



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

EXPLORANDO EL POTENCIAL DE LAS REDES NEURONALES DE GRAFOS  
PARA EXTRAER INFORMACIÓN DE BGP

TESIS PARA OPTAR AL GRADO DE  
MAGÍSTER EN COMPUTACIÓN

MEMORIA PARA OPTAR AL TÍTULO DE  
INGENIERO EN COMPUTACIÓN

VALENTINA FRANCISCA ESTEBAN LAGOS

PROFESORA GUÍA:  
IVANA BACHMANN ESPINOZA

PROFESOR CO-GUÍA:  
SEBASTIÁN FERRADA

SANTIAGO DE CHILE  
2025

*dedicatoria.*

# Índice

<b>1.</b>	<b>Introducción .....</b>	<b>1</b>
	1.1. Hipótesis .....	1
	1.2. Objetivos .....	2
	1.2.1. Objetivo general .....	2
	1.2.2. Objetivos específicos .....	2
	1.3. Metodología .....	2
	1.4. Contribuciones .....	3
	1.5. Estructura del trabajo .....	3
<b>2.</b>	<b>Preliminares .....</b>	<b>4</b>
	2.1. Internet .....	4
	2.1.1. Sistemas Autonomos .....	5
	2.1.2. Ruteo .....	5
	2.2. Border Gateway protocol (BGP) .....	6
	2.2.1. Funcionamiento de BGP .....	6
	2.2.2. BGP Routing Information Base (RIB) .....	7
	2.3. Graph Neuronla Network (GNN) .....	8
	2.4. Trabajo Relacionado .....	9
	2.4.1. Tipos de Relaciones .....	9
	2.4.1.1. Gao .....	10
	2.4.1.2. Lu Rank .....	11
	2.4.1.3. BGP2Vec .....	13
	2.4.2. ProbLink .....	14
<b>3.</b>	<b>Metodologia .....</b>	<b>15</b>
	3.1. benchmark .....	15
	3.2. GNN .....	16
<b>4.</b>	<b>Contrucción grafo Interbet .....</b>	<b>17</b>
<b>5.</b>	<b>Modelos .....</b>	<b>18</b>
<b>6.</b>	<b>Resultados .....</b>	<b>20</b>

<b>7.</b>	<b>Conclusion .....</b>	<b>20</b>
<b>8.</b>	<b>Trabajo Futuro .....</b>	<b>20</b>
	<b>Bibliografía .....</b>	<b>21</b>
	<b>Anexo A. titulo anexo 1 .....</b>	<b>24</b>
	<b>Anexo B. Cosas Extras .....</b>	<b>27</b>
<b>3.</b>	<b>Sistemas Autonomos .....</b>	<b>28</b>
	3.0.1. Internet Exchange Points .....	28
<b>4.</b>	<b>BGP .....</b>	<b>29</b>
<b>5.</b>	<b>GNN .....</b>	<b>30</b>
	5.0.1. Message Passing Neural Networks (MPNN) .....	30
	5.0.2. Graph Convolution Network (GCN) .....	31
	5.0.3. Graph Attention Network (GAT) .....	31
	5.0.4. GraphSAGE (SAmple and aggreGatE) .....	33
<b>6.</b>	<b>Vocabulario .....</b>	<b>34</b>

# Índice de Tablas

# Índice de Ilustraciones

Figura 2.1 Ejemplo de propagación de anuncio. En este caso AS 6453 comienza anunciando su dirección a sus pares. ....	7
Figura 2.2 ???.....	13
Figura 2.3 ???.....	13
Figura 5.1 Flujo de Message passing para un nodo del grafo. ....	31
Figura 5.2 Pipeline del mecanismo de cálculo de las ponderaciones para una GAT. ...	32
Figura 5.3 TODO. ....	34

# Capítulo 1

## Introducción

En la sociedad actual, Internet juega un rol esencial en la vida cotidiana, facilitando la comunicación, la colaboración y el intercambio de información.

Internet se compone de múltiples subredes interconectadas. A través del protocolo BGP, estas subredes comparten información sobre su topología de red, permitiendo que los paquetes de datos enviados por un usuario lleguen a su destino final. Conocer las políticas administrativas y otros aspectos relacionados a estas subredes —mejor conocidas como Sistemas Autónomos (ASes)— es crucial para comprender la topología de Internet y, a su vez, tomar mejores decisiones para mejorar la eficiencia y seguridad de la red. Sin embargo, esta información no es de acceso público.

En este contexto, el uso de Redes Neuronales de Grafos (GNNs) surge como una herramienta poderosa para realizar tareas de aprendizaje sobre datos de Internet, ya que este puede representarse de forma natural como un grafo. En particular, el uso de GNNs para la tarea de inferencia de relaciones comerciales entre Sistemas Autónomos podría darnos una mirada diferente a cómo se ha resuelto previamente dicha tarea y darnos una mejor comprensión de cómo la topología del Internet se relaciona con los acuerdos comerciales entre Sistemas Autónomos, además de características importantes a la hora de inferir las relaciones, para luego tomar luego mejores decisiones para mejorar la seguridad y eficiencia de la red.

### 1.1 Hipótesis

Las Redes Neuronales de Grafos (GNNs) pueden ofrecer un rendimiento superior en comparación con las metodologías del estado del arte [1] para la inferencia del tipo de relación entre Sistemas Autónomos.

## 1.2 Objetivos

### 1.2.1 Objetivo general

El objetivo principal de este estudio es evaluar diversas arquitecturas de Redes Neuronales de Grafos (GNNs) y atributos relevantes de los Sistemas Autónomos para determinar su viabilidad en la inferencia del tipo de relación de tráfico entre Sistemas Autónomos. Para ello, se analizarán características específicas de cada Sistema Autónomo junto con la topología de la red.

### 1.2.2 Objetivos específicos

1. **Obtención de datos:** Recolectar información de fuentes confiables, correspondiente a Sistemas Autónomos representativos de la red de Internet. Esto incluye datos sobre la topología entre Sistemas Autónomos, sus características y relaciones que comparten entre ellos.
2. **Preparación de datos:** Construir un grafo a partir de los datos recolectados, definiendo un formato de entrada específico para nuestros modelos GNN, incluyendo la mejora de calidad de los datos mediante el uso de diferentes técnicas de preprocesamiento, como normalización, conversión de atributos categóricos a numéricos, entre otros.
3. **Diseño e implementación de modelos:** Diseñar e implementar modelos GNN y frameworks específicos que permitan la inferencia del tipo de relación que dos Sistemas Autónomos comparten.
4. **Evaluación de desempeño:** Comparar el rendimiento de diferentes arquitecturas de GNN y técnicas utilizadas anteriormente para esta inferencia, identificando los parámetros de mayor relevancia.
5. **Análisis de resultados:** Interpretar los resultados obtenidos mediante el estudio y su comparación con los valores esperados y el estado del arte[1].

## 1.3 Metodología

El plan de trabajo que se llevó a cabo durante esta investigación constó de cuatro etapas:

**Investigación y familiarización** En esta primera etapa, se llevó a cabo la lectura de artículos académicos relacionados con el uso de GNNs, así como de artículos relevantes sobre la representación de datos de Internet y el problema de inferencia. En paralelo, se realizó una búsqueda de datasets públicos disponibles y el desarrollo de modelos básicos de GNNs para familiarizarse con la herramienta.



**Preparación de datos** Una vez estudiados y recolectado los datos sobre Internet, estos fueron pre-procesados y adaptados a un formato específico a la entrada requerida por el modelo.

**Construcción de modelos y entrenamiento** Una vez se tuvo más claridad sobre la tarea y las herramientas a utilizar, se continuó con la implementación de diversos modelos de GNNs utilizando el conjunto de datos. Además, se llevó a cabo el entrenamiento de estos modelos, ajustando hiperparámetros y realizando modificaciones en los datos de entrada a medida que se aprendía y se realizaban pruebas.

**Construcción de un benchmark** Luego de desarrolladas diversas implementaciones y frameworks de GNNs para la tarea específica, se diseñó un pequeño benchmark para comparar el rendimiento del uso de GNNs con técnicas previas y evaluar sus resultados.

**Análisis de resultados** Con las implementaciones ya creadas y los resultados obtenidos tanto de GNNs como de los otros métodos, se pasó al análisis de los resultados y a la comparación de los desempeños de las diferentes técnicas.

## 1.4 Contribuciones

## 1.5 Estructura del trabajo

Este documento se organiza de la siguiente manera:

- **Capítulo 2:** Explicación de los principales conceptos tratados en esta tesis: Internet, Sistemas Autónomos, BGP y GNNs.
- **Capítulo 2:** Se aborda el problema de inferencia de relaciones entre Sistemas Autónomos y métodos previos que han sido utilizados para resolverlo.

# Capítulo 2

## Preliminares

En este capítulo se presentan conceptos fundamentales para esta tesis. Se comienza con una descripción del funcionamiento de Internet, su estructura y componentes fundamentales. Luego, se explica el protocolo de enrutamiento BGP, a través del cual se obtuvo la información de ruteo para construir el grafo. Finalmente, una explicación del funcionamiento de las Redes Neuronales de Grafos (GNN).

