir que sea programada ([Díaz, 2021](#)).

El Aprendizaje Automático utiliza algoritmos que puedan identificar patrones a través de datos almacenados y procesados para la creación de un modelo que le permita realizar análisis predictivos o de detección, ya que busca replicar la inteligencia humana para ser más precisos y mejoren a través de la práctica, siendo adaptable a cualquier escenario en donde los datos puedan cambiar ([Azure, s.f.](#)).

La aplicación del Aprendizaje Automático es muy amplia, como programas de motores de búsqueda, el diagnóstico automático médico, la detección de fraude en transacciones de una entidad bancaria, análisis de mercado y clasificación de objetos; además del uso de técnicas de reconocimiento facial, voz o escritura ([Díaz, 2021](#)).

Existen tres métodos principales de Aprendizaje Automático:

- **Aprendizaje Supervisado:** Se centra en el desarrollo de algoritmos donde interviene un programador para entrenar al modelo qué conclusiones debe hacer mediante un conjunto de datos etiquetados con un resultado ya definido ([Oracle, s.f.](#)). De esta manera, el modelo puede mapear las entradas con las salidas correctas, mediante una función de pérdida que mida la diferencia entre los verdaderos resultados con el resultado predicho del modelo ([IBM, s.f.-b](#)). Este modelo puede resolver problemas de clasificación y de regresión, y está presente comúnmente en la detección de fraudes, enfermedades o la predicción del precio.
- **Aprendizaje No Supervisado:** Presenta un enfoque más independiente, donde el modelo aprende a identificar patrones o procesos complejos sin la intervención constante de un programador. Dicho aprendizaje implica un entrenamiento mediante datos no etiquetados y sin un resultado esperado. Este aprendizaje utiliza algoritmos de agrupamiento como k-means, análisis de componentes principales o de asociación, también están en las técnicas de visión computacional. Estos algoritmos tienen como fin realizar grupos o agrupamientos de datos que comparten similitudes, como imágenes, las cuales cada

grupo tiene su propia etiqueta ([Oracle, s.f.](#)).

- **Aprendizaje por Refuerzo:** El Aprendizaje por Refuerzo o Recompensa es la retroalimentación que se le da al agente para saber si algo bueno o malo ha ocurrido cuando realiza una acción. Este tipo de aprendizaje está presente en los juegos de ajedrez, donde el refuerzo se realiza cuando finaliza el juego ([Russell & Norvig, 2004](#)). El agente puede interactuar en un entorno dinámico, tomando decisiones y recibimiento retroalimentación mediante una función de recompensa, con el objetivo de aprender una política, el cual le brinda estrategias de selección para tomar acciones que maximicen el total de recompensas ([Comisión Europea, s.f.](#)).

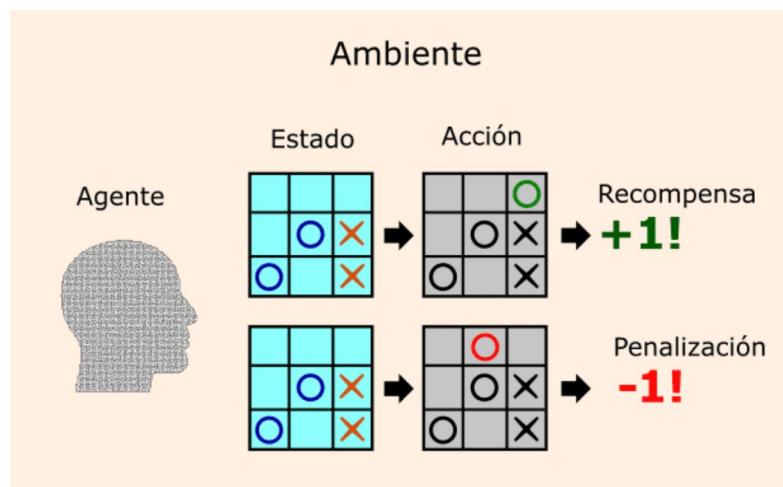


Figura 2.21: Representación del Aprendizaje por Refuerzo

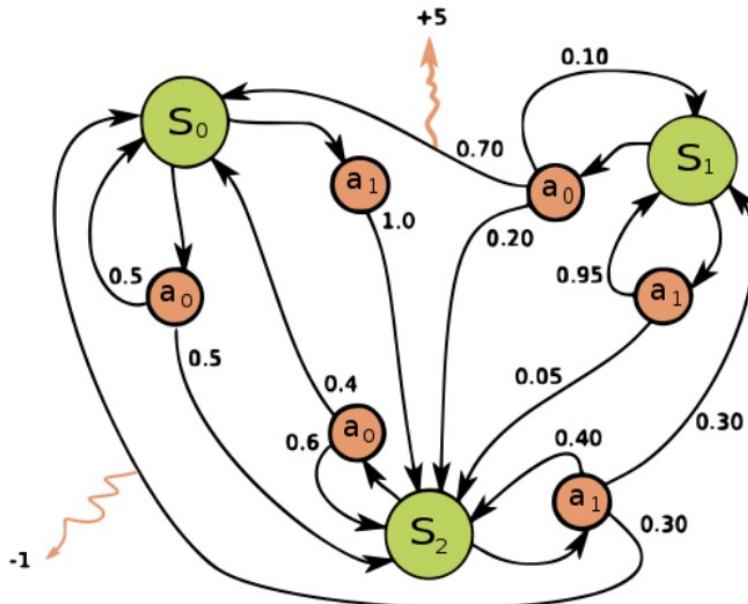
Fuente: ([s.f.](#)). «[Aprendizaje por refuerzo: Concepto, características y ejemplo](#)»

Un concepto fundamental en la que se basa el Aprendizaje por Refuerzo es el Proceso de Decisión de Markov (MDP), el cuál es un marco matemático que le permite realizar un modelo de toma de decisiones en situaciones donde los resultados son aleatorios y depende del control de quien toma las decisiones. Como se muestra en la Figura 3.2,, este es un ejemplo simple de MDP, el cuál en cada paso de tiempo, el proceso inicia en un estado S , y el que toma la decisión puede elegir una acción a que está disponible en dicho estado. El resultado de esto otorga un nuevo estado S_t y una recompensa $R_a(S, S_t)$ ([AcademiaLab, s.f.](#)).

Para que el MDP pueda conocer las probabilidades o dichas recompensas, se define una función el cuál es conocido como Q-learning:

$$Q(s, a) = \sum_{s^t} P_a(s, s^t)(R_a(s, s^t) + \gamma V(s^t)) \quad (\text{Ecuación 2.7})$$

Donde P_a es la probabilidad de que la acción a en el estado s en un tiempo determinado de un nuevo estado s^t , el R_a es la recompensa esperada que recibirá de pasar un estado a

**Figura 2.22:** Proceso de decisión de MarkovFuente: ([s.f.](#)). «[Proceso de decisión de markov](#)»

otro debido a la acción a . Mientras que el V contiene los valores reales del nuevo estado ([s.f.](#)).

2.2.3. Aprendizaje Profundo

Los algoritmos de aprendizaje profundo forman parte del aprendizaje automático, y los cuales han tomado una mayor participación en los últimos años por la alta disponibilidad de grandes datos y los recursos computacionales potentes. Debido a que se cuenta con GPUs más rápidas para el entrenamiento de grandes modelos profundos, los algoritmos de Aprendizaje Profundo supera a los modelos tradicionales en múltiples aplicaciones, como el mejoramiento del rendimiento de modelos de reconocimiento de imágenes, redes neuronales convolucionales profundas al reducir la tasa de error, el rendimiento de sistemas de reconocimiento de voz el cual estuvo estancado varios años; y también aporta enormemente en el campo de la investigación del Procesamiento del Lenguaje Natural (NLP) ([Ma & Tang, 2021](#)).

2.2.3.1. Redes Neuronales Profundas (DNN)

Las Redes Neuronales Profundas, o Deep Neural Network en inglés, permiten al modelo mediante su entrenamiento a realizar tareas más complejas que son difíciles de hacer utilizando redes neuronales tradicionales. Asimismo, dichas redes se inspiran del cerebro humano, teniendo un diseño que no solo se limita a seguir reglas establecidas, sino también predecir y sacar conclusiones (BotPress, 2022). Las Redes Neuronales Profundas si bien son similares a las Redes Neuronales Tradicionales, el término de “Profundo” hace referencia al número de capas ocultas que forma parte de la red, teniendo más nodos de procesamiento, las cuales son conocidas como neuronas (TechEdu, s.f.).

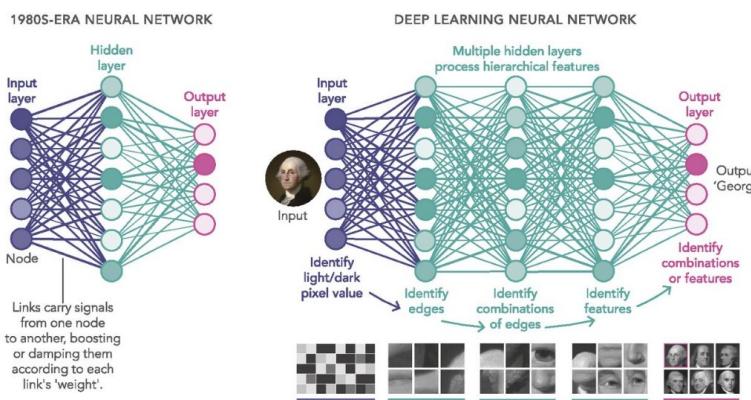


Figura 2.23: Comparación entre una Red Neuronal Artificial y una Red Neuronal Profunda

Fuente: (s.f.). «What Is Deep Learning, Nature Of Machine Learning And Beauty Of Deep Neural Networks ? - New»

Este tipo de red está presente en el reconocimiento de voz, sonidos y grafos, además que utilizan “Big Data” para la resolución de problemas en la que la intervención humana es limitada (BotPress, 2022).

2.2.4. Aprendizaje Profundo por Refuerzo (DRL)

El Aprendizaje Profundo por Refuerzo o Deep Reinforcement Learning (DRL) es la combinación entre técnicas de Redes Neuronales Profundas con Aprendizaje por Refuerzo, esto tiene como beneficio la interacción de forma iterativa en un entorno para tomar decisiones que busquen maximizar una función de recompensas para encontrar estrategias más sofisticadas (GeeksforGeeks, 2023). El Aprendizaje Profundo por Refuerzo es un paso importante en la evolución del aprendizaje de las máquinas que toma la decisión más beneficiosa y que utiliza esa decisión en escenarios futuros (Instituto de Ingeniería del Conocimiento, s.f.).

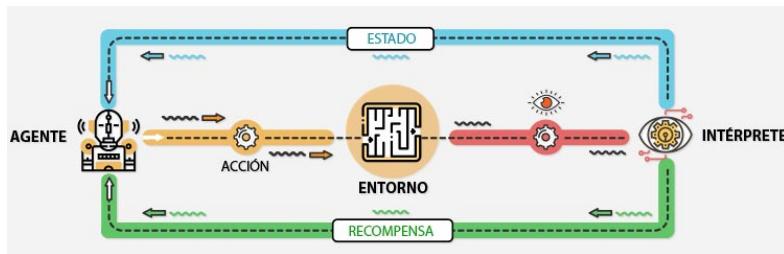


Figura 2.24: Componentes que conforma el Aprendizaje por Refuerzo

Fuente: ([s.f.](#)). «[Aprendizaje profundo por refuerzo](#)»

Un modelo de Aprendizaje Profundo por refuerzo está conformado por un agente, el cual es el que aprende las reglas a seguir y toma decisiones mediante la interacción con el entorno utilizando técnicas de Redes Neuronales Profundas. Se menciona, además, que el DRL está basado en Q-learning, componentes de políticas para que se dirija las decisiones del agente, conceptos claves como lo son la función de valor que calcula las recompensas, las cuales son una señal que muestra el entorno sobre la acción deseada para que el agente cambie el estado de la situación actual del entorno. Son ampliamente utilizados en los sistemas de navegación, tratamientos médicos como la recomendación de medicamentos, perfeccionar los diseños de materiales con el fin de aumentar su efectividad y la personalización del entorno de eCommerce según los gustos de cada cliente ([s.f.](#)).

Con lo que respecta a los sistemas de navegación, se aplican tecnologías de información avanzadas para hacer frente al problema del control adaptativo de las señales de tráfico. Existen muchas investigaciones y propuestas como un enfoque que realiza simulaciones con datos de tráfico reales de la ciudad de Toronto, donde los agentes se coordinan mediante intersecciones vecinas. También la integración de un algoritmo de Aprendizaje por Refuerzo multiagente para enfrentar a los desafíos de estacionariedad y dimensionalidad en el control adaptativo de las señales de tráfico ([Li, 2018](#)).

