



# ANTEPROYECTO TFG

Detección de anomalías en base al tráfico



## Descripción breve

Análisis del tráfico a través de la agregación de flujos para determinar en tiempo real comportamientos anómalos.

Fernando Javier Villar Freire


fernando.villar.freire@udc.es


## Índice


Introducción .....	2
Objetivos .....	2
Acercamiento inicial a las tecnologías NETFLOW e IPXFIX .....	2
Selección e implementación de un algoritmo de ML .....	2
Diseño y despliegue de un sistema distribuido basado en Apache Spark .....	2
Despliegue y pruebas de la solución de análisis en un entorno en tiempo real .....	3
Metodología .....	3
Fases .....	3
1. Planificación .....	3
2. Recopilación y análisis de información .....	3
3. Preparación de los datos .....	3
4. Implementación del algoritmo de ML .....	3
5. Diseño y despliegue del sistema distribuido.....	3
6. Diseño y despliegue de sistema de captura de tráfico .....	4
7. Despliegue y pruebas del modelo final.....	4
8. Extracción de conclusiones y valoración final.....	4
Resultados previstos .....	4
Bibliografía .....	4

## Introducción


En un panorama actual con un aumento del tráfico (debido a la proliferación de IoT, al auge de la filosofía del siempre conectado y al aumento de servicios multimedia, cloud, videoconferencia y su tráfico asociado) junto con la proliferación de ataques y nuevas amenazas, surge la necesidad de modelos de defensa rápidos y adaptables.

La presencia en el mercado de sistemas IDS<sup>i</sup> muestra la preocupación de las organizaciones y el cambio de rumbo frente a la seguridad tomada en los últimos tiempos. 


El uso del enfoque de la agregación para la captura y análisis del tráfico viene a solucionar, en parte, los problemas de las ingentes cantidades de tráfico que soportan a día de hoy las organizaciones. Las tecnologías NETFLOW e IPFIX son dos soluciones similares a la hora de la creación de flujos, pero con funcionamientos que a priori no están pensados para su uso en tareas intensivas de tiempo real. Los análisis asociados a ambas suelen ceñirse a ventanas de 5 minutos, tiempo excesivo a la hora de enfrentarse a  tareas tan críticas como son las de seguridad.

 auge actual de las técnicas de *machine learning* (ML) junto con el crecimiento, disponibilidad y sencillez de uso de la computación distribuida, permite paliar en gran medida los problemas de rendimiento de las soluciones centradas en el análisis de tráfico agregado.

## Objetivos

La  del TFG es la implementación y despliegue de una solución distribuida basada en ML que permita la detección en tiempo real de comportamientos anómalos respecto al tráfico en una red.


Para su consecución introduciremos conceptos transversales necesarios, como pueden ser la captura de flujos, **la algoritmia asociada al ML** o las potencialidades de uso de la computación distribuida.

Concretamente los objetivos a alcanzar son los siguientes: 

### Acercamiento inicial a las tecnologías NETFLOW e IPFIX

- Visión global y estado actual de la tecnología.
- Alternativas de implementación en el mercado.
- Diseño y despliegue de una arquitectura de captura basada en ellas.

### Selección e implementación de un algoritmo de ML

- Selección, transformación y análisis de los datos que compondrán el dataset del estudio. 
- Selección del algoritmo más adecuado para la **tarea de clasificación**, teniendo en cuenta como parámetros de rendimiento la tasa de acierto y la rapidez en la ejecución.
- Evaluación del modelo resultante en un entorno offline sin requerimientos de tiempo real.

### Diseño y despliegue de un sistema distribuido basado en Apache Spark

- Acercamiento a la tecnología distribuida centrándose en la solución Apache Spark.
- Configuración y puesta en marcha de la arquitectura distribuida que dé sustento a las tareas de análisis que centran este trabajo.

## Despliegue y pruebas de la solución de análisis en un entorno en tiempo real

- Despliegue del modelo y evaluación de éste en un entorno online.
- Medir el rendimiento de la solución teniendo como punto de vista, la cantidad de flujos por segundo que es capaz de procesar.

## Metodología



Usaremos una metodología *agile* correspondiendo sus *sprints* a las fases que describiremos a continuación (exceptuando “Planificación” y “Recopilación y análisis de información”).