### 2.1 Internet

El Internet es una red global de redes que interconecta miles de dispositivos en todo el mundo que proporcionan servicios a aplicaciones distribuidas [2]. Entre estos dispositivos se encuentran computadoras, teléfonos celulares, servidores de contenido y muchos otros.

Cuando un dispositivo final intenta establecer una conexión a través de Internet, los datos a enviar son encapsulados en paquetes con cabeceras que contienen la información necesaria para llegar al dispositivo final. En algunos casos los datos pueden ser divididos en diferentes paquetes y ser enviados por distintas rutas hasta llegar al dispositivo final, donde son reensamblados. El recorrido que sigue un paquete desde su origen hasta su destino a través de distintos routers se conoce como ruta.

Los sistemas finales acceden a Internet por medio de proveedores de Internet (ISPs), los cuales, a su vez, son redes conformadas por routers y enlaces de comunicación. Estos ISPs de nivel inferior se conectan a ISPs de nivel superior, nacionales e internacionales como Level 3, Cogent, entre otros [3], que se encuentran en la cima de la jerarquía de Internet, al tener conexiones directas con el *backbone* de Internet. Así, estos proporcionan a los usuarios finales el acceso a proveedores de contenido (CDN, por sus siglas en inglés), sitios web y otros servicios, los cuales también están conectados a la infraestructura de Internet.

### 2.1.1 Sistemas Autonomos

En la estructura de Internet, un Sistema Autónomo (AS) consiste en un conjunto de routers y enlaces de comunicación que comparten una política de enrutamiento común. Estos son operados independientemente por una organizaciones, como un ISPs, universidades, empresas, entre otras.

Los AS utilizan el protocolo BGP para intercambiar información de enrutamiento y así lograr una conexión global entre ASes y, por ende, en Internet. Las conexiones entre los Ases pueden ser de tipo cliente-proveedor o peer-to-peer, pero están influenciadas por acuerdos comerciales. Bajo estos acuerdos, podemos crear un grafo de la infraestructura de Internet, donde los nodos son los ASes y las aristas son las conexiones entre ellos. Cabe destacar que, en este caso, una conexión no implica necesariamente el intercambio de tráfico entre los ASes conectados, ya que el enrutamiento está controlado por BGP, un protocolo de enrutamiento basado en políticas [4].

Ma sinformacion sobre lso Sistmeas Autonomos pueden ser encontrados en el anexo [TODO:], otros conceotop importantes sobre la estructura de Internet son los *Internet Exchange Points* (IXPs), estos corresponden a puntos de interconexion donde multiples ASes se conectan y establecen conexion con otros ASes sin estar directamente conectados.

### 2.1.2 Ruteo

Antes de introducir el protocolo BGP, debemos entender brevemente el ruteo.

El ruteo es el proceso mediante el cual se selecciona el mejor camino que seguirá un paquete dentro de una red para llegar a su destino final. El camino que sigue un paquete se elige según la información de las tablas de enrutamiento (RIBs) de los routers y la información contenida en los encabezados de los paquetes, donde se indica el destino final.

Cuando un paquete llega a un router, se consulta en la tabla de enrutamiento la dirección final para obtener el próximo router o punto de red al cual se debe dirigir el paquete.

Por ejemplo, cuando un usuario accede a una página web desde su hogar, los paquetes viajan desde el computador hasta el router de su casa. Este router luego examina el encabezado del paquete para identificar el destino final, consulta su tabla de enrutamiento y lo reenvía al siguiente punto de la red. Este nuevo router intermedio repite el proceso hasta que el paquete alcanza su destino final.

Existen dos tipos de enrutamiento: estático y dinámico. El enrutamiento estático implica el uso de tablas estáticas, las cuales deben ser modificadas manualmente para cambiar su configuración. Por otro lado, en el enrutamiento dinámico, los routers se encargan de

actualizar las tablas de enrutamiento en tiempo real, ajustándolas según las condiciones de la red. Este proceso se lleva a cabo mediante los protocolos de enrutamiento.

## 2.2 Border Gateway protocol (BGP)

Como se mencionó existen dos tipos de ruteo estático y dinámico, BGP [5] pertenece al tipo dinámico, donde se encarga de analizar todas las posibles rutas a los diferentes destinos y seleccionar la mejor ruta. Luego, comparte esta información con sus routers vecinos, quienes a su vez repiten el proceso. Así, cada Sistema Autónomo genera caminos hacia los distintos prefijos de Internet y cuando ocurren cambios en la red, BGP se encarga de actualizar las rutas mediante el intercambio de mensajes BGP.

### 2.2.1 Funcionamiento de BGP

Una vez que la conexión TCP se ha establecido entre los pares BGP, estos intercambian mensajes *OPEN* para confirmar los parámetros de la sesión. Luego, envían un mensaje *KEEPALIVE* para confirmar la conexión.

La información que se intercambia en primera instancia consiste en una porción de la tabla de información de BGP, llamada *Adj-RIB-Out*, la cual está filtrada de acuerdo a las políticas locales de exportación hacia los diferentes AS. a medida que esta tabla va cambiando, se envían actualizaciones incrementales mediante mensajes *UPDATE*. Además, para asegurar la conexión sigue activa, los pares BGP intercambian cada cierto tiempo mensajes *KEEPALIVE*.

En caso de que se detecte algún tipo de error durante la conexión, se envía un mensaje *NOTIFICATION* el cual indica el tipo de error. Tras su envío la conexión BGP se cierra y todas las rutas aprendidas en la sesión son eliminadas.

La figura Figura 2.1 muestra cómo BGP propaga la información de ruteo entre distintos AS. Cada sistema autónomo anuncia la dirección de destino a sus pares, en este caso específico el AS 64503 anuncia su dirección a sus pares 64502 y 64504, quienes a su vez anuncian la dirección a sus pares, modificando el *AS\_PATH* a medida que la información se propaga. Finalmente, el AS 64501 recibe estos anuncios y, basándose en la información acumulada, determina la mejor ruta para alcanzar la red del AS 64503.

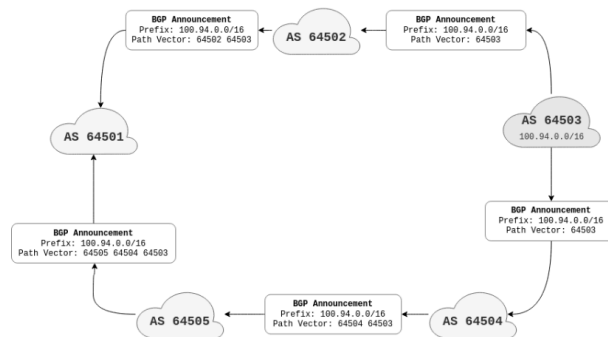


Figura 2.1: Ejemplo de propagación de anuncio. En este caso AS 6453 comienza anunciando su dirección a sus pares.

## 2.2.2 BGP Routing Information Base (RIB)

Un concepto importante son las tablas RIBs a partir de las cuales se extrae la información para construir el grafo de Internet y las aristas que conectan a los ASes.

Cuando se utiliza BGP, los routers BGP reciben mensajes *UPDATE* de sus vecinos, los cuales son analizados y filtrados según las políticas locales de ruteo del AS, para luego anunciar las rutas seleccionadas a sus vecinos. Para esto, BGP utiliza una base de datos denominada Routing Information Base (RIB), que se compone de tres partes:

- **Adj-RIB-In:** Almacena la información de ruteo de los mensajes *UPDATE* recibidos de los vecinos BGP.
- **Loc-RIB:** Contiene la información de ruteo local, es decir, las mejores rutas seleccionadas para cada dirección.
- **Adj-RIB-Out:** Guarda la información seleccionada por el router para ser anunciada a sus pares. Consiste en la información de Loc-RIBs luego de ser aplicadas las políticas internas de ruteo. Esta es la información que se incluye en los mensajes *UPDATE*.

El flujo de información en BGP y el proceso de toma de decisiones de rutas se realiza de la siguiente manera: primero, se reciben los mensajes *UPDATE* de los vecinos, los cuales se almacenan en la Adj-RIB-In, una entrada por cada vecino. Luego, se evalúa el grado de preferencia de cada ruta almacenada en la Adj-RIB-In. En base en esta evaluación, se seleccionan las mejores rutas para cada destino y se instalan en la Loc-RIB. Finalmente, la información contenida en la Loc-RIB se transfiere a la Adj-RIB-Out para ser anunciada a los vecinos BGP, siguiendo las políticas de ruteo locales. Este flujo de información se conoce como el proceso de decisión BGP.

Es importante destacar que las bases de datos que almacenan la información de rutas BGP no son lo mismo que la tabla de ruteo de un router, que es la que el router utiliza para

realizar el forwarding de los paquetes. Las rutas almacenadas en la RIB deben cumplir con ciertos criterios, definidos por el software o el proveedor del router, para ser agregadas a la tabla de ruteo.

## 2.3 Graph Neuronla Network (GNN)

Las GNN son una arquitectura de Redes Neuronales especialmente diseñada para realizar predicciones basadas en datos representativos de grafos. A diferencia de las Redes Neuronales convencionales, las GNNs reciben datos en forma de tensores que pueden representar nodos, atributos de nodos, aristas y atributos de aristas.