2.2.4.1. Deep Q-Network (DQN)

Para que el modelo de Aprendizaje Profundo por Refuerzo pueda aprender, es fundamental tener una función de valor. Esta función está presente en las arquitecturas DQN, la cuál nace en el año 2015 por DeepMind ([2018](#)). Este algoritmo es creado por la necesidad de afrontar la inestabilidad de aprendizaje de la combinación de RL con redes neuronales, ya que almacena todo lo aprendido por el agente y luego lo reproduce aleatoriamente para proporcionar datos de entrenamiento diverso y descorrelacionados ([DeepMind, 2016](#)).

Este algoritmo ha presentado varios avances, estabilizando la dinámica de aprendizaje,

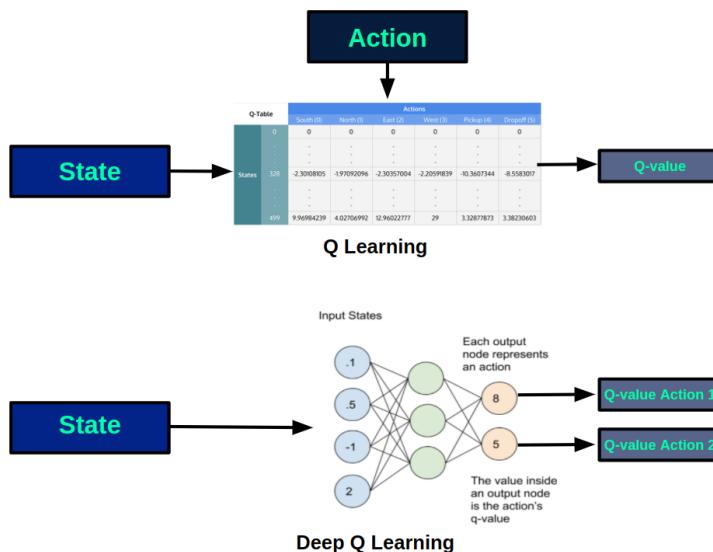


Figura 2.25: Diferencias entre Q-Learning y Deep Q-Learning

Fuente: (2022). «Aprendizaje por refuerzo profundo: DQN»

priorizando las experiencias que ya han sido aprendidas o también conocidas como Experience Replay, además de normalizar, agregar y reescalar los resultados (DeepMind, 2016).

2.2.4.2. Política

Una política es aquella que busca un mapeo óptimo en las acciones realizadas por los estados. Uno de los algoritmos de una política es el algoritmo de actor-crítico que aprende una función de valor de estado para actualizarlo a partir de estimaciones posteriores para reducir la varianza y acelerar el aprendizaje. Además de esto, en el campo de Aprendizaje por Refuerzo y Aprendizaje Profundo por Refuerzo, se tiene un enfoque en el gradiente de política y su optimización, donde el método más popular es el REINFORCE, que en comparación al Q-learning que es más eficiente en el uso de los datos, este tiende a ser más estable. Existen también otras gradientes que son eficientes en situaciones donde las acciones son continuas, una de estas gradientes es la Gradiente de Política Determinista (DPG), el cual se basa en la estimación de la gradiente de la función de valor de acción, siendo más eficiente que los métodos estocásticos de gradiente de política, además, utiliza Redes Neuronales Profundas para una mayor estabilidad del modelo de aprendizaje (Li, 2018).

Para la optimización de las políticas, se utiliza el método Trust Region Policy Optimization (TRPO), que controla las actualizaciones de la política mediante la restricción de región de confianza, mejorando la estabilidad del modelo y la eficiencia computacional (2018).

2.2.4.3. Recompensa

Las recompensas son las que proporcionan una retroalimentación al agente para la toma de decisiones. Esta recompensa es calculada mediante una función que es modificada para facilitar el aprendizaje mientras se obtiene una política óptima. Sin embargo, estas funciones de recompensas no suelen estar presente en mucho de los problemas, por lo que se recurre al aprendizaje por imitación, donde el agente aprende mediante demostraciones de expertos (2018). Este tipo de aprendizaje tiene dos enfoques los cuales son:

- **Deep Q-learning from Demonstrations (DQfD):** Este enfoque combina pérdidas por diferencia temporal (TD), supervisadas y regularizadas. Es entrenado a partir de datos de demostración con el fin de establecer una función de valor y así generar sus propias muestras para el entrenamiento del modelo.
- **Inverse Reinforcement Learning (IRL):** Permite aprender políticas a partir de los datos, evitando aprender una función de recompensa.

2.2.4.4. Modelo y Planificación

El entorno es un modelo que incluye el modelo de transición y un modelo de recompensa, este enfoque de Aprendizaje por Refuerzo basados en Modelos puede aprender de manera eficiente la función de valor y política, teniendo una ventaja al Aprendizaje por Refuerzo sin modelo, ya que no requiere de un gran número de muestras; sin embargo, puede tener problemas a la hora de identificar los modelos estimados y teniendo como resultado que no sean precisos o que el rendimiento sea limitado. Para hacer frente a este problema, la planificación construye una función de valor o una política en base a un modelo (2018).

2.2.4.5. Exploración

Para reducir la incertidumbre sobre la función de recompensa y las probabilidades del entorno, el agente utiliza la exploración como principal herramienta de apoyo. La incertidumbre se puede cuantificar como intervalos de confianza o los propios parámetros del entorno relacionados con el recorrido de estado-acción. Existe más de un enfoque de exploración, como lo es la exploración basada en recuentos de recorridos para guiar el comportamiento del agente con el fin de reducir la incertidumbre, también está el enfoque de la motivación intrínseca que explora solo lo que es más sorprendente en un proceso de aprendizaje basado en los cambios en el error de la predicción (2018).

2.2.5. Fundamentos de grafos

Los grafos son la representación que existe en la relación entre entidades, y que están presentes en varios campos como las ciencias sociales, química, biología y física. Un ejemplo de estos grafos en el campo de la química son los compuestos químicos, en donde los átomos representan los nodos y los enlaces químicos representan las aristas ([Ma & Tang, 2021](#)).

Un grafo puede ser denotado como $G = \{V, E\}$, donde $V = \{v_1, \dots, v_N\}$ es un conjunto de $N = |V|$ nodos y $E = \{e_1, \dots, e_M\}$ es un conjunto de M aristas, además, G representa el tamaño que tiene el grafo. Un aspecto esencial en los grafos son los nodos, los cuales representan a las entidades mientras que el conjunto de aristas representa las conexiones que existen entre los nodos. Existen dos tipos de grafos, los dirigidos donde las aristas van en orden, es decir desde el nodo 1 al nodo 2, mientras que los no dirigidos el orden de los dos nodos no altera el resultado, ya que puede ir del nodo 2 al nodo 1 y viceversa, sin tener diferencias ([2021](#)).

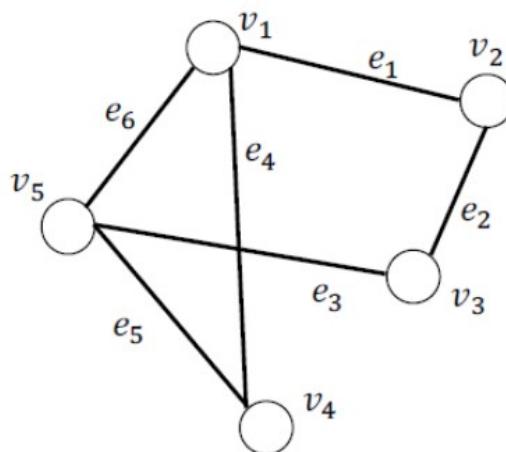


Figure 2.1 A graph with 5 nodes and 6 edges

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Figura 2.26: Propiedades u medidas de un grafo

Fuente: ([2021](#)). *Deep Learning on Graphs*. (p. 19)

Otro aspecto que conforma un grafo es su Matriz de Adyacencia, el cual se denota como $A \in \{0, 1\}^{N \times N}$. Cada A_{ij} representa la relación que hay entre dos nodos. Cabe mencionar que,

en un grafo no dirigido, su matriz de adyacencia es simétrica, es decir, $A_{ij} = A_{ji}$ (2021).

2.2.5.1. Propiedades de un grafo (2021)

El grado indica la cantidad de veces que un nodo está conectado a otro, el cual puede ser calculado de la siguiente forma:

$$d(v_i) = \sum_{v_j \in V} \mathbb{I}_\epsilon(\{v_i, v_j\}), \quad (\text{Ecuación 2.8})$$

$$\mathbb{I}_\epsilon(\{v_i, v_j\}) = \begin{cases} 1 & \text{if } (v_i, v_j) \in \epsilon, \\ 0 & \text{if } (v_i, v_j) \notin \epsilon \end{cases} \quad (\text{Ecuación 2.9})$$

El grado también puede ser calculado mediante su matriz de adyacencia de la siguiente forma:

$$d(v_i) = \sum_{j=1}^N A_{i,j} \quad (\text{Ecuación 2.10})$$

En contraste al grado, los vecinos son el conjunto de nodos que están conectados a un nodo. El camino de un grafo es el recorrido que hay entre los nodos y las aristas y su longitud es el número de aristas que existen. En este camino, se forman rastros que son aristas distintas y la trayectoria que son los nodos distintos. Para que haya conectividad, se necesita al menos que haya un camino entre cualquier par de nodos. Para hallar el camino más corto, se calcula de la siguiente manera:

$$p_{st}^{sp} = \arg \min_{p \in P_{st}} |p| \quad (\text{Ecuación 2.11})$$

Donde p es el camino en p_{st} con su longitud. Para calcular el diámetro de un grafo, es de la siguiente manera:

$$\text{diameter}(G) = \max_{v_s, v_t \in V} \min_{p \in P_{st}} |p| \quad (\text{Ecuación 2.12})$$

2.2.5.2. Procesamiento de señales de Grafos (2021)

Las señales de grafos son las características o atributos asociados en cada nodo, los cuales almacenan la conectividad o estructura de los nodos, como también sus datos. Esta señal consiste en un grafo y una función de mapeo donde se le asigna los valores reales correspondiente a los nodos, el cual es representado de la siguiente manera:

$$f : V \rightarrow \mathbb{R}^{N \times d} \quad (\text{Ecuación 2.13})$$

Donde d es el vector (dimensión del valor) que está asociado a cada nodo.

2.2.5.3. Transformación de Fourier en Grafos (GFT) (2021)

En comparación a la clásica transformación de Fourier, que realiza un análisis de señales establecidas en la dimensión del tiempo o espacio, esta extensión realiza el análisis en la dimensión de un grafo, con el fin de descomponer dicha señal en una serie de componentes espectrales basados en los autovectores de la matriz Laplaciana del grafo, la cual es una matriz que captura la estructura del grafo con el fin de realizar diversas operaciones y análisis. Dicha Matriz Laplaciana (L) es la diferencia de la matriz diagonal de los grados (D) y la matriz de adyacencia del grafo (A).

La Transformación de Fourier en grafos se puede definir matemáticamente de la siguiente manera:

$$f[i] = \sum_{l=1}^N \hat{f}[l] u_l[i] \quad (\text{Ecuación 2.14})$$

Donde:

$\hat{f}[l]$ es el l-ésimo coeficiente de Fourier en grafos.

u_l es el l-ésimo autovector de la matriz Laplaciana L del grafo.

2.2.5.4. Tipo de Grafos (2021)

- **Grafos Heterogéneos:** Son los grafos más simples explicados en los anteriores puntos, en donde solo tienen un tipo de nodos y aristas.
- **Grafos Multidimensionales:** Este tipo de grafo puede compartir múltiples relaciones en simultáneo entre un par de nodos, como por ejemplo cuando los usuarios de YouTube, considerados como nodos, pueden suscribirse entre ellos, donde cada tipo de relación se considera como una dimensión.
- **Signed Graphs:** Este tipo de grafo contiene tanto aristas positivas como negativas, y son popularmente representados en las plataformas sociales en línea, como, por ejemplo, cuando un usuario puede bloquear o dejar de seguir a otros usuarios, dicha acción de seguir o no seguir pueden considerarse como las aristas positivas y negativas, respectivamente.
- **Hypergrafos:** Este tipo de grafo en comparación a los anteriores, este no solo tiene información entre pares de aristas, ya que, en el mundo real, estas relaciones van más allá que las asociaciones entre pares, como por ejemplo los artículos publicados por un autor.

- **Dynamic Graph:** Es un tipo de grafo en el cual sus conexiones entre los nodos pueden cambiar con el tiempo, reflejando un comportamiento y evolución continua entre estos.

