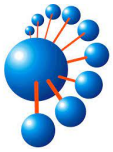
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA
Y CIENCIAS DE LA COMPUTACIÓN
FACULTAD DE INGENIERÍA
UNIVERSIDAD DE CONCEPCIÓN



Detección de anomalías en el tráfico de red entre direcciones IP en intervalos de tiempo utilizando PGSS-BDH Sketch

POR
Luis Andrés Valenzuela Concha

Trabajo semestral de Tópicos en Manejo de Grandes Volúmenes de Datos

Profesora: Cecilia Hernández

Concepción, Diciembre 2023

Índice

1. Introducción

En la era digital, donde la conectividad y el intercambio de información son elementos fundamentales, la seguridad de las redes informáticas se vuelve esencial para preservar la integridad y confidencialidad de los datos. En este contexto, la detección de anomalías en el tráfico de red entre dos direcciones IP específicas emerge como una tarea crítica para identificar posibles amenazas cibernéticas. En respuesta a este desafío, se ha desarrollado un algoritmo innovador que utiliza la implementación de PGSS-BDH Sketch.

La base teórica del algoritmo se cimenta en la metodología propuesta en [1], un enfoque que ha demostrado su eficacia en la síntesis y análisis de flujos de datos complejos. Al integrar esta implementación en un algoritmo, se ha logrado no solo aprovechar las fortalezas de PSGG-BDH, sino también adaptarlas de manera precisa a la detección de anomalías en el contexto específico de las comunicaciones entre dos direcciones IP determinadas, en intervalos de tiempos.

Este informe detallará el enfoque empleado para abordar la detección de anomalías en el tráfico de red entre dos direcciones IP específicas. Se describirá el problema que se busca resolver, soluciones existentes y la metodología, en detalle, detrás de la solución propuesta. Además, se presentarán los resultados de la evaluación experimental y se realizará un análisis teórico de la solución. Finalmente, se presentarán las conclusiones del tema abordado.

2. Descripción del problema

El problema que se aborda en este contexto es la detección de anomalías en el tráfico de red entre direcciones IP específicas, centrándose en intervalos de tiempo y considerando integralmente todo el historial de comunicaciones entre las direcciones IP en cuestión. En este contexto, las anomalías se definen como patrones de comportamiento inusuales o atípicos que pueden indicar posibles amenazas de seguridad, fallos en el sistema o actividades no autorizadas. Estas anomalías pueden manifestarse de diversas maneras, presentando tanto incrementos como decrementos significativos en comparación con el tráfico normal. La capacidad de identificar y comprender estos cambios bruscos en el tráfico contribuye directamente a fortalecer la seguridad y la integridad de la red. Se presenta la Figura 1, que proporciona una representación visual de las anomalías en el tráfico de red entre direcciones IP específicas. En la figura, se destacan los incrementos y descensos significativos en el tráfico, ofreciendo una visualización efectiva de los patrones anómalos considerados en este análisis.

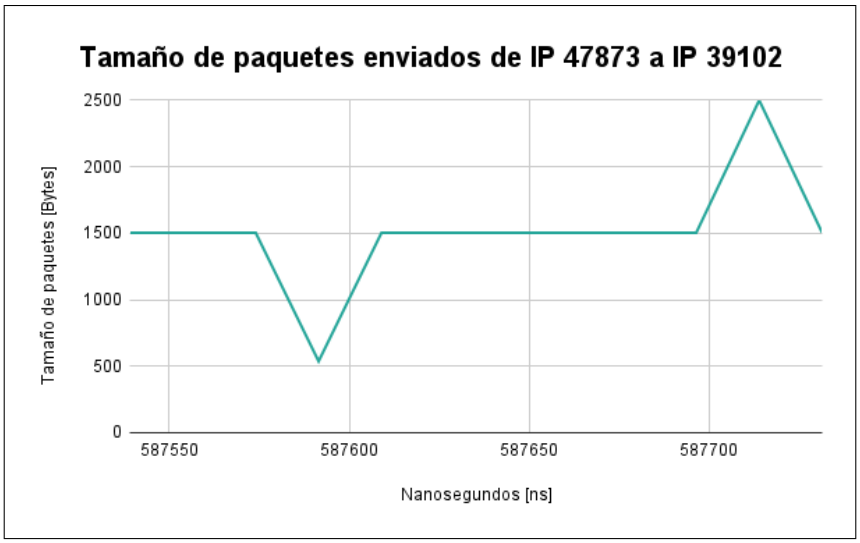


Figura 1: Representación visual de anomalías en el tráfico de red entre dos direcciones IP.