Las GNN tienen una serie de ventajas sobre las Redes Neuronales convencionales cuando se trabaja con datos de grafos. En contraste con los modelos tradicionales, las GNN aprovechan las relaciones entre las entidades que conforman los datos de entrada a el modelo. Estas relaciones pueden incluir aspectos como orden, jerarquía, dependencias o relaciones de otro tipo que son comunes en grafos y se representan a través de las aristas que conectan los nodos.

En cuanto a la adaptabilidad a variaciones en el tamaño de entrada, las Redes Neuronales convencionales requieren que los datos de entrada mantengan un mismo tamaño. Para ello, recurren a técnicas como padding o broadcast, los cuales no tienen efectos significativos en el desempeño de los modelos. Las GNNs, por su parte, ofrecen flexibilidad para adaptarse a distintos tamaños de entrada[6].

Otro motivo para optar por GNNs es su capacidad para manejar el isomorfismo de los grafos, es decir dos grafos pueden lucir diferentes, pero ser estructuralmente iguales. Un modelo tradicional trataría ambos grafos como si fuesen datos diferentes, sin embargo, no lo son. Esto es comparable a lo que sucedería si se le presenta como entrada dos imágenes donde una se encuentra invertida. Es por esta razón que no se puede trabajar directamente con una matriz de adyacencia en una Red Feed Forward, ya que es sensible a estos cambios. Las GNNs utilizan técnicas que son invariantes ante permutaciones, lo que permite trabajar con el isomorfismo en grafos.

Finalmente, el último desafío radica en que la estructura de un grafo no puede ser reducida a un espacio euclidiano, y su conceptualización no puede limitarse a una distancia euclidiana[7]. A diferencia de Redes Neuronales que trabajan, por ejemplo, con imágenes, las cuales pueden ser interpretadas como un grafo, la representación de la información se puede entender en términos de píxeles en un espacio bidimensional.

## 2.4 Trabajo Relacionado

### 2.4.1 Tipos de Relaciones

El ruteo de los paquetes que transitan por Internet no depende únicamente de la infraestructura física, sino también por las políticas de enrutamiento establecidas entre los Sistemas Autónomos, las cuales están principalmente determinadas por las relaciones comerciales entre los ASes [8].

Estas relaciones son las que determinan por donde fluye el tráfico entre los ASes, ya que cada uno define sus políticas en base a sus necesidades y modelos de negocio. Existen dos tipos principales de relaciones entre los Sistemas Autónomos: Customer-Provider (C2P), donde un cliente paga a un proveedor para acceder a partes del internet que no puede llegar y Peering (P2P), donde dos AS intercambian tráfico destinado a sus respectivos clientes de manera mutua y sin pago de por medio.

Gao [4] fue la primera en estudiar las relaciones entre los ASes. En su estudio, sentó las bases para abordar su estudio al proponer una forma de representar las relaciones en Internet mediante un grafo parcialmente dirigido, donde nodos representan los Ases y las aristas las relaciones entre ellos. En este modelo, las relaciones de Customer-Provider se representan mediante una arista dirigida. Estos acuerdos son más comunes en los bordes del grafo de lo que sería Internet, donde la topología estaría compuesta principalmente por «hojas», las cuales están principalmente preocupadas por la entrega de su propio tráfico. En contraste, en el núcleo de Internet, la topología consiste en redes densamente interconectadas [9].

Conocer las relaciones entre los ASes es de vital importancia, ya que permite entender el comportamiento del tráfico de Internet, detectar posibles ataques y fallos en la red, y optimizar el ruteo de los paquetes. Sin embargo estas no son de acceso público, por lo que inferir estas relaciones es un problema de gran relevancia en la actualidad. A través de las rutas anunciadas mediante BGP, es posible inferir el tipo de relación entre los ASes.

La inferencia de estas relaciones ha sido un tema de estudio durante las últimas dos décadas, en las cuales se han propuesto diferentes algoritmos para resolverlo, siendo la mayoría de estos métodos heurísticos y basandose en los datos públicos sobre enrutamiento BGP, como las tablas de enrutamiento (RIBs) y los anuncios BGP.

Gao presentó un primer algoritmo basado en la idea de que el grado de un AS en el grafo suele ser mayor que el de sus clientes, mientras que los ASes peers tienden a tener un grado similar. Su método también incorporó el concepto de *valley-free* (VF) en los caminos de enrutamiento. El concepto de *valley-free* establece que una ruta consistirá en cero o más enlaces C2P, seguido de cero o un enlace de peering, y luego cero o más enlaces

P2C. Esta propiedad captura los incentivos económicos que hasta cierto punto determinan el tráfico entre los ASes, ya que un AS no anunciará las rutas aprendidas de un peer o proveedor, ya que eso implicaría el tránsito gratuito de tráfico, lo que aumentaría los costos de infraestructura sin generar beneficios a cambio [10].

Subramanian et al. [11] formularon la tarea como un problema de optimización, definiéndolo como “the type of relationship (ToR) problem”. Este problema consistía en etiquetar todas las aristas en un grafo de AS para maximizar el número de rutas VF en un conjunto de rutas BGP. Para simplificar el problema, excluyó las relaciones S2S (sibling-to-sibling), que son aquellas que entre dos Ases que pertenecen a un mismo dominio.

Se han realizado más estudios que han ido perfeccionando y profundizando en la comprensión del comportamiento de las relaciones entre ASes [8, 10, 12–14].

Luckie et al. [15] introdujeron el algoritmo *AS-Rank* considerado el estado del arte para inferir relaciones C2P y P2P utilizando datos de BGP. Ellos se basaron en tres supuestos: existe un clique de ASes proveedores de tránsito en la cima de la jerarquía de la topología de Internet, Los clientes establecen relaciones con otros ASes con el fin de ser globalmente alcanzables y no existen ciclos de enlaces C2P para que el enrutamiento converja. Así, introdujeron un nuevo algoritmo para inferir el «cono de clientes» de un AS, que es el conjunto de AS que puede alcanzar utilizando enlaces P2C.

Por último, Shapira y Shavitt [1] emplearon técnicas de Deep Learning para la tarea de clasificación. Utilizaron BGP2Vec con el objetivo de crear representaciones de los Sistemas Autónomos y luego pasaron estos *embeddings* aprendidos a una Red Neuronal Artificial, la cual se encargaba de clasificar los tipos de relaciones entre pares de ASes, obteniendo una precisión del 95.2%. Cabe destacar que el entrenamiento de esta Red Neuronal fue realizado con los datos inferidos del dataset [16].

En nuestro caso desarrollamos esta tarea utilizando redes neuronales de grafos, realizando un experimento similar al de Shapira y Shavitt [1], pero aplicando GNNs en lugar de una Red Neuronal tradicional. Para ello, primero crearemos un pequeño benchmark utilizando algunos de los métodos de inferencia de relaciones entre ASes para luego realizar una comparación entre estas técnicas.

#### **2.4.1.1. Gao**

El algoritmo presentado por Gao fue el primero en abordar el problema de inferencia de los tipos de relaciones entre Sistemas Autónomos a partir de tablas de enrutamiento BGP y la heurística de los grados de cada AS.

Para validar sus resultados, Gao utilizó información interna de AT&T y datos del servicio WHOIS para verificar relaciones de tipo siblings. En este análisis, el 99.1% de los resultados de inferencia fueron confirmados por AT&T.



Su algoritmo consistió en :

1. **Parsear las Routing Tables:** Se parsean las tablas de enrutamiento BGP y se calcula el grado de cada AS.
2. **Identificar el top provider:** Se identifica el top provider de cada AS path.
3. **Clasificar las relaciones:** Se clasifican las relaciones entre los ASes en base a la jerarquía de los ASes y el top provider.

La clasificación de relaciones estaba dado por la lógica de que una vez encontrado el top provider de un as path, los pares AS que se encontraran antes del top provider tendrían una relación de P2C, mientras que los pares que se encontraran después del top provider tendrían una relación de P2P y relación S2S si ambos pares proveían de tránsito para el otro, es decir, podía encontrarse ese par tanto antes como después del top provider.

Pero como no se puede asumir que todos los routers de enrutamiento que pertenecen a un AS estén bien configurados, Gao propuso una mejora para evitar inferencias incorrectas, la cual consistía en contar el número de tablas de enrutamiento que concluyeran que una relación era de un tipo y otra. Es decir, si no más de  $L$  tablas de enrutamiento inferían que un AS  $u$  proveía de tránsito a AS  $v$  y más de  $L$  tablas de enrutamiento inferían que  $v$  proveía de tránsito a  $u$ , entonces se ignoraba que  $u$  proveía de tránsito a  $v$ , siendo  $L$  una constante pequeña.

En el caso de las relaciones de P2P, Gao primero identificó aquellos ASes con los que no podría establecer relaciones de peering, basándose en la idea de que en una relación de peering los ASes no difieren más de  $R$  veces.

#### 2.4.1.2. Lu Rank

Uno de los algoritmos más recientes es el propuesto por Ruan y Susan en 2014 [8]. Hasta ese momento, la mayoría de los métodos para inferir los tipos de relaciones entre sistemas autónomos se basaban en la suposición de que todos los ASes siguen una política uniforme de exportación de rutas. Según esta política, las rutas provenientes de proveedores y peers no son exportadas a AS vecinos que también son proveedores o peers. Bajo esta idea, todos los AS Paths serían *valley-free*.

