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

MIEMBROS DE LA COMISIÓN:
NOMBRE UNO
NOMBRE DOS
NOMBRE TRES

SANTIAGO DE CHILE
2025

dedicatoria.

Índice

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

En la sociedad actual el Internet juega un papel esencial en la vida cotidiana, facilitando la comunicación, la colaboración y el intercambio de información. En los últimos años, su uso y desarrollo ha crecido exponencialmente, convirtiéndose en una herramienta indispensable. Esta creciente interconexión global ha hecho que las redes sean fundamentales para el funcionamiento de la sociedad moderna. En este contexto, es crucial entender cómo está formado el Internet, ya que conocer su funcionamiento y estructura permite preservar su integridad y eficiencia. Además, garantizar un funcionamiento continuo y óptimo.\

El Internet está conformado por miles de Sistemas Autónomos (SA) interconectados entre sí. Cada SA consiste en un conjunto de IPs que comparten un mismo protocolo de enrutamiento y están administradas por una misma entidad, como proveedores de servicios de Internet (ISP), empresas comerciales, universidades, entre otros. A cada uno de estos Sistemas Autónomos se le asigna un número y prefijos de direcciones IP, los cuales anuncia a sus vecinos a través del Border Gateway Protocol (BGP). BGP es un protocolo dinámico de enrutamiento externo en el que los SA anuncian sus tablas de ruteo y cambios en sus AS-Paths para alcanzar direcciones IP específicas. De esta manera, cada SA recibe estos anuncios de todos sus vecinos BGP y toma decisiones sobre la mejor forma de direccionar sus paquetes. \

Dentro del grafo de Sistemas Autónomos que conforma Internet, el camino que un paquete recorre de un nodo a otro no suele ser el más corto debido a los acuerdos comerciales que cada SA, como entidad independiente, establece con sus vecinos. Estos acuerdos se clasifican en tres tipos de relaciones: 1) Provider-to-customer (P2C), en el cual el cliente paga al SA proveedor para que este enlace permita el tráfico de sus paquetes hacia el resto de Internet. 2) Peer-to-peer (P2P), donde los SA intercambian tráfico entre sí y con sus clientes, pero no con sus proveedores u otros pares. 3) Sibling-to-sibling (S2S), cuando dos

SA pertenecen al mismo dominio. Gao [???] propuso reglas para modelar estas relaciones entre SA, que reflejan cómo suelen configurarse en BGP, lo que permite inferir las posibles rutas seleccionadas por este protocolo. Sin embargo, estas soluciones se basan principalmente en el cálculo de heurísticas. Por otro lado, estudios más recientes como el de Shapira y Shavitt[1] proponen técnicas de Deep Learning, creando representaciones de los SA que luego son utilizadas en una Red Neuronal.\

Aquí entra en juego un nuevo enfoque que podría cambiar nuestra forma de analizar datos: el uso de Redes Neuronales de Grafos (GNNs). Las GNNs están diseñadas específicamente para trabajar con datos organizados en forma de grafos. Al ser Redes Neuronales, tienen la capacidad de encontrar representaciones efectivas de la información y descubrir patrones en datos estructurados de esta manera. A diferencia de las Redes Neuronales convencionales, las GNNs pueden aprovechar la información de los nodos vecinos, lo que les permite entender mejor la estructura del grafo y realizar análisis más detallados.\

Así nace esta tesis, con el objetivo de explorar el comportamiento de las Redes Neuronales de Grafos (GNNs) utilizando datos de BGP para representar Internet y las relaciones entre los Sistemas Autónomos que lo componen. Esta área es aún poco explorada y presenta varias dificultades, principalmente debido a la necesidad de aplicar conocimiento tanto en redes como en Deep Learning. En particular, obtener una representación precisa de la topología de Internet requiere comprender a fondo el funcionamiento del protocolo BGP y saber cómo y dónde obtener datos, que no siempre están disponibles públicamente. Así como también tener un conocimiento de teoría de grafos y Deep Learning necesario al momento de la implementación de un modelo de GNNs y diferentes técnicas para la tarea requerida.

1.1 Motivación

(está copiado el problema, es similar pero revisar de cambiar la forma en cómo se plantea)

1.2 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.

1.3 Objetivos

1.3.1 Objetivo general

El objetivo principal de este estudio es evaluar diversas arquitecturas de Redes Neuronales de Grafos (GNNs) para determinar su viabilidad en la inferencia del tipo de relación de tráfico entre dos Sistemas Autónomos. Esto se logrará mediante el análisis de características específicas de cada Sistema Autónomo, la información de actualizaciones BGP, la topología y los cambios en esta.

1.3.2 Objetivos específicos

1. Obtención de datos: Recopilar datos de fuentes confiables como

que correspondan a Sistemas Autónomos representativos de la Red de Internet. Esto implica obtener datos sobre nodos, características y relaciones entre ellos. Asimismo, obtener información relevante sobre flujos de paquetes BGP.

1. Preparación de datos: Mejorar la calidad de los datos mediante el uso de técnicas de normalización, conversión de atributos categóricos a numéricos, manejo de desequilibrio de clases, entre otros.

Además, construir el grafo y definir cómo se proporcionarán los datos de entrada a nuestros modelos GNNs.

1. Diseño e implementación de modelos: Diseñar e implementar modelos GNN y framework específicos que permita la inferencia del tipo de relación que dos Sistemas Autónomos comparten.
2. Evaluación de performance: Comparar el desempeño de diferentes arquitecturas de GNNs en las inferencias, identificando los parámetros de mayor relevancia.
3. Análisis de resultados: Comprender los resultados obtenidos mediante el estudio y la comparación con los valores esperados y estado del arte [1].

1.4 Metodología

El plan de trabajo que se espera llevar a cabo durante esta investigación consta de cuatro etapas:

Investigación y familiarización En esta primera etapa, se llevará a cabo la lectura de artículos académicos relacionados con el uso de GNNs, además de artículos relevantes en la representación de datos de internet, con el objetivo de adquirir conocimiento sobre el problema en cuestión. Al mismo tiempo, se realizará un estudio detallado de datasets representativos de internet y, más importante aún, de la topología de BGP, junto con actualizaciones de estos e información adicional que se puede obtener tan-

to de sistemas autónomos como de los paquetes que intercambian. En paralelo a la investigación, se procederá al desarrollo de modelos básicos de GNNs con el propósito de familiarizarse con las herramientas que se utilizarán a lo largo del proyecto.

Preparación de datos Una vez se tenga información sobre la topología BGP, los Sistemas Autónomos que la componen y los tipos de relaciones de entre ellos, se procederá a convertir los datos a la representación de entrada que nuestro modelo recibirá. Esto también implica el uso de diversas técnicas destinadas a mejorar la calidad de los datos. El enfoque de esta etapa dependerá del estado inicial de los datos, lo que podría implicar acciones como la limpieza de datos, normalización y reducción de la variabilidad, entre otros procesos que se consideren necesarios.

Construcción de modelos y entrenamiento Una vez finalizada la investigación y la familiarización con el problema y las herramientas pertinentes, se dará inicio a la implementación de diversos frameworks y metodologías, utilizando diferentes modelos de GNNs con el conjunto de datos. Posteriormente, se procederá a entrenar los modelos y a ajustar los hiperparámetros o realizar cambios según sea necesario. Se realizará un seguimiento de los resultados, comparándolos con los hallazgos de los artículos académicos previamente revisados. Esto permitirá un proceso de mejora continua, aprendizaje y adaptación en la creación de estos modelos.

Análisis de resultados Una vez terminada la construcción de los modelos, se procederá a analizar los resultados obtenidos para finalmente empezar a escribir el informe de esta tesis.

1.5 Contribuciones

1.6 Estructura del trabajo

la tesis está organizada de la siguiente manera:

- capítuulo 2 blabla
- capítulo 3 blablabla
- capítulo 4 no existe todavía

Capítulo 2

Infrestructura de Internet

El Internet es una red global de redes que interconecta miles de millones de dispositivos en todo el mundo que proporciona 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 traves de Internet, los datos a enviar son encapsulados en paquetes con cabeceras que contienen la información necesaria para llegar a traves de Internet al dispositivo final. En algunos casos los datos puedes ser separados en diferentes paquetes y ser enviados por diferentes 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 provedores de Internet (ISPs), los cuales a su vez son una red conformada por routers y enlaces de comunicacion. 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. Asi estós proporcionan a los usuarios finales el acceso a provedores de contenidos (CDN por sus siglas en ingles), sitios web y otros servicios, los cuales también estan conectados a la infrestructura de Internet

Para el correcto funcionamiento de Internet existen diferentes protocolos, sin embargo, los dos más importantes corresponden al Protocolo de Control de Transmisión (TCP) y el Protocolo de Internet (IP),conocidos colectivamente como TCP/IP. El desarrollo de estos estandares los lleva acabo el Internet Engineering Task Force (IETF) [4], y los documentos se conocen como Requests for Comments (RFCs).

2.1 Sistemas Autonomos

En la estructura de Internet un Sistema Autonomo (AS) consiste en un conjunto routers y enlaces de comunicación que comparten una política de enrutamiento común. Estos son operados de forma independiente por una organización, como un ISP, una universidad, una empresa, entre otros.