2.1. Soluciones existentes

La revisión de las soluciones existentes revela una limitación significativa en la detección de anomalías en el tráfico de red entre direcciones IP. En la mayoría de los casos, las soluciones previas no abordan de manera integral todo el historial de comunicaciones, limitándose a enfoques que no capturan la totalidad de la información relevante. Esta carencia se traduce en la incapacidad para realizar detecciones retrospectivas y análisis exhaustivos en intervalos de tiempo específicos.

3. Descripción de la solución propuesta

La solución propuesta para abordar el desafío de la detección de anomalías en el tráfico de red se fundamenta en la implementación y adaptación de PGSS-BDH Sketch. Este enfoque, originalmente diseñado para el resumen en tiempo real de flujos de datos, ha sido cuidadosamente adaptado para satisfacer, mediante un algoritmo, las necesidades específicas de la detección retrospectiva de anomalías en intervalos de tiempo definidos.

3.1. PGSS-BDH Sketch

La elección del PGSS-BDH Sketch se basó en su capacidad para gestionar y almacenar información a lo largo del tiempo. Esta adaptación estratégica permitió no solo retener todo el historial de comunicaciones entre direcciones IP, sino también realizar consultas retrospectivas en intervalos de tiempo específicos. Esta capacidad de almacenamiento completo del historial y análisis retrospectivo es esencial para la detección precisa de anomalías en el tráfico de red.

3.1.1. Estructura de PGSS-BDH Sketch

El PGSS-BDH Sketch se fundamenta en el funcionamiento del Graph-Sketch, una metodología que logra una marcada reducción del tamaño del grafo original mediante el empleo de funciones hash. En adición a la disminución del tamaño, se lleva a cabo una modificación crucial en la matriz de adyacencia, que originalmente almacena el peso de los arcos. En este contexto, la matriz de adyacencia es transformada en una estructura con hashmaps jerarquizados, los cuales se organizan de acuerdo a intervalos de tiempo específicos. En la Figura 2 se puede apreciar la disminución de tamaño del grafo original mediante funciones hash, y en la Figura 3 se observan la estructura de los hashmaps jerarquizados en cada celda de la martiz.

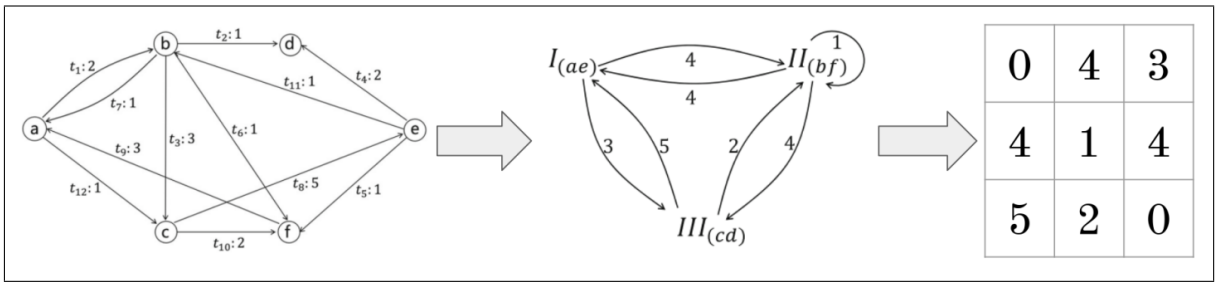


Figura 2: Representación visual de la disminución de tamaño del grafo original.