Los algoritmos previos asumían que todos los AS paths eran *valley-free* o buscaban maximizar el número de caminos que cumplían esa propiedad. Sin embargo, se sabe que un gran número de AS paths en los updates o tablas de enrutamiento BGP no son *valley-free*. Esto hace que al depender de dicha propiedad las relaciones inferidas sean imprecisas.

Ruan y Susan en lugar de basarse en la propiedad de *valley-free*, propusieron un método para clasificar las relaciones observadas entre AS, sustentándose en las relaciones de tránsito entre ellas, a través de los datos de BGP. A diferencia del algoritmo de Gao [4], que se basa principalmente en el grado de los AS, este método utiliza el concepto de grado de

tránsito de los AS lo cual en conjunto con la identificación de los top providers de los AS Path, permiten identificar las relaciones entre AS. .

El algoritmo propuesto consta de tres fases principales:

1. **Preprocesamiento de los datos:** La entrada consiste en un conjunto de BGP routing table dumps obtenidos de RouteViews [17]. Estos datos se sanitizan eliminando loops, descartando ASN inválidos, eliminando ASN duplicados y excluyendo conjuntos de ASN terminales, en caso de que estén presentes en las secuencias.

## 2. Procesamiento de AS paths que contienen AS Tier-1:

En esta fase se definen dos contadores  $\text{cnt}(X, Y)$  y  $\text{cnt}(Y, X)$ , los cuales indican el número de veces que X actúa como proveedor de tránsito para Y, y el número de veces que Y actúa como proveedor de tránsito para X respectivamente. Si X e Y aparecen en un AS path antes del top provider, se interpreta que Y proporciona tránsito a X; de lo contrario, X proporciona tránsito a Y. A partir de esta información y la identificación de ASN Tier-1, las relaciones se clasifican en base a las siguientes reglas:

- Si  $\text{cnt}(X, Y) > 0$  y  $\text{cnt}(Y, X) = 0$ , se establece una relación P2C entre X y Y.
  - Si  $\text{cnt}(X, Y) > 0$  y  $\text{cnt}(Y, X) > 0$ , y al menos uno entre X o Y es un AS Tier-1, la relación entre ellos es P2P.
  - Si  $\text{cnt}(X, Y) > 0$  y  $\text{cnt}(Y, X) > 0$ , y ninguno de los dos es un AS Tier-1, la relación es S2S.
3. **Clasificación de AS paths indeterminados:** En esta fase final, primero se determina el top provider de los AS paths donde no se encontraron AS Tier-1. Para ello, se construye un grafo dirigido en el que cada nodo tiene un atributo denominado *distance*, que indica el camino más corto en hops a un AS Tier-1. Luego, con esta información al igual que en la fase anterior se puede determinar las relaciones de los AS paths bajo las siguientes reglas:
    - Si  $\text{cnt}(X, Y) > 0$  y  $\text{cnt}(Y, X) = 0$ , entonces X y Y tienen una relación provider-to-customer (P2C).
    - Si  $\text{cnt}(X, Y) > 0$  y  $\text{cnt}(Y, X) > 0$ , entonces X y Y tienen una relación peer-to-peer (P2P).

Esta última regla puede corresponder también a una relación S2S, en caso de que en el paso anterior se hayan clasificado como S2S. Si un AS path contiene una relación P2P que no es adyacente a un top provider, entonces se reclasifica como S2S, dado que el enlace es visible a través de un upstream provider. En caso contrario, se mantiene como P2P.

### 2.4.1.3. BGP2Vec

En 2020, Tal Shapira y Yuval Shavitt presentaron un nuevo enfoque [1], para la inferencia de Sistemas Autónomos utilizando por primera vez técnicas de Deep Learning. Este método se basa en la creación de *embeddings* de estos Sistemas Autónomos utilizando únicamente anununcios BGP provenientes de datasets públicos. El modelo alcanzó una precisión del 95.8 % en la clasificación de relaciones entre ASes.

El pipeline que siguieron para esta tecnica consistió en dos etapas: La primera, donde se entrena un modelo de Deep Learning para generar *embeddings* de los ASes, y la segunda, donde se entrena un modelo de Deep Learning para clasificar las relaciones entre los ASes.

El entrenamiento de BGP2Vec para la generación de *embeddings* de Sistemas Autónomos consiste en utilizar como entrada un vector one-hot representante del ASN (Número de Sistema Autónomo) y como salida otro vector one-hot que representa los ASNs del contexto. Luego mediante el descenso de gradiente, se ajustan los pesos de la red para maximizar la probabilidad logarítmica de cualquier ASN de contexto dado el ASN de entrada.

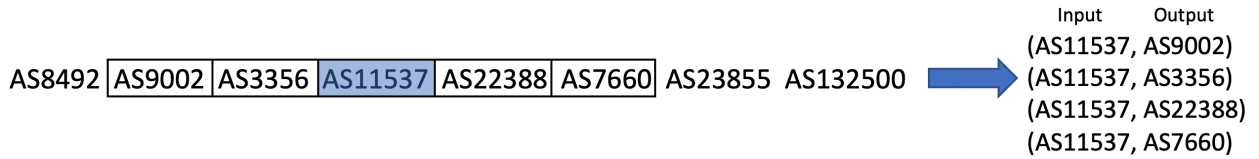


Figura 2.2: ???.

Una vez entrenado los *embeddings* para cada ASN, se continúa con la tarea de clasificación de las relaciones entre los Sistemas Autónomos. Para esto se ocupa una Red Neuronal Artificial de dos capas.

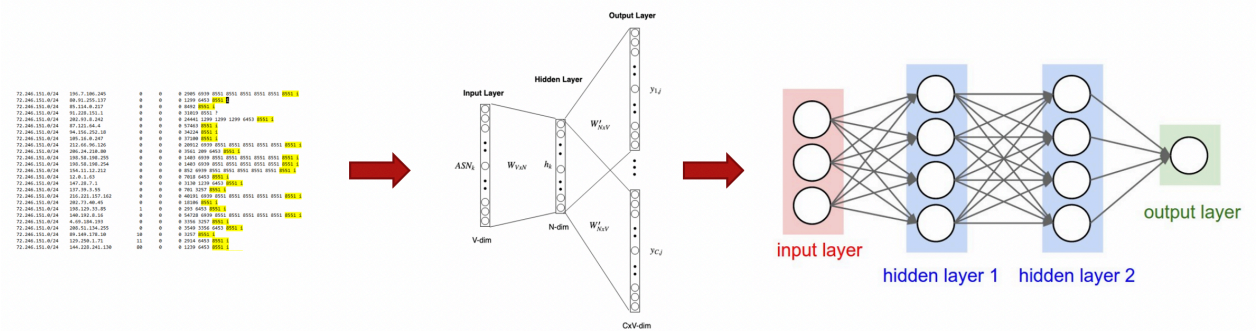


Figura 2.3: ???.

Para la etapa inicial de entrenamiento de BGP2Vec, se utilizaron anuncios extraídos de RouteViews [17], que contenían anuncios de rutas de AS (AS PATH) recolectadas por 19 colectores. Este dataset recopiló 3,600,000 rutas de AS con 62,525 vértices de AS y 113,400 enlaces no dirigidos.

Para el entrenamiento de la Red Neuronal encargada de la clasificación de las relaciones entre ASes, se utilizó el conjunto de datos de relaciones entre AS de CAIDA [16], que contiene relaciones P2P y P2C/C2P. Además, para algunos experimentos, se empleó el dataset ToR de BGProtect (www.BGProtect.com) [18], basado en el trabajo de Shavitt et al. [13] .

## 2.4.2 ProbLink

ProbLink es un algoritmo probabilístico para inferir las relaciones entre Sistemas Autónomos propuesto por Yuchen Jin et al. [10]. Este permite el uso de atributos con valores estocásticos. Toma en cuenta información sobre los links y caminos que atraviesan . ProbLink no impone un orden específico en el que los ASes y los enlaces deben ser analizados, en su lugar itera con continuamente hasta alcanzar una convergencia.

Para cada atributo se calcula la distribución de probabilidad condicional basada en los datos observados y el conjunto inicial de etiquetas. Luego en cada iteración, se actualiza las probabilidades de los tipos de cada enlace ejecutando su algoritmo probabilístico y se recalcula las distribuciones de las características utilizando los valores de probabilidad actualizadas de cada enlace. Se repite el proceso hasta la convergencia, es decir , hasta que el porcentaje de enlaces que cambia de etiqueta caiga por debajo de un umbral.

Las características utilizadas para calcular la probabilidad de cada relación son:

- **Atributo Triplet:** Analiza secuencias de tres enlaces en una ruta BGP y modela la *valley-freeness* de manera probabilística.
- **Atributo Non-path:** Mide la probabilidad de que un enlace tenga enlaces adyacentes p2p o p2c que no aparezcan antes en ninguna ruta. Captura la propiedad de que un enlace es menos probable que sea p2c si tiene muchos enlaces adyacentes p2p/p2c sin aparecer previamente en las rutas.
- **Atributo Distance to clique:** Identifica que los ASes de alto nivel están más cerca entre sí en términos de saltos AS y que estos suelen actuar como proveedores, mientras que los ASes de bajo nivel tienden a ser peers.
- **Atributo Vantage point:** El número de puntos de vista (VPs) que observan un enlace puede indicar su tipo. Se basa en la suposición de que los enlaces p2c son más propensos a ser observados por múltiples VPs en comparación con los enlaces p2p y c2p.
- **Atributo Co-located IXP and co-located private peering facility:** Usa datos de PeeringDB para identificar ASes que comparten IXPs o instalaciones, lo que sugiere una mayor probabilidad de que estén en una relación de peering.