2.2.6. Métodos de Redes Neuronales Profundas basadas en Grafos

2.2.6.1. Graph Neural Networks (2021)

Las Redes Neuronales Gráficas (GNNs) son métodos que aplican Redes Neuronales Profundas a datos estructurados representados en grafos. Este método puede ser representado como un proceso de aprendizaje de representación en grafos y que están enfocados en aprender las características representativas de cada nodo con el fin de facilitar las tareas del grafo. Dicho proceso se puede representar de la siguiente manera:

$$F^{(of)} = h(A, F^{(if)}) \quad (\text{Ecuación 2.15})$$

Donde A representa la matriz de adyacencia del grafo con N nodos y $F^{(if)}$ y $F^{(of)}$ representan las matrices de características de entrada y salida, las cuales también tienen las dimensiones del grafo. Los subíndices “ if ” y “ of ” son los filtros que se realizan dentro del grafo y que en la figura está representada el proceso tradicional de filtrado del grafo a sus características de los nodos:

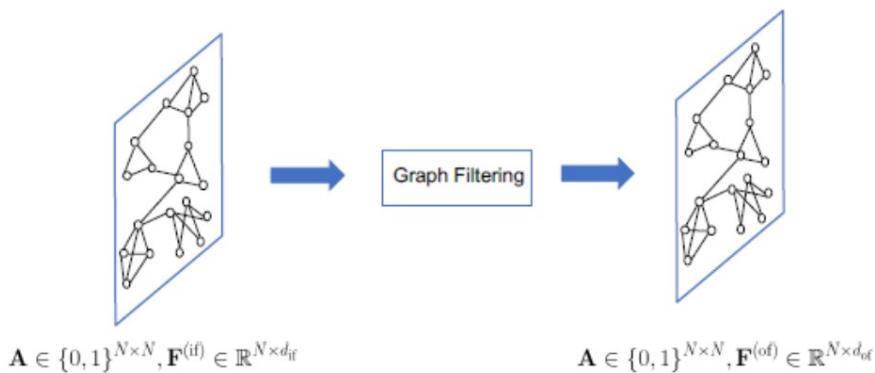
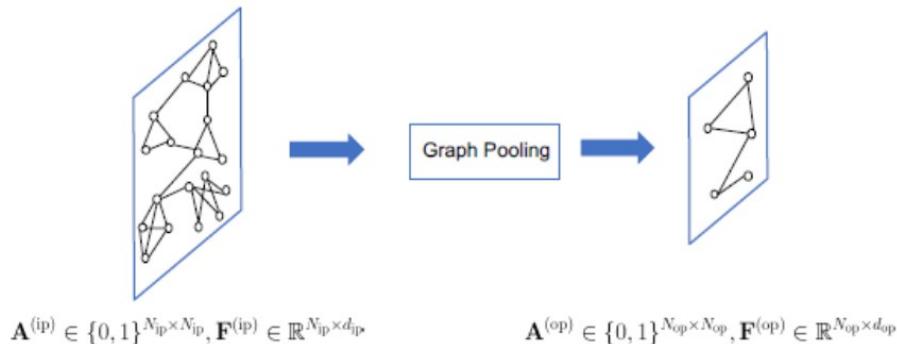


Figura 2.27: Operación de filtración de grafos

Fuente: (2021). *Deep Learning on Graphs*. (p. 108)

Existen otras operaciones que son necesarias para las tareas centradas en grados y generalmente son operaciones de agrupación a partir de las características de los nodos. Este proceso de agrupación toma un grafo como entrada y luego producen un grafo reducido con pocos nodos, como se observa en la siguiente figura:

**Figura 2.28:** Operación de filtración de grafosFuente: (2021). *Deep Learning on Graphs*. (p. 109)

Dicha operación se puede describir de la siguiente manera:

$$A^{(op)}, F^{(op)} = pool(A^{(ip)}, F^{(ip)}) \quad (\text{Ecuación 2.16})$$

Los subíndices *ip* y *op* indican la entrada y salida de la agrupación.

2.2.6.2. Robust Graph Neural Networks

Al igual que las DNNs tradicionales, estas son vulnerables a ataques adversarios, generando perturbaciones en la manipulación de la estructura del grafo y la características de los nodos para engañar a los modelos de GNNs. Por esta limitación, Robusta Graph Neural Networks es una extensión de las GNNs que adoptan medidas críticas de seguridad, las cuales pueden estar presentes en los sistemas financieros y la gestión de riesgo. La RGNNs comúnmente utiliza un enfoque de limpieza del grafo perturbado, los cuales han sido víctimas de una violación de ciertas propiedades que presenta el grafo real (Jin et al., 2020).

2.2.6.3. Scalable Graph Neural Networks

Las GNNs tradicionales presentan problemas de escalabilidad que impide la adopción de grados a gran escala, debido a que utiliza un método de gradiente para minimizar la función de pérdida en cada etapa de representación de nodos, requiriendo una gran cantidad de recursos computacionales y operaciones de cálculos (Libro). Para abordar dichos problemas, se utilizan las técnicas y enfoques de las SGNNs para que las GNNs sean más eficientes a medida que el tamaño y la complejidad de los grafos van aumentando. Una de las principales técnicas de las SGNNs es el Graph Coarsening que utiliza la reducción de grafos para el entrenamiento escalable de estas (Huang et al., 2021).

2.3. Marco Conceptual

2.3.1. Aprendizaje Profundo

Subconjunto del Aprendizaje automático, con la capacidad de reconocer patrones con un gran volumen de datos, estructurados y no estructurados, entrenados por redes neuronales artificiales que intentan emular el cerebro humano y que, a comparación del aprendizaje automático, no requiere de la intervención humana para el modelo ([IBM, s.f.-a](#)).

2.3.2. Aprendizaje Profundo (RL)

Subconjunto del Aprendizaje Automático, su objetivo es determinar qué acciones debe un agente del software elegir para maximizar una función de recompensa mediante la interacción dinámica del entorno. Su principal ventaja frente a otros algoritmos de Aprendizaje Automático, es que permite que el modelo aprenda sin necesidad de que intervenga el programador ([Techopedia, 2024](#)).

2.3.3. Aprendizaje Profundo por Refuerzo (DRL)

Técnica que combina el aprendizaje profundo con aprendizaje por refuerzo. Se utiliza una red neuronal de aprendizaje profundo que aprende por cada acción en tiempo real se presente, un ejemplo de esto es en un juego de Ajedrez ([Gamco, s.f.](#)).

2.3.4. Redes Neuronales Profundas (DNN)

Las Redes Neuronales Profundas, o Deep Neural Networks (DNN) en inglés, es una de las categorías de modelos del Aprendizaje Profundo, esta red está compuesta por varias capas de nodos que tienen como objetivo simular las neuronas del cerebro humano, su estructura está conformada por capas de entrada, de red y de salida, cada capa realiza cálculos de aprendizaje ([Talaei et al., 2023](#)).

2.3.5. Graph Neural Network (GNN)

Técnica que utiliza la potencia de predicción del aprendizaje profundo para predecir objetos(nodos) que están relacionados (bordes). Tiene la capacidad de predecir a nivel de nodo,

edges o gráficos([Nvidia, 2022](#)).

2.3.6. Zona Urbana

Lugar compuesto por ciudades, con alto nivel de población, además de estar compuesta por calles, avenidas, edificación y comercios. Además de que se encuentra la mayoría de servicios básicos, como también empresas y espacios burocráticos del Estado, las zonas urbanas se caracterizan por tener una gran variedad de autopistas, vías de ferrocarril, estaciones y puertos([Enciclopedia Humanidades, s.f.](#)).

2.3.7. Delincuencia

Es la acción de cometer actos ilegales que van en contra de las leyes impuestas por el estado, son cometidas a nivel individual o por grupos delictivos. Existen diversos tipos de delincuencias, entre ellas, el robo, violencia doméstica, suplantación de identidad, asalto, entre otros([Conceptualia, s.f.](#)).

2.3.8. Ruta

Trayecto el cual una persona puede transitar desde un lugar a otro, el cuál puede variar según el propósito de la persona. También una ruta puede definirse como un camino, vía o carretera de diferentes lugares geográficos([Significado, s.f.](#)).

2.3.9. Riesgo

Es la probabilidad de que ocurra un evento y sus efectos negativos frente a una persona u objeto, como el daño o la situación de peligro. El riesgo puede ser representado mediante una medida probabilística con el fin de saber qué evento puede amenazar con el cumplimiento de un objetivo de una organización o, en otros contextos, para saber el nivel de seguridad ciudadana([ConceptoDefinicion, 2023](#)).

Capítulo 3

METODOLOGÍA DE LA INVESTIGACIÓN

3.1. Diseño de la investigación

El diseño del presente trabajo de investigación es de tipo experimental puro, ya que se analizarán las variables, procesando los valores con técnicas de regresión para obtener un vector característico que sirva como entrada al modelo de Aprendizaje Profundo, específicamente un modelo de Redes Neuronales Profundas basadas en Grafos.

3.1.1. Tipo de investigación

El tipo de investigación del presente trabajo es de alcance experimental, ya que el desarrollo de un algoritmo de Aprendizaje Profundo para la detección de rutas seguras en la ciudad de Los Ángeles busca una explicación a partir de patrones que predigan qué rutas son las más seguras basándose en los datos de geolocalización de incidentes delictivos; para ello, se establece una relación de causa efecto, donde a través de una función de riesgo determine qué rutas debe evitar para determinar la opción más segura.

3.1.2. Enfoque de investigación

El enfoque que presenta este trabajo es cuantitativo, ya que utiliza instrumentos de identificación y medición de riesgo en los datos de incidencias, cuyo resultado numérico servirá para entrenar al modelo de Aprendizaje Profundo para detectar qué rutas debe evitar.

3.2. Población y muestra

Categoría	Descripción
Población	Registro de crímenes extraídos de la página oficial del Departamento de Policía de Los Ángeles, Estados Unidos, desde el año 2020 hasta el año 2024.
Muestra	Aproximadamente más de 680,000 registros de crímenes de la página oficial del Departamento de Policía de Los Ángeles. Para la selección de la muestra, se utilizó muestreo no probabilístico o dirigido, ya que se seleccionaron los registros que tienen como descripción aquellos crímenes perpetrados en las calles, tiendas o bancos; como por ejemplo asaltos a mano armada, violación, robo de vehículos, acoso sexual, vandalismo, entre otros.
Unidad de análisis	Latitud y Longitud donde ocurrió el crimen registrado.
Variable y tipo de análisis	Variable cuantitativa discreta, ya que el presente trabajo se centra en variables numéricas y la medición de riesgo de las rutas.

3.3. Operacionalización de Variables

3.4. Técnicas de recolección de datos

Para el presente trabajo se recolectó un conjunto de datos de registros públicos de crímenes en la ciudad de Los Ángeles, que se componen de variables tanto cuantitativas (longitud, latitud, fecha, área, número de documento, edad, entre otros), como cualitativas (descripción del crimen, dirección, sexo, entre otros). Para obtener este conjunto de datos se siguió un flujo de búsqueda, descarga y segmentación de los registros de crímenes en base al tipo de crimen realizado como se muestra en la figura. Cabe mencionar que en proceso de segmentación, el conjunto de datos contiene una columna llamada “Crm Cd Desc”, que son los tipos de crímenes registrados, por lo que se filtra aquellos crímenes que involucren agresión física o verbal, como también vandalismos en las zonas urbanas de una ciudad. Asimismo, se extraerá en formato XLSX solo las columnas necesarias para el entrenamiento del modelo, los cuales son: Crm Cd Desc, Date Rptd, LAT, LON. “LAT” y “LON” representan la latitud y longitud donde ocurrió el

DEFINICIÓN DE VARIABLES		
VARIABLE Y DEFINICIÓN	INDICADOR	FÓRMULA
RUTAS SEGURAS Trayecto el cual evita las zonas con alto nivel de delincuencia para una persona.	Indicador de Riesgo	$R = \sum_{i=1}^n (W_i \times C_i)$ <p>Donde:</p> <ul style="list-style-type: none"> ■ n es el número de factores considerados. ■ W_i son los pesos asignados a cada factor. ■ C_i son los valores normalizados de cada factor.
ALGORITMO DE APRENDIZAJE PROFUNDO Tipo de algoritmo de la subcategoría del Aprendizaje Automático, el cual es el Aprendizaje Profundo, que utiliza Redes Neuronales Artificiales con múltiples capas para el procesamiento y análisis de datos complejos.	Indicador de Eficiencia	$TTM = N_{ep} \times T_{ep}$ <p>Donde:</p> <ul style="list-style-type: none"> ■ N_{ep} es el número de épocas del modelo. ■ T_{ep} el tiempo promedio necesario para completar una época en el entrenamiento.
	Indicador de Precisión	$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$ <p>Donde:</p> <ul style="list-style-type: none"> ■ n es el número total de observaciones o datos. ■ y_i es el valor observado en la posición i. ■ \hat{y}_i es el valor predicho en la posición i.
	Tasa de éxito del agente	$\frac{\text{Number of Successful Routes}}{\text{Total Number of Routes}}$ $\frac{\text{Success Rate}}{=}$

crimen. Además de la extracción del conjunto de datos de registro de crímenes, es importante extraer la cartografía de la ciudad de Los Ángeles, como se muestra en la figura, haremos uso