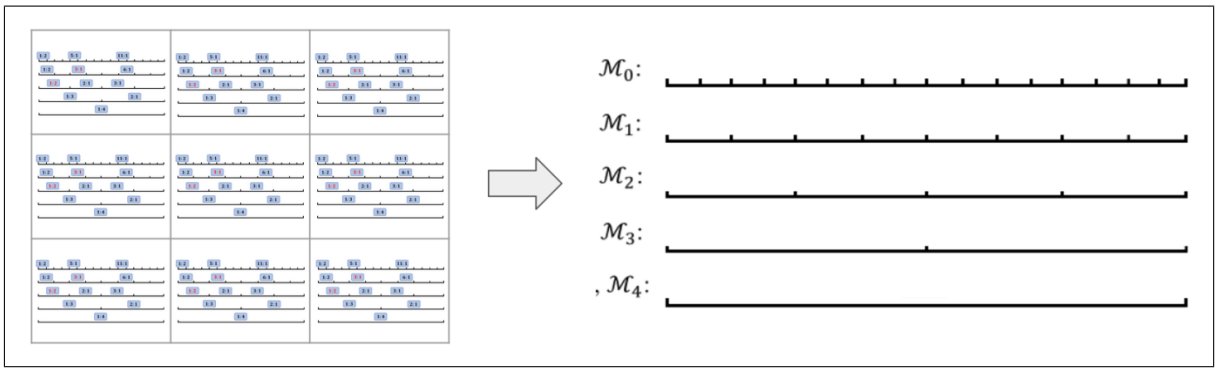


Figura 3: Representación visual hashmaps jerarquizados
dentro de cada celda de matriz de adyacencia.

3.1.2. Inserción de aristas en PGSS-BDH Sketch

Para la inserción de aristas en el sketch se utiliza cada función hash para determinar la posición de la arista en la matriz de adyacencia. Luego, se almacena el peso de la arista en cada uno de los hashmap correspondiente al intervalo de tiempo actual. Si se da el caso de que ya haya una inserción previa en el mismo intervalo de tiempo, se suma el peso de la arista actual al peso de la arista previa. De esta manera, se logra mantener la información de todo el historial de comunicaciones entre direcciones IP. El formato de inserción es de la forma: inserción $(IP_1, IP_2, peso, marca\_de\_tiempo)$. Se muestra un ejemplo de inserciones, donde se insertarán los valores asociados a la Figura 1. En la Figura 4, se lleva a cabo la inserción inicial de la arista $(47873, 39102, 1504, t_1)$. Posteriormente, tras la inserción de la arista $(47873, 39102, 1504, t_2)$ en la Figura 5, se continúa este proceso sucesivamente hasta completar la inserción de todas las aristas, cada una con su peso asociado y respectiva marca de tiempo. El resultado final se ilustra en la Figura 6, donde las anomalías se destacan en color rojo para una mejor apreciación.

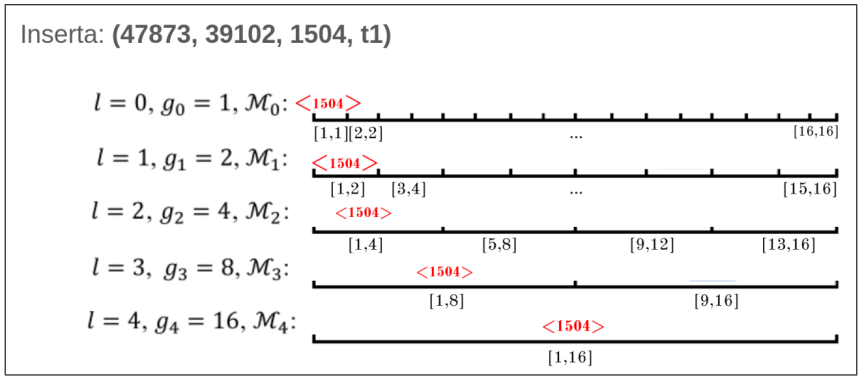


Figura 4: Inserción de la arista $(47873, 39102, 1504, t_1)$.

