Desarrollo de extractor de características de tráfico de redes orientado a la identificación y análisis en la detección de ataques

Santiago Ríos Guiral

Trabajo de grado como requisito para optar al título de:
Ingeniero Electrónico

Asesor:
Jaime Alberto Vergara Tejada

Universidad de Antioquia
Facultad de Ingeniería, Departamento de Ingeniería Electrónica y Telecomunicaciones
Ingeniería Electrónica
Medellín, Antioquia, Colombia
2022

TABLA DE CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

RESUMEN

En la actualidad la necesidad de implementar nuevos mecanismos de protección a las redes de computadores ha ganado relevancia debido a la globalización y la complejidad de los nuevos dispositivos de cómputo. Con el objetivo de establecer mecanismos de seguridad se ha ido desarrollando la seguridad informática, un área de la ingeniería electrónica y de las telecomunicaciones cuyo objetivo es crear, diseñar y desarrollar técnicas y herramientas capaces de proteger las redes de computadores y sus elementos ante las vulnerabilidades del mundo informático. Para satisfacer estas necesidades se están desarrollando nuevas herramientas que haciendo uso de métodos de aprendizaje de máquina e inteligencia artificial permiten analizar el tráfico de red y así crear modelos y sistemas capaces de detectar cuando se presenta un ataque y proveer sistemas de prevención ante estos.

En este trabajo se propuso la modificación de un extractor de tráfico de red orientado a la detección de ataques con el propósito de analizar qué propiedades del tráfico de red influyen en la discriminación del tráfico normal con respecto a los ataques informáticos y así lograr obtener valores de precisión diferentes en la clasificación del tráfico cuando se compara su funcionamiento con la versión no modificada. Para lograr estos objetivos se realizó un estudio de las bases de datos disponibles conformadas por tráfico de red, centrándose en los formatos de creación ya sea como paquetes o flujos de red, su disponibilidad y su relevancia dentro del área de la seguridad. Además, se hizo un estudio de los extractores disponibles centrándose en que estos fueran de código abierto y que pudiesen ser adaptados a los requerimientos de este trabajo y cuya orientación este en la clasificación de tráfico y asimismo se realizó un estudio de las características y propiedades del tráfico que permiten diferenciar los ataques del tráfico normal. A partir de los desarrollos previamente descritos se hizo la modificación del extractor CICFlowMeter, se realizaron pruebas sobre su correcto funcionamiento y se utilizaron diferentes bases de datos con tráfico en formato de paquetes. Con el extractor modificado se obtuvieron archivos CSV y haciendo uso de estos se realizó la clasificación del tráfico usando modelos de aprendizaje de máquina. Cuando se realizó el comparativo de las 2 versiones del extractor se

encontró que las modificaciones hechas influyen sobre la precisión de la clasificación y que para determinados tipos de ataques la capacidad de identificar el tráfico anómalo incremento con las nuevas características.

palabras clave **– Extractor, tráfico de red, clasificación de tráfico, tráfico anómalo y tráfico normal, redes de computadores.**

ABSTRACT

In today's world, the need to implement new security mechanisms for computer networks has gained relevance due to globalization and the complexity of new computing devices. To establish a protection mechanism, computer security has been in constant development. This is the area of electronic and telecommunications engineering whose objective is to create, design, and develop techniques and tools capable of protecting computer networks and associated devices from vulnerabilities. To meet these needs new tools are being developed, some of these tools involve machine learning and artificial intelligence to allow them the analysis of network traffic and thus create models and systems capable of detecting when an attack occurs and then be able to present prevention systems against these intrusions.

In this work, the modification of a traffic flow extractor is made, and its performance in a traffic classification experiment is evaluated. The purpose is to analyze which properties of network traffic influence the discrimination of normal traffic in comparison to computers attacks and thus obtain different values in the classification accuracy for the modified extractor. To achieve these objectives, a study of the available databases built with network traffic was carried out focusing on the data format, either packets or network flows, their availability, and their relevance within the security field. Also, a study of available extractors was made with an emphasis on their availability to be modified, the properties they capture, their traffic variety, and their building orientation aimed at network security and traffic classification. Therefore, the CICFlowMeter extractor was modified by adding new extraction characteristics, tests were carried out to verify the CICFlowMeter correct operation and different databases that include traffic on packet format were used to obtain CSV files and finally perform the traffic classification with the use of diverse machine learning models. By comparing the two versions of the extractor, it was found that the modifications do influence the classification's accuracy, and for certain types of attacks, the ability to identify anomalous traffic increases with the addition of the new traffic characteristics.

keywords **– Extractor, network traffic, traffic classification, anomalous traffic, normal traffic, computers network.**

I. INTRODUCCIÓN

A. Contexto

La globalización y la complejidad de los nuevos sistemas de comunicación han sido el eje fundamental del avance y crecimiento de las redes de computadores. En la actualidad se ha vuelto imperativo el diseño y la implementación de sistemas de seguridad con el objetivo de proteger la información de los usuarios. El mundo informático sigue avanzando a gran velocidad, el uso de redes de computadores hace parte de la vida cotidiana de las personas, es fundamental para suplir las necesidades básicas como para las actividades de ocio y entretenimiento y se han vuelto indispensables en el continuo desarrollo de la humanidad. Ante esta situación, hay un constante crecimiento de dispositivos conectados a las redes, lo cual a su vez genera un aumento en la cantidad de información que se intercambia y almacena. Todas estas pautas suponen nuevos retos en el área de la seguridad y es un tema fundamental por tratar, ya que un principio de la seguridad de la información es la toma de decisiones y acciones que permitan proteger la información de los usuarios y así mantener la integridad de esta.

El desarrollo de nuevas técnicas de prevención y detección de ataques son fundamentales con el propósito de establecer la confidencialidad, la integridad y la disponibilidad de la información ante el surgimiento de nuevas modalidades de ataques informáticos que se aprovechan de las vulnerabilidades presentes en las redes de computadores [1]. Cada año se evidencia la aparición de nuevas modalidades y tipos de ataques informáticos que involucran tanto *hardware* como *software* que vulneran la privacidad de las redes y también se observa un mayor número de ataques ejecutados. Con el objetivo de contrarrestar esta problemática de seguridad, el área de la seguridad informática ofrece un constante desarrollo de nuevas herramientas que ofrecen la protección de la información que se transmite a través de los dispositivos electrónicos.

Para lograr la prevención y detección de ataques se están implementando nuevos sistemas de intrusión y detección de ataques, los cuales utilizan mecanismos de inteligencia artificial. Estos métodos se basan en el análisis y estudio del tráfico de red con el objetivo de crear modelos y sistemas capaces de detectar cuando se presenta un ataque, así como elaborar pólizas de seguridad que permitan la prevención de estos mismos. Estos sistemas se fundamentan en el uso de bases de datos y repositorios con contenido correspondiente a tráfico normal y anómalo. Estás bases de datos son actualizadas constantemente con la inclusión de nuevos ataques y de esta forma se mantienen al día las pólizas de seguridad.

El tráfico puede organizarse en paquetes o flujos de datos y contienen información relevante de la transmisión y de la red de computadores. En este proceso de recaudar el tráfico de red participan los extractores de características o administradores de tráfico. Estas son herramientas fundamentales que permiten obtener información relevante del tráfico y son utilizados para extraer y capturar propiedades del tráfico que son usados en la construcción de modelos de clasificación de tráfico.

En la actualidad existen distintos tipos de extractores que permiten el análisis y la recolección del tráfico. Además, se cuenta con un amplio repertorio de bases de datos y en combinación son usados en el desarrollo de modelos de detección de ataques [2]. No obstante, es necesario continuar desarrollando nuevos modelos de clasificación de tráfico y para esto se requieren nuevas bases de datos, partiendo de la creación y modificación de extractores que permitan extraer características del tráfico relevantes para diferenciar entre el tráfico normal y el tráfico anómalo [3], [4].

B. Contribución de este trabajo

En este trabajo se presenta el proceso de estudio requerido en el área de la seguridad de la información para modificar un extractor de tráfico y evaluar su precisión en la clasificación de tráfico. Se realiza un estudio de las bases de datos disponibles con tráfico relevante, se hace un

estudio de extractores de código abierto disponibles y se hace la modificación de un extractor. A partir del extractor modificado se hace un análisis comparativo con la versión original con el objetivo de determinar la influencia de los cambios realizados en el proceso de clasificación de tráfico mediante la implementación de varios modelos de aprendizaje de máquina. En la **figura 1** se muestra una nube de palabras que permite conocer los términos claves y relacionados al tema durante el desarrollo de este trabajo.

Figura 1. Nube de palabras con términos claves de este trabajo de grado.

II. OBJETIVOS

A. Objetivo general

Implementar un extractor de características de tráfico de red, a partir de la modificación de una herramienta de código abierto y libre distribución, que permita obtener propiedades que faciliten la identificación de ataques informáticos.

B. Objetivos específicos

- Realizar un estudio del estado del arte de los extractores de características de tráfico de red disponibles, identificando adicionalmente las características más relevantes dentro de la detección de ataques informáticos.
- Seleccionar el extractor de características a modificar. Los cambios se centrarán en agregar nuevas características propuestas y modificar o eliminar existentes, evaluando el desempeño de diferentes combinaciones de características en la detección de ataques.
- Implementar el extractor modificado, que permite extraer diferentes características que faciliten la detección de tráfico malicioso Realizar las pruebas sobre su funcionamiento usando bases de datos que contengan tráfico de red.

III. MARCO TEÓRICO

A. Tráfico de red

El tráfico de red se define como el conjunto de datos que se mueven a través de diferentes redes de computadores en distintos instantes de tiempo. Normalmente este tráfico se captura ya sea en un formato de paquete o de flujo. El proceso de captura de dicha información varía dependiendo del formato, en caso de ser un paquete, su captura se logra al reflejar los puertos de red de los distintos dispositivos y existen diferentes programas que permiten la captura y análisis de la información de red. En el caso de un flujo de tráfico, un grupo de paquetes de red se agrupan en un evento, donde estos paquetes tienen unas características en común dentro de una ventana de tiempo predefinida. El tráfico de red constituye la información que es transmitida por 2 o más usuarios y que permite interconectar los usuarios ya sea en una misma red o en redes diferentes. En la **figura 2** se muestra un diagrama básico de una red de computadores que se conecta a una red global como lo es internet [2].

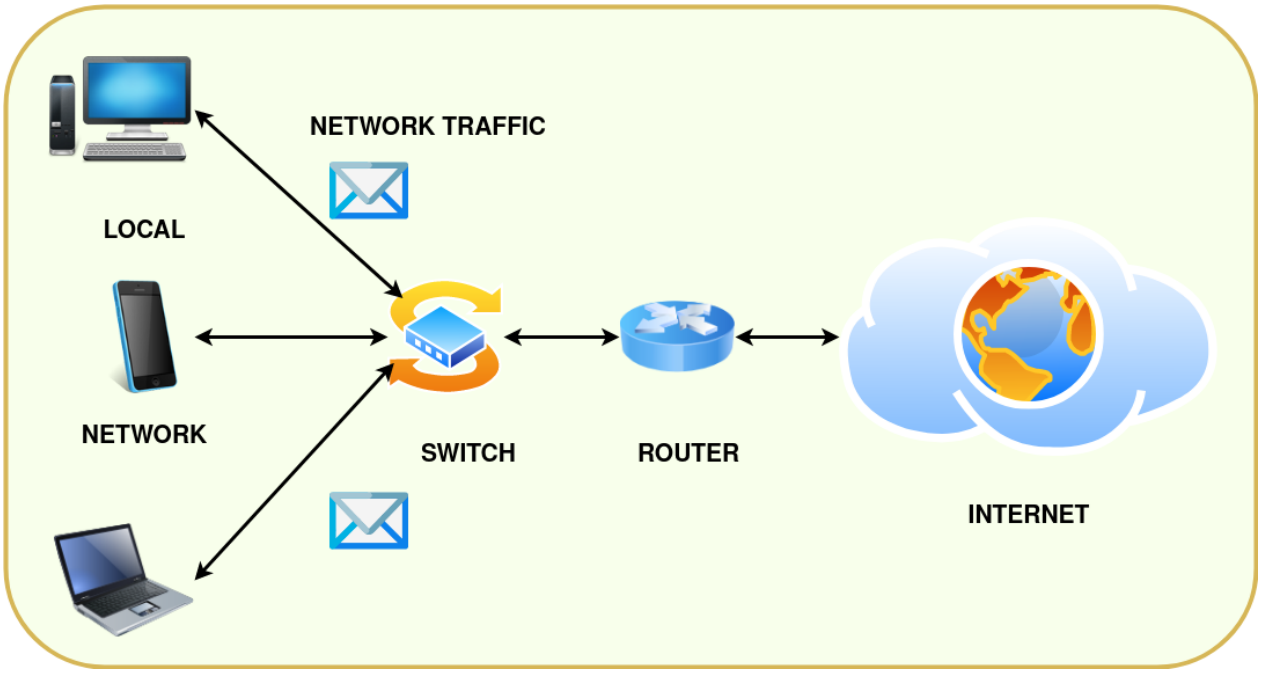

Figura 2. Diagrama básico de una red local de computadores y su acceso a una red global.