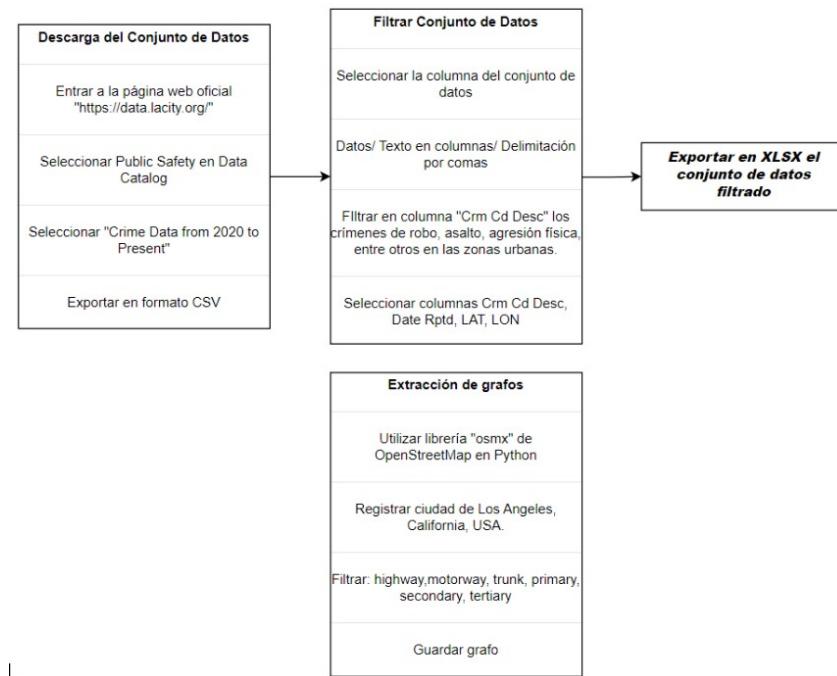


Figura 3.1: Elaboración propia: Flujo de recolección de datos

de la librería “osmnx” de OpenStreetMaps en Python para extraer los datos de una zona que buscas según lo que necesitas. Para el presente trabajo, se busca la región de la ciudad de Los Ángeles, el cual la librería nos brindará todas los nodos que son las calles, estaciones, avenidas, entre otros.

3.5. Técnicas para el procesamiento y análisis de la información

Para la metodología propuesta, se usó de base la metodología CRISP-DM(Cross-Industry Standard Process for Data Mining), que está orientado mayormente al campo de la minería de datos, es un proceso o un ciclo de vida del desarrollo de un modelo, y el cuál es altamente personalizable y flexible en cualquier tipo de situación o desarrollo (IBM, s.f.-c). Este ciclo de vida del modelo consta de 6 fases, las cuales son:

- Entendimiento del modelo: La comprensión de los objetivos y requisitos definidos del proyecto para el usuario final, con el fin de convertir dicho conocimiento en una definición técnica del problema.
- Comprensión de los datos: Se realiza un análisis exploratorio en los datos para tener una visión general de lo que se puede conseguir de estas.

- Preparación de los datos: En esta fase se cubre todas las actividades requeridas para la construcción del conjunto de datos, junto con el preprocesamiento de los datos que serán empleados para la siguiente fase.
- Modelado de datos: Fase en la cual tiene que ver con la implementación de los modelos requeridos del proyecto que derivarán a los próximos resultados y conclusiones de este.
- Evaluación: Se mide los resultados y calidad del modelo, junto con su efectividad para el cumplimiento de los requisitos establecidos.
- La implementación del modelo en un entorno operativo, la planificación del mantenimiento y monitoreo de este para que siga cumpliendo con los objetivos del proyecto.

Estableciendo la base teórica en lo que consiste la metodología CRISP-DM, también se usó de ejemplo la metodología propuesta por Levy et al. en su artículo «[SafeRoute: Learning to Navigate Streets Safely in an Urban Environment](#)», ya que estos trabajan también con conjuntos de datos de delitos y la obtención del mapa de las ciudades en OpenStreetMaps, además que implementa una función de recompensa basada en riesgos para entrenar su modelo de Aprendizaje por Refuerzo. Otro artículo de inspiración para el desarrollo de la metodología, es de Esquivel et al. en su artículo «[Predicting Motor Vehicle Theft in Santiago de Chile using Graph-Convolutional LSTM](#)», ya que proponen la aplicación de regresión LOESS para el preprocesamiento de los datos que serán utilizadas para el entrenamiento de su modelo LSTM basado en grafos. Por último, también tomará de guía el flujo de condición para la elección de rutas seguras propuesta por Soni et al. en su artículo «[Route-The Safe: A Robust Model for Safer Route Prediction Using Crime and Accidental Data](#)». La metodología del presente trabajo de investigación se puede visualizar de manera gráfica en la Figura 3.2.

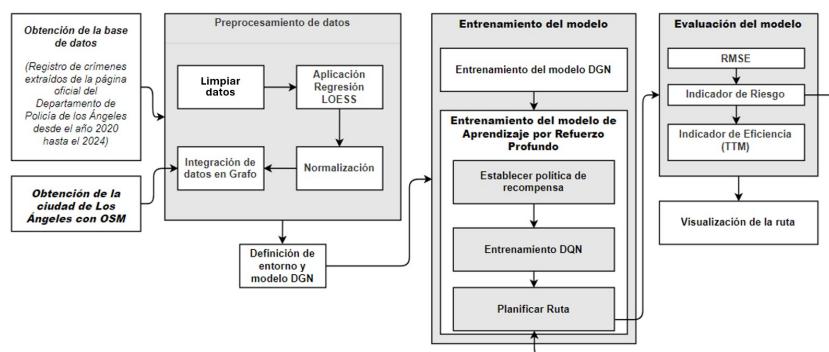


Figura 3.2: Elaboración propia: metodología propuesta

Como ya se mencionó, esta metodología viene de la derivación de la metodología tradicional CRISP-DM junto con las metodologías propuestas en los antecedentes. Se puede en-

contrar una relación entre la comprensión del negocio con el objetivo del presente artículo el cuál es la minimización del riesgo en la detección de rutas seguras. Asimismo, en la fase de la comprensión de los datos, se encuentra la recolección de los registros de crímenes en la ciudad de Los Ángeles y la obtención de la ciudad de Los Ángeles en OpenStreetMap. La fase de preparación de los datos es más que la fase del preprocesamiento de los datos con sus respectivos pasos propuestos por la metodología. Mientras que el Modelado es la creación y entrenamiento del modelo de Deep Graph Network (DGN) para la representación del grafo de la ciudad y del entrenamiento del modelo de Aprendizaje por Refuerzo Profundo para establecimiento de las políticas de recompensa y la planificación y detección de rutas seguras. Por último, las fases de evaluación y despliegues son las métricas propuestas en la metodología que determinará qué ruta segura debe elegir el usuario según el índice de riesgo que presente.

3.5.1. Metodología de la implementación de la solución

3.5.1.1. Adquisición

En esta fase se describe la obtención del conjunto de datos, como ya se mencionó anteriormente en el punto de las técnicas de recolección, se necesita registros de crímenes realizados en una zona urbana, para esto, se decidió trabajar con el conjunto de datos abiertos de registros de crimen en la ciudad de Los Ángeles desde el 2020 hasta el 2024. Dicho conjunto de datos tiene más de novecientos veinticinco mil registros aproximadamente y cuenta con 28 columnas las cuales se detallan a continuación:

Nombre de Columna	Descripción	Tipo de dato
DR_NO	Número de División de Registros: Número de archivo oficial compuesto por un año de 2 dígitos, un ID de área y 5 dígitos	Texto
Date Rptd	MM/DD/YYYY Floating Timestamp	Texto
DATE OCC	MM/DD/YYYY Floating Timestamp	Texto
TIME OCC	En horario militar de 24 horas	Texto

Continúa en la siguiente página

Tabla 3.1 – Continuación de la página anterior

Nombre de Columna	Descripción	Tipo de dato
AREA	El LAPD tiene 21 comisarías comunitarias denominadas áreas geográficas dentro del departamento.	Texto
AREA NAME	Las 21 áreas geográficas o divisiones de patrulla también reciben una designación de nombre que hace referencia a un punto de referencia o a la comunidad circundante.	Texto
Rpt Dist No	Un código de cuatro dígitos que representa una subárea dentro de un Área Geográfica.	Texto
Part 1-2	Sin descripción	Número
Crm Cd	Indica el delito cometido. (Igual que el Código Penal 1)	Texto
Crm Cd Desc	Define el Código Penal previsto.	Texto
Mocodes	Modus Operandi: actividades asociadas con el sospechoso en la comisión del delito.	Texto
Vict Age	Numérico de dos caracteres	Texto
Vict Sex	F - Mujer M - Hombre X - Desconocido	Texto
Vict Descent	Código de ascendencia: A - Otro asiático ... Z - Indio asiático	Texto
Premis Cd	El tipo de estructura, vehículo o lugar donde ocurrió el delito.	Número
Premis Desc	Define el Código de Premisa proporcionado.	Texto
Weapon Used Cd	El tipo de arma utilizada en el delito.	Texto
Weapon Desc	Define el código de arma utilizada proporcionado.	Texto

Continúa en la siguiente página

Tabla 3.1 – Continuación de la página anterior

Nombre de Columna	Descripción	Tipo de dato
Status	Estado del caso. (IC es el valor pre-determinado)	Texto
Status Desc	Define el código de estado proporcionado.	Texto
Crm Cd 1	Indica el delito cometido. El Código Penal 1 es el principal y el más grave.	Texto
Crm Cd 2	Puede contener un código para un delito adicional, menos grave que el Código Penal 1.	Texto
Crm Cd 3	Puede contener un código para un delito adicional, menos grave que el Código Penal 1.	Texto
Crm Cd 4	Puede contener un código para un delito adicional, menos grave que el Código Penal 1.	Texto
LOCATION	Dirección del incidente del crimen redondeada a la centena de cuadra más cercana para mantener el anonimato.	Texto
Cross Street	Cruce de Calle de Dirección Redondeada	Texto
LAT	Latitud	Número
LON	Longitud	Número

Tabla 3.1: Descripción de las columnas del conjunto de datos de LAPD

Fuente: ([s.f.](#)). «Crime Data from 2020 to Present».

Dicho conjunto de datos se seleccionarán las columnas importantes que servirán para el entrenamiento del modelo propuesto, los cuales son las columnas CRM Cd Desc, Date Rptd, LAT y LON. Con respecto al tipo de crimen, este conjunto de datos contiene múltiples registros que van desde vandalismo callejero a fraude financiero, por lo que se decidió filtrar manualmente en la columna CRM Cd Desc lo siguiente: VEHICLE - STOLEN, BURGLARY FROM VEHICLE, BIKE - STOLEN, SHOPLIFTING-GRAND THEFT (\$950.01 &

OVER), BATTERY - SIMPLE ASSAULT, CRM AGNST CHLD (13 OR UNDER) (14-15 & SUSP 10 YRS OLDER), ASSAULT WITH DEADLY WEAPON, AGGRAVATED ASSAULT, CRIMINAL THREATS - NO WEAPON DISPLAYED, THEFT FROM MOTOR VEHICLE - PETTY (\$950 & UNDER) BURGLARY, THEFT PLAIN - PETTY (\$950 & UNDER), THEFT FROM MOTOR VEHICLE - GRAND (\$950.01 AND OVER), ROBBERY, BUNCO, GRAND THEFT, BATTERY WITH SEXUAL CONTACT, SHOPLIFTING - PETTY THEFT (\$950 & UNDER), VANDALISM - FELONY (\$400 & OVER, ALL CHURCH VANDALISMS), RAPE, FORCIBLE, VANDALISM - MISDEAMEANOR (\$399 OR UNDER), OTHER ASSAULT, PICKPOCKET, DISTURBING THE PEACE, BUNCO, ATTEMPT, PEEPING TOM, ATTEMPTED ROBBERY, CHILD STEALING, INDECENT EXPOSURE, STALKING, BURGLARY, ATTEMPTED, RAPE, ATTEMPTED, DISCHARGE FIREARMS/SHOTS FIRED, VEHICLE - ATTEMPT STOLEN, BURGLARY FROM VEHICLE, ATTEMPTED THEFT, PERSON, VEHICLE, STOLEN - OTHER (MOTORIZED SCOOTERS, BIKES, ETC), THEFT FROM PERSON - ATTEMPT, BOMB SCARE, ASSAULT WITH DEADLY WEAPON ON POLICE OFFICER, SHOTS FIRED AT INHABITED DWELLING KIDNAPPING - GRAND ATTEMPT, SHOTS FIRED AT MOVING VEHICLE, TRAIN OR AIRCRAFT, THROWING OBJECT AT MOVING VEHICLE KIDNAPPING, CRIMINAL HOMICIDE, PURSE SNATCHING, THEFT FROM MOTOR VEHICLE - ATTEMPT, SHOPLIFTING - ATTEMPT, PROWLER, MANSLAUGHTER, NEGLIGENT, PURSE SNATCHING - ATTEMPT, BIKE - ATTEMPTED STOLEN, PICKPOCKET, ATTEMPT.