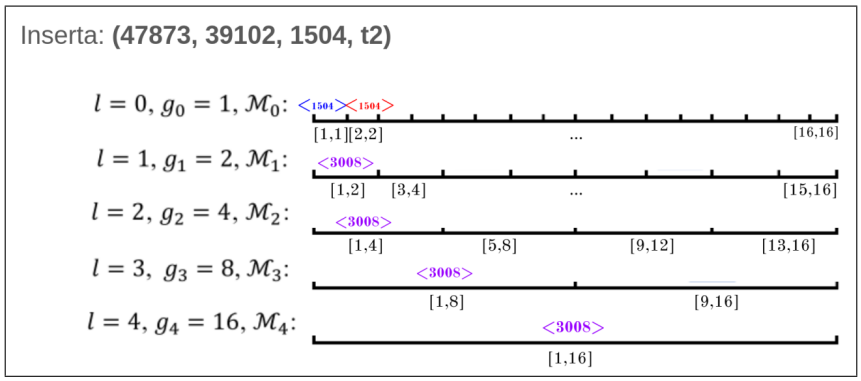


Figura 5: Inserción de la arista $(47873, 39102, 1504, t_2)$.

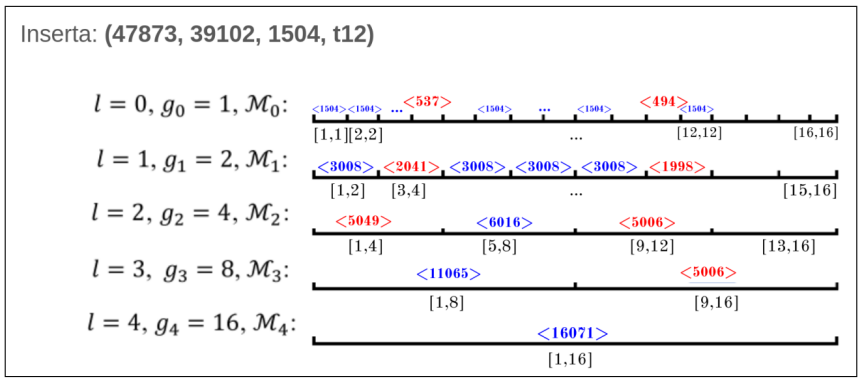


Figura 6: Resultado final de insertar aristas hasta t_{12}

3.1.3. Consulta de peso en PGSS-BDH Sketch

La operación de consulta de peso en un intervalo de tiempo dado recibe como argumentos los vértices v_{origen} y $u_{destino}$, junto con un intervalo de tiempo $[t_s, t_e]$, y devuelve el resultado con una complejidad temporal de $O(k \log_2(t_e - t_s))$, donde k representa la cantidad de funciones hash asociadas a la creación del sketch.

3.2. Algoritmo

Utilizando la implementación de PGSS-BDH Sketch, se ha desarrollado un algoritmo que resuelve el problema planteado anteriormente. El algoritmo recibe como entrada las direcciones IP origen y destino, el intervalo de tiempo en el que se desea realizar la detección y la precisión de la misma. La precisión se define como el número de desviaciones estándar que se consideran para determinar si un valor es anómalo. El algoritmo devuelve como salida un vector que contiene los intervalos de tiempo en los que se detectaron anomalías.

Algoritmo 1 muestra en detalle la consulta de encontrar anomalías. El algoritmo en cuestión, comienza creando subintervalos de longitud uniforme que cubran el intervalo de tiempo de la consulta. Luego, utilizando la estructura de PGSS-BDH Sketch, se obtiene el promedio del flujo entre las direcciones IP en el historial completo de comunicaciones. Luego, se calcula la desviación estándar del flujo entre las direcciones IP. Finalmente, se recorren los subintervalos creados previamente y se consulta el flujo entre las direcciones IP en cada uno de ellos. Si el flujo en un subintervalo es menor que el promedio menos la precisión por la desviación estándar o mayor que el promedio más la precisión por la desviación estándar, se considera que hay una anomalía en el subintervalo y se agrega al vector de salida.