El uso del tráfico de red varía de acuerdo con los objetivos establecidos por un administrador de red, este puede ser usado desde ámbitos como el análisis de datos para visualizar el desarrollo de nuevas tecnologías hasta el campo de la seguridad informática donde se usa para evaluar las características de un tráfico normal a las de un ataque y así construir modelos que permiten su identificación [2].

1) Paquete

Un paquete de tráfico de red es la unidad de información de red que se almacena de acuerdo con un evento. Los paquetes son capturados en un formato de archivo conocido como pcap (*packet capture*) el cual funciona como una interfaz de captura del tráfico de red. El paquete contiene las características de la transmisión (*Header*), la cual permite hacer un control sobre la comunicación e incluye información sobre los protocolos usados para transmitir la información de un usuario. Además, los paquetes contienen la información de usuario que se está intercambiando (*Payload*) y esta puede ser encriptada para respetar la privacidad del usuario. Los datos de cabecera del paquete son establecidos por la red y los protocolos de transporte y existen diferentes protocolos, sin embargo, los más importantes son: TCP, UDP, ICMP e IP [2].

2) Flujo

Los flujos de red son formatos condensados de información que están conformados principalmente por la información sobre las conexiones dentro de la red. El flujo se crea al agregar un grupo de paquetes transmitidos dentro de una ventana de tiempo. Los flujos se generan a partir de características similares entre un grupo de paquetes, pueden ser creados a partir del protocolo usado, el intercambio de información entre 2 puertos de red, el intercambio de información entre 2 direcciones IP o de acuerdo con una especificación dada por el desarrollador de las herramientas que permiten la captura del tráfico. Usualmente el flujo no contiene la información que se está intercambiando y los paquetes dentro del flujo conforman un evento. Normalmente el flujo contiene tanto la dirección IP fuente y destino, el puerto fuente y destino y el protocolo de transporte. Los flujos pueden ser clasificados en 2 categorías:

unidireccionales y bidireccionales. Los flujos se generan al capturar el tráfico directamente de la red usando herramientas como nfdump, entre otras o puede ser construido usando extractores y analizadores de tráfico que leen archivos pcap [2], [5].

Una herramienta comúnmente usada para la creación de flujos es Netflow. Este es un conjunto de servicios prestados por Cisco IOS que permite a los administradores de red acceso a la información concerniente a los flujos IP dentro de la red. La salida básica se conoce como un "*flow record*" y su formato de captura depende de las plantillas disponibles y de las cuales el usuario utiliza de acuerdo con los objetivos que se plantea al utilizar la herramienta [5], [6].

B. Bases de datos de tráfico

Una base de datos con contenido de tráfico de red corresponde a la recopilación de las comunicaciones entre diferentes dispositivos pertenecientes a una red. Estas bases de datos se crean con el objetivo de proveer herramientas para el estudio y la clasificación del tráfico. El tráfico se puede guardar en diferentes formatos de archivos. Normalmente los paquetes de red se almacenan en archivos pcap mientras que los flujos son almacenados en archivos csv (*Comma Separated Values*). Para su creación se requiere el uso de extractores o administradores de tráfico que permitan capturar los paquetes de una red además de establecer las características que se están extrayendo [7].

Las bases de datos cumplen un papel fundamental en el área de la seguridad informática ya que una correcta recopilación del tráfico de red permite optimizar el desarrollo de sistemas de detección y prevención de ataques. Para esto, es importante tener en consideración la gran variedad de tráfico normal como anormal, con las características del tráfico especificadas, que el tráfico se encuentre debidamente anotado, que los ataques estén claramente identificados y anotados, que exista la documentación necesaria explicando el método creación y uso, que sean

constantemente actualizados y que estén a la disposición de las distintas investigaciones que requieran su acceso [7], [8].

C. Seguridad informática

La seguridad informática es el área de la seguridad que se encarga de la protección de las redes, los dispositivos computacionales y todo dispositivo asociado. Su objetivo se centra en la creación, el diseño y el desarrollo de técnicas y herramientas que resguarden la información que los usuarios están transmitiendo. Su finalidad se centra en preservar la integridad, la confidencialidad y la protección de la información que se utiliza en las redes de computadores. Por lo tanto, en este campo se implementa un constante desarrollo de herramientas de *hardware* y *software* que evita el uso indebido y no autorizado de los dispositivos de red [1], [9].

1) Sistemas de detección de intrusiones

Un sistema de detección de intrusiones (*IDS - Intrusion Detection System*) es un dispositivo o varios que monitorean el tráfico de red y que realiza un análisis de esté partiendo de unas pólizas de seguridad previamente planteadas. Tiene como objetivo diferenciar el tráfico normal del tráfico anómalo y así detectar los ataques partiendo de la clasificación obtenida cuando se hizo una búsqueda de las anomalías en el tráfico. Este mecanismo de seguridad permite detectar cuando hay un intento de ataque a una red de computadores que se está monitoreando, también permite discriminar el comportamiento de los ataques y agrega robustez a la red al asegurar integridad y seguridad en las comunicaciones que se llevan a cabo. La **figura 3** muestra el diagrama de una red con la presencia de un sistema de detección de intrusiones. Usualmente este dispositivo está integrado con el módem o el conmutador (*switch*) dentro de una red [10], [11].

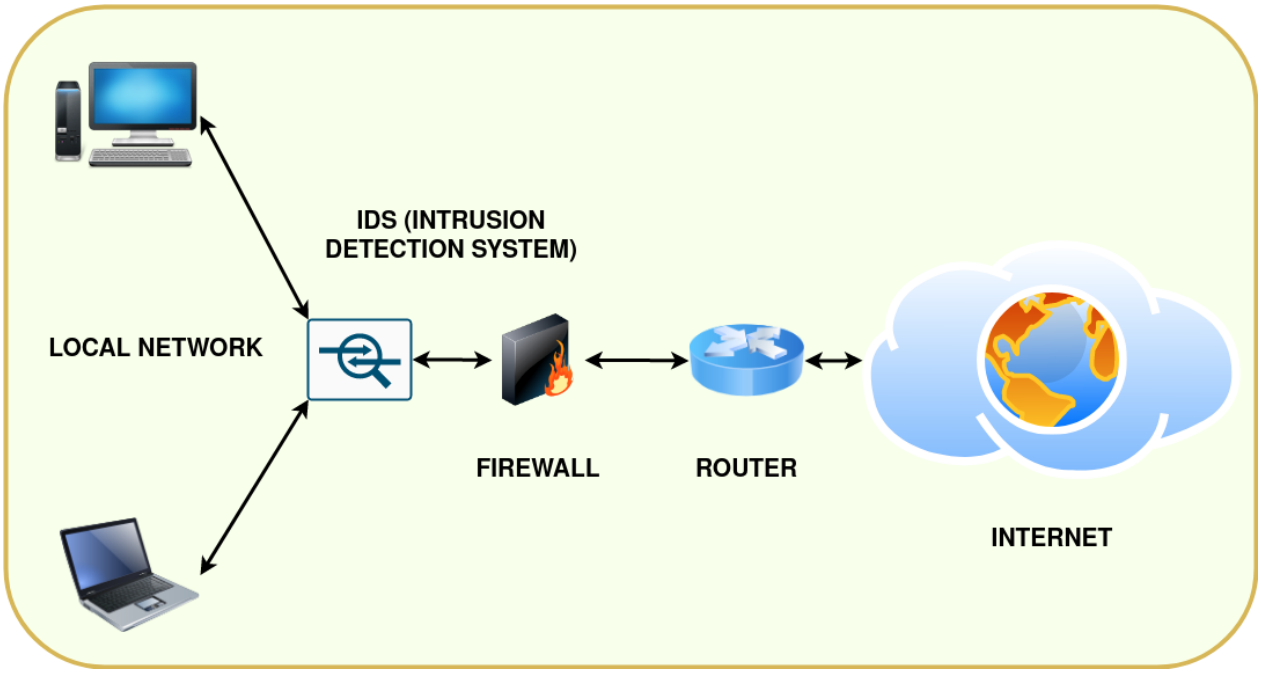

Figura 3. Diagrama de un sistema de detección de intrusiones.

2) Sistemas de prevención de intrusiones

El sistema de prevención de intrusiones (*IPS - Intrusion Prevention System*) es un dispositivo o varios que analizan el tráfico dentro de una red de computadores y cuyo objetivo se centra en descartar todo tráfico malicioso a la vez que permite bloquear los ataques informáticos. El sistema realiza un análisis profundo del tráfico siguiendo las pólizas de seguridad preestablecidas con anterioridad. Entre las actividades que realiza se encuentra el trazado de los protocolos utilizados, inspección de las características del tráfico, análisis estadístico, análisis del encabezado de los paquetes, entre otros. En caso de la presencia de un ataque, el dispositivo permite tomar decisiones basadas de acuerdo con la seriedad de estos. Las medidas tomadas pueden incluir desde desconectar la comunicación, descartar el paquete hasta la generación de alarmas. En la **figura 4** se muestra su diagrama en una red de computadores y similar a un sistema de detección de intrusiones se encuentra ubicado en el módem o conmutador de la red [10].

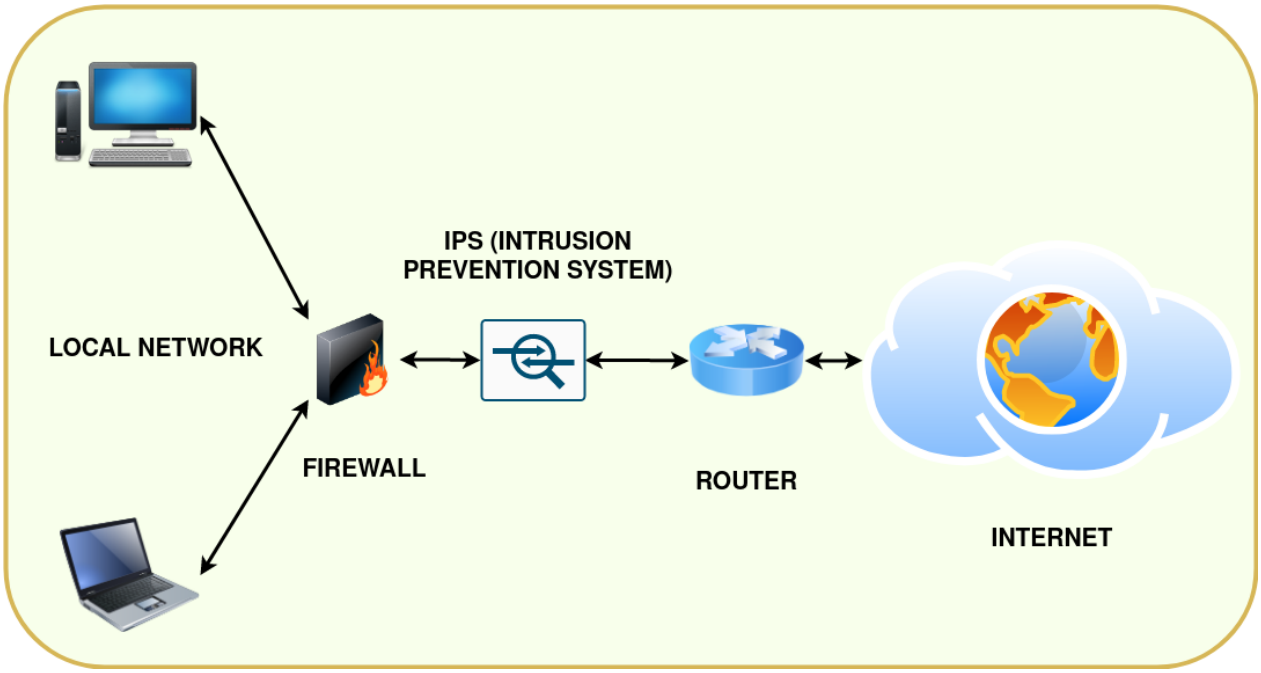

Figura 4. Diagrama de un sistema de prevención de intrusiones.