# Capítulo 3

## Metodologia

### 3.1 benchmark

En conjunto a la tarea principal de este trabajo se decidio crear un benchmark con algunos de los algoritmos de inferencia de relaciones comerciales entre Sistemas Autonomos, estos fueron los de GAO [4], RUAN [8], problink [10] y bgp2vec [1]. Este ultimo siendo un enfoque similar al que se tomo en este trabajo.

El primer paso consistio en la extraccion de las RIBs publicas que se encuentran en diferentes dataset de internet como lo son las organizaciones de CAIDA [19], RIPE RIS [20] y Routeviews [17]. Tanto RIPE como Routeviews mantienen colectores BGP en partes estrategicas de la infraestructura de Internet como los puntos de intercambio de trafico (IXP) [citar] los cuales se encargan de recolectar las rutas BGP de los diferentes sistemas autonomos que se encuentran conectados a ellos.

Lo que se hizo fue con la libreria BGPStream recolectar las RIBs (Routing information Bases, por sus siglas en ingles) de los colectores de RIPE y Routeviews. En un primer momento se recolectaron las de un dia completo, sin embargo por razones de espacio y memoria se decidio bajar esto a 12 hrs del dia 1/07/2022 y 2023.

41805|8262|198202  
42473|8529|6461|46887|31980  
55720|4637|3257|30456|198990  
3561|209|3356|9002|50494  
13760|6939|19782|1767|19217

Los algoritmos fueron sacados de diferentes fuentes, donde fueron previamente implementados en [21], [22, 23].

Se crearon archivos notebook para cada metodo, los cuales luego de extraidas las ribs y convertidas a un archivo que las contienen son importadas a los notebooks, luego se corren los metodos de inferencia, se ajustan los resultados para que todos tengan el formato tipo

```
29838|26941|2|2
46416|34927|1|0
149478|58752|1|1
203201|204760|1|2
```

donde el formato consiste en

```
<provider AS>|<customer-AS>|2|<real relationship>
<customer-AS>|<provider AS>|1|<real relationship>
<peer-AS>|<peer-AS>|0|<real relationship>
```

Donde en el primer caso etiquetamos como 2 a los ASes que consideramos con esa relación pero `real relationship` consiste en el valor que se tomó como ground truth en nuestro caso que es AS Rank.

Luego con esta información se pudieron sacar los siguientes resultados comparando:

TODO: poner gráficos y resultados.

TODO: debo poner cómo se corren las cosas?

TODO: Decir que también en nuestro código dejamos que se pueda aceptar archivos oix

TODO: Decir que las implementaciones de estos algoritmos fueron sacados de sus mismas implementaciones originales y donde fueron sacadas pero modificadas para aceptar las rutas que yo saque

## 3.2 GNN

Teníamos que crear los embeddings para los sistemas autónomos para esto se partió construyendo un grafo (DGL y networkx) a partir de las rutas BGP colectadas los primeros 3 días de Julio del año (2022-2023).

Luego se visualizó y se sacó información de esta.

Estas mismas son también las que se ocuparon como input de los algoritmos/métodos de inferencia previos presentes en el benchmark.

Adiferencia de otros métodos como node2vec, Deepwalk, bgp2vec GNN no solo estaría incorporando en estos embeddings la topología del grafo sino también los atributos de cada nodo.

Se tomaron 3 enfoques GNN al momento de crear los embeddings de los nodos:

- Encode-Decoder perspective

En un primer intento se intentó recolectar los paths de las RIBS correspondientes a los 2 primeros días de julio de 2022/2023, sin embargo por problemas de memoria se decidió

tomar unicamente el primer dia de julio de 2022/2023, proque no pod[ia por memoria trabajar con una matriz densa o dispersa para el caso de la matriz de adjacencia negativa. Por otro lado los metodos del benchmark tampoco podians er corridos por lo mismo y como la idea tambeon es poder crear un benchmark y al ceacion de un grafo que otros puedan usar se decidio tomar unicamente un unico dia, pensado q talvez se pueda tener mejores resultados o una topologia de internet m[as completa si esque se toman mas dias]]

La idea del uso de GNN es preentrenara un modelo con a large dataset y luego fine tuned en un datset (labeled) mas chico. a la task siguiente se le llama downstream task.

Puede que no siempre endra sentido preentrenar representaciones de los nodos. [<https://arxiv.org/pdf/2207.06010>] todo encaja para q en este caso no sea super mejor. Su conclusion fue q no hubo/encontro un significant improvemnet en downstream tasks, pape rtmb dosciusses potenciales causas

## Capítulo 4

# Contrucción grafo Interbet

Para esta parte se decidio contruir un grafo de la topologia donde cada nodo consta de los Sistemas Autonomos , las aristas las conecciones BGP existenete sentre ellas. Ademas se le suma atributos a cada SA. Esro con el fin de facilitar el input para diferentes tasks de GNN usando la informacion de internet.

La topologia se saco de las mismas BGP routes sacadas con BGPStream (libreria) que saca las RIBs de los colectores de Routeviews y RIPE NCC.

En cuanto a lso atributos de los nodos se sacaron del dataset de peeringDB, especificamente los atributos:

- atributo 1
- atributo 2

- atributo 3
- atributo 4
- atributo 5

Las snapshots diarias de la informacion de peeringDB se encuentran en formato sql, sqlite y/o json.

para esto primero se descargo los archivos diarios de peeringDB, luego se exploraron estos , viendo los potenciales atributos que se podrian usar para los nodos.

## Capítulo 5

### Modelos

Priemro se creo con GNN un modelo para crear los emebddings de los nodos/AS con un modelo encode Encode-Decoder para predecir laa existencia o no de links

La idea asi en cada forward se incluyen tanto los parametros de SAGEConv como los de la MLPredictor.

```
class GraphSAGE(nn.Module):
    def __init__(self, in_feats, hidden_feats, out_feats):
        super().__init__() # ✓ Esto es lo correcto
        self.conv1 = SAGEConv(in_feats, hidden_feats, 'mean')
        self.conv2 = SAGEConv(hidden_feats, out_feats, 'mean')

        self.decoder = MLPredictor(out_feats)

    def encode(self, g, in_feat):
        h = self.conv1(g, in_feat)
        h = F.relu(h)
        h = self.conv2(g, h)
        return h
```



```

def decodeDotProduct(self, g, h):
    with g.local_scope():
        g.ndata["h"] = h
        # Compute a new edge feature named 'score' by a dot-product between
the
        # source node feature 'h' and destination node feature 'h'.
        g.apply_edges(fn.u_dot_v("h", "h", "score"))
        # u_dot_v returns a 1-element vector for each edge so you need to
squeeze it.
        return g.edata["score"][:, 0]

def decodeMLP(self, g, h):
    return self.decoder(g, h)

class MLPPredictor(nn.Module):
    def __init__(self, h_feats):
        super().__init__()
        self.W1 = nn.Linear(h_feats * 2, h_feats)
        self.W2 = nn.Linear(h_feats, 1)

    def apply_edges(self, edges):
        h = torch.cat([edges.src["h"], edges.dst["h"]], 1)
        return {"score": self.W2(F.relu(self.W1(h))).squeeze(1)}

    def forward(self, g, h):
        with g.local_scope():
            g.ndata["h"] = h
            g.apply_edges(self.apply_edges)
            return g.edata["score"]

```

Probamos diferentes casos, donde se empezaba con attr iniciales lo atributos sacados de PeeringDB, luego unicamnete in y out degree, luego tambien iniciando con puros 1s.

Vamos a probar crear los embeddings a paratr de node2vec, randomwalk

## Capítulo 6

## Resultados

## Capítulo 7

## Conclusion

## Capítulo 8

## Trabajo Futuro

# Bibliografía

- [1] T. Shapira y Y. Shavitt, «Unveiling the Type of Relationship Between Autonomous Systems Using Deep Learning», en *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, 2020, pp. 1-6. doi: 10.1109/NOMS47738.2020.9110358.
- [2] J. F. K. y Keith W. Ross, *Computer networking : a top-down approach [7th edition]*. 2017.
- [3] «CAIDA AS Rank». [En línea]. Disponible en: <https://asrank.caida.org/asns>
- [4] L. Gao, «On inferring autonomous system relationships in the internet», *IEEE/ACM Trans. Netw.*, vol. 9, n.º 6, pp. 733-745, dic. 2001, doi: 10.1109/90.974527.
- [5] Y. Rekhter, T. Li, y S. Hares, «A Border Gateway Protocol 4 (BGP-4)», ene. 2006. [En línea]. Disponible en: <https://datatracker.ietf.org/doc/html/rfc4271>
- [6] D. Duvenaud *et al.*, «Convolutional networks on graphs for learning molecular fingerprints», en *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2*, en NIPS'15. Montreal, Canada: MIT Press, 2015, pp. 2224-2232.
- [7] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, y P. Vandergheynst, «Geometric Deep Learning: Going beyond Euclidean data», *IEEE Signal Processing Magazine*, vol. 34, n.º 4, pp. 18-42, 2017, doi: 10.1109/MSP.2017.2693418.
- [8] L. Ruan y J. S. Varghese, «Computing observed autonomous system relationships in the internet». 2014.
- [9] B. Woodcock y M. Frigino, «2016 Survey of Internet Carrier Interconnection Agreements».
- [10] Y. Jin, C. Scott, A. Dhamdhere, V. Giotsas, A. Krishnamurthy, y S. Shenker, «Stable and practical AS relationship inference with problink», en *Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation*, en NSDI'19. Boston, MA, USA: USENIX Association, 2019, pp. 581-597.
- [11] L. Subramanian, S. Agarwal, J. Rexford, y R. Katz, «Characterizing the Internet hierarchy from multiple vantage points», en *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, 2002, pp. 618-627. doi: 10.1109/INFCOM.2002.1019307.