3.3. Análisis teórico

Se demostrará la complejidad temporal de la consulta de encontrar anomalías. Para ello, se considerará que la cantidad de funciones hash utilizadas para crear el sketch es k y que el intervalo de tiempo de la consulta es $[T_s, T_e]$ de rango T y los subintervalos creados son de la forma $[t_s, t_e]$ de rango t. Debido a la estructura de PGSS-BDH Sketch, la complejidad temporal para encontrar el promedio del flujo entre las direcciones IP en el historial completo es $O(k)$. Luego, el cálculo de la desviación estándar y detección las anomalias del flujo tienen una complejidad temporal de $O(\frac{T}{t} k \log_2(t))$. Por lo tanto, la complejidad temporal de la consulta de encontrar anomalías es $O(k + 2\frac{T}{t} k \log_2(t))$. En la práctica, los subintervalos $[t_s, t_e]$ que se generan al dividir el intervalo de consulta, contienen $t_s = t_e$. Esto implica que consultar el peso en el sketch sea de $O(k)$. En consecuencia, se infiere que la complejidad de tiempo del algoritmo se reduce a $O(k+kT)$, siendo linealmente proporcional al tamaño del intervalo de tiempo de la consulta.

Algorithm 1 find_anomalias(IP_s, IP_d, t_s, t_e, precision)

Input: IP_s: IP origen, IP_d: IP destino, t_s: tiempo inicial, t_e: tiempo final, *precision*: precisión de la detección

Output: A: vector que contiene los intervalos de tiempo de las anomalías detectadas en el flujo entre las direcciones IP

$A \leftarrow \emptyset$;
$r \leftarrow t_e - t_s + 1$;
$\mathcal{I} \leftarrow \emptyset$;
for $i \leftarrow 1$ **to** r **do**
    $\mathcal{I} \leftarrow \mathcal{I} \cup [t_s + i, t_s + i]$;
$x \leftarrow h_1(IP_s)$, $y \leftarrow h_1(IP_d)$;
$p \leftarrow \infty$
for $i \leftarrow 1$ **to** k **do**
    $M \leftarrow$ hashmaps en la posición $A[x][y]$ del sketch S_i;
    $M_l \leftarrow$ hashmap en la posicion $\lceil \log_2(T) \rceil$ de M;
    $p \leftarrow \min(p, M_l[1])$;
$d \leftarrow 0$;
for each $I \in \mathcal{I}$ **do**
    $w \leftarrow$ PGSS-BDH-query$(IP_s, IP_d, I.t_s, I.t_e)$;
    $d \leftarrow d + (w - p)^2$;
$d \leftarrow \sqrt{d/r}$;
for each $I \in \mathcal{I}$ **do**
    $w \leftarrow$ PGSS-BDH-query$(IP_s, IP_d, I.t_s, I.t_e)$;
    if $w < p - precision * d$ **or** $w > p + precision * d$ **then**
        $A \leftarrow A \cup [I.t_s, I.t_e]$;
return A;

Todos los códigos asociados a estas implementaciones se encuentran disponibles en el siguiente repositorio de GitHub: `https://github.com/Luis-Valenzuela-Concha/PGSS-Sketchs`.

4. Evaluación experimental

Se presentan los resultados de una serie de experimentos destinados a evaluar el rendimiento del algoritmo de detección de anomalías en el tráfico de red. El objetivo principal es identificar los contextos en los que el algoritmo muestra mayor efectividad. Para llevar a cabo los experimentos, se utilizaron datasets que contenían información que incluía, marca de tiempo, direcciones IP (IP1 e IP2), y el peso asociado a cada comunicación.

4.1. Experimento de Tiempo de Cómputo

Se realizó un experimento variando el tamaño del intervalo de consulta y manteniendo fijo el número de funciones hash y cantidad de vértices del sketch. La Figura 7 ilustra que el tiempo de cómputo es linealmente proporcional al tamaño del intervalo de consulta, corroborando las observaciones teóricas.