Los AS utilizan el protocolo BGP, para intercambiar información de enrutamiento y así tener una conexión global entre ASes y por ende del Internet. Las conexiones entre los Ases pueden ser de cliente-proveedor o de peer-to-peer, pero están influenciadas por acuerdos comerciales. Bajo estos acuerdos, podemos crear un grafo de la infrestructura de Internet, donde los nodos son los ASes y las aristas son las conexiones entre ellos. Cabe destacar en este caso conexión no implica necesariamente que haya un intercambio de trafico entre los ASes conectados, ya que el enrutamiento está controlado por BGP, un protocolo de enrutamiento basado en politicas (reachability)[5].

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

Los ISP, quienes puedes estar conformados por uno o varios Sistemas Autonomos, se dividen comunmente en tres niveles de jerarquía. El Tier-1, donde se encuentran los AS que conformaan el «backbone» de Internet, estos intercambian paquetes entre si 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 [6]. Tambien Luckie et al.[7] analizó los AS y propuso una métrica para indicar que tan global es un AS, donde si el customer cone es mayor o igual a 200 se considera Tier, si es mayor a 2000 se considera Tier 2 y si es menor a 200 se considera Tier 3.

Una ultima separación que se puede hacerentre AS es segun sus conexiones a otros Sistemas Autonomos, en single-homed y multi-homed. Los AS single-homed tienen solo una conexión a otro AS, mientras que un multi-homed tiene conexiones a más de un AS.

2.2 Internet Exchange Points

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

Un IXP consiste generalmente en un switch de alta velocidad y capacidad que conecta routers de diferentes AS, permitiendo el intercambiodirecto de tráfico sin necesidad de atravesar por redes intermedias. Mejorando asi el intercambio de tráfico haciendolo más eficiente y de bajo costo.

Las relaciones de tráfico en un IXP seestablecen mediante el protocolo BGP, por ende a pesar que los AS esten fisicaamente conectados a tarves del IXP, no se establecen un relación a no ser que se configure explicitamente.

Al igual que los AS, los IXPs tienen un ASN. Sin embargo, de forma generalizada, este ASN se extrae de los AS PATH en BGP. Algunos IXP lo mantienen visible como metodo de debbuging [5].

2.3 Ruteo

El ruteo es el proceso mediante el cual se selecciona el camino que seguirá un paquete dentro de una red para llegar a su destino mediante la mejor ruta. La comunicación entre dos dispositivos de la red puede establecerse mediante diferentes rutas, lo que permite conectar dispositivos de red sin una conexión directa a través de dispositivos intermedios.

Un enrutador o router es un dispositivo de red encargado de seleccionar las rutas que seguirán los datos enviados a traves de la red. El camino que sigue un paquete se elige según la información de las tablas de enrutamiento (RIBs) de los routers y en la información contenida en los encabezados de los paquetes, donde se indica el destino final. Cuando llega un paquete a un router, se consulta en la tabla de enrutamiento la dirección final para obtener el proximo router o punto de red al cual se debe dirigir el paquete. Cuando un paquete llega a un router, este consulta su tabla de enrutamiento para determinar el siguiente salto, es decir, el próximo router o punto de red al que debe enviarlo.

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 reenvia 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 ir

actualizando 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.

También se pueden clasificar los protocolos de enrutamiento en dos categorías: ruteo interno y ruteo externo. El ruteo Interno se encarga de gestionar las rutas a seguir de un paquete dentro de un Sistema Autónomo. Algunos d elos protocolos son:

- **OSPF (Open Shortest Path First):** Utiliza el algoritmo de Dijkstra para determinar las rutas más cortas entre nodos [8].
- **RIP (Routing Information Protocol):** Utiliza un enfoque de vector de distancia para calcular la ruta más optima, basándose en la cantidad de saltos [9].

El ruteo externo se centra en la gestión de rutas entre los Sistemas Autónomos que conforman el Internet. En este caso, se usan protocolos de enrutamiento externo. Algunos protocolos de enrutamiento externos son:

- **BGP (Border Gateway Protocol):** Tiene un enfoque de vector de distancia. Utiliza un enfoque de vector de distancia y toma decisiones basadas en políticas de red para intercambiar información eficientemente [10].
- **IS-IS (Intermediate System to Intermediate System):** Protocolo deenrutamiento de estado de enlace, similar a OSPF [11].

Capítulo 3

Border Gateway protocol (BGP)

Como se mencionó en la sección anterior, BGP [10] es un protocolo de enrutamiento utilizado para intercambiar información de rutas entre Sistemas Autónomos en Internet. El cual utiliza TCP [12] 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.

Como sabemos el Internet esta formado por miles de redes privadas, públicas, corporativas y gubernamentales que están interconectadas mediante protocolos estandarizados entre sí. BGP se encarga de analizar todas la posibles rutas a los diferentes destinos y seleccionar la mejor ruta.

A medida que un paquete viaja por las diferentes redes de Internet, cada Sistema Autonomos decide el siguiente salto por el cual se enviara un el mensaje. Esta desición se toma en base a la información de ruteo recolectada por en intercambio de mensaje BGP.

Por ejemplo, cuando un usuario en Chile carga una página web con servidores en Argentina, BGP permite que la comunicación se establezca ya que asegura que los paquetes sigan la mejor ruta disponible a través de múltiples redes interconectadas.

3.1 Funcionaqmiento 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 mensajes *KEEPALIVE* para confirmar la conexión.

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

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

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*.

A continuación la figura Figura 3.1 muestra un ejemplo de como BGP propaga la información de ruteo a través de los diferentes AS. quienes repiten el proceso de anunciar la dirección de destino a sus pares modificando la informacipon de AS PATH y asi sucesivamente hasta qie la información llega al AS 64501, el cual gracias a la informacion acumula podra sabra como llegar a la red del AS 64503 y podrá eleir entre los dos posibles rutas que tiene almacenda dentro de sus rutas .

La figura Figura 3.1 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 especifico 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, determinar la mejor ruta para alcanzar la red del AS 64503.

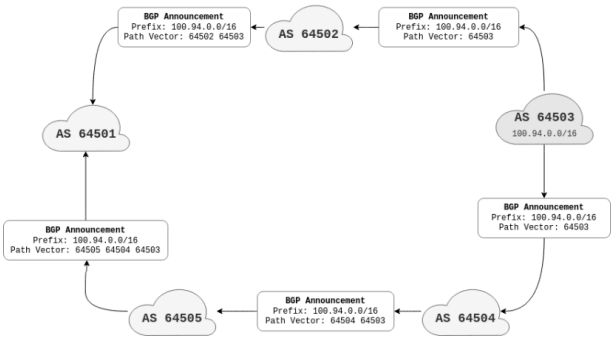

Figura 3.1: Ejemplo de propagación de anuncion. En este caso AS 6453 comienza anunciando su dirección a sus pares.

Mensaje OPEN

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

En este mensaje se especifica la versión de BGP ha 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* una sesión antes de que la conexión sea cerrada.

Mensaje UPDATE

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

Además, estos mensajes incluyen path atributes, los cuales proporcionn información sobre las rutas que se están anunciando para que luego cada AS pueda decidir la mejor ruta a los ddifernetes 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 unicamente del *HEADER* de un mensaje BGP, donde se indica que corresponde a un *KEEPALIVE* por medio del campo *type*, cuyo valor es 2el cual corrsponde 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 es cerrada.

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.

3.2 BGP Routing Information Base (RIB)

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 las rutas seleccionadas ser anunciados 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 infomración de ruteo local, es decir las mejores rutas seleccionadas para cada dirección.
- **Adj-RIB-Out:** Guarda la información selecionada por router para ser anunciada a sus pares. Consiste en la información de Loc_RIBs luego de ser aplicadas las politicas 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. Con 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.

3.3 EXTRAS

- Export policies:
  - PErmitir/denegar una ruta
  - Asignar multiples exit discriminator
  - Agregar community values
  - Prepending

Capítulo 4

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 [13].

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

Gao [5] 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 parcialemente 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 acuerdo 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 [14].

Conocer las relaciones entre los ASes es de vital imporcia, ya que permite entender el comportamiento del trafico 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 metodos heurísticos y basandose los datos públicos sobre enrutamiento BGP, como las tablas de enrutamiento (RIBs) y los anuncios BGP.

Gao presento un primer algoritmo basando 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 metodo 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 en trafico 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 [15].

Subramanian et al. [16] formuló la tarea como un problema de optimización, definiendolo 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, excñuyó 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 [17, 18, 15, 19, 13].

Luckie et al. [20] introdujeron el algoritmo *AS-Rank* considerado el estado del arte para inferir C2P y P2P utilizando datos de BGP. Ellos se basarón 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. Basándose en estos supuestos, 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 ultimo, Shapira y Shavitt [21] 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 [22].

En nuestro caso desarrollamos esta tarea utilizando redes neuronales de grafos, realizando un experimento similar al de Shapira y Shavitt [21], 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 relizar uan comparacion entre estas tecnicas.

4.1.1 Gao

El algoitmo presentado por Gao Fue la primera en abordar el problema de inferencia de los tipos de relaciones entre Sistemas Autónomos. a partir de tablas de enrutamiento BGP y la heuristica 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. Ene ste análisis, el 99.1% de los resultados de inferencia fueron confirmados por AT&T.