3) Diferencia entre un IDS e IPS

La mayor diferencia entre estos dos dispositivos se encuentra en su ubicación. Un dispositivo IPS se encuentra ubicado entre el "firewall" y el dispositivo de red (enrutador) fuera de la red local con el objetivo de bloquear el tráfico de red cuando se detecta un ataque. En cambio, un IDS se encuentra ubicado dentro de la red local con el objetivo de servir de alarma ante la ocurrencia de un ataque. En cuanto al funcionamiento, un IPS analiza el tráfico de red y determina el propósito del paquete y a partir de este estudio define si bloquea el tráfico o continua la transmisión de este. En el caso del IDS este se centra en analizar toda la información dentro de una red local y de esta forma se ajustan las diferentes pólizas de seguridad establecidas con el fin de evitar la que se presenten ataques futuros [10], [11].

4) Detección de intrusiones

La detección de intrusiones se define como el proceso de monitoreo a una red de comunicaciones con el objetivo de detectar actividades sospechosas. Este procedimiento requiere la colección de información de las redes de computadores y se implementa un análisis del tráfico

para identificar qué actividades se clasifican como sospechosas. Existen 3 categorías para identificar el tráfico malicioso [11].

a) Detección basada en firmas

Una firma es la característica previamente conocida de un ataque que ya está identificado. Esta técnica se centra en comparar las actividades maliciosas con tráfico de red que pertenecía a los ataques ya identificados. Es una técnica bastante utilizada, sin embargo, presenta una desventaja, ya que depende de la comparación con ataques ya conocidos y esto impide que esta técnica sea utilizada con nuevas modalidades de ataques informáticos [2].

b) Detección basada en anomalías

Una anomalía se refiere a las propiedades del tráfico que al comparar con un tráfico normal se logra visualizar una diferencia marcada en sus características. Esta técnica se centra en identificar los patrones del tráfico en una red y mediante un análisis profundo se determinan las actividades sospechosas cuando se compara con el tráfico normal. En esta categoría se utilizan técnicas basadas en el análisis estadístico del tráfico, técnicas basadas en el conocimiento de actividad anómala y técnicas orientadas a la inteligencia artificial [12].

c) Detección híbrida

Las técnicas de detección híbridas buscan solventar las desventajas de los 2 métodos anteriores. Este método realiza un análisis de relación cruzada entre los resultados de las detecciones basadas en firmas y anomalías. Se busca evitar inconvenientes ante la detección de ataques nuevos por parte de la técnica de detección basada en firmas y se busca evitar que el tráfico anormal sea ocultado como tráfico normal cuando las propiedades del tráfico son similares. Está técnica ofrece mejores resultados en la implementación de sistemas de detección y prevención de intrusiones al compararlo con los 2 métodos anteriores [12].

D. Extracción y análisis de tráfico

1) Extractores de tráfico

El análisis de tráfico consiste en la captura de información de los dispositivos de red con el objetivo de realizar una inspección exhaustiva y determinar qué está sucediendo dentro de las infraestructuras de red. Un extractor de tráfico es una herramienta fundamental para cumplir con esta tarea ya que captura la información de la red. Los extractores son herramientas que monitorean y analizan el tráfico de red con el objetivo de extraer la información y las características de comunicación entre los dispositivos de red. Existen diferentes tipos de extractores y estos varían de acuerdo con los objetivos establecidos por el desarrollador. Estos pueden ser orientados a la recolección de tráfico, al análisis del comportamiento de la infraestructura de red, a la creación de bases de datos o la investigación en el campo de la seguridad informática [2].

Los extractores utilizan librerías que permiten el procesamiento del tráfico en la capa de red y se encargan de la captura de paquetes pertenecientes a la comunicación entre los diferentes dispositivos. El tráfico se almacena en formatos de archivo que permita su fácil acceso y posterior análisis. Este puede ser capturado en formato de paquetes (pcap) o en formato de flujos (csv). Además, el extractor permite extraer diferentes propiedades y características del tráfico y de la red donde se realiza su instalación. Las propiedades del tráfico pueden ser obtenidas de la información transmitida en los paquetes o son generadas a partir cálculos estadísticos con los datos ya recopilados [2], [4].

2) Clasificación del tráfico

La clasificación de tráfico es un proceso fundamental por realizar por parte de los operadores de red ya que permite tener un control sobre las operaciones que se realizan dentro de la red. Entre las diferentes actividades que conforman la clasificación se encuentra el área de la seguridad informática y el desarrollo de herramientas para la detección y prevención de ataques.

La clasificación requiere asociar el tráfico de red a unas clases predefinidas (tráfico normal y ataques) y su metodología puede ser descompuesta en 4 categorías: identificación del puerto de red, contenido del tráfico, comportamiento del dispositivo de red y las características del flujo de red [4].

a) Clasificación basada en la identificación de puertos

El tráfico es clasificado teniendo en cuenta el puerto de red que se está usando en la transmisión. De acuerdo con el número de puerto identificado por la IANA (*Internet Assigned Numbers Authority*) y previamente registrado a una aplicación ya definida es posible asignar una clase de servicio a la comunicación de red. En la actualidad este método de clasificación se está volviendo ineficiente debido al uso de dinámicas de negociación de puertos, procesos de tunelización y el uso indebido de números de puerto asignados a aplicaciones conocidas buscando evitar la detección por parte de los "firewalls" y ocultar el tipo de tráfico que se transmite [4].

b) Clasificación basada en el comportamiento del dispositivo de red

Es un método de clasificación que consiste en analizar el contenido del tráfico de red en busca de características que identifican los servicios que se están utilizando en la comunicación. Es un método alternativo a la clasificación por identificación del puerto de red, sin embargo, requiere una mayor capacidad computacional y hacer uso de mayores espacios de almacenamiento. Es un método todavía en uso pero que tiene una gran desventaja proveniente de la necesidad de incrementar la seguridad de la información perteneciente al usuario y por ende el contenido del tráfico es usualmente encriptado y su acceso se prohíbe de acuerdo con las leyes de privacidad informática establecidas [4].

c) Clasificación basada en el contenido del tráfico

La clasificación a través del comportamiento de los dispositivos se basa en determinar las características principales de los dispositivos de red y de esta forma predecir cuál es el tipo de

aplicación y servicio que están ejecutando durante la comunicación. Esta técnica define un punto de análisis en los extremos de la red y en este lugar examina el tráfico. Entre los datos que extrae del tráfico se encuentran la cantidad de dispositivos conectados, el protocolo de transporte, los puertos de red que se están utilizando, las direcciones IP, entre otros. Esta clasificación se fundamenta en las características del tráfico establecidas por las aplicaciones y los protocolos y a partir de diferentes estudios se asigna el tráfico a una clase predefinida. Es un método con un alto índice de precisión, no obstante, la desventaja de este radica en que la precisión depende de la ubicación del dispositivo de monitoreo [4].

d) Clasificación basada en las características del flujo de tráfico

Esta técnica de clasificación está basada en la creación de sesiones de comunicación, las cuales están conformadas por una dupla completa de flujos de tráfico. Un flujo completo es un intercambio de información unidireccional de paquetes consecutivos en una red computacional. El paquete debe contener el puerto y la dirección IP tanto de la fuente como del destino, además debe indicar el protocolo que se está implementando. A partir del flujo se puede obtener un subflujo el cual consiste en una porción del flujo completo que se obtiene al definir una ventana de tiempo dentro de la sesión de comunicación.

Los flujos están compuestos por características, las cuales representan las propiedades únicas de los paquetes dentro de la sesión. En la clasificación mediante las características del flujo se utilizan las propiedades del tráfico como discriminadores para mapear los flujos a las clases de interés. En esencia, la clasificación de tráfico basada en características de flujo explota la diversidad y las características distinguibles de la huella de tráfico generada por diferentes aplicaciones como lo son los extractes [4].

E. Aprendizaje de máquina

El aprendizaje de máquina (*ML - Machine Learning*) es el área de la computación que estudia la creación, el desarrollo y la aplicación de algoritmos matemáticos y técnicas que permitan buscar patrones con datos de referencia. Estas técnicas se basan en mejorar automáticamente sus resultados a través de la experiencia y el uso de los datos disponibles. Hace parte de la rama de la inteligencia artificial y su objetivo se centra en la construcción de modelos capaces de predecir resultados de acuerdo con las muestras de entrada. Existen 3 métodos de aprendizaje, el aprendizaje supervisado, el aprendizaje no supervisado y aprendizaje reforzado [3].

El aprendizaje supervisado utiliza las muestras de entrada que están debidamente anotadas y las separa en 2 grupos. El primer grupo es utilizado para entrenar el modelo haciendo uso de los diferentes algoritmos planteados y el segundo grupo se utiliza para evaluar la precisión del Modelo. En el aprendizaje no supervisado las muestras de entrada son utilizadas sin una clase destino y el análisis se realiza sobre las similitudes de los datos y a partir de la estructura encontrada se determina una distribución de los datos. Por último, en el aprendizaje de refuerzo, los algoritmos se implementan con el objetivo de crear un modelo inicial de predicción y que estos sean capaces de utilizar lo aprendido en la práctica y de esta forma mejorar las predicciones realizadas [3], [13].

El aprendizaje de máquina es utilizado en una gran variedad de áreas de investigación. En el campo de la seguridad informática se utiliza en el desarrollo de algoritmos de clasificación de tráfico con el objetivo de tener modelos capaces de discriminar los ataques del tráfico normal. Usualmente se utilizan algoritmos de aprendizaje supervisado y en la **tabla I** se presentan algunos de los algoritmos con mayor implementación en el análisis del tráfico de red [3].

TABLA I

ALGORITMOS DE APRENDIZAJE DE MÁQUINA UTILIZANDO EN EL ANÁLISIS DE TRÁFICO

Algoritmo	**Síntesis**
AdaBoost (AB)	Es un clasificador que consiste en crear un número n de predictores en secuencia donde los predictores siguientes al actual se ajustan de tal forma que se reduzcan las predicciones incorrectas al trabajar sobre parámetros de mayor complejidad en las muestras de entrada [14].
K Nearest Neighbor (KNN)	Es un clasificador que determina la clase de pertenencia de una muestra de acuerdo con las muestras vecinas en un espacio donde las características en común permiten su definición. Este modelo genera un contorno alrededor de las clases de salida [15].
Linear Discriminant Analysis (LDA)	Clasificador que utiliza una frontera lineal para determinar las clases a las que pertenecen los datos de entrada. Utiliza el teorema de Bayes sobre la probabilidad condicional. El modelo utiliza una distribución Gaussiana [16].
Random Forest (RM)	Este algoritmo utiliza una combinación de árboles de clasificación para determinar la predicción de un subconjunto de los datos de entrada y así usar el promedio de las clasificaciones por parte de los estimadores para determinar las clases de las muestras [17].
Support Vector Machine (SVM)	Esté algoritmo genera un número de regiones n igual al número de clases de salida. Clasifica las muestras de entrada de acuerdo con su posición en el espacio y genera hiperplanos que separan las clases de acuerdo con sus características [18].

Con los modelos de clasificación se puede generar una matriz de confusión como se presenta en la **tabla II** la cual ofrece una gama de medidas de desempeño. A continuación, se presenta la terminología usada para describir los resultados.

- Positivos verdaderos (*True Positive* - TP): Número de registros correctamente clasificados como tráfico normal
- Falsos verdaderos (*False Positive* - FP): Número de registros incorrectamente clasificados como tráfico normal

- Falsos negativos (*False Negative* - FN): Número de registros correctamente clasificados como tráfico anómalo.
- Positivos negativos (*True Negative* - TN): Número de registros incorrectamente clasificados como tráfico anómalo.