- [12] X. Dimitropoulos *et al.*, «AS relationships: inference and validation», *SIGCOMM Comput. Commun. Rev.*, vol. 37, n.º 1, pp. 29-40, ene. 2007, doi: 10.1145/1198255.1198259.
- [13] U. Weinsberg, Y. Shavitt, y E. Shir, «Near-Deterministic Inference of AS Relationships», en *IEEE INFOCOM Workshops 2009*, 2009, pp. 1-2. doi: 10.1109/INFCOMW.2009.5072167.
- [14] G. Di Battista, T. Erlebach, A. Hall, M. Patrignani, M. Pizzonia, y T. Schank, «Computing the types of the relationships between autonomous systems», *IEEE/ACM Trans. Netw.*, vol. 15, n.º 2, pp. 267-280, abr. 2007, doi: 10.1109/TNET.2007.892878.
- [15] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas, y k. claffy, «AS relationships, customer cones, and validation», en *Proceedings of the 2013 Conference on Internet Measurement Conference*, en IMC '13. Barcelona, Spain: Association for Computing Machinery, 2013, pp. 243-256. doi: 10.1145/2504730.2504735.
- [16] «CAIDA. 2022. AS-relationships dataset». [En línea]. Disponible en: <https://publicdata.caida.org/>
- [17] «University of Oregon RouteViews Project, <https://www.routeviews.org/routeviews/>».
- [18] «BGProtect, [www.BGProtect.com](http://www.BGProtect.com)».
- [19] «CAIDA». [En línea]. Disponible en: <https://www.caida.org/>
- [20] «RIPE NCC Routing Information Service (RIS), <https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/>».
- [21] Y. Jin, «ProbLink».
- [22] T. Shapira y Y. Shavitt, «BGP2VEC».
- [23] A. Mobilia, «AS-ToR-Inference».
- [24] G. Huston, «Interconnection, Peering and Settlements», 2003, [En línea]. Disponible en: <http://websrv.cs.fsu.edu/~xyuan/cis6930/huston99.pdf>
- [25] J. Postel, «Transmission Control Protocol», sep. 1981. [En línea]. Disponible en: <https://www.rfc-editor.org/rfc/rfc793.txt>
- [26] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, y G. E. Dahl, «Neural message passing for Quantum chemistry», en *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, en ICML'17. Sydney, NSW, Australia: JMLR.org, 2017, pp. 1263-1272.

- [27] T. Kipf y M. Welling, «Semi-Supervised Classification with Graph Convolutional Networks», p. , 2016, doi: 10.48550/arXiv.1609.02907.
- [28] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio', y Y. Bengio, «Graph Attention Networks», *ArXiv*, 2017, [En línea]. Disponible en: <https://api.semanticscholar.org/CorpusID:3292002>
- [29] W. L. Hamilton, R. Ying, y J. Leskovec, «Inductive representation learning on large graphs», en *Proceedings of the 31st International Conference on Neural Information Processing Systems*, en NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 1025-1035.

# Anexo A

## titulo anexo 1

Collectors RIPE NCC:

Nombre	Ubicación	Typo	Sponsors
RRC00	Amsterdam, NL	multihop	RIPE NCC
RRC01	London, GB	IXP	LINX, LONAP
RRC03	Amsterdam, NL	IXP	AMS-IX, NL-IX
RRC04	Geneva, CH	IXP	CIXP
RRC05	Vienna, AT	IXP	VIX
RRC06	Otemachi, JP	IXP	DIX-IE, JPIX
RRC07	Stockholm, SE	IXP	Netnod
RRC10	Milan, IT	IXP	MIX
RRC11	New York, NY, US	IXP	NYIIX
RRC12	Frankfurt, DE	IXP	DE-CIX
RRC13	Moscow, RU	IXP	MSK-IX
RRC14	Palo Alto, CA, US	IXP	PAIX
RRC15	Sao Paulo, BR	IXP	PTTMetro-SP
RRC16	Miami, FL, US	IXP	Equinix Miami
RRC18	Barcelona, ES	IXP	CATNIX
RRC19	Johannesburg, ZA	IXP	NAP Africa JB

Nombre	Ubicación	Typo	Sponsors
RRC20	Zurich, CH	IXP	SwissIX
RRC21	Paris, FR	IXP	France-IX Paris and France-IX Marseille
RRC22	Bucharest, RO	IXP	InterLAN
RRC23	Singapore, SG	IXP	Equinix Singapore
RRC24	Montevideo, UY	multihop	LACNIC region
RRC25	Amsterdam, NL	multihop	RIPE NCC
RRC26	Dubai, AE	IXP	UAE-IX, Datamena

Collectors de ROuteViews:

- Collectors:

Host	Ubicación
amsix.ams.routeviews.org	AMS-IX Amsterdam, Netherlands
cix.atl.routeviews.org	CIX-ATL Atlanta, Georgia
decix.jhb.routeviews.org	DE-CIX KUL, Johor Bahru, Malaysia
iraq-ixp.bgw.routeviews.org	IRAQ-IXP Baghdad, Iraq
pacwave.lax.routeviews.org	Pacific Wave, Los Angeles, California
pit.scl.routeviews.org	PIT Chile Santiago, Santiago, Chile
pitmx.qro.routeviews.org	PIT Chile MX, Querétaro, Mexico
route-views.routeviews.org	Cisco IPv4 U of Oregon, Eugene Oregon
route-views.amsix.routeviews.org	AMS-IX AM6, Amsterdam, Netherlands
route-views.bdix.routeviews.org	BDIX, Dhaka, Bangladesh

Host	Ubicación
route-views.bknix.routeviews.org	BKNIX, Bangkok, Thailand
route-views.chicago.routeviews.org	Equinix CH1, Chicago, Illinois
route-views.chile.routeviews.org	NIC.cl Santiago, Chile
route-views.eqix.routeviews.org	Equinix DC, Ashburn, Virginia
route-views.flxix.routeviews.org	FL-IX, Miami, Florida
route-views.fortaleza.routeviews.org	IX.br (PTT.br), Fortaleza, Brazil
route-views.gixa.routeviews.org	GIXA, Ghana, Africa
route-views.gorex.routeviews.org	IGOREX, Guam, US Territories
route-views.jinx.routeviews.org	JINX, Johannesburg, South Africa
route-views.kixp.routeviews.org	KIXP, Nairobi, Kenya
route-views.linx.routeviews.org	LINX, London, United Kingdom
route-views.mwix.routeviews.org	FD-IX, Indianapolis, Indiana
route-views.napafrika.routeviews.org	NAPAFrika, Johannesburg, South Africa
route-views.nwax.routeviews.org	NWAX, Portland, Oregon
route-views.ny.routeviews.org	DE-CIX NYC, New York, USA
route-views.paix.routeviews.org	PAIX, Palo Alto, California
route-views.perth.routeviews.org	West Australian Internet Exchange, Perth, Australia
route-views.peru.routeviews.org	Peru IX, Lima, Peru
route-views.phoix.routeviews.org	University of the Philippines, Diliman, Quezon City, Philippines



Host	Ubicación
route-views.rio.routeviews.org	IX.br (PTT.br), Rio de Janeiro, Brazil
route-views.saopaulo.routeviews.org	SAOPAULO (PTT Metro, NIC.br), Sao Paulo, Brazil
route-views2.saopaulo.routeviews.org	SAOPAULO (PTT Metro, NIC.br), Sao Paulo, Brazil
route-views.sfmix.routeviews.org	San Francisco Metro IX, San Francisco, California
route-views.siex.routeviews.org	Sothern Italy Exchange (SIEX), Rome, Italy
route-views.sg.routeviews.org	Equinix SG1, Singapore, Singapore
route-views.soxrs.routeviews.org	Serbia Open Exchange, Belgrade, Serbia
route-views.sydney.routeviews.org	Equinix SYD1, Sydney, Australia
route-views.telxatl.routeviews.org	TELXATL, Atlanta, Georgia
route-views.uaeix.routeviews.org	UAE-IX, Dubai, United Arab Emirates
route-views.wide.routeviews.org	DIXIE (NSPIXP), Tokyo, Japan

## Anexo B

### Cosas Extras

- ¿Cómo se ve el sobreajuste?
- Heat map para analizar de la importancia de los atributos
- ¿Por qué se obtienen mejores resultados con GraphSAGE que con CGN?
- ¿Por que no se ocupo un enfoque transductivo y unicamente uno inductive Learning?
- Transductive
- pros y contra de cada metrica de evaluacion.