Su algotimo 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 clasificaciónde relaciones esaba dado por la logica de que una ves encontrado el top provider de un as path, los pares AS que se encontrasen antes del top provider tendrian una relación de P2C, mientras que los pares que se encontrasen despues del top provider tendrian una relación de P2P y realcion S2s si ambos pares proveian de transito para each other es decir, podia encontrarse ese par tanto antes como despues del top provider.

Pero como no se puede asumir que todods lso speaging routers que pertences a un AS pueden estar bien configurados, Gao propuso un amejora para evitar inferncias incorrectas la cual consistia en contar un numero de Routing tbales que que concluian que una relacion era de un tipo y otra. es decir, si no más de L routing tables inferian que un AS u preoveia de transito a AS vy más de L routing tables inferian que v proveia d etransito a u entonces se ignoraba que u proveia de trnasito a v, sendo L una constante pequeña.

En caso de las relaciones de P2P Gao primero identifico aquellos Ases con qquienes no podría establecer relaciones de peering las identifico bajo la idea de que en una relacion de peering los ASes no differ more than R times.

4.1.2 Lu Rank

Uno de los algoritmos más recientes es el propuesto por Ruan y Susan en 2014 [13]. 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 AS siguen una política uniforme de exportación de rutas. Según esta política, las rutas provenientes de proveedores y peers no son exportadas AS vecinos que también son proveedores o peers. Bajo este 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 número 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»*, prupusieron un método para clasificar las relaciones observadas entre AS, sustentandose en las relacioens de tránsito entre ellas, a tarves de los datos de BGP. A diferencia del algoritmo de Gao [5] [TODO:], que se basa principalmente en el grado de los AS, este metodo utiliza el concepto de grado de tránsito de los ASlo cual en conjunto con la ideantificación 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 [23]. 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 cnt(X, Y) y 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 suiguientes reglas:

- Si cnt(X, Y) > 0 y cnt(Y, X) = 0, se establece una relación P2C entre X y Y.
- Si cnt(X, Y) > 0 y cnt(Y, X) > 0, y al menos uno entre X o Y es un AS Tier-1, la relación entre ellos es P2P.
- Si cnt(X, Y) > 0 y 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 infomación al igual que en la fase anterior se puede determianr las relaciones de los AS paths bajo las siguientes reglas:
   - Si cnt(X, Y) > 0 y cnt(Y, X) = 0, entonces X y Y tienen una relación provider-to-customer (P2C).
   - Si cnt(X, Y) > 0 y 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. caso contrario se mantiene como P2P.

4.1.3 BGP2Vec

En 2020, Tal Shapira y Yuval Shavitt presentaron un nuevo enfoque [21], para la inferencia de Sistemas Autónomos utilizando por primera vez técnicas de Deep Learning. Este metodo se realiza se basa en la creacion de empebings de estos Sistemas autonomos utilizando unicamente anuncion 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 consistio 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 generar embeddigs 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 logaritmica de cualquier ASN de contexto dado el ASN de entrada.

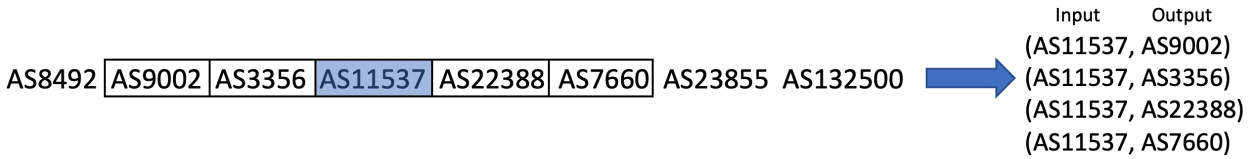

Figura 4.1: ???.

Una vez entrenado los embeddings para casa ASN, se continua con la tarea de classificación de las relaciones entre los Sistemas Autónomos. Para esto se ocupa un Red Neuronal Artificial de dos capas.

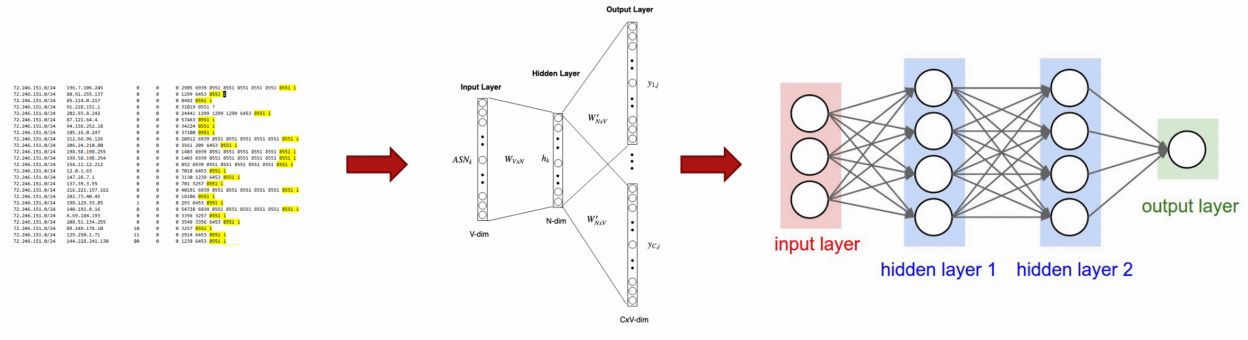

Figura 4.2: ???.

Para la etapa inicial de entrenamiento de BGP2Vec, se utilizaron anuncios extraídos de RouteViews [23], 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 [22], que contiene relaciones P2P y P2C/C2P. Además para algunos experimentos, se empleó el dataset ToR de BGProtect (www.BGProtect.com) [24], basado en el trabajo de Shavitt et al. [25]..

4.2 ProbLink

ProbLink es un algoritmo probabilistico para inferir las relaciones entre Sistemas Autonomos propuesto por Yuchen Jin et al. [15]. Este permite el uso de atributos con valores estocásticos. Toma en cuenta información sobre los links y caminos qure atraviesan . Pro-

bLink no impone un orden específico en que los ASes y lso enlances deben ser analizados, en su lugar itera con tinuamente hasta alcanzar una convergencia.

Para cada atributo se calcula la distribución de probabilidad condicional basada en lso 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 probabilistico y se recalcula las distribuciones d elas caracteristicas utilizando los valores de probaboilidad actualizadas de cada enlace. Se repite el proceso hasta la convergencia, es decir , ahsta que el porcentaje de enlaces que cambia de etiqute 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.

4.3 Desafíos en la inferencia de Relaciones entre Sistemas Autónomos

Capítulo 5

Redes Neuronales

5.1 Grafos

Un grafo (Figura 5.1) es una estructura discreta formada a partir del conjunto de vértices (también conocido como nodos) y aristas las cuales son las uniones entre estas [26]. De forma más sencilla un grafo es una representación visual de una relación binaria. Un gráfo G se caracteriza mediante la pareja de conjuntos (V, E) donde V es el conjunto no vacío de vértices y E denota el conjunto de aristas, este último es, a su vez, es un conjunto de pares de nodos. Así, la definición de un grafo está dada por $G = (V, E)$. Usamos $v_i \in V$para denotar que un nodo forma parte del grafo y $e_{ij} = (v_i, v_j) \in E$ para indicar que existe una arista entre el nodo v_i y v_j . Cada nodo v_i tiene vecinos con los cuales comparte una arista, estos se representan de la forma $N(v_i) = \{v_j \in V : (i, j) \in E\}$. El número de vertices y aristas en un grafo se representan mediante $n = |V|$ y $m = |E|$.

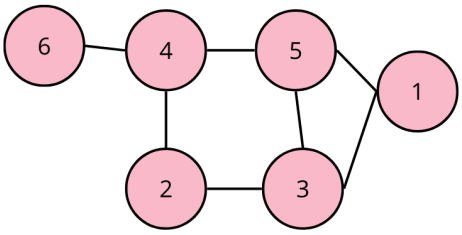

Figura 5.1: Representación de un grafo no dirigido de 6 nodos numerados.

Una forma de representar un grafo es mediante una matriz de adyacencia denotada $A \in \mathbb{R}^{n \times n}$, donde un el valor $A_{ij} = 1$ si $e_{ij} \in E$ y $A_{ij} = 0$ si $e_{ij} \notin E$. Si la matriz es simétrica, el grafo es no dirigido; de lo contrario, se trata de un grafo dirigido.

Nodos y Aristas de un grafo pueden contener atributos. De esta manera, los atributos de los nodos pueden ser representados mediante una matriz $H_n \in \mathbb{R}^{n \times d}$ donde cada fila representa un vector de características de un nodo. En el caso de los atributos de las aristas, estos pueden ser representados por la matriz de adyacencia, en la cual, en lugar de contener 1 y 0, contiene dichos atributos.

Además, los grafos pueden clasificarse en diferentes categorías que ofrecen información adicional y características distintivas. A continuación, se presentan algunas categorías comunes:

- Grafos dirigidos/no dirigidos: En un grafo dirigido, cada arista tiene una dirección específica, indicando un flujo unidireccional entre los nodos conectados. A diferencia de un grafo no dirigido, donde las aristas no tienen una orientación definida, lo que representa conexiones bidireccionales entre nodos.
- Grafos homogéneos/heterogéneos: En un grafo homogéneo, tanto nodos como aristas son del mismo tipo, en contraste de grafos heterogéneos donde los nodos y aristas pueden ser diferentes y por tanto representar cosas diferentes.
- Grafos estáticos/dinámicos: Un grafo dinámico experimenta cambios en su estructura a medida que transcurre el tiempo, a diferencia de un grafo estático, que mantiene una topología constante en función del tiempo.