Todos los tipos de crímenes mencionados están relacionados con robos, asaltos, vandalismo, agresión física o sexual en las zonas públicas, asalto a pequeñas tiendas, acoso, intento de robo, entre otros. La principal razón por la que se debe filtrar este tipo de crímenes es para que el modelo pueda predecir las rutas seguras en base a crímenes perpetrados en las calles. Esto da más de seiscientos mil registros.

Entregable: Conjunto de datos con columnas requeridas para el entrenamiento del modelo filtrado por crímenes ocasionados en las calles de la ciudad.

Además de la adquisición del conjunto de datos, se debe obtener el mapa de la zona de la ciudad de Los Ángeles, con el fin de tener las calles, avenidas, estaciones y vías que servirá al modelo para la creación del grafo. Para esto se hará uso de la librería de OpenStreetMap llamada “OSMX” y qué en base a una consulta realizada en Python filtrado con los parámetros “highway”, “motorway”, “trunk”, “primary”, “secondary”, “tertiary”, se obtendrá los nodos y vértices de la ciudad.

Entregable: Grafo de las calles de la ciudad de Los Ángeles obtenidos de la librería OSMX.

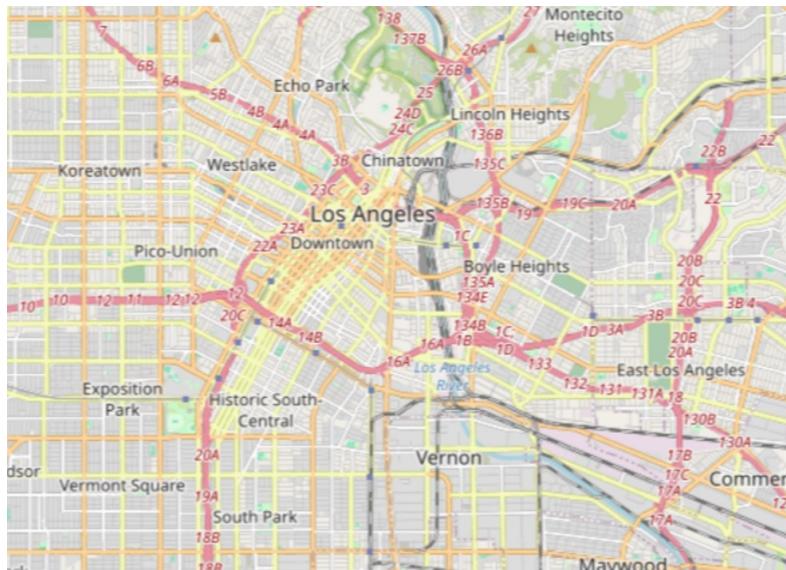


Figura 3.3: Zona de la ciudad de Los Ángeles en OpenStreetMap

3.5.1.2. Preprocesamiento de datos

En esta fase se realizará 4 pasos para que la carga de la información esté lista y limpia para hacer su uso en la etapa de entrenamiento del modelo, por lo que se detallará a continuación los siguientes pasos.

a. Limpieza de dato

Este primer paso ayudará en primera instancia a eliminar los datos vacíos que se encuentran en la base de datos de registro de crímenes. Esto puede causar problemas a la hora de entrenar el modelo y reducir la precisión de este, por lo que es esencial mantener la integridad del conjunto de datos, para esto, Python cuenta con múltiples librerías o lista de comandos que le permite eliminar los datos faltante según los parámetros que se le imponga, una de las más conocidas es “.dropna()”.

El segundo paso el cambio de tipo de variable, ya que el conjunto de datos está compuesto mayoritariamente por variables de tipo texto que no deberían ser consideradas de esa manera, hacer un correcto cambio en el tipo de variables permitirá hacer un análisis de series de tiempo más eficiente y preciso o aplicarla correctamente en los pasos posteriores como la aplicación de regresión LOESS. Python también ofrece múltiples listas de comandos que permiten cambiar el tipo de variable según las necesidades requeridas del proyecto.

Entregable: Conjunto de datos completo con la correcta clasificación de sus variables.

b. Aplicación Regresión LOESS

La Regresión LOESS es un método que combina la regresión lineal con la regresión no lineal, ajustando modelos sencillos sobre subconjuntos locales de datos para la creación de una función que describa la variación de los datos. En cada punto del conjunto de datos se ajusta una regresión polinómica dando más peso a los puntos cercanos y menos peso a los más lejanos, esto puede ayudar para la determinar la relación entre entre las variables y sobre todo, el manejo de datos ruidosos, es decir, si presenta una gran variabilidad en los datos, por lo que puede resaltar las tendencias para entender mejor las áreas donde hay más crímenes, proporcionando una visión más clara de los patrones de los datos y mejorando la identificación de zonas de alto riesgo ([MAXIMA FORMACION, s.f.](#)).

Para esto, se selecciona primero cada punto de interés X_i , para la asignación de pesos se utiliza la siguiente fórmula:

$$w_i(x) = \left(1 - \left|\frac{x-x_i}{d_{max}}\right|^3\right)^3 \quad (\text{Ecuación 3.1})$$

Donde d_{max} es la distancia máxima dentro del vecindario. Adicionalmente a esto, se realiza el ajuste del Polinomio local en término de matrices, el cual se hace minimizando la suma ponderada de los errores cuadrado:

$$\sum_i w_i(x)(y_i - (\beta_0 + \beta_1(x_i - x)))^2 \quad (\text{Ecuación 3.2})$$

Donde y_i son los valores de respuesta.

Entregable: Conjunto de datos suavizados de los crímenes con baja variabilidad de estos.

c. Normalización

Dado a que se trabaja con la latitud y longitud, es importante el proceso de la normalización para que tengan una escala similar, esto ayuda a mejorar la eficiencia y precisión del modelo. Se hará uso de la normalización Min-Max, ya que asegura que todos los atributos están en el rango de 0 a 1, asimismo ayuda a preservar las relaciones entre los valores originales y es compatible con muchos algoritmos([Medium, 2023a](#)).

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (\text{Ecuación 3.3})$$

Donde:

x es el valor original.

x_{min} es el valor mínimo de la característica.

x_{max} es el valor máximo de la característica.

x' es el valor normalizado.

Entregable: Conjunto de datos normalizados en un mismo rango.

d. Normalización

Este paso es necesario para el análisis y toma de decisiones del modelo, por lo que se debe contar con la estructura geográfica de la ciudad representada en grafo de OpenStreetMap, junto con el conjunto de datos ya normalizados. Para la evaluación del riesgo de una ruta, se debe asociar cada registro de crimen con el vértice más cercano (intersección) o con la arista (calle) dónde ocurrió en la ciudad, para esto se utiliza la distancia euclídea, el cual se representa de la siguiente manera:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (\text{Ecuación 3.4})$$

Dónde (x_1, y_1) son las coordenadas del crimen y (x_2, y_2) son las coordenadas del vértice del grafo.

Entregable: Conjunto de datos integrado en el grafo de la ciudad para su uso en la fase de entrenamiento.

3.5.1.3. Definición de entorno y modelo DGN

La definición de un entorno para el entrenamiento del modelo es fundamental para proporcionar una plataforma robusta y flexible para encontrar rutas seguras en el grafo de la ciudad, para esto se hará uso del entorno Gym, que es un paquete proporcionado por Python y que da la facilidad de establecer reglas y la capacidad de interactuar con el entorno del grafo de la ciudad para la toma de decisiones y recibir recompensas basadas en el riesgo ([Universitat Oberta de Catalunya, s.f.](#)). Asimismo, Gym puede facilitar la implementación de algoritmos DGN y diferentes algoritmos DRL, además de ser compatibles con diversas bibliotecas de Aprendizaje Profundo como TensorFlow, Pytorch, entre otros que facilita la integración de los algoritmos de DRL como DQN. Para la elección de un modelo DGN, se optó por utilizar el modelo Graph Convolutional Network (GCN), por su capacidad de modelar relaciones complejas entre los nodos y aristas en grafos para saber cómo están distribuidos los crímenes a lo largo de las calles y vecindarios. Asimismo, GCN tiene la capacidad de procesar eficientemente datos espaciales en el grafo que le permite integrar los datos de los registros de crímenes ya que se manejan según coordenadas. Debido a que la ciudad de Los Ángeles está compuesta por muchas calles

y avenidas que se interceptan entre sí, esto hace que sea un grafo grande y complejo, el cual un modelo GCN es flexible y adaptable a ese sin necesidad de establecer características específicas ([Bernstein M., s.f.](#)).

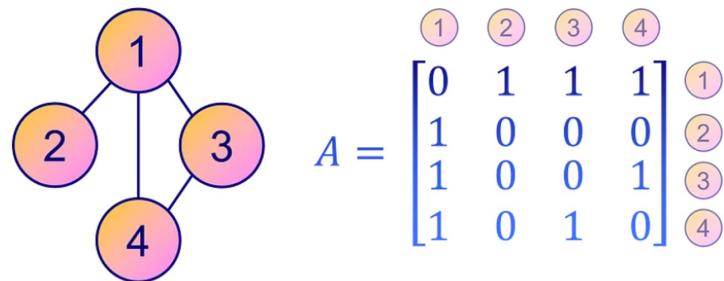


Figura 3.4: Estructura simple de un grafo

Fuente: ([2023b](#)). «Graph Convolutional Networks: Introduction to GNNs».

Entregable: Entorno de trabajo en Gym utilizando el modelo GCN para la fase del entrenamiento.

3.5.1.4. Entrenamiento del Modelo

Para el entrenamiento del modelo DGN, este será realizado en el entorno de TensorFlow, el cual permite la construcción de modelos de Aprendizaje Profundo complejos y personalizados, definiendo capas personalizadas, funciones de activación y de pérdida. Otra de las razones es porque está optimizado para realizar cálculos en grandes volúmenes de datos y que está respaldado por Google y una gran comunidad de desarrolladores, brindándonos abundante contenido y documentación para la implementación efectiva del GCN

a. Entrenamiento del modelo DGN

Para el entrenamiento del modelo DGN, este será realizado en el entorno de TensorFlow, el cual permite la construcción de modelos de Aprendizaje Profundo complejos y personalizados, definiendo capas personalizadas, funciones de activación y de pérdida. Otra de las razones es porque está optimizado para realizar cálculos en grandes volúmenes de datos y que está respaldado por Google y una gran comunidad de desarrolladores, brindándonos abundante contenido y documentación para la implementación efectiva del GCN.

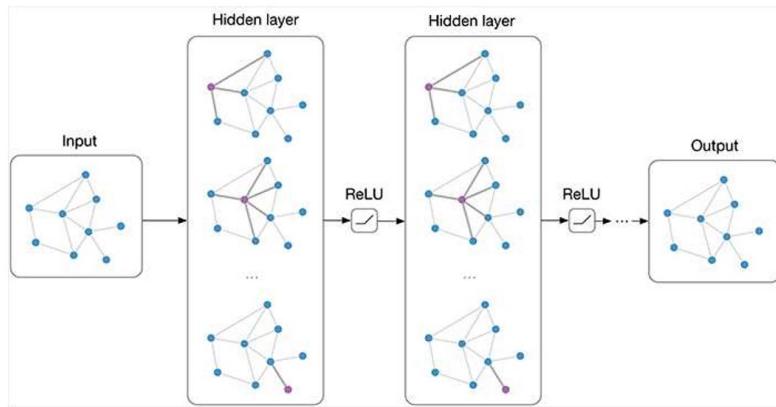


Figura 3.5: Estructura de GCN

Fuente: (2020). «Graph Neural Networks - An overview».