TABLA II

MATRIZ DE CONFUSIÓN

Matriz de confusión		Valores predichos	
		Normal	Ataque
Clase	Normal	TP	FN
	Ataque	FP	TN

Nota: Tomado de (Ring, M., 2019).

Para evaluar la clasificación se utiliza la **ecuación 1** que describe la precisión (*accuracy*) en la discriminación del tráfico de red. Esta ecuación indica el porcentaje de registros que fueron clasificados correctamente con respecto a todas las muestras de entrada.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \qquad (1)$$

También existen medidas que permiten evaluar la clasificación con respecto a las clases de salida, es decir, discriminar el tráfico como normal o como ataque. Se tiene la medida de precisión que determina el porcentaje de clasificación correcta de una determinada clase y la **ecuación 2** presenta su cálculo.

$$P = \frac{TP}{TP + FP} \qquad (2)$$

La sintonía (*recall*) es una medida que calcula la proporción de predicciones correctas de una clase sobre todas las posibilidades positivas o negativas y en la **ecuación 3** se presenta la forma de calcularlo.

$$R = \frac{TP}{TP + FN} \qquad (3)$$

Por último, se cuenta con la medida F1, la cual corresponde al promedio armónico de la medida de precisión y de sintonía para las clases de salida. En la **ecuación 4** se muestra la forma como se hace su cálculo.

$$F1 = 2 * \frac{P * R}{P + R} \qquad (4)$$

Tanto la matriz de confusión como las medidas previamente descritas permiten determinar el rendimiento de la clasificación y así realizar un comparativo del comportamiento general y un análisis específico cuando se clasifica tanto tráfico normal y anormal. Con estas medidas de rendimiento se puede hacer una comparación de los algoritmos, de las bases de datos utilizadas y de las muestras de tráfico utilizadas [3].

IV. METODOLOGÍA

En el proceso de modificación de un extractor orientado a la detección de ataques es necesario realizar un amplio proceso de investigación donde se hace un estudio de las bases de datos disponibles y de los extractores disponibles, se realiza un análisis de las características y propiedades del tráfico, un análisis del proceso de clasificación de tráfico y por último se hace la modificación del extractor y se evalúa su funcionamiento ante la clasificación del tráfico. En la **figura 5** se presenta un diagrama de flujo con las etapas que se siguieron en el desarrollo de este proyecto.

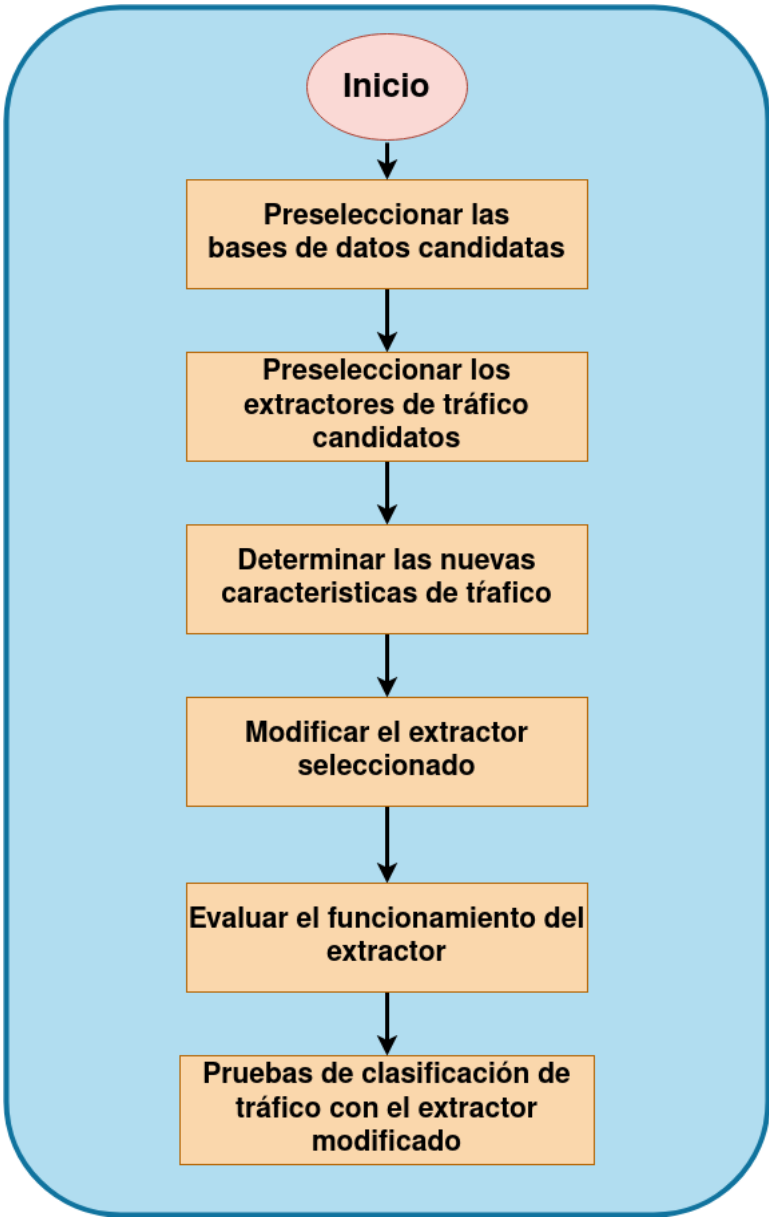

Figura 5. Procedimiento por seguir para modificar un extractor de tráfico.

A. Selección de las bases de datos

El proceso de selección está conformado por el estudio y el análisis de las bases de datos disponibles que se puedan orientar a la clasificación de tráfico. Es de gran importancia determinar qué bases de datos se van a utilizar en el momento de realizar la evaluación del extractor

modificado, ya que deben ser orientadas al desarrollo de sistemas de detección y prevención de ataques. De la calidad de las bases de datos a utilizar depende la calidad de los resultados obtenidos en la implementación de un modelo de clasificación. Es fundamental analizar qué bases de datos ofrecen la mayor variedad de tráfico, tanto normal como para diferentes tipos de ataques, además las bases de datos deben recopilar diferentes características que identifiquen el tipo de información que se transmite en una red. Las bases de datos deben tener un formato flexible que permita hacer diferentes pruebas de clasificación, deben ser continuamente actualizadas, deben estar anotadas y requieren una amplia variedad de tráfico [1]. En la **tabla III** se presenta una lista de bases de datos a utilizar en el proceso de selección y depuración para su posterior uso con el extractor modificado.

TABLA III

BASES DE DATOS CON CONTENIDO DE TRÁFICO DE RED UTILIZADAS EN EL PROCESO DE PRESELECCIÓN PARA SU POSTERIOR USO EN LA EVALUACIÓN DEL EXTRACTOR MODIFICADO

Base de datos	Formato	Año de creación	Anotado	Tráfico normal/ ataques
AWID	Sesión	2015	SI	SI/SI
Booters	Paquete	2013	NO	NO/SI
Botnet	Paquete	2014	SI	SI/SI
CIC-Bell-DNS	Paquete/Flujo	2021	SI	SI/SI
CIC-Bell-DNS-EXT	Paquete/Flujo	2021	SI	SI/SI
CIC DoS	Paquete	2017	SI	SI/SI
CIC-IDS 2017	Paquete/Flujo	2017	SI	SI/SI
CIC-IDS 2018	Flujo bidireccional	2018	SI	SI/SI
CIDDS-001	Flujo bidireccional	2017	SI	SI/SI

CIDDS-002	Flujo bidireccional	2017	SI	SI/SI
CIRA-CIC-DoHBrw	Flujo bidireccional	2020	SI	SI/SI
CDX	Paquete	2009	NO	SI/SI
CTU-13	Paquete/Flujo	2013	SI	SI/SI
DARPA	Paquete	1998-2000	SI	SI/SI
DDOS-2016	Paquete	2016	SI	SI/SI
DDOS-CIC	Flujo bidireccional	2019	SI	SI/SI
ISCX	Paquete/Flujo	2012	SI	SI/SI
ISOT	Paquete	2010	SI	SI/SI
KDD CUP 99	Paquete	1999	SI	SI/SI
Kyoto 2006+	Sesión	2006-2008	SI	SI/SI
LBLN	Paquete	2004	NO	SI/SI
LITNET	Flujo bidireccional	2020	SI	SI/SI
NDSec-1	Paquete	2016	SI	SI/SI
NGIDS-DS	Paquete	2016	SI	SI/SI
NSL-KDD	Paquete	1998	SI	SI/SI
SSHCure	Paquete/Flujo	2014	NO	SI/SI
TRAbID	Paquete	2017	SI	SI/SI
TRAbID DDoS	Paquete	2017	SI	SI/SI
TUIDS	Paquete/Flujo	2017	SI	SI/SI
Twente	Flujo unidireccional	2008	SI	NO/SI
UGR-16	Flujo unidireccional	2016	SI	SI/SI
UNSW-NB15	Paquete	2015	SI	SI/SI

B. Selección del extractor

Esta etapa se centra en el análisis de los extractores de tráfico disponibles, que se orienten a la investigación en el campo de la seguridad informática y que sean de código libre, donde la licencia permite su modificación en aras de contribuir en este campo de la academia. Durante el proceso de búsqueda de un extractor de tráfico se debe realizar un proceso de selección reiterativo que incluye un proceso de depuración de acuerdo con los objetivos de este trabajo.

1) Preselección de extractores

Este proceso se centra en la investigación y análisis de los extractores disponibles para su modificación. La selección de estos extractores se basó en su disponibilidad, la posibilidad de hacer modificaciones, el formato de captura del tráfico de red y la orientación que se le puede dar a la clasificación de tráfico.

2) Depuración de los extractores

En esta etapa se busca depurar los extractores encontrados durante la etapa de preselección con el propósito de escoger el extractor que se ajuste de la mejor forma posible a los objetivos planteados en esté trabajo. Este procedimiento requiere que se presente una comparación de los extractores encontrados, donde se enuncian las ventajas y desventajas de su implementación y modificación. Después de que se realice este proceso de depuración se llega a un extractor seleccionado para su modificación.

C. Estudio del extractor seleccionado

Posterior a la etapa de selección del extractor a modificar, se presenta la etapa de estudio del extractor. En esta etapa se profundiza el análisis sobre los requerimientos de instalación y el lenguaje de programación utilizado, se estudia cómo es su funcionamiento y cuál es el procedimiento por realizar durante el proceso de modificación. Este proceso requiere un estudio

de las librerías que se emplean para capturar el tráfico de red y como es la conexión con la herramienta de extracción. A partir del conocimiento generado en esta etapa se establece como será implementado la modificación del extractor.

D. Modificación del extractor seleccionado

La modificación del extractor busca generar, eliminar o modificar las características del tráfico extraídas con el propósito de brindar nuevas alternativas en la discriminación del tráfico y así tener resultados de clasificación diferentes al extractor sin modificaciones.

1) Agregar características del tráfico

Se agregan estadísticas como estimaciones estadísticas de las propiedades captadas por las librerías que trabajan directamente con la capa de red en la transmisión de la información dentro del computador. Para agregar una característica se debe conocer cómo es su implementación dentro del código y se debe realizar un análisis de su importancia en la caracterización del tráfico.

2) Eliminar características del tráfico

El proceso de eliminación de características se realiza en caso de encontrar características que no ofrecen la suficiente información para hacer una discriminación del tráfico normal en comparación a los ataques informáticos. La eliminación de una característica debe estar acompañada de un análisis de porque no es necesaria su inclusión en los modelos de clasificación de tráfico.

3) Modificar características del tráfico

Modificar una característica consiste en la modificación de los parámetros utilizados en la extracción del tráfico de red. Este proceso se orienta a cambiar la formación de los flujos de red dentro de la configuración de una ventana de tiempo y como es la organización de sus paquetes de acuerdo con unas propiedades en común.

E. Pruebas de clasificación de tráfico

En la etapa final del trabajo se hace la clasificación de tráfico y se evalúa el funcionamiento del extractor modificado con respecto a la nueva implementación de las características.