5.2 Inteligencia Artificial

Inteligencia Artifial (IA) es un campo de la informatica que busca simular el comportamiento de la inteligencia humanana, es decir, intenta replicar y automatizar la capacidad del ser humano para tomar desiciones.

Dentro del área de la Inteligencia Artificial, nos encontramos con el Machine Learning, diciplina que a través del desarrollo de algoritmos y modelos busca que las máquinas aprendan patrones por medio de la experiencia, la cual incluye datos de entrenamiento y retroalimentación. El objetivo es entrenar una máquina para una tarea especifica sin la necesidad de programar explícitamente un algoritmo.

Finalmente, dentro de Machine Learning se encuentra el campo de Deep Learning, un área que se centra en el uso de arquitecturas de Redes Neuronales profundas para aprender representaciones de datos de manera jerárquica. A diferencia de Machine Learning convencional, donde las características se extraen manualmente de los datos y se proporcionan al modelo, en Deep Learning, estas representaciones se aprenden de forma automática mientras el modelo lleva a cabo la tarea asignada. Una característica distintiva de esta disciplina es el uso de Redes Neuronales, estructuras compuestas por múltiples capas entre la entrada y la salida. Cada capa procesa la información y extrae características cada vez más abstractas a medida que se profundiza en la Red. Permitiéndole al modelo así capturar patrones y características complejas en los datos.

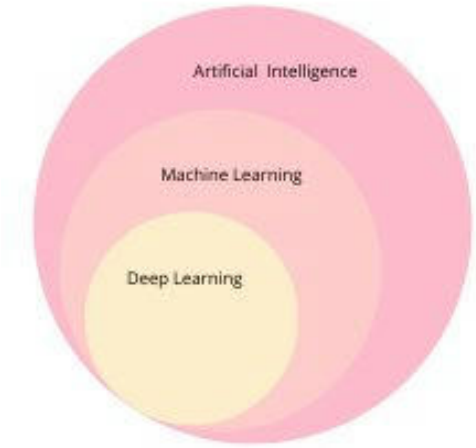

Figura 5.2: jerarquía conceptual entre Inteligencia Artifial, Machine Learning y Deep Learning.

1. Machine learning es una rama de la inteligencia artificial, que ya no depende de unas reglas y u programador, sino que la computadora puede establecer sus propias reglas y aprender por si misma
2. Aprendizaje Supervisado:

5.3 Redes Neuronales

Una Red Neuronal es un modelo computacional compuesto de neuronas (perceptrones), dispuestas en capas y conectadas entre si con el fin de aprender patrones mediante el intercambio de información ponderada por pesos. Estos pesos se ajustan en base a los datos de entrada, asignando valores en función del reconocimiento de patrones, que permiten una salida esperada.

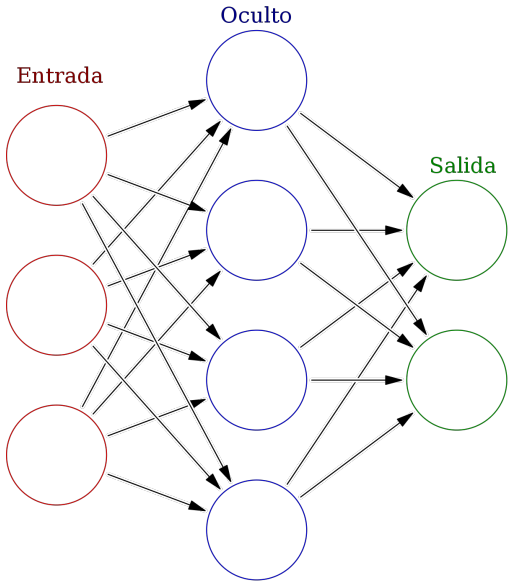

Figura 5.3: Arquitectura básica de una Red Neuronal Fully Connected de una capa.

En una Red Neuronal, cada unidad toma entradas, las pondera por separado, suma los valores y pasa esta suma a través de una función para producir una salida, la cual es compartida con otras neuronas a las cuales está conectada. El perceptrón, que funciona como una representación matemática de una unidad básica en la Red, realiza cálculos para determinar tendencias en los datos de entrada, asignándole diferentes pesos a cada valor de entrada en base a patrones entre los datos para realizar tareas específicas.

Un perceptrón está compuesto por cuatro elementos distintivos (Figura 5.4), i) los valores de entrada definidos como $x_i, x_{i+1}, ..., x_{n-1}, x_n$ donde cada x_j corresponde a un vector de tamaño d, ii) los pesos definidos como $w_j \in \mathbb{W}^{n \times d}$ donde $\mathbb{W}$ corresponde a la matriz de pesos los cuales son ajustados durante la etapa de entrenamiento de la Red, iii) la suma $z = \sum_{j=1}^{d} w_j x_j + b$ y iv) la función de activación, la cual establece un umbral de salida para evitar que los valores de salida se disparen. Esta función de activación permite incluir más capas y, por ende, mayor complejidad en las arquitecturas de redes que se construyan. Las funciones de activación tienen la capacidad de mejorar el aprendizaje de patrones en los datos[27]. Algunas de las funciones de activación comúnmente empleadas incluyen la Sigmoide, la Tangente Hiperbólica (tanh), la Rectified Linear Unit (ReLU) y la Leaky ReLU.

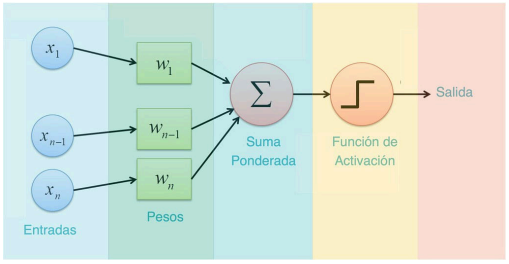

Figura 5.4: Estructura general de un perceptrón.

La técnica comúnmente utilizada para el entrenamiento de Redes Neuronales es el *back-propagation*, que tiene como objetivo ajustar los pesos de los parámetros de la Red para minimizar la función de pérdida[28]. Esta función cuantifica la diferencia entre las predicciones hechas y los valores reales. Una vez que se ha calculado la pérdida, el proceso de optimización se centra en modificar los pesos para mejorar la precisión general de la Red.

Durante el entrenamiento de las Redes Neuronales, se emplea el descenso de gradiente, un método que implica el cálculo de la derivada de la función de perdida con respecto a los pesos de la Red. Este cálculo determina la dirección y magnitud en la que los parámetros de un modelo deben ser ajustados para minimizar la función de perdida. Por ende, es fundamental que esta función sea continua y derivable. En problemas de regresión, se suele utilizar funciones como el Mean Squared Error y Mean Absolute Error, mientras que en problemas de clasificación, destaca la Cross-Entropy Loss.

5.3.1 Redes Neuronales Feed Forward (FFNN)

También conocida como *multilayer perceptrons*, esta arquitectura representa la forma más simple y fundamental de una Red Neuronal, sirviendo como la base de la mayoría de los modelos de Deep Learning. En esta arquitectura la información fluye exclusivamente hacia «adelante», sin bucles o conexiones hacia atrás.

El flujo de información comienza en la capa de entrada, donde se reciben los datos, seguida de las capas ocultas (*hidden layers* en ingles), siendo las Fully Connected las más comunes (Figura ???fully-connected), donde cada neurona está conectado a cada neurona de la capa anterior. De esta manera, las salidas de cada perceptrón generan una salida que, al estar conectada con otros nodos, funcionan como entrada para la siguiente capa, continuando así hasta llegar a la capa de salida.

El objetivo principal de una Red Feed Forward es aproximar alguna función $f(x)$. Por ejemplo, en un problema de regresión, se busca modelar la relación $y = f(x)$.

5.3.2 Redes Neuronales Recursivas (RNN)

Las Redes Neuronales Recurrentes (RNN) son una variante de las Redes Neuronales Feed Forward, diferenciándose por su capacidad para retener y utilizar información previa, es decir, poseen «memoria».

A diferencia de las Redes Neuronales Feedforward convencionales, que asumen que los datos de entrada en cada capa son independientes entre sí, las Redes Neuronales Recurrentes (RNN) introducen conexiones entre las salidas previas y la salida actual, generando así un proceso de retroalimentación.

Esta característica en las RNN las hace particularmente eficientes para trabajar con datos secuenciales, como en aplicaciones de procesamiento del lenguaje natural incluyendo traducción, generación de texto y la predicción de series temporales.

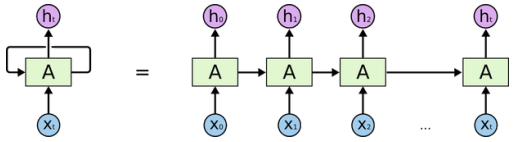

Figura 5.5: Arquitectura básica de las Redes Neuronales Recurrentes.

La imagen de arriba es una representación simple de una Red Neuronal Recurrente (Figura 5.5). En el lado izquierdo se encuentra la notación abreviada y, en el lado derecho, la notación desplegada para representar RNNs. Donde x_t es un vector que representa la entrada en el instante de tiempo t. A el estado oculto con el paso del tiempo t y actúa como la «memoria» de la Red, calculando en función del estado oculto anterior y la entrada en el paso actual.