En cuanto a las funciones de activación, se utilizará la función de activación ReLU (Rectified Linear Unit), ya que el modelo GCN requiere de funciones de activación no lineales para aprender y modelar las relaciones complejas entre los datos y mejora el rendimiento del modelo teniendo una mejor capacidad de identificar zonas de alto riesgo de crimen. Esta función de activación es representada matemáticamente como:

$$\text{ReLU} = \max(0, x) \quad (\text{Ecuación 3.5})$$

Donde x es un valor de entrada, y que la función devuelve 0 si en caso x es negativo y si es positivo devuelve x .

Esta función de activación ReLU será empleada en cada capa oculta del modelo.

En cuanto a la función de activación de la capa de salida, se utilizará la función Sigmoid, el cuál comprime los valores de entrada a un rango entre 0 y 1, útil para predecir la probabilidad de riesgo de crimen para cada nodo del grafo y que de esta manera el agente pueda determinar qué ruta tomar en modelo DRL. La función de activación es representada matemáticamente de la siguiente manera:

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (\text{Ecuación 3.6})$$

Se emplea la fórmula de propagación el cual permite actualizar las características de cada nodo en función de las características de sus vecinos, con el fin de capturar relaciones estructurales en el grafo, este aprenderá los pesos con el fin de minimizar la función de pérdida, ajustando para mejorar la predicción del riesgo. Esta fórmula es representada de la siguiente manera:

$$h_v^{l+1} = \sigma \left(\sum_{u \in N(v)} \frac{1}{c_{uv}} h_u^l W^l \right) \quad (\text{Ecuación 3.7})$$

Donde:

h_v^l es la representación del nodo v en la capa l .

$N(v)$ son los nodos vecinos de v .

c_{uv} es el factor de la normalización de la arista entre u y v .

W^l es la matriz de pesos en la capa l .

σ es la función de activación.

Por último se emplea un optimizador para el entrenamiento del modelo en TensorFlor, siendo el más conocido el Optimizador Adam (Adaptive Moment Estimation), que funciona en diversos problemas, ajustando automáticamente la tasa de aprendizaje en función de su estimación del momento ([InteractiveChaos, s.f.](#)).

Entregable: Representaciones de las características de riesgo de crimen en un grafo para el entrenamiento del modelo DRL y la probabilidad de riesgo de cada nodo.

b. Entrenamiento del modelo de Aprendizaje por Refuerzo Profundo

- Establecer política de recompensa

Para el establecimiento de una política de recompensa, la cual le servirá al agente determinar qué ruta será la más segura, se hará uso de las probabilidades de riesgo de cada nodo obtenidas en el entrenamiento del modelo DGN, además de establecer un umbral, que es de 0.6 y que representa el valor máximo que puede tomar el agente para elegir qué ruta tomar. Para esto, se propone la siguiente fórmula:

$$\text{Recompensa} = \begin{cases} R_{safe} & \text{si Indice de Riesgo} \leq \theta \\ R_{risk} & \text{si Indice de Riesgo} > \theta \end{cases} \quad (\text{Ecuación 3.8})$$

En cada nodo que el agente visite, este determinará mediante la función de recompensa si debe tomar dicho nodo, si en caso el índice de riesgo es menor al umbral, lo selecciona, caso contrario, lo evita.

Entregable: Función de recompensa para el entrenamiento del modelo DQN.

- Entrenamiento DQN

Para que el modelo aprenda a elegir la ruta más segura basándose en la política de recompensa ya establecida anteriormente, se decide entrenar un agente en un modelo DQN, ya que tiene la capacidad de manejar problemas de toma de decisiones mediante la interacción en entornos complejos y de alta dimensionalidad como lo

es el grafo de la ciudad de Los Ángeles. Para el correcto entrenamiento del modelo DQN, se debe considerar los siguientes parámetros:

- **Replay Buffer Size:** Este parámetro tiene como propósito el de almacenar las experiencias del agente, es decir, es una memoria que incluye el estado, la acción tomada, la recompensa recibida y el siguiente estado resultando, haciendo que el entrenamiento sea más estable y eficiente ([RAY, s.f.](#)). El tamaño que se define comúnmente es de 100,000 experiencias. Sin embargo, el establecimiento del tamaño de experiencia depende de la memoria y tiempo de muestreo durante el entrenamiento.
- **Minibatch Size:** Número de experiencias muestreadas del parámetro anterior de cada red neuronal y ayuda a estabilizar el aprendizaje con un correcto uso de los recursos computacionales. Los valores típicos son 32 o 64.
- **Learning rate:** Es la tasa de aprendizaje del modelo y el encargado de ajustar o controlar el tamaño de los pesos de la red neuronal. El valor comúnmente utilizado es 0.001. Se debe evitar utilizar una tasa de aprendizaje mayor ya que puede volver inestable el modelo, asimismo, si la tasa es muy baja puede aumentar el tiempo de aprendizaje del modelo DRL ([Medium, 2020](#)).

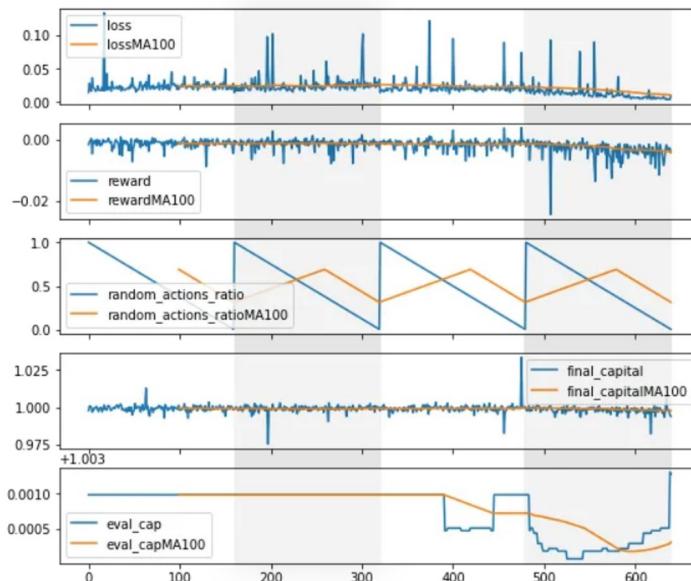


Figura 3.6: Experimentación de tasa de aprendizajes

Fuente: ([2020](#)). «[Finding a learning rate in Deep Reinforcement Learning](#)».

- **Discount Factor:** Determina la importancia de las recompensas, controlando la valoración que le asigna el agente a la recompensa frente a las recompensas futuras. El valor de recompensa varía de 0 a 1, siendo los valores típicos entre

0.9 y 0.99; este valor determina que tan importante es para el agente las recompensas futuras, por lo que si es bajo, valorará más las recompensas inmediatas. Esto puede ayudar al modelo de detección de rutas seguras para saber si vale la pena seguir al siguiente nodo.

- **Exploration Rate:** Es un parámetro que controla la probabilidad de que el agente tome una acción, esto facilita el equilibrio entre la política aprendida y la exploración de nuevas acciones. Este paso es importante para el modelo, porque le permitirá explorar más las diversas calles de la ciudad, que serían los nodos, para que el agente descubra nuevas estrategias. Usualmente se inicia con un valor alto de 1 en exploración y mediante se va entrenando el modelo va disminuyendo gradualmente.
- **Target Network Update Frequency:** Es la frecuencia con la que se actualiza la red con el fin de estabilizar el entrenamiento con número de pasos establecidos. Se debe seleccionar la cantidad de pasos según lo requiera el modelo, ya que una frecuencia de actualización más alta puede hacer que la red se desactualice o si en caso es muy baja, se desestabilice el entrenamiento debido a los cambios realizados por la red. Los valores típicos suelen ser de 10000 pasos.

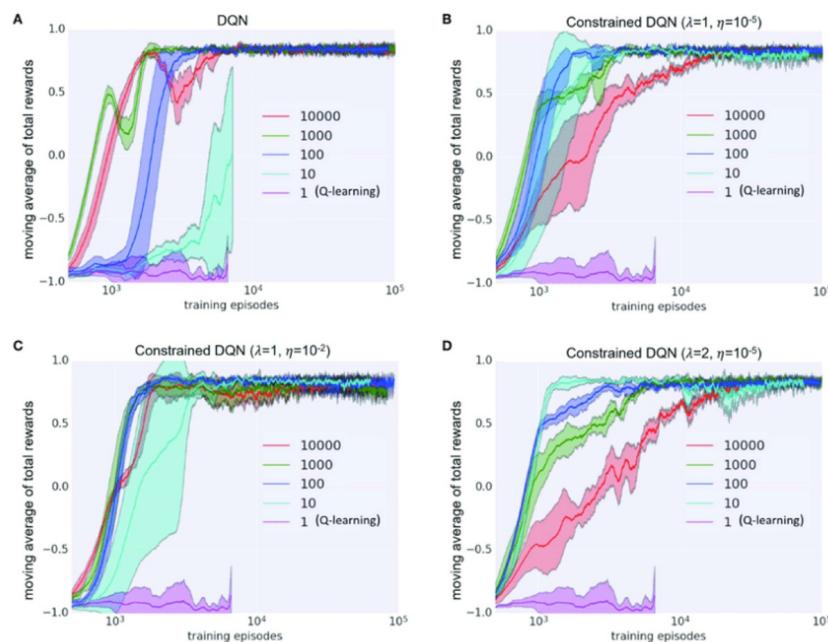


Figura 3.7: Efectos de la frecuencia de actualización de la red objetivo en la tarea del laberinto MNIST

Fuente: (2019). «Constrained Deep Q-Learning Gradually Approaching Ordinary Q-Learning».

Entregable: Modelo DQN entrenado con una política optimizada de mapeo de estado del entorno a la acción que maximiza la recompensa esperada.

c. Planificar Ruta

Tomando en cuenta el entrenamiento del modelo DQN y la política optimizada, el agente utiliza dicha política desde el punto de origen para decidir el siguiente nodo en la ruta. Para esto, se utiliza una propagación hacia delante, donde cada paso que realiza el agente consulta a la política para saber si la acción minimiza el riesgo de crimen, asegurando que siempre se elija la mejor opción. Luego este proceso se realiza iterativamente hasta que el agente alcance el destino, construyendo una ruta completa y coherente. Además, se registra los índices de riesgo de cada nodo o arista seleccionada por el agente y se representa matemáticamente de la siguiente manera:

$$R_{total} = \sum_{i=1}^N r_i \quad (\text{Ecuación 3.9})$$

Donde r_i es la probabilidad de riesgo de cada nodo o arista i , lo que proporciona una medida cuantitativa del riesgo para evaluar la efectividad del modelo.

Entregable: Subgrafo de la ruta resultante y el índice total de riesgo.

3.5.2. Metodología para la medición de resultados

3.5.2.1. Evaluación del Modelo

- Root Mean Squared Error (RMSE)

Se utiliza el RMSE para medir la precisión del modelo DGN propuesto en cuanto a las probabilidades de riesgo, cuantificando la desviación de las probabilidades de riesgo de cada nodo. Para esto, se utiliza la fórmula tradicional que es la siguiente:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (\text{Ecuación 3.10})$$

Donde:

N es el total de nodos del grafo.

y_i es la probabilidad de riesgo objetivo para nodo i .

\hat{y}_i es la probabilidad de riesgo predicha por el modelo DGN para nodo i .

Si el valor es bajo, quiere decir que las estimaciones de riesgo son precisas y asegura que el modelo propuesto es efectivo para la identificación de zonas de alto riesgo para que el agente pueda evitar dichas zonas en el proceso de entrenamiento DQN.

Entregable: Índice de efectividad del modelo DGN.

- Indicador de Riesgo

Se utiliza la función de riesgo total establecido para la planificación de la ruta, y se evalúa el riesgo a cada ruta propuesta. Asimismo se establece un umbral que servirá de comparación con el riesgo total de una ruta, el cual determinará si la ruta se acepta o no. A continuación, se presenta el flujo condicional para la aceptación de la ruta:

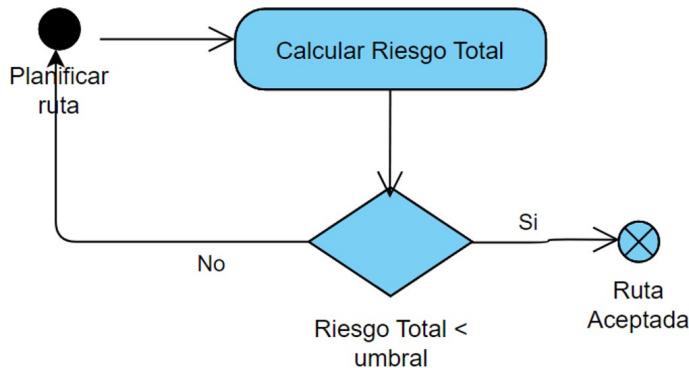


Figura 3.8: Elaboración propia: Flujo de condición de elección de ruta

Entregable: Ruta Segura con el menor índice de riesgo aceptada.