En cada uno de los *sprints* realizaremos tareas de análisis, documentación, desarrollo y pruebas, entregando a su finalización un producto funcional correspondiente a la fase en la que se engloba.

## Fases

A la hora de abordar la tarea distinguimos 8 fases diferentes. Las 2 primeras corresponden a los trabajos preparatorios que dan sustento a las 6 posteriores.

### 1. Planificación

En esta primera fase nos centraremos en la asignación de cargas de trabajo y orden de ejecución de las fases posteriores.

### 2. Recopilación y análisis de información

Debido a que en gran medida la materia del estudio es desconocida por parte del autor, este bloque tendrá asignado una buena parte de la carga de trabajo total.

En un primer momento nos centraremos en la recopilación de los conceptos principales sobre los que versan las tecnologías implicadas, los cuales nos permitirán comprender en un segundo paso los conceptos más complejos que harán falta para completar el TFG y la elaboración de la memoria final.

### 3. Preparación de los datos

Consistirá en la preparación previa del dataset para su posterior uso en el entrenamiento de los algoritmos a evaluar.

El dataset escogido es *Intrusion detection evaluation dataset* (*ISCXIDS2012*). De uso muy extendido, nos permitirá en último término valorar nuestra solución contra otras ya publicadas.

### 4. Implementación del algoritmo de ML

Inicialmente acometeremos la selección e implementación del algoritmo de clasificación teniendo como punto de partida la colección que nos ofrece el componente de Apache Spark ML. Tras la elección del modelo procederemos a su evaluación contra el conjunto de datos de validación.

Todos los procesos asociados a esta fase se realizarán en un entorno offline. El modelo final será cargado posteriormente en el sistema distribuido.

### 5. Diseño y despliegue del sistema distribuido

En un primer momento el diseño, implementación y despliegue del sistema distribuido se realizará en local corriendo sobre una única máquina.

Sabiendo que ésta no es la forma de sacar todo el partido a la tecnología, dependiendo del tiempo restante del proyecto y el coste asociado, se valorará desplegar el sistema sobre la plataforma AWS de Amazon.

## 6. Diseño y despliegue de sistema de captura de tráfico

Se centrará en la configuración e instalación de la infraestructura necesaria para realizar la captura y agregación de tráfico que sirve de punto de entrada a nuestro sistema.

## 7. Despliegue y pruebas del modelo final

A continuación, cargaremos el modelo algorítmico en nuestro sistema distribuido y evaluaremos su comportamiento ante un entorno en tiempo real.

## 8. Extracción de conclusiones y valoración final

Por último, tras estudiar los datos de la fase 7, podremos valorar si los objetivos indicados en este documento han sido realizados y la adecuación de éstos a nuestras expectativas iniciales.

Finalmente se desarrollará una memoria final con las conclusiones sacadas del trabajo realizado.

## Resultados previstos

En un primer lugar entendemos que la memoria de por sí, sirve como un acercamiento inicial a las diferentes tecnologías de las que hace uso este trabajo.

El grueso del producto final será el sistema distribuido desplegado que permitirá el análisis del tráfico en tiempo real en busca de posibles anomalías.

Esperamos que el trabajo sea una muestra del potencial que el uso de técnicas de ML supone para la realización de trabajos de monitorización y securización de redes.

## Bibliografía

- *Flow Monitoring Explained: From Packet Capture to Data Analysis With NetFlow and IPFIX*. Rick Hofstede, Pavel Celeda, Brian Trammell, Idilio Drago, Ramin Sadre, Anna Sperotto, Aiko Pras
- *Flow-based Compromise Detection*. Rick Hofstede
- *Network Anomaly Detection using Artificial Neural Networks*. Andropov Sergey, Budko Marina, Budko Mikhail, Guirik Alexeki
- *Real-time Analysis of NetFlow Data for Generating Network Traffic Statistics using Apache Spark*. Milan Cermak, Tomas Jirsik, Martin Lastovicka
- *Network Traffic Classification Using Apache Spark*. Marc Muntanyola Pros

<sup>i</sup> <https://www.juniper.net/us/en/products-services/what-is/ids-ips/>

<sup>ii</sup> <https://www.unb.ca/cic/datasets/ids.html>