5.3.3 Redes Neuronales Convolucionales (CNN)

Las Redes Neuronales Convolucionales (Figura 5.6) son un tipo especializado de modelo de Red Neuronal diseñado especialmente para procesar información en forma de grilla [29]. Su aplicación principal se encuentra en el análisis de imagenes, en el reconocimineto de objetos, clases y categorías.

Las CNN se componen de una capa de entrada, una capa de salida y varias capas ocultas intermedias. Estas capas ocultas llevan a cabo operaciones de convolución, lo que les permite aprender características específicas de las imágenes. En el proceso de convolución, se aplican filtros, a través de matrices de pesos. Estos filtros aprenden a detectar diversas

características como bordes, patrones, colores, entre otros. Así a medida que se avanza en las capas de la CNN, la red es capaz de reconocer elementos más complejos.

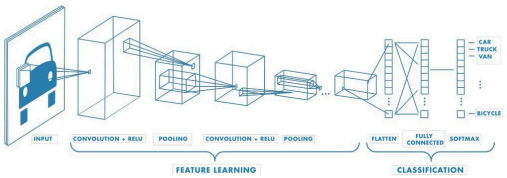

Figura 5.6: Estructura de un modelo de Convolucional con tres capas.

5.3.4 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.

Existen diferenets enfoques, dependiendo de la tarea de aprendizaje que se quiere llevar a cabo, estos son:

- Nivel de nodo: Incluye tareas como clasificación, regresión y clustering de nodos. Se realizan inferencias a partir de las conexiones con otros nodos.
- Nivel de aristas: Se abordan tareas de clasificación y predicción de aristas. Por ejemplo, determinar la existencia de una relación entre dos nodos.
- Nivel de grafo: Se encuentran tareas de clasificación, regresión y matching de grafos para las cuales el modelo debe ser capaz de aprender una representación para el grafo completo.

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 sig-

nificativos en el desempeño de los modelos. Las GNNs, por su parte, ofrecen flexibilidad para adaptarse a distintos tamaños de entrada[30].

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[31]. 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.

Así, la esencia detrás del uso de GNNs radica en su capacidad de entrenar un modelo que pueda procesar un grafo, sus nodos y relaciones, logrando identificar patrones relevantes en la topología para lograr de forma efectiva la tarea asignada. Por ejemplo, en el ámbito de las redes sociales, las GNNs pueden ser utilizadas para clasificar usuarios según sus interacciones, identificando así grupos afines. Otra aplicación puede ser la recomendación de contenido de interés de un usuario, basándose en sus conexiones y preferencias históricas. En el campo de la biología, es posible predecir el tipo de moléculas basándose en sus características estructurales y propiedades.

El diseño de una GNN se hace por medio de la combinación de diferentes módulos:

- Módulo de propagación: Este módulo se utiliza para propagar información entre los nodos capturando tanto la topología como los atributos de los nodos. Esto se logra combinando los datos de cada nodo con los de sus vecinos.
- Módulo de muestreo: Cuando los grafos son muy grandes, se utiliza generalmente un módulo de muestreo con el fin de seleccionar un subconjunto del grafo, aportando de este modo en la capacidad de generalización de un modelo y reducción de complejidad. Se combina generalmente con un módulo de propagación.

- Módulo de pooling: Cuando se necesita representaciones de subgrafos su utiliza este módulo para extraer información de los nodos. Se utiliza para reducir la dimensionalidad de las representaciones de nodos.

Un modelo GNN se construye generalmente combinando estos módulos. A continuación (Figura 5.7), se ilustra el pipeline de una arquitectura GNN. El modelo recibe como entrada un grafo, y en la capa GNN, se emplea un operador convolucional, un módulo de muestras y una operación skip-connection que se fusionan para propagar la información y extraer detalles de alto nivel mediante el módulo de pooling. Después de pasar por todas las capas intermedias, se obtiene una salida en forma de embeddings, a los cuales se les aplica una función de pérdida para obtener los resultados del ajuste del modelo en base a la tarea asignada.

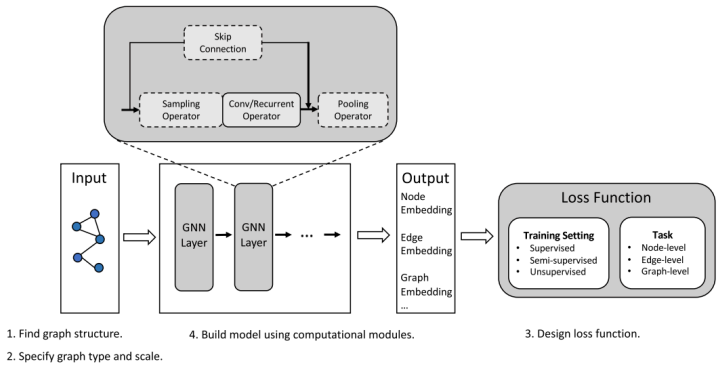

Figura 5.7: Pipeline básico de una arquitectura GNN.

5.3.4.1. 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[32].

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

1. Propagacion de mensajes entre nodos: Cada nodo envía un mensaje que contine 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 5.8), 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.

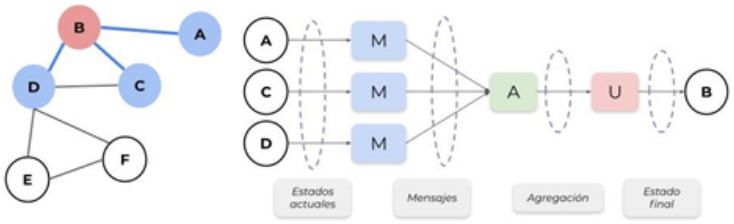

Figura 5.8: Flujo de Message passing para un nodo del grafo.

5.3.4.2. 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[33]. 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:

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

Donde σ se define como la función de activación, $H^{(l)}$ la matriz de caracteristicas 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 asi 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.

5.3.4.3. 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[34].

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 5.9).

Así tenemos:

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

$$e_{ij}^{(l)} = \text{LeakyReLU}\left(\boldsymbol{a}^{(l)T}\left(\boldsymbol{z}_i^{(l)} \parallel \boldsymbol{z}_j^{(l)}\right)\right) \tag{5.3}$$

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

$$h_i^{(l+1)} = \sigma\left(\sum_{jinN(i)} \alpha_{ij}^{(l)} z_j^{(l)}\right) \tag{5.5}$$

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 $a^{(l)}$ 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 σ como la función de activación que se aplicará a la capa.

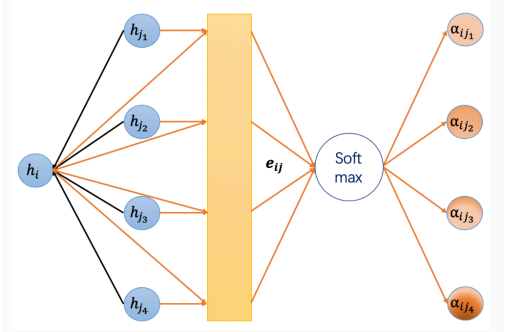

Figura 5.9: Pipeline del mecanismo de cálculo de las ponderaciones para una GAT.

5.3.4.4. 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 infomeación de todos sus nodos vecinos, utilizando factorización de matrices, GraphSAGE «aprende» a crear las representaciones de sus nodos, es decir graphSAGE utiliza las caracteristicas de nodos de su vecindario y la topología para aprender una funcion que genera los embeddings en base a un muestreo de nodos vecinos. Ayudado de esta forma a generalizar sobre nodos no vistos naturalmente [35]. GraphSAGE no necesita de todos sus vecinos durante el entrenamiento para crear una representacion de el mismo, sino que a traves de un subconjunto de estos aprendera 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 decri no necesesita conocer todo el grafo ni todos lso nodos para crear estars representaciones. Este enfoque es util principalmente a la hora de trabajar con grafos dinámicos, batching/sampling, etc. Representando asi a nodos een un vector de baja dimensionalidad y generalizondo para luego nodos no vistos.

El proceso de creación de embeddings para los nodos del garfo estan dados por las siguientes ecuaciones:

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

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

$$h_i^{(l+1)} = \text{norm}\Big(h_i^{(l+1)}\Big) \tag{5.8}$$

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 traves de una funcion de agregación combian estos nodos vecinos (por ejemplo promedio, suma, lstm, etc). Luego tenemos $h_i^{(l+1)}$ correspondiente a la concatenación de la representación anterios del nodos i y la de las caracteristicas de nodos vecinos de la capa $l+1$, correspondiente a lo previamente calculado. Finalmente tenemos $\text{norm}\Big(h_i^{(l+1)}\Big)$ la cal se encarga de normalizar las caracteristicas del nodo i en la capa $l+1$.

A continuación tenemos Figura 5.10, el cual ilustra el proceso de creacion de las representaciones de los nodos. Dado primero 1) por la selección de un numero fijo de vecinos de un nodo, 2) Luego la agregación y concatenacińf de las caracteristicas de estos nodos al nodo dst junto con normalizacion, 3) Fianlemente el paso de prediccion y ajuste de valores de los pesos de la red.