Definiciones:

## Sistemas Autonomos

Como se hablo previamente en la seleccion [TODO:]XXXXX los Sistema sAutonomos son un ocnjunto de dispositivos que comparten un mismo protocolo de ruteo y estan adminis-trados por una misma entidad, y su interconexion forma el Internet.

Los AS se identifican con un número de Sistema Autónomo (ASN) de 16 bits y controlan un conjunto de direcciones IP. Esta asignación es llevada a cabo por los Registros Regionales de Internet (*Regional Internet Registry*, RIRs), quienes reciben bloques de IPs por la *Internet Assigned Numbers Authority* (IANA) y los distribuyen a los Registros Locales de Internet (*Local Internet Registries*, LIR) y a los usuarios finales.

Los ISP, quienes pueden estar conformados por uno o varios Sistemas Autónomos, se dividen comúnmente en tres niveles de jerarquía. El *Tier-1*, donde se encuentran los AS que conforman el *backbone* de Internet, estos intercambian paquetes entre sí sin un costo asociado. Los *Tier-2* son generalmente operadores nacionales que compran tránsito a los *Tier-1* y venden tránsito a los *Tier-3*. Finalmente los *Tier-3* son los operadores locales que pagan por el tránsito para proporcionar acceso a Internet a los usuarios finales [24]. También, Luckie et al. [15] analizó los AS y propuso una métrica para indicar qué tan global es un AS, donde si el customer cone es mayor o igual a 200, se considera *Tier-1*; si es mayor a 2,000, se considera *Tier-2*; y si es menor a 200, se considera *Tier-3*.

Una última separación que se puede hacer entre AS es según sus conexiones a otros Sistemas Autónomos, clasificándolos en *single-homed* y *multi-homed*. Los AS *single-homed* tienen solo una conexión a otro AS, mientras que los *multi-homed* tiene conexiones a más de un AS.

### 3.0.1 Internet Exchange Points

Los *Internet Exchange Points* (IXPs) son puntos de interconexión donde múltiples ASes pueden establecer relaciones de peering con otros ASes [2].

Un IXP consiste generalmente en un switch o router de alta velocidad y capacidad que conecta routers de diferentes ASes, permitiendo el intercambio directo de tráfico sin necesidad de atravesar redes intermedias. Mejorando así el intercambio de tráfico, haciéndolo más eficiente y de bajo costo.

Las relaciones de tráfico en un IXP se establecen mediante el protocolo BGP, y el IXP solo actua como un intermediario. De esta forma por ejemplo un ISP en vez de estar

conectados a multiples ASes en una coneccion punto a punto, puede conectarse unicamente al IXP y como estos otros ASes tambien estaran conectados al IXP se ahorra recursos fisicos y de administracion.

Al igual que los ASes, los IXPs tienen un ASN. Sin embargo, de forma generalizada, este ASN se extrae de los AS PATH en BGP. Algunos IXPs lo mantienen visible como método de debugging [4].

## BGP

BGP utiliza TCP [25] como su protocolo de transporte, lo que significa que no necesita preocuparse por la fragmentación de paquetes, la confirmación de recepción (ACK), la retransmisión de datos, entre otros aspectos de confiabilidad.

Los mensajes BGP solo son procesados una vez que han sido completamente recibidos. Su tamaño mínimo es de 19 octetos, correspondiente al *HEADER*, mientras que el tamaño máximo es de 4096 octetos. Existen 4 tipos de mensajes en BGP: *OPEN*, *UPDATE*, *KEEPALIVE* y *NOTIFICATION*.

### Mensaje OPEN

Luego de establecida la conexión TCP entre los pares BGP, el primer mensaje que se envía en un mensaje *OPEN*, mediante el cual ambos lados confirman los parámetros de la conexión.

En este mensaje se especifica la versión de BGP a utilizar, el número de Sistema Autónomo (AS) del emisor, el Hold Time, el BGP identifier y los parámetros opcionales.

El Hold Time define el tiempo máximo que puede pasar sin recibir un mensaje *KEEPALIVE* o *UPDATE* en una sesión antes de que la conexión sea cerrada.

### Mensaje UPDATE

Se utiliza para transferir información de enrutamiento entre los pares BGP. A través de este mensaje se anuncian nuevas rutas y se notifican los withdrawals de aquellas que ya no son válidas.

Además, estos mensajes incluyen path attributes, los cuales proporcionan información sobre las rutas que se están anunciando para que luego cada AS pueda decidir la mejor ruta a los diferentes destinos. Algunos ejemplos de estos atributos son *ORIGIN*, *AS\_PATH*, *NEXT\_HOP*, *MULTI\_EXIT\_DISC*, *LOCAL\_PREF*, *ATOMIC\_AGGREGATE* y *AGGREGATOR*.

## Mensaje **KEEPALIVE**

El intercambio de mensajes *KEEPALIVE* dentro del protocolo BGP se utiliza para confirmar que la conexión entre ambos pares sigue activa, evitando que el Hold Time expire.

Este mensaje consta únicamente del *HEADER* de un mensaje BGP, donde se indica que corresponde a un *KEEPALIVE* por medio del campo *type*, cuyo valor es 2, el cual corresponde al valor 2. Dicho mensaje tiene un tamaño de 19 octetos.

## Mensaje **NOTIFICATION**

El mensaje *NOTIFICATION* se envía cuando se detecta un error en la comunicación BGP. Una vez enviado, la conexión se cierra.

Este mensaje incluye un código de error y un subcódigo de error, los cuales indican en qué tipo de mensaje se produjo el error y la zona específica del problema. Además, contiene un campo de datos que proporciona información adicional sobre el error detectado.

# GNN

## 5.0.1 Message Passing Neural Networks (MPNN)

Es la arquitectura de Red Neuronal para grafos más utilizada. Su funcionamiento radica en la idea de que cada nodo en un grafo puede intercambiar información con sus vecinos de manera que cada nodo podrá actualizar su representación en base a la información acumulada por su entorno.

La información se propaga entre nodos a través de mensajes. Cada nodo envía mensajes a sus nodos vecinos y recibe mensajes de ellos. Estos mensajes pueden contener información sobre el nodo emisor y se utilizan para actualizar la representación del nodo receptor[26].

Se emplea un mecanismo denominado *message passing*, el cual consta de tres pasos:

1. Propagación de mensajes entre nodos: Cada nodo envía un mensaje que contenga su representación actual a sus nodos vecinos.
2. Aplicación de una función de agregación: Luego de la propagación de mensajes, se aplica una función de agregación para combinar la información recibida de los nodos vecinos.

Esta función puede adoptar diversas formas como la suma o la media.

1. Actualización de la representación: La representación de cada nodo se actualiza mediante la información agregada proveniente de sus nodos vecinos, así como a partir de su representación previa.

A continuación ( Figura 5.1), se presenta el comportamiento de una capa de MPNN para un nodo. El proceso inicia con el envío de un mensaje  $M$  por parte de cada nodo vecino de  $B$ .  $B$  recibe estos mensajes y los agrega mediante una operación generando una representación  $A$ . Finalmente, el nuevo estado del nodo  $B$  se calcula mediante una última función que toma el valor  $A$  y su propia representación para crear su nueva descripción  $U$ .

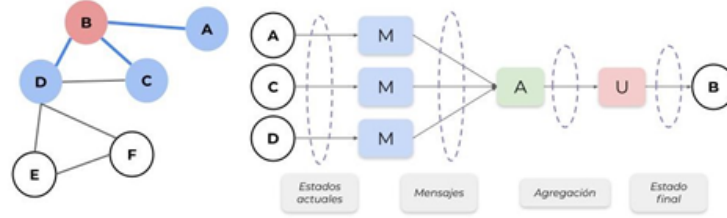


Figura 5.1: Flujo de Message passing para un nodo del grafo.

### 5.0.2 Graph Convolution Network (GCN)

Es un tipo específico de MPNN, donde se utilizan convoluciones de grafos para agregar información de los nodos adyacente de un nodo en un grafo.

La operación de convolución en el grafo produce la suma normalizada de las características de los nodos vecinos[27]. Esta normalización garantiza que la información agregada sea ponderada correctamente, es decir, evitar que un nodo con gran cantidad de vecinos tenga una representación desproporcionada y que luego tenga una influencia mayor en la representación otros nodos en las siguientes capas.

La notación de los embeddings de los nodos está dado por:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}\right) \quad (5.1)$$

Donde  $\sigma$  se define como la función de activación,  $H^{(l)}$  la matriz de características de los nodos en la capa  $l$ ,  $W^{(l)}$  la matriz de aprendizaje de pesos, con dimensionalidades dada por el tamaños de atributos entrantes y de salida por capa y  $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$  la matriz de adyacencia normalizada.

Es así como GCN permite la creación de embeddings para los nodos de un grafo dada la matriz de adyacencia de este, lo que quiere decir que debe conocer el grafo completo para poder realizar la tarea de aprendizaje. Este es un enfoque transductivo, en contraste a otros enfoques inductivos como GraphSAGE.