- Indicador de Eficiencia (TTM)

El indicador de Eficiencia (TTM) evalúa el rendimiento del modelo DRL, con el fin de proporcionar una medida cuantitativa para el tiempo de procesamiento del modelo y que se determina de la siguiente manera:

$$TTM = N_{ep} \times T_{ep} \quad (\text{Ecuación 3.11})$$

Donde:

N_{ep} es el número de épocas del modelo.

T_{ep} el tiempo promedio necesario para completar una época en el entrenamiento.

Asimismo, se evalúa otro indicador de Eficiencia (TTM) que es para determinar el tiempo promedio de la ruta seleccionada, y puede ser representada de la siguiente manera:

$$TTM = \frac{\text{Distancia de la ruta}}{\text{Velocidad Promedio}} \quad (\text{Ecuación 3.12})$$

Entregable: Indicador de eficiencia del modelo DRL y de la ruta seleccionada.

3.5.2.2. Visualización de la ruta

Debido a que se trabaja con datos geoespaciales y el empleo de un modelo de planificación de rutas seguras, se propone utilizar la herramienta Folium en Python, ya que permite crear mapas interactivos que pueden ser explorados desde un navegador web, asimismo, se integra fácilmente con otras bibliotecas como Pandas, y personalizable. Esto es de ayuda para identificar patrones espaciales de riesgo y como la ruta seleccionada evita las zonas de alto riesgo.

Entregable: Visualización de la ruta segura seleccionada.

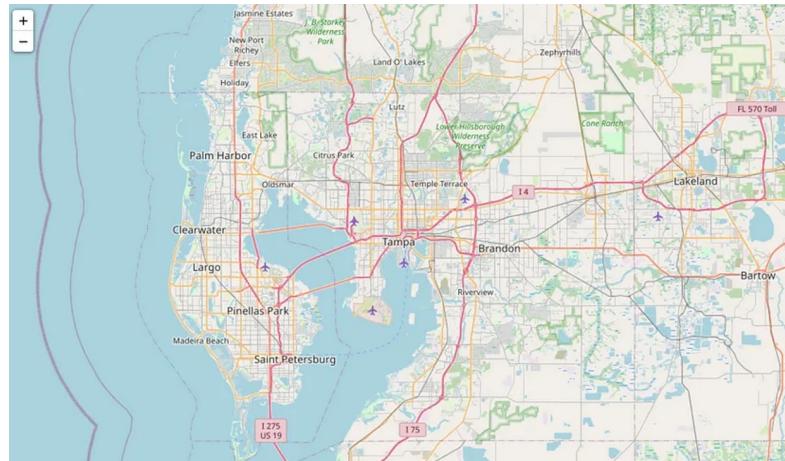


Figura 3.9: Mapa de Tampa, Florida, creado en Folium y OpenStreetMap

Fuente: (2023c). «How to Generate Interactive Maps with Folium».

3.6. Cronograma de actividades y presupuesto

Se elaboró un cronograma de actividades de toda la investigación, mostrada en la Figura 3.10, contemplando desde el inicio del mes de abril 2024 hasta la sustentación del trabajo estimado para inicios de diciembre 2024.

ACTIVIDADES / TAREAS	Abril				Mayo				Junio				Julio				Agosto				Setiembre				Octubre				Noviembre				Diciembre							
					Mes 1				Mes 2				Mes 3				Mes 4				Mes 5				Mes 6				Mes 7				Mes 8				Mes 9			
	Semana	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4			
INICIO																																								
Selección de tema de Investigación																																								
Redacción del título																																								
Elaboración del Plan de Investigación																																								
Identificación de Realidad Problemática																																								
Planteamiento del Problema																																								
Planteamiento de los Objetivos																																								
Justificación y Delimitación																																								
Planteamiento de Hipótesis																																								
MARCO TEÓRICO																																								
Revisión Bibliográfica																																								
Bases teóricas																																								
Marco Conceptual																																								
METODOLOGÍA																																								
Recolección de los Datos																																								
Preparación de los Datos																																								
Desarrollo de la Solución																																								
Desarrollo de la Evaluación del modelo																																								
Discusión sobre los Resultados																																								
Despliegue																																								
Conclusiones y Recomendaciones																																								
CIERRE																																								
Elaboración del Informe Final																																								
Presentación																																								
Sustentación																																								

Figura 3.10: Cronograma de actividades de la investigación.

Fuente: Elaboración propia.

A continuación, se muestra los costos personales del autor para la resolución del proyecto.

Tabla 3.2: Presupuesto de los costos personales del autor.

Item	Tiempo usado (horas)	Costo (soles)	Subtotal
Recursos materiales			
Laptop HP Pavilion Gaming Laptop 15 core i7 9750h 16GB		S/.3,208.50	S/.3,208.50
Servicios generales			
Internet + luz (9 meses)	110	S/.327.00	S/.2,943.00
Total			S/.6,151.50

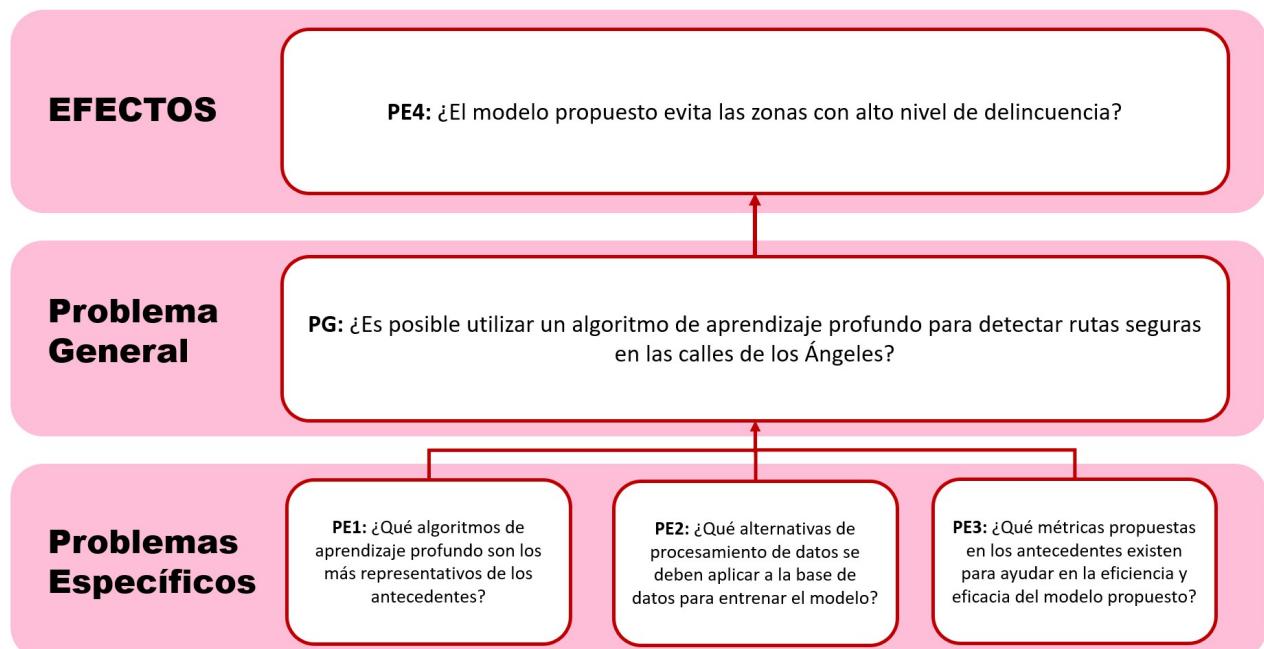
Fuente: Elaboración propia.

Anexos

Anexos A

Anexo I: Matriz de Consistencia

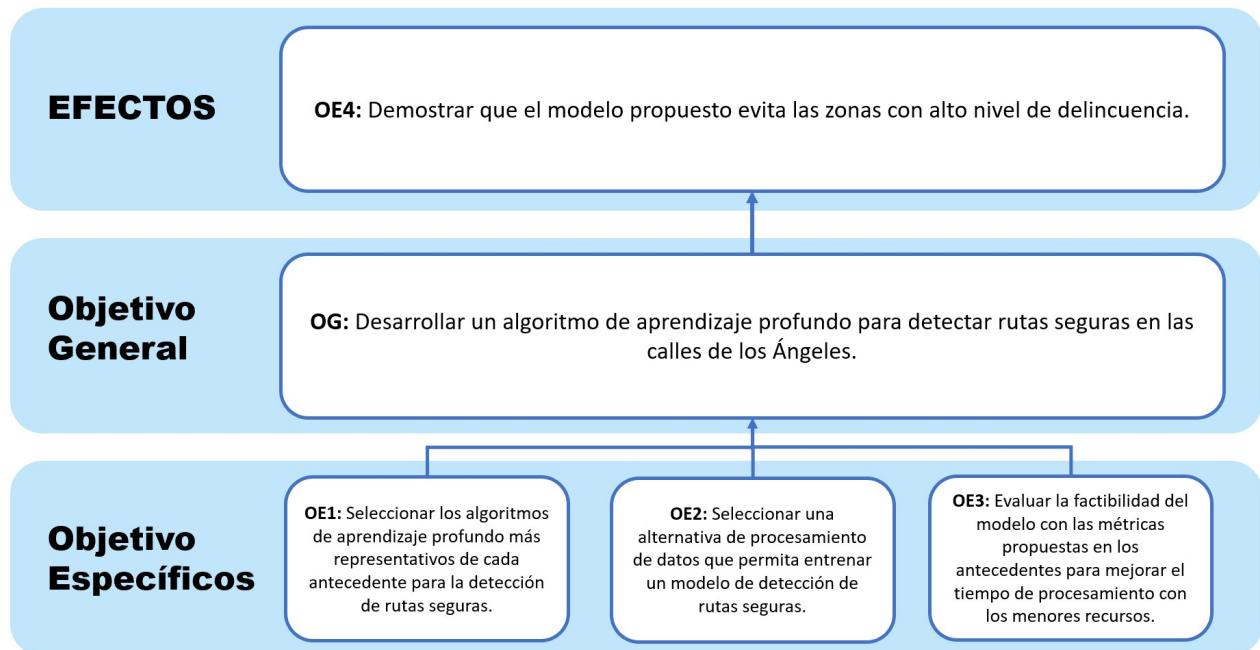
A.1. Árbol de Problemas



PROBLEMAS	OBJETIVOS	HIPÓTESIS
Problema General	Objetivo General	Hipótesis General
Problemas Específicos	Objetivos Específicos	Hipótesis Específicas
¿Es posible utilizar un algoritmo de aprendizaje profundo para detectar rutas seguras en las calles de los Ángeles?	Desarrollar un algoritmo de aprendizaje profundo para detectar rutas seguras en las calles de los Ángeles.	Mediante el desarrollo de un algoritmo de aprendizaje profundo se logra detectar rutas seguras en las calles de los Ángeles.
¿Qué algoritmo de aprendizaje profundo es el más representativo de los antecedentes?	Determinar qué algoritmo de aprendizaje profundo es el más representativo de cada antecedente para la detección de rutas seguras.	La selección del algoritmo de aprendizaje profundo más representativos de cada antecedente mejora significativamente la precisión y eficiencia al detectar rutas seguras.
¿Qué alternativas de procesamiento de datos se deben aplicar a la base de datos para entrenar el modelo?	Seleccionar una alternativa de procesamiento de datos que permita entrenar un modelo de detección de rutas seguras.	La identificación de una alternativa de procesamiento de datos influye positivamente para poder entrenar un modelo de detección de rutas seguras.
¿Qué métricas propuestas en los antecedentes existen para ayudar en la eficiencia y eficacia del modelo propuesto?	Evaluando la factibilidad del modelo con las métricas propuestas en los antecedentes para mejorar el tiempo de procesamiento con los menores recursos.	La utilización de métricas propuestas en los antecedentes es factible para optimizar el tiempo de procesamiento del modelo.
¿El modelo propuesto evita las zonas con alto nivel de delincuencia?	Demostrar que el modelo propuesto evita las zonas con alto nivel de delincuencia.	Se demuestra que el modelo propuesto evita las zonas con alto nivel de delincuencia.