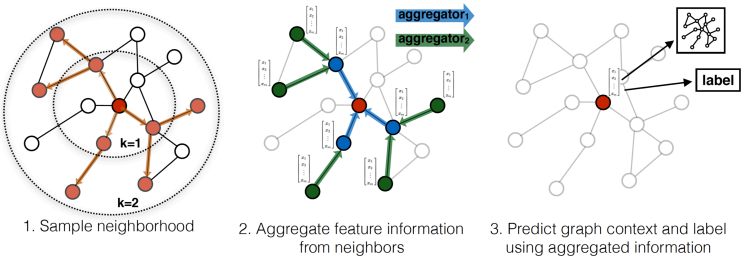

Figura 5.10: TODO.

Capítulo 6

Internat Data collection

6.1 CAIDA

El Centro de Análisis de Datos de Internet Aplicados o CAIDA por sus siglas en inglés [36] es una organización de investigación cuyo principal objetivo es la recopilación y análisis de datos relacionados a la infrestructura, rendimiento y trafico del Internet. Este funciona como un canal de difusión donde se publican y comparten tanto herramientas como investigaciones relacionado a las redes de Internet.

En el contexto de nuestro problema de **inferencia de Relaciones entre Sistemas Autonomos**, CAIDA ofrece el dataset **CAIDA AS Relationships** [22]. Este clasifica las relaciones entre Sistemas Autonomos en dos principales tipos: Customer-to-provider (C2P) y Peer-to-peer (P2P). Estas relaciones son inferidas a partir de datos públicos de ruteoBGP ofrecidos por RouteViews[23] y RIPE RIS[37]. El dataset **CAIDA AS Relationships** está compuesto por dos subdatasets:

- **serial-1:** Contiene las relaciones inferidas utilizando un método similar al descrito en el estudio de M. Lukie et al.[7] .Esta disponible dasde 1998 hasta la actualidad, con un archivo mensual. Cada archivo incluye un grafo completo derivado de instantáneas de las tablas BGP de RouteViews y RIPE RIS, tomadas cada 2 horas durante un período de 5 días.
- **serial-2:** Basado en el dataset Serial-1, este agrega enlaces inferidos mediante BGP communities, utilizando el método descrito por Vasileios et al. [38] . Está disponible desde octubre de 2015 hasta el presente, con archivos generados mensualmente.

Otro proyecto relevante de CAIDA es Archipelago (Ark) Measurement Infrastructure, una plataforma de medición distribuida a nivel global. Esta cuenta con nodos de medición ubicados en diversas zonas geográficas y topológicas para proporcionar una visión integral del estado y funcionamiento de Internet. Entre las mediciones realizadas a través de Ark se encuentran proyectos como The Spoofer Project ,Scamper, Internet Topology Discovery, entre otros.

Además, CAIDA ofrece BGPStream [39], una librería de código abierto diseñada para manejar grande svolumenes de datos BGP. Esta herramienta permite acceder tanto a datos BGP en tiempo real como a históricos, obtenidos de los colectores de RouteViews y RIPE NCC. Esto facilita la investigación y el análisis de eventos relacionados con el enrutamiento, como interrupciones, fugas de rutas o ataques BGP, de manera eficiente y rápida.

6.2 RouteViews Project

El proyecto RouteViews [23] de la Universidad de Oregon nació en un principio como una herramienta los operadores de Internet, con el fin de que estos pudiesen obtener información BGP en tiempo real sobre el sistema de enrutamiento global desde las perspectivas de varios backbones y ubicaciones en Internet.

Sin embargo hoy en dia, su uso se ha extendido a multiples tareas, tales como la visualización de rutas AS, el estudio de la utilizacion del espacio de direcciones IPv4 y otros aspectos relacionados a la infrestructura y enrutamiento de Internet.

El proyceto cuenta con alrededor de 41 recolectores de enrutamiento (ver en ANEXO) ubicados en puntos clave al rededor del mundo, como Ámsterdam, Londres, Sídney, Singapur, São Paulo, Santiago, Nairobi, Sudáfrica, Chicago, California y otras ubicaciones, especialmente en Internet Exchange Points (IXP). Estos recolectores obtienen información de enrutamiento de más de 260 organizaciones difernetes que compaten sus datos con RouteViews, incluyendo muchos de los ISP más grandes del mundo.

Los recolectores establecen sesiones de peering BGP con diferentes sistemas autónomos, a través de estas conexiones, los recolectores reciben actualizaciones de las tablas BGP (RIS) y UPDATES del protocolo BGP, obteniendo asi información sobre las rutas globales y las conexiones entre AS. En estas actualizaciones incluyen infomración como:

- **AS Path:** Camino de Sistemas Autonomos que un paquete debe seguir para llegar a una red destino.
- **Prefijos IP:** Rangos de direcciones IP que están siendo anunciados por los AS.

- **Next Hop:** Siguiente salto (router o red) que los paquetes deben tomar para continuar hacia su destino.

El proyecto también cuenta con una compilación de archivos en dos formatos, los obtenidos por Cisco y por el software de enrutamiento FRR. El primero recopila información en intervalos de 2 horas, comenzando a las 00:00 UTC. El segundo obtiene RIBs y actualizaciones, los RIBs corresponden a snapshots recogidas cada 2 horas, mientras que los Updates son archivos continuos que se actualizan cada 15 minutos. Estos archivos históricos están en formato MRT y están disponibles para cada recolector.

6.3 RIPE NCC RIS

RIPE (Reseaux IP Européens) Network Coordination Centre es uno de los cinco Registros Regionales de Internet (RIRs) que proporciona servicios de registro de Internet y coordinación de recursos de Internet para Europa, Oriente Medio y partes de Asia Central. RIPE NCC opera el Routing Information Service (RIS) [37], una plataforma de recopilación de datos de enrutamiento, que tiene como ojetivo mejorar la comprensión y visibilidad del sistema global de enrutamiento de Internet.

Al igual que RouteViews, RIS recopila datos a traves de un conjunto de más de 26 colectores (ANEXO) remos de rutas (RRCs) distfribuidos globalmente, generalmente ubicados en Puntos de Intercambio de Internet (IXPs). Voluntarios se conectan con los RRCs mediante BGP, y RIS almacena los mensajes que recibe de estos peering.

Para acceder a los datos, existen dos formas: directamente a través de archivos en formato MRT, RIS Live y RISwhois, o mediante herramientas que utilizan los datos de RIS, como RIPEstat, donde se puede encontrar información ya organizada, como la exploración del enrutamiento por país y la localización de información de contacto sobre abusos.

6.4 The peering DB

PeeringDB [40] es una base de datos de acceso libre, una iniciativa sin fines de lucro, gestionada y promovida por voluntarios, donde los usuarios comparten información sobre redes. Esta información facilita la interconexión global de redes en Puntos de Intercambio de Internet (IXPs), centros de datos, y además proporciona datos valiosos para investigaciones.

Fue creada con el fin de facilitar he resource was created to help support peering between networks and peering coordinators, and today, it includes a wide range of interconnection data from networks,

6.5 Otroas Gerramientas

6.5.1 Hurricane Electric Internet Services

6.5.2 Servidores Looking glass

6.5.3 BGGPView

Capítulo 7

Experimentos

7.1 Benchmark

Para abordar este problema se comenzó por la creación de un Benchmark. …..

Para esto se probaron los siguientes metodos anteriores:

1. Gao [sacado de BGP2VEC] Paper - [5]
2. Ruan [sacado de BGP2VEC]: ntroduced by Ruan

and Susan Varghese [13]

7.2 Datos

- Se uso x durante lso X displaying
- ¿Cuanto se demoró?

7.3 modelos

Para bordar este problema se utiulizaron ……

7.4 Dataset de Evaluación

Las Relaciones entre SA son privadas, esto hace que en algoritmos de inferencia sea dificir validar, y para casos de machine learning, entrenar el modelos.

Lo primero que podriamos hacer es buscar un SA que publique sus AS relaciones. esto fue loq ue hoz la personsa de la tesis y creo un dataset a partir de la informacion que provee Hurricane's BGP Toolkit.

Capítulo 8

Experimentos

Para abordar este problema se tomarón dos enfoque diferentes.

- Clasificación Binaria: Se tomaron p2c y c2p como una misma clase y las otras siendo p2p.
- Clasificación Multiclase: Las relaciones p2c y c2p se tomaron como clases diferentes.

8.1 Dataset de Validación

Uno de los podibles caminos a ocupar fue ocupar como dataset de validación de la Tesis «BGP Data Analysis: Exploring Solutions for Au- tonomous Systems Relationships Inference»

Este utiliza el atributo BGP communities más la información RIB, esto con la idea de que hay algunos AS que publican sus políticas de BGP communities. Entonces se creó un JSON que contiene reglas de ciertos SA y reglas (BGP communies policy) con las que podría saber el tipo de relación que se tendría con un vecino.

Por ejempo TODO: Agregar un ejemplo.

8.2 Experimento 1:

- GNN -> GCN

- Predictor -> DOtProduct y MLP
- Optimizador -> Adam
- Función de pérdida -> CrossEntropyLoss
- Split de edges para entrenamiento y validacion
- Stochastic Gradient Descent (SGD) con un learning rate de XXX.

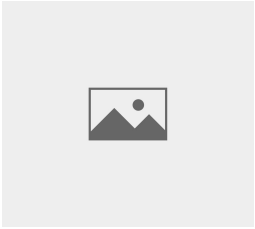

Figura 8.1: Resultados.

- Problemas con el modelo:
  - Posible overfitting.