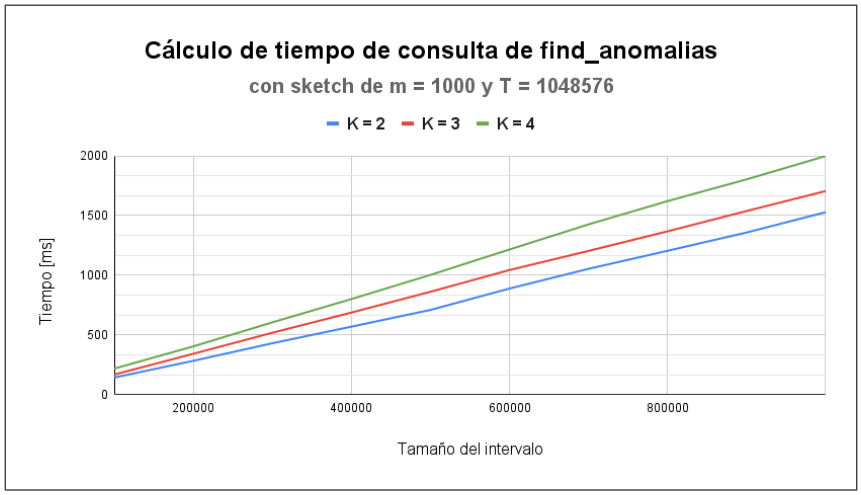


Figura 7: Cálculo de tiempo de consulta de find_anomalias.

4.2. Experimentos de Precisión, Recall y Espacio Utilizado

La realización de estos experimentos tiene como finalidad identificar el número óptimo de funciones hash en la creación del PGSS-BDH Sketch para maximizar la eficacia del algoritmo. Para ello, se realizaron experimentos variando el número de funciones hash y manteniendo fijo el tamaño del intervalo de consulta. En teoría, debido a la estructura del sketch, se espera que un incremento del número de funciones hash se aumentaría la precisión y recall del algoritmo, aunque también aumentaría el espacio utilizado por el sketch. No obstante, los resultados evidencian en la Figura 8 y la Figura 9 que tanto la precisión como el recall se mantienen estables independientemente de la cantidad de

funciones hash. Paralelamente, la Figura 10 revela que el espacio ocupado por el sketch aumenta de manera lineal con el incremento en el número de funciones hash.

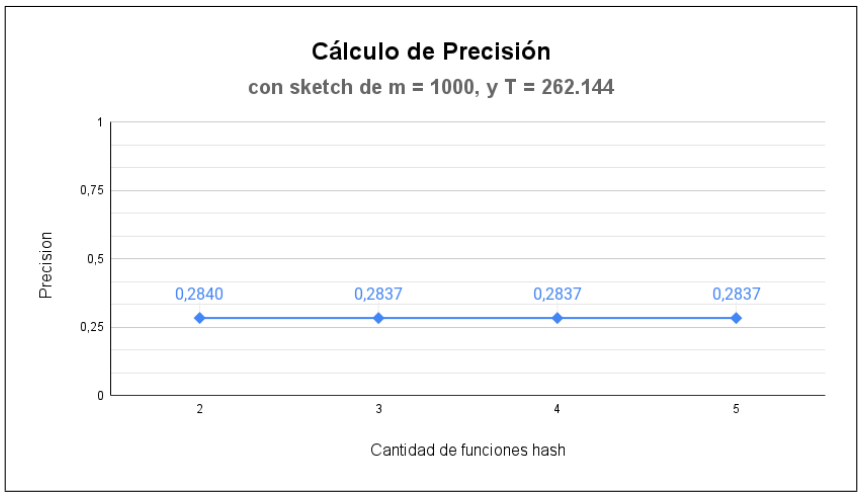


Figura 8: Cálculo de Precision.