Tabla A.1: Matriz de consistencia. Fuente: Elaboración propia

A.2. Árbol de Objetivos



Anexos B

Anexo II: Resumen de Papers investigados

Tipo	Nº	Título	Autor	Año	País	Fuente
Problema	1	Predicting motor vehicle theft in Santiago de Chile using graph-convolutional LSTM.	Esquivel, N., Nicolis, O., & Márquez, B. P	2020	Chile	39th International Conference of the Chilean Computer Science Society (SCCC)
	2	SafeRoute: Learning to navigate streets safely in an urban environment	Levy, S., Xiong, W., Belding, E., & Wang, W. Y.	2020	USA	ACM Transactions on Intelligent Systems and Technology (TIST)
Propuesta	3	Route-the safe: A robust model for safest route prediction using crime and accidental data.	Soni, S., Shankar, V. G., & Chaurasia, S.	2019	India	Int. J. Adv. Sci. Technol
	4	Real-time deep reinforcement learning based vehicle navigation	Koh, S., Zhou, B., et al.	2020	Reino Unido	Applied Soft Computing
	5	An Enhanced Motif Graph Clustering-Based Deep Learning Approach for Traffic Forecasting	Zhang, C., Zhang, S., et al.	2020	China	IEEE
Técnica	6	Algoritmo para calcular la ruta más segura y óptima	Cano, S., & Tabares, S. A. A.	2022	Colombia	OSF Repository
	7	A reinforcement learning-based routing algorithm for large street networks. International Journal of Geographical Information Science	Li, D., Zhang, Z., Alizadeh, B., et al.	2024	USA	International Journal of Geographical Information Science
	8	Real-time safest route identification: Examining the trade-off between safest and fastest routes	Ghoul, T., Sayed, T., & Fu, C.	2023	Canada	Analytic methods in accident research

Tabla B.1: Cuadro Resumen de Papers investigados. Fuente: Elaboración propia

BIBLIOGRAFÍA

- AcademiaLab. (s.f.). Proceso de decisión de markov. https://academia-lab.com/encyclopedia/proceso-de-decision-de-markov/%5C#google_vignette
- AI SUMMER. (2020). Graph Neural Networks - An overview. https://theaisummer.com/Graph_Neural_Networks/
- Areiza, S., Cano, S., Serna, A., & Riascos, J. (2022). ALGORITMO PARA CALCULAR LA RUTA MÁS SEGURA Y ÓPTIMA. <https://doi.org/10.31219/osf.io/wd73v>
- Azure. (s.f.). ¿Qué es el aprendizaje automático? <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-machine-learning-platform>
- Bernstein M. (s.f.). Graph convolutional neural networks. <https://mbernste.github.io/posts/gcn/>
- BotPress. (2022). ¿Qué es una red neuronal profunda? <https://botpress.com/es/blog/deep-neural-network>
- Castillo, L. (2023). Robo de autos en Perú: ¿Cuántos autos se roban al día? <https://www.tracklink.pe/contenido/blog/robo-de-autos-peru-cuantos-autos-roban-dia/>
- CEPLAN. (2023, marzo). Análisis del crecimiento y expansión urbana a nivel nacional y el impacto a nivel regional. https://geo.ceplan.gob.pe/uploads/Analisis_crecimiento_expansion_urbana.pdf
- CEUPE. (s.f.). Aprendizaje por refuerzo: Concepto, características y ejemplo. <https://www.ceupe.com/blog/aprendizaje-por-refuerzo.html?dt=1717951617262>
- Comisión Europea. (s.f.). El aprendizaje basado en refuerzos mejora la toma de decisiones. <https://cordis.europa.eu/article/id/190536-better-decisions-through-reinforcement-learning/es>
- ConceptoDefinicion. (2023). Definición de Riesgo. <https://conceptodefinicion.de/riesgo/>
- Conceptualia. (s.f.). Delincuencia. <https://conceptualia.es/cultura-y-sociedad/sociedad/delincuencia/>
- DAMAVIS. (2022). Aprendizaje por refuerzo profundo: DQN. <https://blog.damavis.com/aprendizaje-por-refuerzo-profundo-dqn/>
- DeepMind. (2016). Deep Reinforcement Learning. <https://deepmind.google/discover/blog/deep-reinforcement-learning/>

- Díaz, J. (2021). Machine Learning and Deep Learning. *Ingeniare*, 29(2), 182-183. <https://doi.org/10.4067/S0718-33052021000200180>
- Enciclopedia Humanidades. (s.f.). Zona urbana. <https://humanidades.com/zona-urbana/>
- Esquivel, N., Nicolis, O., & Márquez, B. P. (2020). Predicting Motor Vehicle Theft in Santiago de Chile using Graph-Convolutional LSTM. *2020 39th International Conference of the Chilean Computer Science Society (SCCC)*, 1-7. <https://doi.org/10.1109/SCCC51225.2020.9281174>
- Gamco. (s.f.). ¿Qué es Aprendizaje por refuerzo profundo? <https://gamco.es/glosario/aprendizaje-por-refuerzo-profundo/>
- GeeksforGeeks. (2023). A Beginner's Guide to Deep Reinforcement Learning. <https://www.geeksforgeeks.org/a-beginners-guide-to-deep-reinforcement-learning/>
- Ghoul, T., Sayed, T., & Fu, C. (2023). Real-time safest route identification: Examining the trade-off between safest and fastest routes. *Analytic Methods in Accident Research*, 39, 100277. <https://doi.org/https://doi.org/10.1016/j.amar.2023.100277>
- Hernández, L. (2022). ¿Cuántos usuarios consultan Waze mensualmente? <https://noticias.autocosmos.com.co/2022/08/29/cuantos-usuarios-consultan-waze-mensualmente>
- Huang, Z., Zhang, S., Xi, C., Liu, T., & Zhou, M. (2021). Scaling Up Graph Neural Networks Via Graph Coarsening.
- IBM. (s.f.-a). ¿Qué es Deep Learning? <https://www.ibm.com/es-es/topics/deep-learning>
- IBM. (s.f.-b). ¿Qué es el aprendizaje supervisado? <https://www.ibm.com/es-es/topics/supervised-learning>
- IBM. (s.f.-c). Conceptos básicos de ayuda de CRISP-DM. <https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview>
- ILAPD. (s.f.). Crime Data from 2020 to Present. https://data.lacity.org/Public-Safety/Crime-Data-from-2020-to-Present/2nrs-mtv8/about_data
- Instituto de Ingeniería del Conocimiento. (s.f.). Aprendizaje profundo por refuerzo. <https://www.iic.uam.es/aprendizaje-profundo-por-refuerzo/>
- InteractiveChaos. (s.f.). Adam. <https://interactivechaos.com/es/manual/tutorial-de-machine-learning/adam>
- Jin, W., Ma, Y., Liu, X., Tang, X., Wang, S., & Tang, J. (2020). Graph Structure Learning for Robust Graph Neural Networks.
- Koh, S., Zhou, B., Fang, H., Yang, P., Yang, Z., Yang, Q., Guan, L., & Ji, Z. (2020). Real-time deep reinforcement learning based vehicle navigation. *Applied Soft Computing*, 96, 106694. <https://doi.org/https://doi.org/10.1016/j.asoc.2020.106694>
- Levy, S., Xiong, W., Belding, E., & Wang, W. (2020). SafeRoute: Learning to Navigate Streets Safely in an Urban Environment. *ACM Trans. Intell. Syst. Technol.*, 11(6). <https://doi.org/10.1145/3402818>

- Li, Y. (2018). DEEP REINFORCEMENT LEARNING: AN OVERVIEW. <https://arxiv.org/pdf/1701.07274>
- Ma, Y., & Tang, J. (2021). *Deep Learning on Graphs*. Cambridge University. https://yaoma24.github.io/dlg_book/
- Magnus Lofstrom, Brandon Martin. (2023). *Tendencias del Crimen en California* (inf. téc.). PPIC.
- MAXIMA FORMACION. (s.f.). ¿QUÉ ES LA REGRESIÓN LOCAL LOESS O LOWESS? <https://www.maximaformacion.es/blog-dat/que-es-la-regresion-local-loess-o-lowess/>
- Medium. (2020). Finding a learning rate in Deep Reinforcement Learning. <https://nieznam.medium.com/finding-a-learning-rate-in-deep-reinforcement-learning-27ed725f6b90>
- Medium. (2023a). ¿Por qué la normalización es clave e importante en Machine Learning y Ciencia de Datos? <https://jorgeiblanco.medium.com/por-qu%C3%A9-la-normalizaci%C3%93n-es-clave-e-importante-en-machine-learning-y-ciencia-de-datos-4595f15d5be0>
- Medium. (2023b). Graph Convolutional Networks: Introduction to GNNs. <https://towardsdatascience.com/graph-convolutional-networks-introduction-to-gnns-24b3f60d6c95>
- Medium. (2023c). How to Generate Interactive Maps with Folium. <https://towardsdatascience.com/how-to-generate-interactive-maps-with-folium-b232778758c4>
- Melo, Y. (2024). Este es el día que más robos de celulares se registraron en el 2023, según reporte de Osiptel. <https://www.infobae.com/peru/2024/02/05/este-es-el-dia-que-mas-robos-de-celulares-se-registraron-en-el-2023-segun-reporte-de-osiptel/>
- Numbeo. (2023). América: Índice de Criminalidad 2023. <https://es.numbeo.com/criminalidad/clasificaciones-por-region?region=019&title=2023>
- Nvidia. (2022). ¿Qué Son las Graph Neural Networks? <https://la.blogs.nvidia.com/blog/que-son-las-graph-neural-networks/>
- Oracle. (s.f.). ¿Qué es el machine learning? <https://www.oracle.com/pe/artificial-intelligence/machine-learning/what-is-machine-learning/>
- Perú, A. (2024). Venta de vehículos creció 2.4% en el 2023. <https://amcham.org.pe/news/venta-de-vehiculos-crecio-2-4-en-el-2023/#:~:text=La%20industria%20automotriz%20volvi%C3%A9a%20a,%20comparaci%C3%B3n%20con%20el%202022>
- Ramírez, S. (2023). Cada hora roban 200 celulares en el país: ¿Qué hay detrás del aumento de este delito? <https://elcomercio.pe/lima/osiptel-cada-hora-roban-200-celulares-en-el-pais-que-hay-detrás-del-aumento-de-este-delito-operadoras-reportaron-723375-sustracciones-de-celulares-muertes-por-robo-de-celular-noticia/?ref=ecr>
- RAY. (s.f.). Replay Buffers. <https://docs.ray.io/en/latest/rllib/rllib-replay-buffers.html>

- ResearchGate. (2019). Constrained Deep Q-Learning Gradually Approaching Ordinary Q-Learning. https://www.researchgate.net/publication/337860304_Constrained_Deep_Q-Learning_Gradually_Approaching_Oldinary_Q-Learning
- Russell, S., & Norvig, P. (2004). *Inteligencia Artificial: Un Enfoque Moderno* (J. M. Corchado Rodríguez, F. Martín Rubio, J. M. Cadenas Figueredo, L. D. Hernández Molinero, E. Paniagua Arís, R. Fuentetaja Pinzán, M. Robledo de los Santos & R. Rizo Aldeguer, Trad.; 2.^a ed.). Pearson Educación, S.A. <https://luismejias21.files.wordpress.com/2017/09/inteligencia-artificial-un-enfoque-moderno-stuart-j-russell.pdf>
- Shopping. (s.f.). What Is Deep Learning, Nature Of Machine Learning And Beauty Of Deep Neural Networks ? - New. <https://www.the-sdh-v.top/ProductDetail.aspx?iid=1086844820%5C&pr=38.88>
- Significado. (s.f.). Definición de Ruta. <https://significado.com/ruta/>
- Soni, S., Gauri Shankar, V., & Sandeep, C. (2019). Route-The Safe: A Robust Model for Safest Route Prediction Using Crime and Accidental Data. 28, 1415-1428.
- Statista Research Department. (2023). Crímenes con mayor tasa de víctimas en áreas urbanas de Perú entre julio y diciembre del 2023. <https://es.statista.com/estadisticas/1290030/tasa-de-victimas-por-hecho-delictivo-peru/>
- Tableau. (s.f.). Qué es la inteligencia artificial: definición, historia, aplicaciones y futuro. <https://www.tableau.com/es-mx/data-insights/ai/what-is>
- Talaei, T., Ould, H., & Kaabouch, N. (2023). Deep learning: systematic review, models, challenges, and research directions. *Neural Computing and Applications*, 35(31), 23103-23124. <https://doi.org/10.1007/s00521-023-08957-4>
- TechEdu. (s.f.). Red neuronal profunda. <https://techlib.net/techedu/red-neuronal-profunda/>
- Techopedia. (2024). Aprendizaje por refuerzo. <https://www.techopedia.com/es/definicion/aprendizaje-por-refuerzo>
- Universitat Oberta de Catalunya. (s.f.). Gym. <https://datascience.recursos.uoc.edu/es/gym/>
- Zhang, C., Zhang, S., Yu, J. J. Q., & Yu, S. (2020). An Enhanced Motif Graph Clustering-Based Deep Learning Approach for Traffic Forecasting. *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 1-6. <https://doi.org/10.1109/GLOBECOM42002.2020.9322104>