8.3 Experimento 2:

- GNN -> GraphSAGE
- Predictor -> MLP
- Optimizador -> Adam
- Función de pérdida -> CrossEntropyLoss
- Split de edges para entrenamiento y validacion
- Neighbou sampling.

Figura 8.2: Resultados.

- ¿Por que ocupamos GraphSAGE y no GCN?

Como estamos entreando de forma transductive, es decir ocupando un mismo grafo para entrenar y validar, puede ocurrir que se este overfiteando el grafo y por eso obteniendo buenos resultados, estod ebdo a que al probar con epoch muuuy grandes los valores de loss todo el rato eran muy similares, y lo que se espera obtener en este caso es q llegase a un punto dond ela loss era similar , pero luego empezaran adiverger, que seria el punto en donde el modelo se esta empezando a aprender los datos de memoria, pero esto no

paso???. Para evitar esto se decidio ocupar otras forma sd eentrenamiento para majear que el modelo pudeese generalizar y no se estuviese validando mal (porq al final mas q na es problema de q noe stamos validando con datos diferentes). Entrena runasdo ampling.

8.4 Experimento 3:

- GNN -> GraphSAGE
- Predictor -> MLP
- Optimizador -> Adam
- Función de pérdida -> CrossEntropyLoss
- Split de edges para entrenamiento y validacion
- Otro sampling

Figura 8.3: Resultados.

8.5 Experimento 4:

Agregar paquetes de flujo

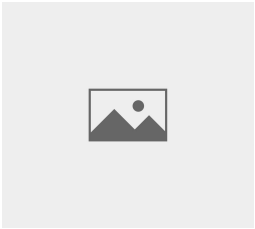

Figura 8.4: Resultados.

Destacar que en nuestro enfoqeu fue de Regresion, es decir la salida final de los modelos corresponde a un valor uncaente al cual luego se calcula un umbral para clasificar en las clases correspondientes. Esto porque l tener clases desbalanceadas, el modelo desidiria en solo clasificar un solo tipo (la q es mayor) y aun asi no se tienen resultados tan malos. Por esto no se tomo ese enfoque ya que es una de als fallas a la que queda propenso.

GCN VS GraphSAGE

- Comparamos el caso de GCN con GraphSAGE usando como Predictor el DotProduct y ocupando todas las features:

GCN:
- Con 100 epoch, un % de train de 0.6 se obtiene una accuracy de 0.81, sin embargo se puede observar overfitting mas o menos desde el epoch 70.
- Ocupando un modelo que agrega entre cada capa GNN dropout se octiene una accuracy 0.8. No es muy diferentes.
- Drop out ayuda harto, sin embargo hace q no sea tan smoth el converger.

GraphSAGE:
- Con epoch 100, un % de experience

Podemos ver que GraphSAGE super en performance a GCN. Tambien podemos ver que para ambos modelos el overfitting se presenta en diferentes casos, esto por la naturaleza de los calculos de cada uno en la aggregación. Pues en GCN para obtener un mensaje al nodo , se necesitan todos sus vecinos haciendo que se afacil reconocer un nodo de otro ya si mas fcil de que se reconosca overfitting, en vambio en graphSAGE el mensaje al nodo en como un promedio de los vecinos, por lo que es mas dificil que se reconosca overfitting.

Se realizo la mism acomparación pero en vez de DotProduct se ocupo MLPPredictor. En el caso de GCN La accuracy subió a un 0.8222(0.9221 cmabiando porte capas) y GraphSAGE a un 0.92 (con el mismo numero entre capas) 0.9546 (cambiando nuemero de capas). Sin embargo con 100 epoch no cambiaba mucho al final. Quise ver que ocurria si ocupaba un porcentaje muy bajo de train, para evr si habia overfitting, pero lo que psao fue q igual habia buenos resultados…??? ocupe el modelo sin droppout y ahi se podia ver un poco el overfitting. con una cantidad de X ejmplos de edge para train.

[Cachar porque no me esta encajando el numero de true y falses en el train msakk al ahcer split del datset]

como sea Es mejro con MLP que con DotProduct, por lo que se ocupara MLP en los siguientes experimentos.

OCUPAR TODAS LAS FEATURES ES NECESARIO?

Luego para ver que tan esenciales en la tarea era los features recolectados se porbo con graphSAGE y MLP como predictor (con ahsta el momento los mejores resultados que s e ahan obtenido), con 100 epoch y un % de train de 0,6 .

Se partio ocupando unicamente como atributos de los ndoos su grado in y grado out por nodo. Obteniendo una Accuracy de 0.8724 con DotProduct y 0.9290 con MLPpredcitor. Con esto vemos que si bien se dan buenos rsutltodos agregar las feeatures Mejora la accuracy. ¿Pero son de ayuda las que le estamos pasando? Prque si bien ya tenemos que son una ayuda luego de hacer una exploracion de esta (En anexo mas info) podemos notar que para muchos features par ala mayria de los nodos no se tiene información y por ende no serian muy relevantes?.

PAra esto se decidio incluir unicamete aquellas features cuya información para todos los nodos estuviera sobre 80%. Es decir existe info de la feature para el 80 % de los nodos. Estoso consistieron en :

- AS_rank_numberAsns
- AS_rank_customer
- AS_rank_peer
- peeringDB_ix_count
- peeringDB_fac_count
- cti_top

Con estos usando GraphSAGE y MLP como predictor se obtuvo una accuracy de 0.9452 lo que s eve que tener todas las otras valores lo mejora pero no tanto, es decir no son tan relevantes para dicha tarea.

otros caso fue elegir unicamente .....[COMPLETAR]...

IDEAS FALLIDAS

- Dentrro de otras ideas que se intentaron pero no funcionaron fue crear grafos a partir de los recolectores RRC, de sta forma tener diferentes grafos a partri d elos cuales algunos se elijan para training y otro diferente para testeo. Sin embargo falle, no se si retomar o no.
- otra idea fue que RIPEstat tineen una API de la cual se puede obtener información, sin embrago se demora demasiado para la cantidad de nodos ques e tienen por grafos.

Continuando con los experimentos una de los problemas que mas miedo tenia era que se entrenara y se esttuviera overfitteoando el grafo, porque no tenemso mas grafo que el de esa fechaa, pues esos datos (los atributos corresponden a sacados por otro paper) . Es decir estabamos tomando un enfoque inductivo. Es por esto que una podible solución que se nos ocurrio fue proabr diferentes tecnicas de samplingm¿, area que aun sigue en investigación e inovacion en el area de GNN.

Para esto se partio con random neighbour sampling[explicado en XXX marco teorico] y se obtuvo una accuracy de 0.9414 , lo que no mejoro mucho los resultados obtenidos anteriormente. ya desde el segundo epoch hay overfitiing.

se decidió probar con otro tipo dxxxe meodo el cual corrspondiente a ClusterGCN [explciado marco teorico seccion XXX] obteniendo una accuracy 0.9696 !!!!!!!!!

Luego para comparar GNN con otros metods para resolver dicha tarea, se probo resolver la atrae de clasificación con PAgeRank, DeepWalk y BGP2Vec. Obteniendo Resultados XX, 0.9270 (entrenando) y XX respectivamente.

La idea es ver que tan bien se comporta el modelo en comparación con otros metodos que se han ocupado para resolver la misma tarea.

Caso	Accuracy
GCN + DotProductPredictor	0.81
GraphSAGE + DotProductPredictor	0.9
GCN + MLPPredictor	0.9221
GraphSAGE + MLPPredictor	0.9546
GraphSAGE + MLPPredictor + in_degree y out_degree	0.9290
GraphSAGE + MLPPredictor + features sobre 80%	0.9452
GraphSAGE + MLPPredictor + Random neighbour sampling	0.9414
GraphSAGE + MLPPredictor + ClusterGCN	0.9696

Caso	Accuracy
PageRank	0.8746
DeepWalk	0.9270
BGP2Vec	X
Crear de 0	0.9068

Con GCN el overfitting es más facil que con GraphSAGE, es de esperarxe por la formula de cada una. GraphSAGE para que se vea overfitting % de train debe ser más bajo a diferencia de GCN que puede ser alto .