1) Generar los archivos de prueba

En esta etapa se generan los archivos de tráfico con los que se harán las pruebas del extractor utilizando diferentes algoritmos de clasificación. El principal objetivo de este procedimiento se centra en lograr el correcto funcionamiento del extractor con las modificaciones implementadas. Para esto se utilizan diferentes bases de datos las cuales son utilizadas para probar la extracción de características y la correcta generación de un nuevo archivo en formato csv con el tráfico capturado. En esta etapa se utilizan las bases de datos escogidas de acuerdo con un proceso de depuración ya mencionado y que estén de acuerdo con los requerimientos del extractor. Asimismo, se realiza el proceso de extracción tanto con el extractor original como con el extractor modificado para una posterior comparación.

2) Comparación de los modelos de clasificación de tráfico

En esta etapa se realiza la comparación del extractor original con respecto al extractor modificado. Este proceso requiere el uso de Python y la implementación de modelos de aprendizaje de máquinas con el objetivo de establecer una clasificación de tráfico usando el método de aprendizaje supervisado. Se utilizan los archivos csv generados como entradas de los modelos, los cuales deben ser debidamente anotados y así diferenciar el tráfico normal de los ataques. Se realiza la creación de modelos para las 2 versiones del extractor, se divide el tráfico en grupos de datos (*datasets*) para el entrenamiento y las pruebas sobre el modelo. El resultado obtenido define el grado de precisión del modelo para clasificar el tráfico. Posteriormente se hace una comparación de la clasificación obtenida con los extractores y se determinan las conclusiones de acuerdo con los resultados obtenidos.

V. RESULTADOS Y ANÁLISIS

A. Bases de datos disponibles

El análisis de las bases de datos presentados en la **tabla III** permitió hacer una depuración con las bases de datos que se van a utilizar en la evaluación de los modelos de clasificación. Se encontró que hay preferencias a utilizar atributos o características que se pueden obtener directamente de la captura de los paquetes sobre las características generadas a través de cálculos secundarios, ya que esto optimiza el proceso de detección de ataques en una red. Se encontró una mayor utilización cuando la información del tráfico se encuentra en formatos de paquetes (archivos pcap) y el uso de estas depende de la variedad de ataques y tráfico normal que se incluyen debidamente anotados. En la **tabla IV** se presentan las bases de datos seleccionadas a utilizar en la clasificación del tráfico.

TABLA IV
BASE DE DATOS SELECCIONADAS EN LA EVALUACIÓN DEL EXTRACTOR

Base de datos	**Formato**	**Año de creación**	**Anotado**	**Tráfico normal/ ataques**
CIC-Bell-DNS-EXT [19],[20]	Paquete/Flujo	2021	SI	SI/SI
CTU-13 [21],[22]	Paquete/Flujo	2013	SI	SI/SI
ISOT [23],[24]	Paquete	2010	SI	SI/SI
TRAbID [25],[26]	Paquete	2017	SI	SI/SI
TRAbID DDoS [25],[26]	Paquete	2017	SI	SI/SI

B. Extractores de tráfico disponibles

1) Extractores preseleccionados

En la investigación que se llevó a cabo de los extractores de tráfico, se encontró una gran variedad de estas herramientas dedicadas al análisis de la infraestructura de red como lo son Wireshark y tcpdump, sin embargo, también se encontraron extractores orientados a la clasificación del tráfico. En la **tabla V** se presentan los extractores preseleccionados para ser modificados.

TABLA V

SÍNTESIS DE LOS EXTRACTORES PRESELECCIONADOS

Extractor	**Síntesis**
CICFlowMeter	Extractor y analizador de tráfico ethernet. Genera flujos bidireccionales. Trabaja tanto leyendo archivos pcap y en vivo. Creado por el instituto de ciberseguridad de Canadá y es usado en la detección de tráfico anómalo. Está codificado en Java [27], [28].
Network classification	Extractor de características orientado a la clasificación de tráfico. Su funcionamiento se centra en la lectura de archivos pcap de manera recursiva y así extraer los atributos en un formato de flujo. Está escrito en lenguaje C [29].
ML based network traffic classifier	Extractor de tráfico que utiliza NFStream en el proceso de captura. Utiliza modelos de aprendizaje de máquina para realizar el proceso de clasificación. Está escrito en C [30].
Joy	Paquete para capturar y analizar flujos de tráfico, está orientado a la investigación y el monitoreo de las redes de computadores. Está escrito en C y fue desarrollado por Cisco [31].
Kitsune	Programa desarrollado en Python que se especializa en la captura de tráfico ya sean usando archivos pcap o mediante la captura en vivo [32].
Libprotoident	Librería diseñada por la universidad de Waikato para la identificación de protocolos de la capa de aplicación mediante la inspección de paquetes. Incluye herramientas útiles para el análisis de tráfico. Está escrito en C++ [33].
Malcom	Herramienta para el análisis de tráfico que incluye módulos para la extracción y captura. El tráfico es capturado en archivos pcap y

	requiere el acceso a internet para su compilación. Usa un servidor local para su funcionamiento [34].
NFStream	Es un extractor y analizador de tráfico que genera flujos bidireccionales. Está desarrollado para Python. Trabaja leyendo archivos pcap o con capturas en vivo. Puede ser implementado como una librería o como un programa descargado desde la fuente [35], [36].
Nprint	Es un caracterizador de tráfico de red orientado a los algoritmos de aprendizaje máquina para la clasificación del tráfico. Su objetivo es representar paquetes TCP y UDP. Está escrito en C++ [37].
OpenFPC	Es una herramienta orientada a la recolección de paquetes de red. Su objetivo es desplegar una herramienta capaz de grabar las interacciones dentro de una red. Está escrito en Perl [38].
Peafowl	Es un extractor de paquetes que utiliza el método de inspección de tráfico. Está orientado a la identificación de protocolos de la capa de aplicación y fue escrito en C y C++. Desarrollado para Linux [39].

2) Depuración de los extractores disponibles

A partir de los extractores preseleccionados en la **tabla V** se obtuvo un comparativo entre estos, donde se especifican las ventajas y desventajas de su implementación y modificación. En la **tabla VI** se muestra la comparación entre los extractores y de acuerdo con esto se obtienen los resultados de la depuración donde se descartan los extractores que no cumplen con los objetivos planteados.

TABLA VI

COMPARATIVO DE LOS EXTRACTORES PRESELECCIONADOS

Extractor	**Ventajas**	**Desventajas**	**Decisión**
CICFlowMeter	Codificado en Java, uso de jnetpcap, estructura clara en su codificación, posee una gran variedad de atributos y es orientado a la clasificación.	Uso de un entorno de desarrollo.	Preselección

Network classification	Está escrito en lenguaje C y cuenta con una buena estructura. Está orientado a la clasificación.	No es de código abierto.	Descartado
ML based network traffic classifier	Uso de NFStream y conexión directa entre la captura, la extracción y la clasificación.	Pocas características.	Descartado
Joy	Utilizado por CISCO.	No es de código abierto.	Descartado
Kitsune	Escrito en Python y con una codificación limpia.	Solo trabaja con archivos pcap.	Descartado
Libprotoident	Orientado a la extracción y clasificación de tráfico.	Genera atributos usando la inspección del tráfico.	Descartado
Malcom	Captura desde archivos pcap o en vivo. Unifica el proceso de extracción y clasificación de tráfico.	Requiere de un servidor web local para implementar la interfaz.	Descartado
NFStream	Escrito en Python, captura en vivo o desde archivos pcap y las características del tráfico son definidas como clases.	Posee algunos módulos obsoletos cuando se instala desde la fuente.	Preselección
Nprint	Caracteriza paquetes TCP y UDP, extrae desde archivos pcap y en vivo.	Exceso en el número de características de tráfico.	Descartado
OpenFPC	Extrae y graba el tráfico presente en la red.	No está orientado a la clasificación de ataques.	Descartado
Peafowl	Extracción de la información de capa de red y de diferentes protocolos.	Clasifica usando la inspección del tráfico.	Descartado

C. Selección de extractor

1) Candidatos finales

En el proceso de depuración se obtuvieron 2 candidatos finales para su modificación. El primero es el extractor CICFlowMeter del instituto de ciberseguridad canadiense y el segundo es NFStream, un extractor diseñado para trabajar como librería de Python o de forma independiente.

a) CICFlowMeter

CICFlowMeter es una herramienta que permite generar y analizar los flujos de tráfico en una red de computadores. Está orientado al análisis y la clasificación de tráfico con el objetivo de ofrecer herramientas capaces de discriminar anomalías del tráfico normal. Se puede utilizar para generar flujos bidireccionales donde el primer paquete en cada dirección determina el sentido del flujo unidireccional. Además, se encuentra codificado en java y hace uso de la librería jnetpcap desarrollada en el lenguaje C, la cual se encarga de la parte de captura del tráfico. En la **figura 6** se presenta la interfaz que aparece al iniciar el proceso.

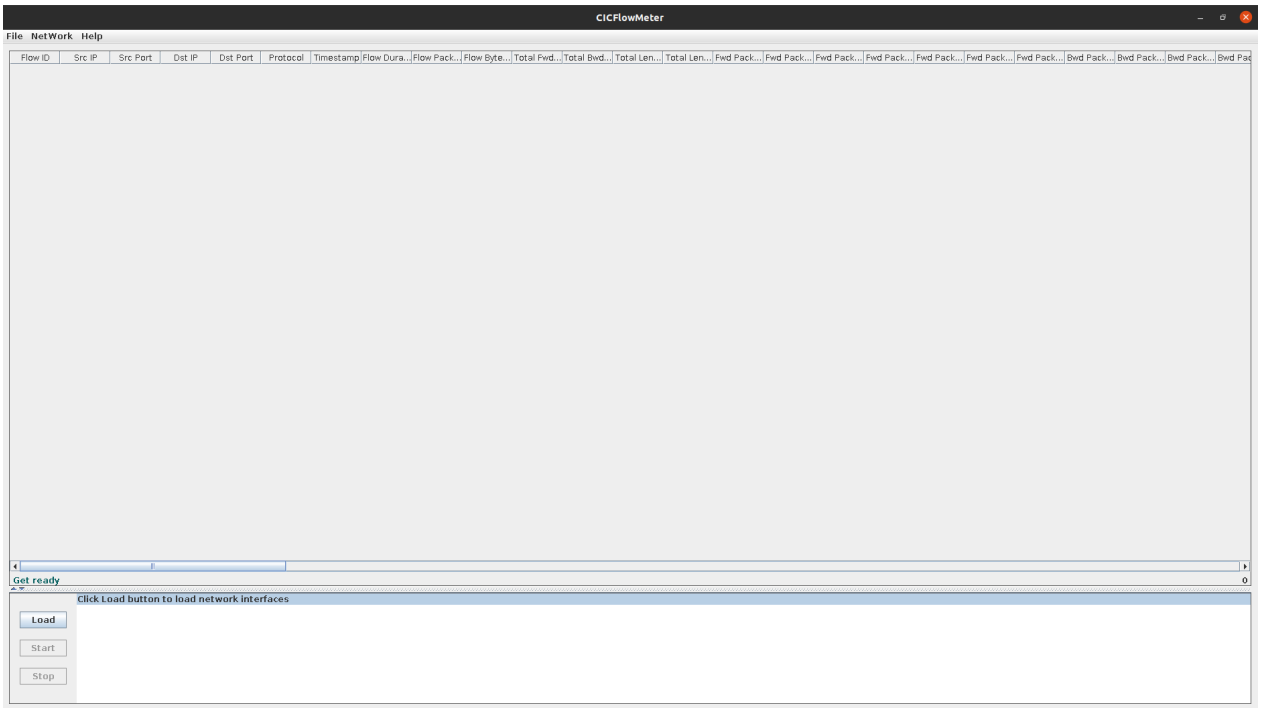

Figura 6. Interfaz gráfica del extractor CICFlowMeter.