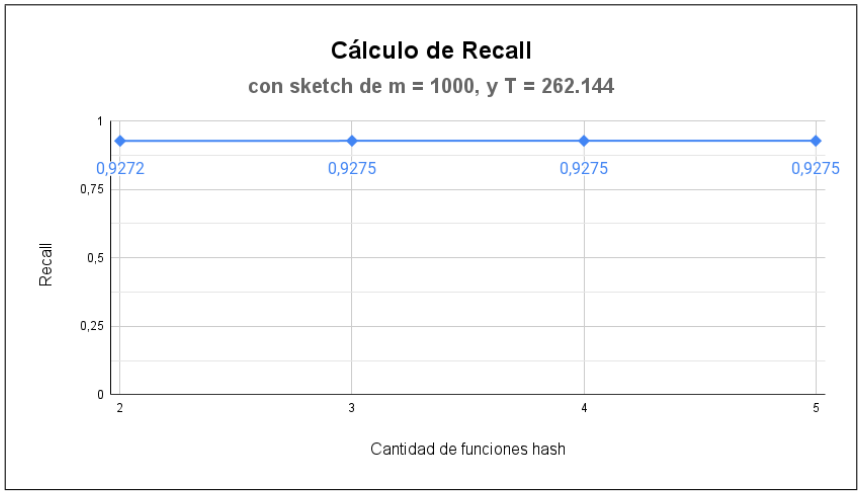


Figura 9: Cálculo de Recall.

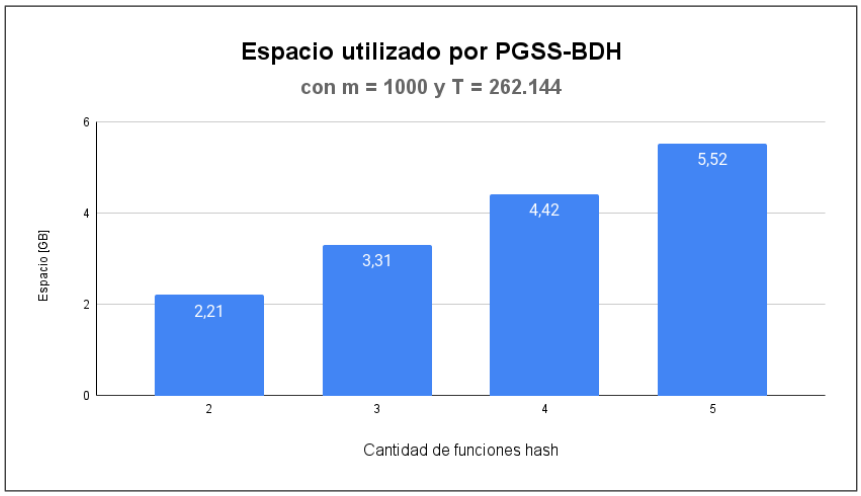


Figura 10: Espacio utilizado por PGSS-BDH.

En resumen, los resultados experimentales revelan que la implementación óptima del algoritmo se logra con dos funciones hash, ya que se obtiene una precisión y recall estables, sin aumentar significativamente el espacio utilizado por el sketch. Como resultado de este experimento, se recomienda utilizar dos funciones hash para equilibrar eficazmente la precisión del algoritmo con la eficiencia en términos de recursos computacionales y espacio de almacenamiento.

5. Conclusiones

En conclusión, se logra con éxito la implementación de un algoritmo de detección de anomalías en el tráfico de red basado en el PGSS-BDH Sketch. La metodología propuesta se ha aplicado de manera exitosa, permitiendo la creación de un algoritmo funcional y eficaz. A continuación, se presenta una evaluación exhaustiva de su desempeño a través de una serie de experimentos diseñados para analizar la variación de parámetros clave, como el número de funciones hash, y su impacto en la precisión y recall, y el espacio utilizado del sketch.

En estos experimentos, se identificaron configuraciones específicas del sketch que brindan un equilibrio óptimo entre precisión y eficiencia computacional al algoritmo. En particular, al usar dos funciones hash, el algoritmo muestra una precisión y recall estables sin aumento significativo en el espacio ocupado, en comparación con configuraciones de mayor cantidad de funciones hash. Estos hallazgos subrayan la utilidad y adaptabilidad del algoritmo propuesto en contextos prácticos, estableciendo una base sólida para futuras implementaciones y desarrollos en la detección de anomalías en el tráfico de red.

Referencias

[1] Yan Jia, Zhaoquan Gu, Zhihao Jiang, Cuiyun Gao, and Jianye Yang. Persistent graph stream summarization for real-time graph analytics. 2023.