### 5.0.3 Graph Attention Network (GAT)

Otra variante de MPNN son las Graph Attention Networks (GAT). A diferencia de una Red Neuronal de Convolución, GAT incorpora un mecanismo de atención que permite que

cada nodo pondere de forma diferenciada, indicando la importancia de las representaciones de cada vecino para la actualización de las características de un nodo[28].

Los coeficientes se calculan por un mecanismo el cual calcula un puntaje para cada par de nodos. Luego estos puntajes se normalizan por medio de la función SoftMax ( Figura 5.2).

Así tenemos:

$$z_i^{(l)} = W^{(l)} h_i^{(l)} \quad (5.2)$$

$$e_{ij}^{(l)} = \text{LeakyReLU} \left( \mathbf{a}^{(l)T} \left( z_i^{(l)} \parallel z_j^{(l)} \right) \right) \quad (5.3)$$

$$\alpha_{ij}^{(l)} = \frac{e_{ij}^{(l)}}{\sum_{\{k \in N(i)\}} \exp(e_{ik}^{(l)})} \quad (5.4)$$

$$h_i^{(l+1)} = \sigma \left( \sum_{j \in N(i)} \alpha_{ij}^{(l)} z_j^{(l)} \right) \quad (5.5)$$

Donde la Ecuación (5.2) corresponde a la transformación lineal del embedding de la capa anterior  $h_i^{(l)}$  con  $W_i^{(l)}$  una matriz de pesos entrenable.

La Ecuación (5.3) calcula un puntaje de atención entre dos vecinos. Primero concatena los embeddings  $z$  de dos nodos. Luego realiza el producto punto entre este y una matriz entrenable  $\mathbf{a}^{(l)}$  y aplica una función LeakyReLU al final.

En Ecuación (5.4) se aplica una función softmax, con el objetivo de normalizar los puntajes de atención en las aristas entrantes de cada nodo.

Finalmente, en la Ecuación (5.5), al igual que en GCN, se lleva a cabo la agregación de los nodos vecinos, pero en este caso, se escala por el puntaje de atención. Se utiliza  $\sigma$  como la función de activación que se aplicará a la capa.

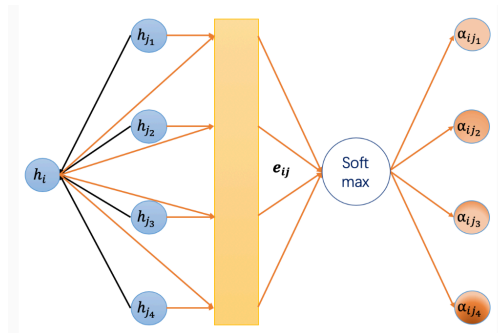


Figura 5.2: Pipeline del mecanismo de cálculo de las ponderaciones para una GAT.

### 5.0.4 GraphSAGE (SAmple and aggreGatE)

Es un framework de aprendizaje inductivo el cual nos permite aprender representaciones de los nodos de un grafo. A diferencia de los enfoques anteriores los cuales son inherentemente transductivos donde se crean las representaciones de los nodos por medio de la recolección de la información de todos sus nodos vecinos, utilizando factorización de matrices, GraphSAGE «aprende» a crear las representaciones de sus nodos, es decir GraphSAGE utiliza las características de nodos de su vecindario y la topología para aprender una función que genera los embeddings en base a un muestreo de nodos vecinos. Ayudado de esta forma a generalizar sobre nodos no vistos naturalmente [29]. GraphSAGE no necesita de todos sus vecinos durante el entrenamiento para crear una representación de el mismo, sino que a través de un subconjunto de estos aprenderá a crear un embedding, que representa su rol local y global en un grafo.

¿Qué significa que sea Inductivo?

Que sea inductivo significa que puede crear embeddings para nodos no vistos durante el entrenamiento. Es decir no necesita conocer todo el grafo ni todos los nodos para crear estas representaciones. Este enfoque es útil principalmente a la hora de trabajar con grafos dinámicos, batching/sampling, etc. Representando así a nodos en un vector de baja dimensionalidad y generalizando para luego nodos no vistos.

El proceso de creación de embeddings para los nodos del grafo están dados por las siguientes ecuaciones:

$$h_{N(i)}^{(l+1)} = \text{aggregate} \left( \{h_j^l, \forall j \in N(i)\} \right) \quad (5.6)$$

$$h_i^{(l+1)} = \sigma \left( W \cdot \text{concat} \left( h_i^l, h_{N(i)}^{(l+1)} \right) \right) \quad (5.7)$$

$$h_i^{(l+1)} = \text{norm} \left( h_i^{(l+1)} \right) \quad (5.8)$$

Donde  $h_{N(i)}^{(l+1)}$  de la Ecuación (5.6) representa las características de nodos vecinos de un nodo  $i$  en la capa  $l + 1$  el cual a través de una función de agregación cambian estos nodos vecinos (por ejemplo promedio, suma, lstm, etc). Luego tenemos  $h_i^{(l+1)}$  correspondiente a la concatenación de la representación anterior del nodo  $i$  y la de las características de nodos vecinos de la capa  $l + 1$ , correspondiente a lo previamente calculado. Finalmente tenemos  $\text{norm} \left( h_i^{(l+1)} \right)$  la cual se encarga de normalizar las características del nodo  $i$  en la capa  $l + 1$ .

A continuación tenemos Figura 5.3, el cual ilustra el proceso de creación de las representaciones de los nodos. Dado primero 1) por la selección de un número fijo de vecinos de un nodo, 2) Luego la agregación y concatenación de las características de estos nodos al

nodo dst junto con normalizacion, 3) Finalmente el paso de prediccion y ajuste de valores de los pesos de la red.

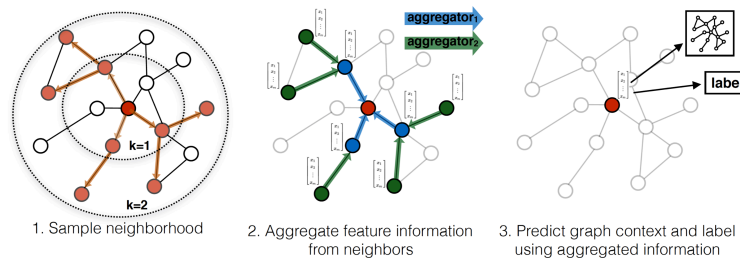


Figura 5.3: TODO.

## Vocabulario

- **Protocolos:** Un protocolo en Internet ... TODO: Completar
- **IRR:** Internet Routing Registry. Base de datos que contiene información sobre los prefijos IP y los sistemas autónomos que los anuncian. Los IRRs son utilizados por los operadores de red para filtrar rutas BGP y prevenir el anuncio de rutas falsas.
- **RIR::** Regional Internet Registry. Son organizaciones responsables de la asignación y administración de direcciones IP y números de Sistemas Autónomos (ASNs) en determinadas regiones geográficas. ARIN (American Registry for Internet Numbers): Norteamérica, el Caribe y África Subsahariana; RIPE NCC (Réseaux IP Européens Network Coordination Centre): Europa, Oriente Medio y Asia Central; APNIC (Asia-Pacific Network Information Centre): Asia y el Pacífico; LACNIC (Latin American and Caribbean Network Information Centre): América Latina y el Caribe; AfriNIC (African Network Information Centre): África.
- **AS::** Autonomous System
- **BGP::** Border Gateway Protocol
- **RRC::** Route Collectors
- **IXP::** Internet Exchange Point
- **Internet Carrier:** Empresas que operan infraestructuras de Red de gran escala. Ofrecen servicios de intercambio de datos.

Forman parte del núcleo de Internet. ejemplos de carrier son: AT&T.

- **ISP:** Internet Service Provider



- **CDN:** Content Delivery Network
- **ASN:** Número de sistema autónomo, un número entero de 32 bits que identifica de forma única una red. Por ejemplo, uno de los ASN de Cloudflare (tenemos varios) es 13335.
- **content provider network:** Red privada que conecta sus centros de datos a Internet, a menudo evitando los ISP regionales de nivel 1.[2]
- **Internet Service Provider (ISP):** Proveedor de servicios de Internet. Empresa que ofrece servicios de acceso a Internet a usuarios finales. Se encuentran en la parte más periférica de la red de Internet
- **Prefijos IP:** un prefijo IP es un rango de direcciones IP, agrupadas en potencias de dos. En el espacio IPv4, dos direcciones forman un prefijo /31, cuatro forman un /30, y así sucesivamente, hasta /0, que es la abreviatura de «todos los prefijos de IPv4». Lo mismo aplica para IPv6, pero en lugar de agregar 32 bits como máximo, puede agregar hasta 128 bits. La siguiente figura muestra esta relación entre los prefijos IP, a la inversa: un /24 contiene dos /25 que contienen dos /26 y así sucesivamente
- **Clique:** Un conjunto de Sistemas Autónomos que están interconectados, es decir, cada AS dentro del clique tiene una conexión directa con todos los demás AS dentro del clique. Este tipo de estructura suele encontrarse en la capa más alta de la jerarquía de Internet, conocida como Tier-1 ASes.
- **IANA:**