CICFlowMeter tiene 2 modalidades de trabajo, la primera es mediante la lectura de archivos pcap y la segunda mediante la captura en vivo. En la lectura de archivos pcap se permite escoger la duración de la ventana de tiempo que agrupa los flujos y así tener un control sobre la generación. En cuanto a la captura en vivo, el usuario especifica la interfaz donde se desea capturar el tráfico. El archivo de salida está formado por los flujos extraídos y estos son almacenados en un formato csv para su posterior análisis. Permite calcular más de 80 propiedades estadísticas del tráfico y además cuenta con la posibilidad de agregar, modificar y eliminar características, permite controlar la duración de la ventana de tiempo que determina el evento (flujo) y permite convertir tráfico desde el formato de paquetes a un flujo tanto unidireccional o bidireccional [27], [40], [41].

El extractor define un flujo de datos como una secuencia de paquetes con los mismos valores para la dirección IP de fuente, dirección IP de destino, puerto fuente, puerto destino y con el protocolo implementado. Está escrito en Java y ofrece una amplia flexibilidad en el momento de seleccionar las características a utilizar en la captura del tráfico [27].

b) NFStream

NFStream es una librería desarrollada para Python orientada al análisis y estudio del tráfico de red. Permite capturar los paquetes del tráfico de red y extraer las características y atributos que lo conforman. La herramienta posee 2 métodos de extracción, mediante la lectura de archivos pcap o con la captura en vivo de los paquetes. NFStream está diseñado para ser una herramienta rápida al utilizar el lenguaje de programación C en el proceso de captura del tráfico y al usar Python para la extracción de la información relevante de la red. NFStream proporciona 2 métodos en la extracción de características estadísticas basadas en flujo. El primer método incluye características estadísticas basadas en cálculos secundarios que se obtienen luego de la generación del flujo de tráfico y el segundo método genera características de flujo temprano, es decir, que se obtienen desde el momento en el que el paquete fue capturado.

El extractor puede ser instalado desde la fuente cuando se descarga el repositorio desde GitHub y esta modalidad ofrece un mayor control sobre las modificaciones que se van a realizar ya que las características de red pueden ser modificadas y las nuevas características se agregan como clases. También puede ser instalado como librería de Python, usando el comando de consola "pip", sin embargo, la desventaja de este método es que trae limitaciones cuando se va a realizar su modificación debido a que no se puede modificar el código preexistente y sólo permite agregar características implementadas como clases [36].

2) Extractor seleccionado

A partir de los resultados obtenidos en el proceso de depuración se determinó que el extractor seleccionado el cual se va a modificar es el CICFlowMeter versión 4 desarrollado por el instituto de ciberseguridad canadiense. Se escogió este extractor debido a su amplio uso dentro del campo de la seguridad informática, debido al uso de Java como lenguaje de programación que además incluye la librería jnetpcap y debido a que el programa se encuentra modularizado, lo cual facilita el proceso de análisis y modificación. Además, CICFlowMeter permite la extracción desde archivos pcap y captura en vivo con un formato de salida como archivo csv, lo que facilita el proceso de análisis de resultados usando Python con sus respectivas librerías de aprendizaje de máquina. En la **figura 7** se muestra un diagrama a gran escala de la estructura del extractor.

D. Análisis del extractor

En esta etapa se realizó el análisis de las características del tráfico que se pueden implementar dentro del extractor y como es su modificación. Se encontró que el extractor implementa las estadísticas a partir de las propiedades del tráfico y a la vez utilizando una librería de Java llamada "org.apache.commons.math3.stat.descriptive.SummaryStatistics", la cual permite generar cálculos descriptivos de un conjunto de datos.

1) Análisis del funcionamiento

Posterior a la selección del extractor se realizó el estudio de este con el propósito de determinar el procedimiento a seguir para su modificación. CICFlowMeter utiliza la librería jnetpcap para capturar la información del tráfico con los atributos disponibles por parte de los protocolos establecidos en la comunicación. Posteriormente el módulo de lectura de paquetes almacena todos los paquetes que tengan atributos comunes de la 5-tupla que limita un flujo o que se encuentren dentro del rango de la ventana de tiempo. Con estos paquetes el módulo de generación de flujos crea un flujo bidireccional y el siguiente módulo calcula las características del tráfico. En la siguiente etapa se hacen los cálculos de las características resultantes de cálculos estadísticos y en el paso final se concatenan los datos en una cadena de caracteres y se añaden a una fila de un archivo csv. En el momento que termina el proceso de extracción el usuario cuenta con un archivo CSV con los flujos bidireccionales capturados. En la **figura 7** se presenta un esquema de funcionamiento del extractor.

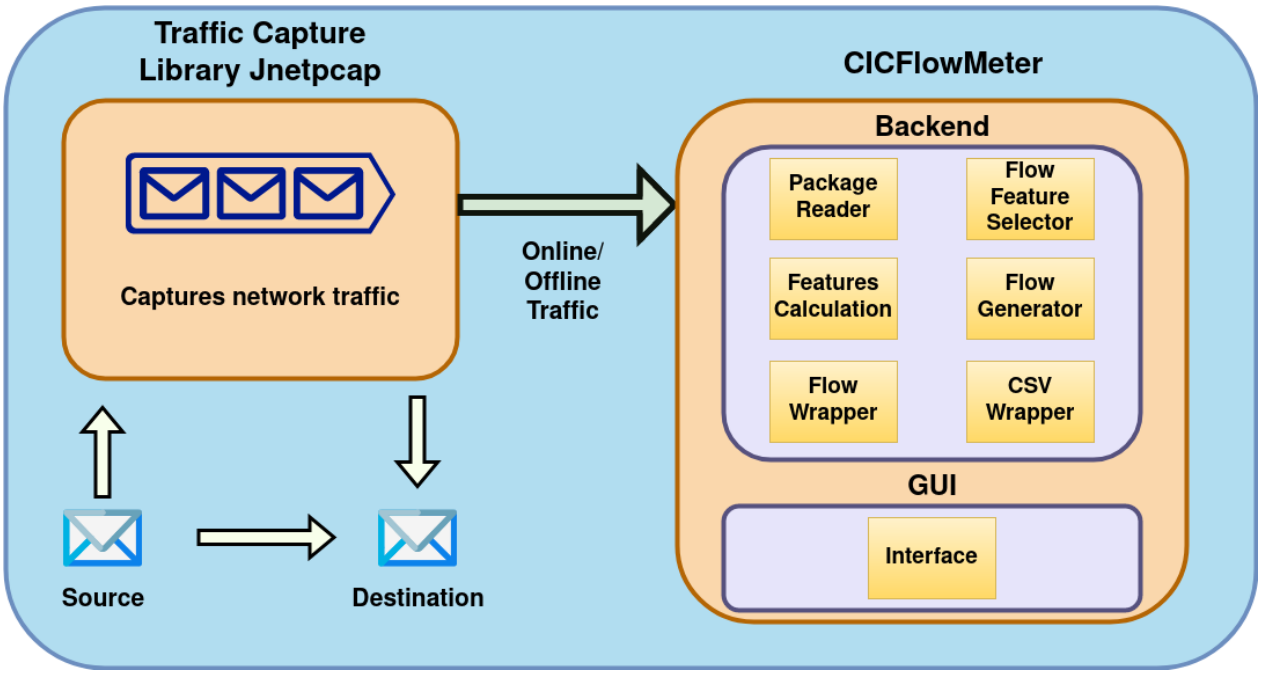

Figura 7. Resumen estructural del extractor CICFlowMeter.

2) Características del tráfico agregadas

De acuerdo con el estudio realizado sobre la importancia de las características estadísticas en el tráfico, a partir de una análisis de las estadísticas descriptivas que permiten describir un conjunto de datos y teniendo en cuenta lo encontrado en los artículos *Bayesian Neural Networks for Internet Traffic Classification* y *The significant Features of the UNSW-NB15 and the KDD99 Data Sets for Network Intrusion Detection Systems*, se decidió agregar estadísticas descriptivas y estadísticas que describen el tamaño de los paquetes y su relación con el uso que se le está dando dentro de una red [42], [43].

Al extractor CICFlowMeter se agregaron 29 características del tráfico donde 24 características son descriptivas del comportamiento del tráfico para esto se usó la librería *Descriptive Statistic*s disponible en Java y 5 son cálculos determinados por la información capturada de la capa de aplicación por parte de Jnetpcap.

En la **tabla VII** se muestran 24 atributos añadidos correspondientes a propiedades estadísticas descriptivas de distintos parámetros del tráfico que son: Los paquetes en el sentido de avance y de regreso en los flujos unidireccionales, el tiempo de llegada entre los paquetes del flujo, el tiempo de llegada entre paquetes de avance y los paquetes de regreso, del flujo bidireccional formado y del tiempo en el cual la transmisión se encuentra en los estados de activo e inactivo. Para estos parámetros del flujo bidireccional generado, se agregaron estadísticas descriptivas como lo son: la desviación estándar relativa, la mediana y el rango intercuartil, ya que estas estadísticas permiten tener un mayor conocimiento sobre el comportamiento del tráfico.

TABLA VII

CARACTERÍSTICAS DESCRIPTIVAS AGREGADAS AL EXTRACTOR

Características estimativas del tráfico
1. *Forward Packet Relative Standard Deviation*
2. *Forward Packet Median*
3. *Forward Packet IQR*
4. *Backward Packet Relative Standard Deviation*
5. *Backward Packet Median*
6. *Backward Packet IQR*
7. *Flow IAT Relative Standard Deviation*
8. *Flow IAT Median*
9. *Flow IAT IQR*
10. *Forward IAT Relative Standard Deviation*
11. *Forward IAT Median*
12. *Forward IAT IQR*
13. *Backward IAT Relative Standard Deviation*
14. *Backward IAT Median*
15. *Backward IAT IQR*
16. *Flow Relative Standard Deviation*
17. *Flow Median*
18. *Flow IQR*
19. *Flow Active Relative Standard Deviation*
20. *Flow Active Median*
21. *Flow Active IQR*
22. *Flow Idle Relative Standard Deviation*
23. *Flow Idle Median*
24. *Flow Idle IQR*

Las 5 características restantes que se añadieron al extractor corresponden a cálculos basados en los paquetes recolectados por la librería jnetpcap y que permiten hacer una mejor caracterización del tipo de tráfico que se está transmitiendo en la red. En la **tabla VIII** se presentan estos atributos.

TABLA VIII
CARACTERÍSTICAS AGREGADAS AL EXTRACTOR A NIVEL DE LA CAPA DE APLICACIÓN

Características de tráfico
25. *Total Packet Count in the bidirectional flow*
26. *Total Bytes Count in the bidirectional flow*
27. *Average Bytes per Packet in the flow*
28. *Average Bytes per Packet in the forward packets*
29. *Average Bytes per Packet in the backward packets*

E. Modificación del extractor CICFlowMeter

En el extractor se agregaron 29 características entre atributos descriptivos estadísticos que permiten incrementar la profundidad del análisis del tráfico y atributos a nivel de capa de red que permiten estudiar diferentes tipos de intrusiones. Para determinar el correcto funcionamiento del código incluido y del extractor en general se realizó una prueba capturando el tráfico de red de una red local utilizando un computador portátil con sistema operativo Linux . En la **figura 8** se muestra la captura de tráfico y se evidencia el correcto funcionamiento del extractor con los cálculos de las características agregadas y la creación de forma correcta del archivo de salida con los flujos [25], [26].