Capítulo 9

Resultados y Analisis

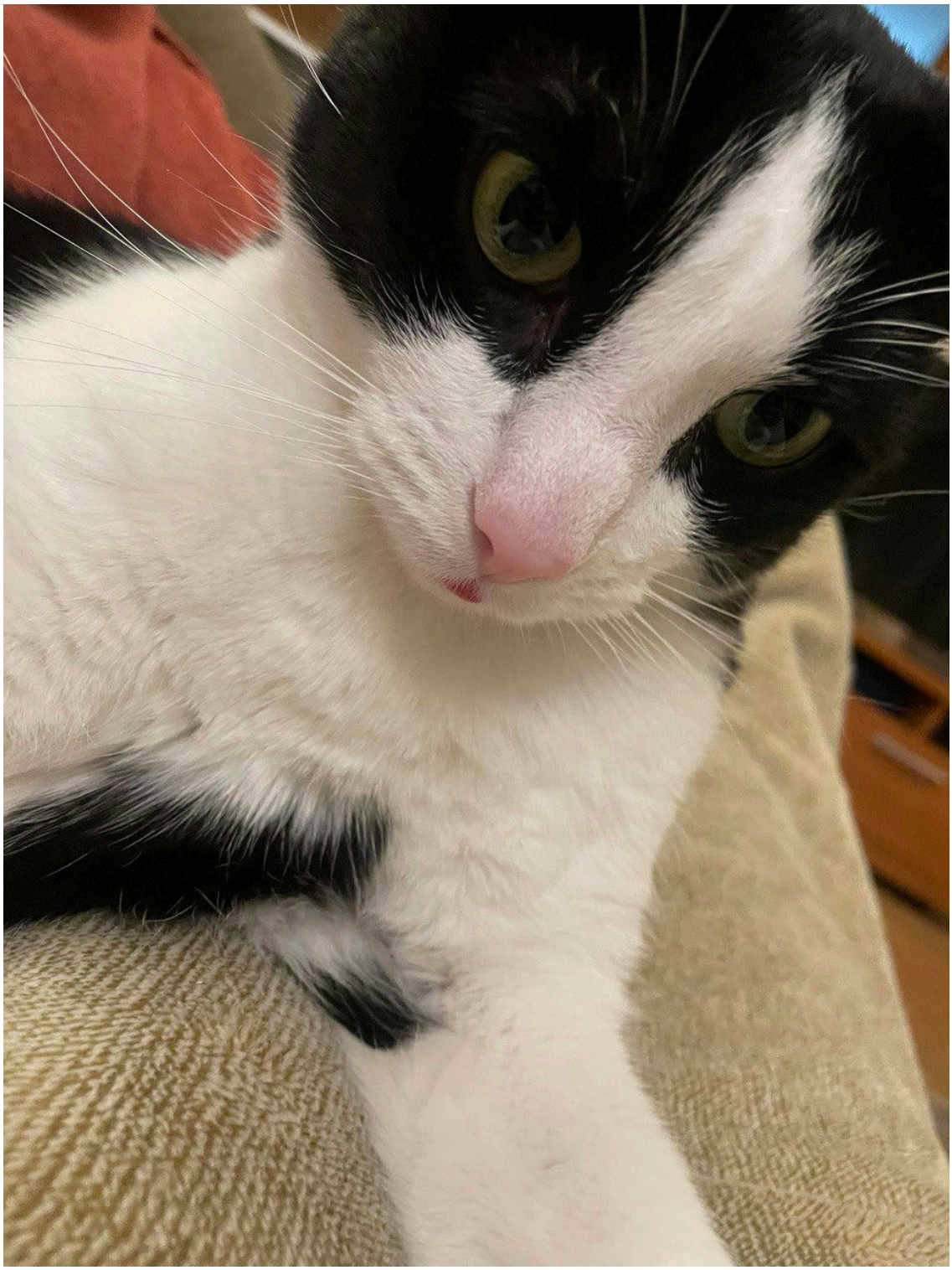

Figura 9.1: El gatito más bello del mundo.

Bibliografía

[1] Y. S. Tal Shapira, «Unveiling the Type of Relationship Between Autonomous Systems Using Deep Learning», *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, 2020.

[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] «Internet Engineering Task Force (IETF)». [En línea]. Disponible en: https://www.ietf.org/

[5] L. Gao, «On inferring autonomous system relationships in the internet», *IEEE/ACM Transactions on Networking*, 2001.

[6] G. Huston, «Interconnection, Peering and Settlements», 2003, [En línea]. Disponible en: http://websrv.cs.fsu.edu/~xyuan/cis6930/huston99.pdf

[7] k. c. A. D. M. Luckie B. Huffaker y V. Giotsas, «AS Relationships, Customer Cones, and Validation», *in Internet Measurement Conference (IMC)*, 2013.

[8] J. Moy, « OSPF Version 2», *RFC 2328, https://www.rfc-editor.org/rfc/rfc2328.html*, 1998.

[9] « RIP Version 2», *RFC 2328, https://datatracker.ietf.org/doc/html/rfc2453*, 1998.

[10] «A Border Gateway Protocol 4 (BGP-4).», *RFC 4271*, 2006.

[11] «IS-IS for IP Internets (isis), http://www.ietf.org/html.charters/isis-charter.html».

[12] « TRANSMISSION CONTROL PROTOCOL». 1981.

[13] L. Ruan y J. Susan Varghese, «Computing observed autonomous system relationships in the internet». 2014.

[14] B. Woodcock y M. Frigino, «2016 Survey of Internet Carrier Interconnection Agreements».

[15] A. D. Yuchen Jin Colin Scott y V. Giotsas, «Stable and Practical AS Relationship Inference with ProbLink». 2019.

[16] J. R. R. H. K. Lakshminarayanan Subramanian Sharad Agarwal, «Characterizing the Internet hierarchy from multiple vantage points», *IEEE INFOCOM*, 2002.

[17] M. F. B. H. Y. H. k. c. X. Dimitropoulos D. Krioukov y G. Riley, «AS Relationships: Inference and Validation», *ACM SIGCOMM Computer Communication Review (CCR), v.37, n.1, pp.29-40*, 2007.

[18] U. Weinsberg Y. Shavitt y E. Shir, «Near-deterministic inference of AS relationships», *ConTel 2009*, 2009.

[19] G. D. Battista M. Patrignani y M. Pizzonia, «Computing the types of the relationships between autonomous systems,». 2003.

[20] A. D. V. G. e. a. M. Luckie B. Huffaker, «AS relationships, customer cones, and validation.». 2013.

[21] T. S. y Yuval Shavitt, «Unveiling the Type of Relationship Between Autonomous Systems Using Deep Learning». 2020.

[22] «CAIDA. 2022. AS-relationships dataset». [En línea]. Disponible en: https://publicdata.caida.org/

[23] «University of Oregon RouteViews Project, https://www.routeviews.org/routeviews/ ».

[24] «BGProtect, www.BGProtect.com».

[25] U. Weinsberg Y. Shavitt y E. Shir, «Near-deterministic inference of AS relationships». 2009.

[26] S. H.,. Z.,. Y. Z. L. L. W. C. L. M. S. Jie Zhou Ganqu Cui, «Graph neural networks: A review of methods and applications», 2018.

[27] W. I. A. G. S. M. Nwankpa C., «Activation Functions: Comparison of trends in Practice and Research for Deep Learning. », 2018.

[28] G. E. H. R. J. W. Rumelhart D. E., «Learning representations by back-propagating errors», 1986.

[29] A. C. I. Goodfellow Y. Bengio, *Deep Learning*. MIT Press, 2016.

[30] J. A.-I. R. G.-B. T. H. A. A.-G. R. P. A. David Duvenaud Dougal Maclaurin, «Convolutional Networks on Graphs for Learning Molecular Fingerprints», 2015.

[31] Y. L. A. S. P. V. Michael M. Bronstein Joan Bruna, «Geometric deep learning: going beyond Euclidean data», *IEEE Signal Processing Magazine*, 2017.

[32] P. F. R. O. V. G. E. D. Justin Gilmer Samuel S. Schoenholz, «Neural Message Passing for Quantum Chemistry», 2017.

[33] M. W. Thomas N. Kipf, «SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS», 2017.

[34] A. C. A. R. P. L. Y. B. Petar Velickovic Guillem Cucurull, «GRAPH ATTENTION NETWORKS», *International Conference on Learning Representations*, 2018.

[35] R. Y. William L. Hamilton y J. Leskovec, «Inductive Representation Learning on Large Graphs», 2018.

[36] «CAIDA». [En línea]. Disponible en: https://www.caida.org/

[37] «RIPE NCC Routing Information Service (RIS), https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/».

[38] M. L. Vasileios Giotsas Shi Zhou y kc claffy, «Inferring Multilateral Peering».

[39] «BGPStream ,https://bgpstream.caida.org/».

[40] «PeerinDB. 2022. The Interconnection database». [En línea]. Disponible en: https://www.peeringdb.com/

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 Paolo, 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

Host	Ubicación
route-views.bdix.routeviews.org	BDIX, Dhaka, Bangledesh
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, Virgina
route-views.flix.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.napafrica.routeviews.org	NAPAfrica, 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

Host	Ubicación
route-views.phoix.routeviews.org	University of the Philippines, Diliman, Quezon City, Philippines
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 analisar 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:

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 asignacion y administracion de direcciones IP y numeros de Sistemas Autonomos (ASNs) en determinadas regiones geograficas. ARIN (American Registry for Internet Numbers): Norteamerica, el Caribe y Africa Subsahariana; RIPE NCC (Réseaux IP Européens Network Coordination Centre): Europa, Oriente Medio y Asia Central; APNIC (Asia-Pacific Network Information Centre): Asia y el Pacifico; LACNIC (Latin American and Caribbean Network Information Centre): America Latina y el Caribe; AfriNIC (African Network Information Centre): Africa.
- **AS:**: Autonomous System
- **BGP:**: Border Gateway Protocol
- **RRC:**: Route Collectors
- **IXP:**: Internet Exchange Point
- **Internet Carrier**: Empresas que operan infrestructras de Red de gran escala. Ofrecen servicios de intercambio de datos.

Forman parte del nucleo 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 encuentras en la parte msa oeriferica de la red de Internt
- **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 Autonomos que estan interconectados, es decir,cada AS dentro del clique tiene una conexion directa con todos los demas AS dentro del clique. Este tipo de estructura suele encontrarse en la capa más alta de la jerarquía de tinterent, conocida como Tier-1 ASes.
- **IANA:**