Modelo predictivo ataques ddos

## Definición y caso de uso

Este documento explica el proceso del trabajo que hemos realizado para crear modelos predictivos de ataques DDoS. Hemos utilizado un [dataset](https://www.unb.ca/cic/datasets/ddos-2019.html) (CSVs/CSV-03-11.zip) de la universidad de New Brunswick que contiene 87 columnas y millones de filas de las cuales hemos limpiado y nos hemos quedado con, alrededor de 400 miles de filas.

Los datos contienen valores como IP’s, protocolos, puertos, tamaño de paquetes, tiempo entre paquetes, etc. Estos se obtuvieron de un PCAP a través de la aplicación Wireshark durante 2 días. Se simuló actividad benigna de 25 usuarios a través de protocolos HTTP, HTTPS, FTP, SSH y email. Contienen datos de 6 tipos de ataques diferentes además de datos de tránsito de red benignos.

Los ataques DDoS son una amenaza sobre la seguridad de las redes desde hace años y hasta día de hoy. Este se basa, a grandes rasgos, en colapsar servidores mediante el envío malicioso de un gran número de paquetes. Este es un problema que amenaza a la seguridad en las redes desde hace años y hasta el día de hoy. Hay muchas herramientas capaces de detectar ataques DDoS, sin embargo, consideramos que una capaz de hacerlo en tiempo real, y con capacidad de aprender de nuevos tipos de ataques son parece muy prometedor.

Un buen modelo predictivo se podría incorporar en un sistema de seguridad. Si este es capaz de detectar ataque DDoS en tiempo real sería de gran contribución a la seguridad de la red. Además, si es un modelo explicativo, y con capacidad de clasificación del tipo de ataque sufrido aportaría una capa de información extra sobre el caso.

Además, se podría entrenar con nuevos tipos de ataques que surjan para que se vaya con gran facilidad sin tener que definir las reglas explícitamente.

# Feature engineering

El [dataset](https://www.unb.ca/cic/datasets/ddos-2019.html) es un conjunto de 7 archivos, uno por cada tipo de ataque mezclados con tránsito de red benigno. Hemos limpiado y clasificado los datos para unificarlos y balancearlos y nos hemos quedado con un bloque de datos que contiene alrededor de 680 mil casos con cada tipo de ataque igual de representado que el resto.

Este caso es bastante particular, ya que hay gran cantidad de columnas que, al representar datos sobre el flujo, se mantienen en valores bajos en casi todos los casos, sin embargo, en cierto tipo de ataque se eleva enormemente ya que es la cualidad de ese tipo de ataque. Esto hace imposible tratar los outliers ya que nos haría quedarnos sin la información que nos indica que tipo de ataque es. Estas sería las columnas más importantes:

* Puerto fuente: Número de puerto en el dispositivo que envía los datos.
* Puerto destino: Número de puerto en el dispositivo que recibe los datos.
* IP’s fuente: Dirección IP del dispositivo que origina la comunicación.
* IP’s destino: Dirección IP del dispositivo que recibe la comunicación.
* Protocolo: Reglas que definen cómo se transmiten los datos (e.g., TCP, UDP).
* Etiqueta con el tipo de ataque: Descripción del tipo de ataque detectado (e.g., NETBIOS, LDAP).
* Información acerca de los paquetes: Detalles de los paquetes capturados

(e.g., tamaño, marcas de tiempo, contenido).

* Información acerca del flujo: Datos sobre la serie de paquetes en una comunicación

(e.g., cantidad de paquetes, duración).

## Capa RAW:

* Unificación de los 7 archivos diferentes en un solo parquet.
* Balanceo de la representación de cada clase. Eliminamos columnas de cada clase hasta que tengan todas la misma cantidad.
* Refactorizamos los encabezados de las columnas para poder trabajar mejor con ellos.

## Capa STAGING

* Eliminamos columnas que no nos aportan ningún valor.
* Eliminamos valores de filas que no tienen valor.
* En “Flow\_Packets/s" y "Flow\_Bytes/s" hay valores infinitos y nulos. Son valores que se rellenaron al colapsar la red debido a un ataque por lo que los reemplazamos con el valor máximo encontrado en la columna.

## Capa BUSSINESS

* Clasificamos las IP´s en base a si son públicas o privadas.
* Convertimos a índices las etiquetas de ataques y protocolos.
* Agrupamos los puertos por su tipo y los convertimos a índices.

Tras las transformaciones anteriores nos quedamos con un archivo parquet de un tamaño de 15 MB de 72 columnas y 400.000 filas. Con esto hemos realizado el entrenamiento de los modelos

## Pre-Entreno

* Aplicado el MinMaxScaler a todo el set de datos menos las columnas categóricas.
* Hemos aplicado una reducción de la dimensionalidad, pasando de 72 columnas a 7, las cuales contienen un 100 % de la información. Para ello usamos el algoritmo LDA.
* Hemos dividido el dataset de forma estratificada para que cada clase tenga la misma representación tanto en el set de entreno como en el de test.

# modelos

## **XGBoost**

|  |  |  |
| --- | --- | --- |
| Tiempo de entrenamiento: 1m 7.9seg  Precision media: 89%  Recall media: 88%  F1 media: 88%  Precisión de benignos: 99% | Optimización de hiperparametros a través de una búsqueda bayesiana:  booster='gbtree'  learning\_rate=0.23962854155043442  n\_estimators=211  max\_depth=8  min\_child\_weight=7  subsample=0.6142975198934528  colsample\_bytree=0.8348945145303123 |  |

## **KNN**

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

|  |  |
| --- | --- |
| Tiempo de entrenamiento: 1m 7.9seg  Precision media: 89%  Recall media: 88%  F1 media: 88%  Precisión de benignos: 99% |  |

## **ResNet**

|  |  |  |
| --- | --- | --- |
| Tiempo de entrenamiento: 8m 30seg (usando GPU)  Precision media: 86%  Recall media: 86%  F1 media: 85%  Precisión de benignos: 98% | Optimización de hiperparametros a través de una búsqueda bayesiana:  loss\_function = CrossEntropyLoss  optimizer = Adam  epochs=20  batch\_size=512  learning\_rate=0.0005109  weight\_decay=0.0007081 |  |