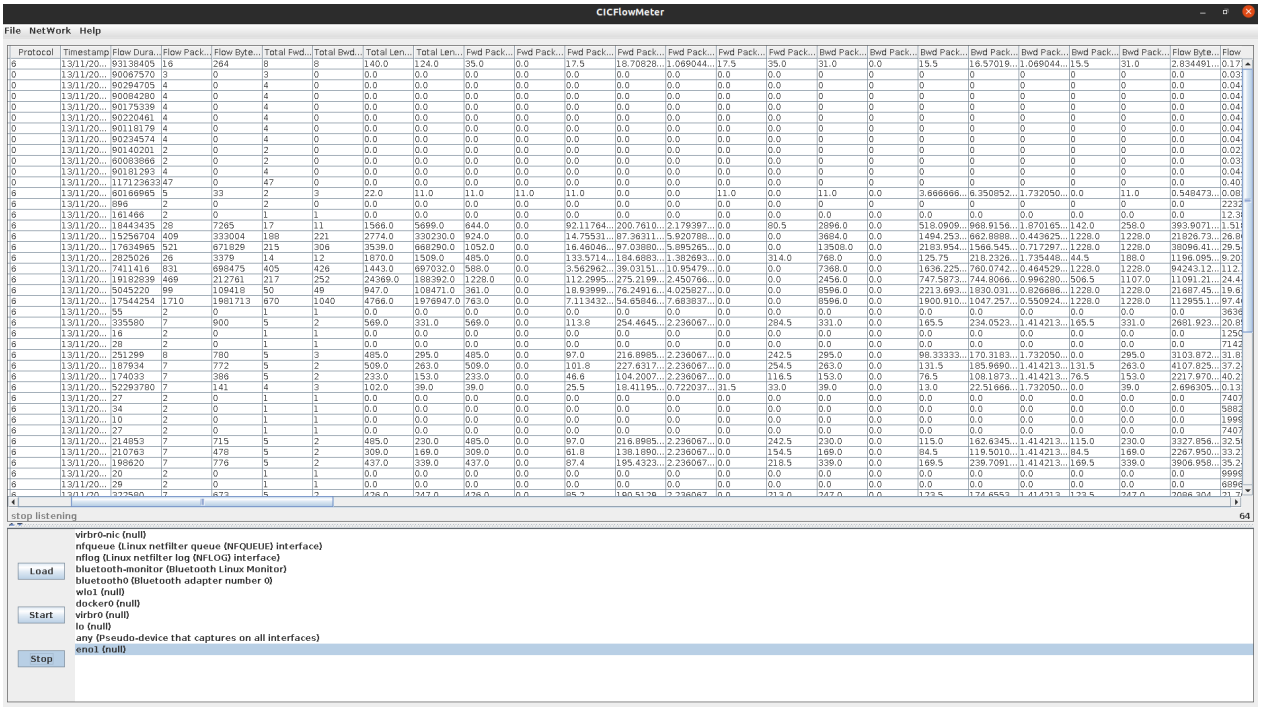

Figura 8. Extracción del tráfico capturando desde la interfaz de red ethernet.

Además, se realizó una prueba capturando los flujos bidireccionales desde un archivo pcap. En la **figura 9** se presenta el resultado de la extracción del tráfico perteneciente a la base de datos TRAbID, que está orientada al desarrollo de sistemas de detección de intrusiones y contiene tráfico normal, contenido de servicios conocidos, contenido de servicios normales y contenido de servicios para el tráfico anómalo [25], [26].

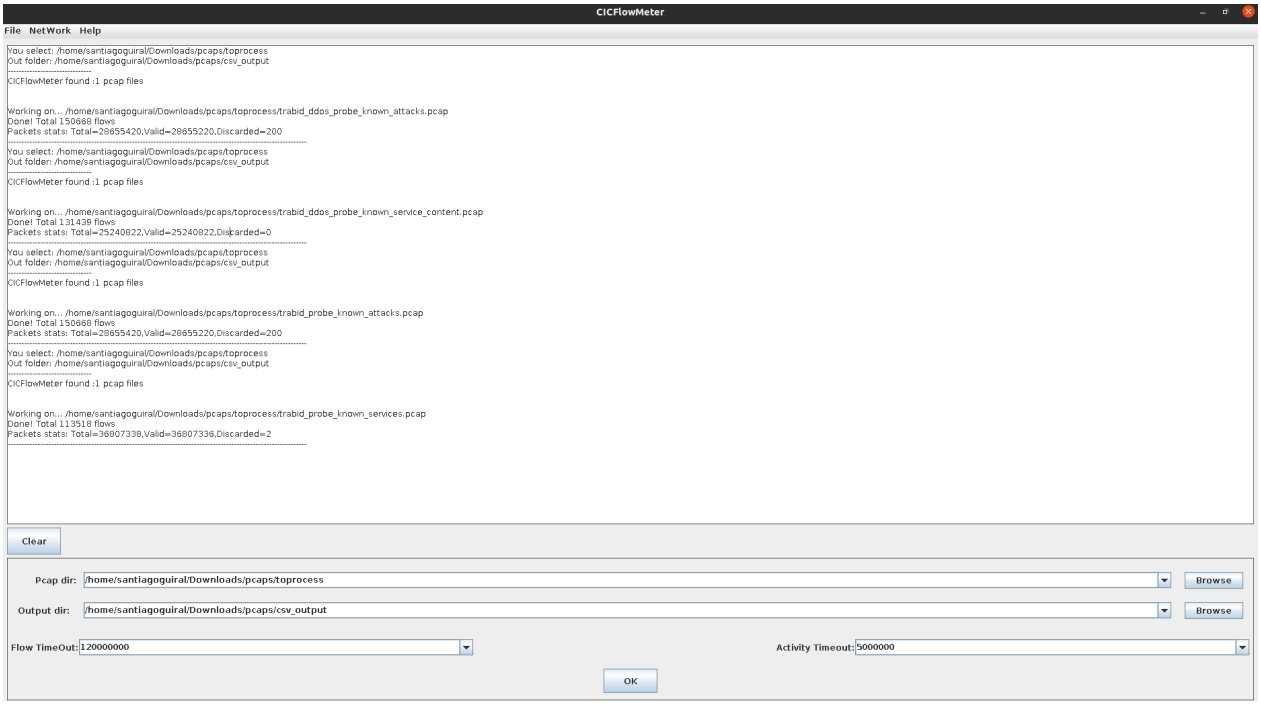

Figura 9. Extracción del tráfico mediante la lectura de archivos pcap.

F. Clasificación del tráfico de red

La clasificación del tráfico que fue extraído con CICFlowMeter se realizó haciendo usó de las bases de datos incluidas en la **tabla II**. También se hizo uso de Python, en particular de la librería Scikit-Learn la cual provee diferentes algoritmos de aprendizaje de máquina con la cual fue posible construir los modelos de clasificación. En aras de comparar el funcionamiento de las 2 versiones del extractor se utilizaron modelos de clasificación con el objetivo de medir el porcentaje de precisión en la detección y discriminación del tráfico normal y el anómalo.

1) CIC-Bell-DNS EXT

La base de datos *CIC-Bell-DNS Ext* contiene tráfico DNS (*Domain Name System*) obtenido de navegar por internet. Contiene tráfico normal y diferentes tipos de ataque de exfiltración. En la **tabla IX** se presenta la comparación de los extractores y se observa como la precisión mejora ligeramente para el extractor modificado con 4 de 5 algoritmos exceptuando el

de *Support Vector Machine*. De acuerdo con esto se puede decir que las nuevas características mantienen el nivel de precisión para el tráfico DNS y los ataques de exfiltración. De acuerdo con el análisis de esta tabla, la precisión puede incrementarse al agregar características que permitan caracterizar el tipo de ataques presente en esta base de datos. La precisión más alta se obtuvo con el algoritmo de *AdaBoost*.

TABLA IX

COMPARACIÓN DE LA PRECISIÓN CON LAS 2 VERSIONES DEL EXTRACTOR UTILIZANDO DIFERENTES ALGORITMOS DE CLASIFICACIÓN DE TRÁFICO EN LA BASE DE DATOS CIC-BELL-DNS EXT

Algoritmos	**Extractor Original**	**Extractor Modificado**
	Precisión (%)	**Precisión (%)**
AdaBoost (AB)	84.20	84.23
K Nearest Neighbor (KNN)	82.05	82.17
Linear Discriminant (LDA)	84.10	84.16
Random Forest (RM)	83.13	83.30
Support Vector Machine (SVM)	80.29	72.30

Además, en la **tabla X** se presenta la comparación de los 2 extractores cuando se evalúa con mayor profundidad el rendimiento de la clasificación. Se observa que la precisión para clasificar ataques y tráfico normal se mantiene constante para todos los algoritmos, excepto para *Support Vector Machine* cuando se hace la comparación de ambas versiones del extractor. También se observa que la sintonía y el puntaje F1 se mantiene constante en 4 de los 5 algoritmos y su valor es superior tanto para el tráfico anómalo como para el tráfico normal, lo que indica una buena relación de ataques detectados con respecto a la cantidad de muestras disponibles.

TABLA X

MEDIDAS DE RENDIMIENTO EN LA COMPARACIÓN DE LOS EXTRACTORES PARA LA BASE DE DATOS CIC-BELL-DNS EXT

Algoritmos		**Extractor Original**			**Extractor Modificado**		
		Precisión (%)	**Sintonía (%)**	**F1 (%)**	**Precisión (%)**	**Sintonía (%)**	**F1 (%)**
AB	Normal	99	56	71	99	56	71
	Ataque	81	100	89	81	100	89
KNN	Normal	79	67	72	79	66	72
	Ataque	83	90	87	83	91	87
LDA	Normal	99	55	71	99	55	71
	Ataque	80	100	89	80	100	89
RF	Normal	86	62	72	87	62	72
	Ataque	82	95	88	82	95	88
SVM	Normal	98	45	62	96	22	36
	Ataque	77	99	87	70	100	82

Nota: *AB - AdaBoost, KNN - K Nearest Neighbor, LDA - Linear Discriminant, RF - Random Forest, SVM - Support Vector Machine.*

En la **figura 10** se muestran las matrices de confusión para los algoritmos, en la columna de la izquierda se tiene el extractor original mientras que en la columna de la derecha el extractor modificado. En todos los casos se puede observar el incremento del número de ataques clasificados correctamente con la adición de las nuevas características en el extractor modificado, Asimismo se observa una constancia en 4 de los 5 algoritmos cuando se clasifica tráfico normal ya que se evita el incremento de tráfico normal clasificado como anómalo.

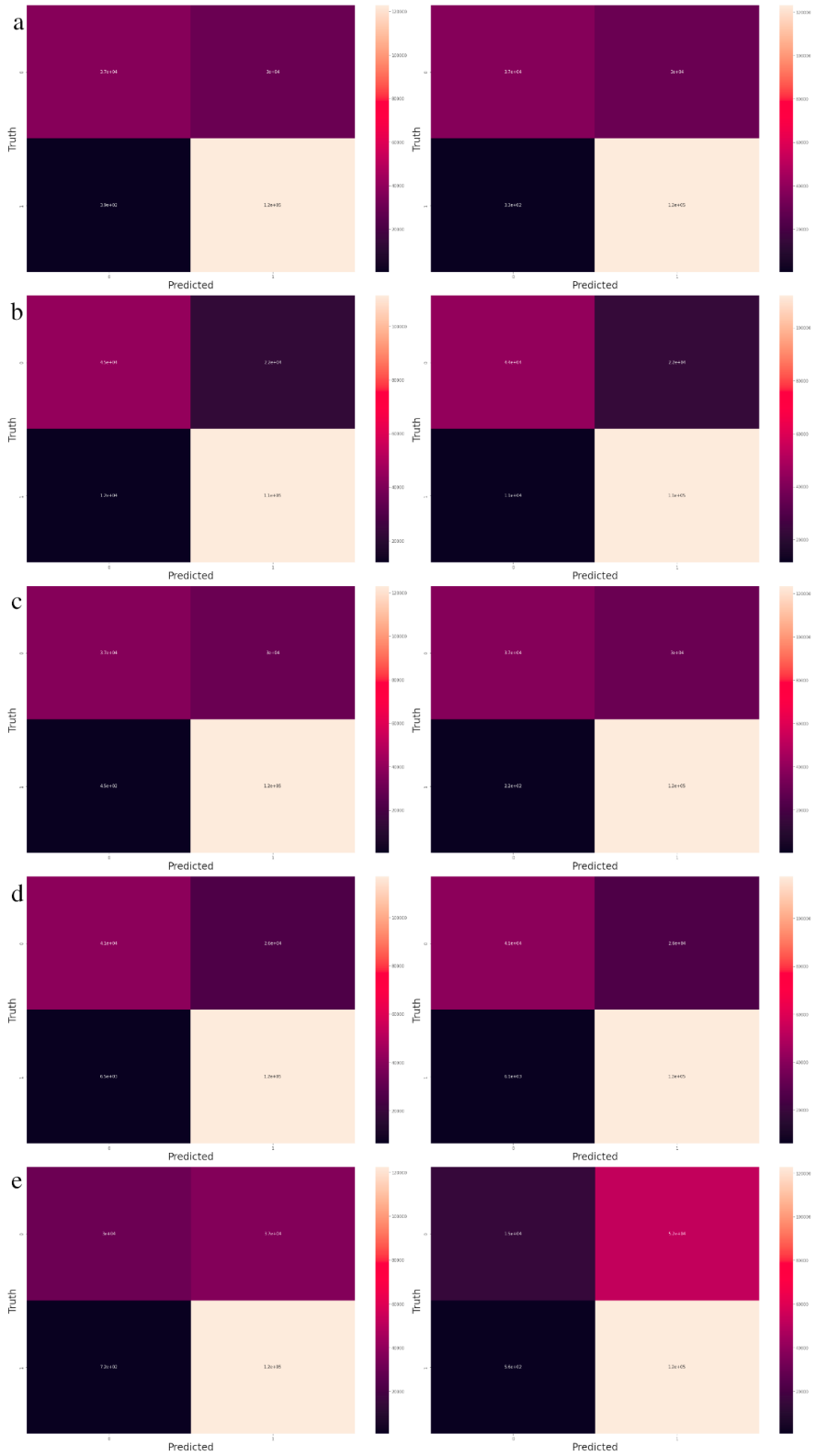

Figura 10. Matrices de confusión para CIC-Bell-DNS EXT. a) *AdaBoost.* b) *K Nearest Neighbor.* c) *Linear Discriminant.* d) *Random Forest.* e) *Support Vector Machine.*

2) CTU-13

La base de datos CTU-13 está compuesta por tráfico normal y por tráfico botnet, es decir, ataques que permiten el control remoto de los dispositivos de red. En la **tabla XI** se presenta la comparación de los extractores y se observa que todos los algoritmos exceptuando *Support Vector Machine* poseen un incremento en la precisión de la clasificación con el extractor modificado. También se destaca que los algoritmos *AdaBoost, K Nearest Neighbor* y *Random Forest* poseen una precisión por encima del 99% logrando que estos modelos son indicados para clasificar tráfico con la base de datos CTU-13. En cuanto al algoritmo *Support Vector Machine* se observa una baja precisión y al incrementar el número de características extraídas su precisión desciende y esto puede ser consecuencia del incremento de la dimensionalidad del espacio.

TABLA XI
COMPARACIÓN DE LA PRECISIÓN CON LAS 2 VERSIONES DEL EXTRACTOR UTILIZANDO DIFERENTES ALGORITMOS DE CLASIFICACIÓN DE TRÁFICO EN LA BASE DE DATOS CTU-13

Algoritmos	**Extractor Original**	**Extractor Modificado**
	Precisión (%)	**Precisión (%)**
AdaBoost (AB)	99.91	99.93
K Nearest Neighbor (KNN)	99.35	99.47
Linear Discriminant (LDA)	96.48	97.06
Random Forest (RM)	99.95	99.95
Support Vector Machine (SVM)	77.07	76.73

En la **tabla XII** se muestra más a fondo el rendimiento de los extractores. En todos los algoritmos se observa que para el extractor modificado con nuevas características la precisión incrementa o se mantiene constante cuando se hace la detección de ataques. Asimismo, en 3 de los 5 algoritmos exceptuando *Linear Discriminant* y *Support Vector Machine* la precisión en la detección de tráfico normal disminuye. Para las medidas de sintonía y el puntaje F1 hay mejoras

o los valores se mantienen constantes para 4 de los 5 algoritmos exceptuando *Support Vector Machine* cuando se ha realizado la modificación del extractor.

TABLA XII

MEDIDAS DE RENDIMIENTO EN LA COMPARACIÓN DE LOS EXTRACTORES PARA LA BASE DE DATOS CTU-13

Algoritmos		**Extractor Original**			**Extractor Modificado**		
		Precisión (%)	**Sintonía (%)**	**F1 (%)**	**Precisión (%)**	**Sintonía (%)**	**F1 (%)**
AB	Normal	100	100	100	100	100	100
	Ataque	100	100	100	100	100	100
KNN	Normal	99	99	99	99	99	99
	Ataque	99	100	99	100	100	100
LDA	Normal	100	91	95	99	93	96
	Ataque	95	100	97	96	100	98
RF	Normal	100	100	100	100	100	100
	Ataque	100	100	100	100	100	100
SVM	Normal	64	97	77	63	98	77
	Ataque	97	64	77	98	63	77

Nota: *AB - AdaBoost, KNN - K Nearest Neighbor, LDA - Linear Discriminant, RF - Random Forest, SVM - Support Vector Machine.*

En la **figura 11** se muestran las matrices de confusión para los algoritmos, en la columna de la izquierda se tiene el extractor original y en la columna de la derecha el extractor modificado. Se observa como la clasificación de tráfico normal mejora en todos los casos disminuyendo la clasificación de este como tráfico anómalo. Además, para el extractor que se ha modificado, se tiene una mejoría en la precisión para clasificar ataques con los algoritmos de *AdaBoost* y *K Nearest Neighbor*.

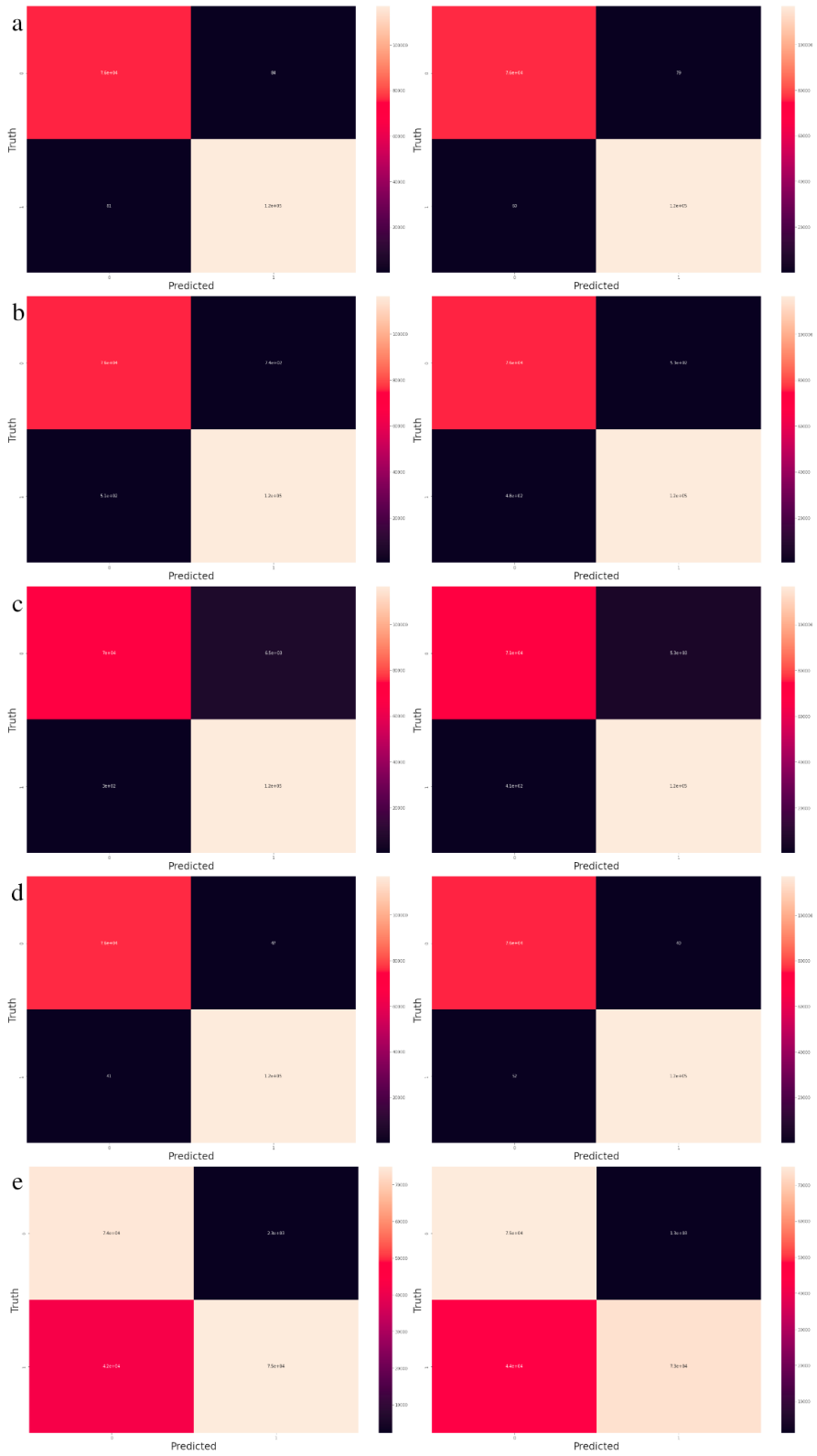

Figura 11. Matrices de confusión para CTU-13. a) *AdaBoost.* b) *K Nearest Neighbor.* c) *Linear Discriminant.* d) *Random Forest.* e) *Support Vector Machine.*

3) ISOT

La base de datos ISOT está orientada a la detección de tráfico botnet en una red. Los resultados de clasificación se mantienen uniformes con la implementación de los 5 algoritmos como se ve en la **tabla XIII**. Hay una ligera mejora con los modelos de *K Nearest Neighbor*, *Linear Discriminant* y Random Forest con las nuevas características implementadas, de lo contrario hay una ligera disminución de la precisión para los algoritmos de *AdaBoost* y *Support Vector Machine*. Por lo tanto, las nuevas características añadidas no son factores de gran influencia en los cambios en la precisión de la clasificación del tráfico. Por lo tanto, el algoritmo con mayor índice de precisión es *Random Forest* debido a la clasificación de acuerdo con las características extraídas del tráfico.

TABLA XIII
COMPARACIÓN DE LA PRECISIÓN CON LAS 2 VERSIONES DEL EXTRACTOR UTILIZANDO DIFERENTES ALGORITMOS DE CLASIFICACIÓN DE TRÁFICO EN LA BASE DE DATOS ISOT

Algoritmos	**Extractor Original**	**Extractor Modificado**
	Precisión (%)	**Precisión (%)**
AdaBoost (AB)	94.10	94.05
K Nearest Neighbor (KNN)	94.10	94.20
Linear Discriminant (LDA)	87.13	87.18
Random Forest (RM)	95.78	95.81
Support Vector Machine (SVM)	86.11	86.03

En la **tabla XIV** se presenta un análisis más profundo del rendimiento cuando se comparan los extractores. Se observa que la precisión para detectar ataques mejora o se mantiene igual con las nuevas características agregadas. También se tienen mejores resultados de rendimiento para la sintonía y el puntaje F1 con el extractor modificado tanto para el tráfico

normal como el anómalo y esto conlleva una menor cantidad de falsos positivos cuando se realiza la detección de ataques.

TABLA XIV

MEDIDAS DE RENDIMIENTO EN LA COMPARACIÓN DE LOS EXTRACTORES PARA LA BASE DE DATOS ISOT

Algoritmos		**Extractor Original**			**Extractor Modificado**		
		Precisión (%)	**Sintonía (%)**	**F1 (%)**	**Precisión (%)**	**Sintonía (%)**	**F1 (%)**
AB	Normal	83	81	82	84	80	82
	Ataque	96	97	96	96	97	96
KNN	Normal	82	83	83	83	83	83
	Ataque	96	96	96	97	97	97
LDA	Normal	69	43	53	68	44	54
	Ataque	89	96	93	90	96	93
RF	Normal	87	88	88	87	88	88
	Ataque	98	97	97	98	97	97
SVM	Normal	87	20	32	87	20	32
	Ataque	86	99	92	86	99	92

Nota: *AB - AdaBoost, KNN - K Nearest Neighbor, LDA - Linear Discriminant, RF - Random Forest, SVM - Support Vector Machine.*

En la **figura 12** se presentan las matrices de confusión para los algoritmos implementados. En la columna izquierda se muestran los resultados del extractor original y en la columna derecha los resultados pertenecen al extractor modificado. De esta gráfica se nota la necesidad de incluir una mayor cantidad de muestras con tráfico normal con el objetivo de mejorar los valores de sintonía de la clasificación. Además, se observa una mejoría en la clasificación de ataques con los algoritmos *K Nearest Neighbor* y *Random Forest* y se mantiene constante para los algoritmos *AdaBoost, L*inear *Discriminant* y *Support Vector Machine*.

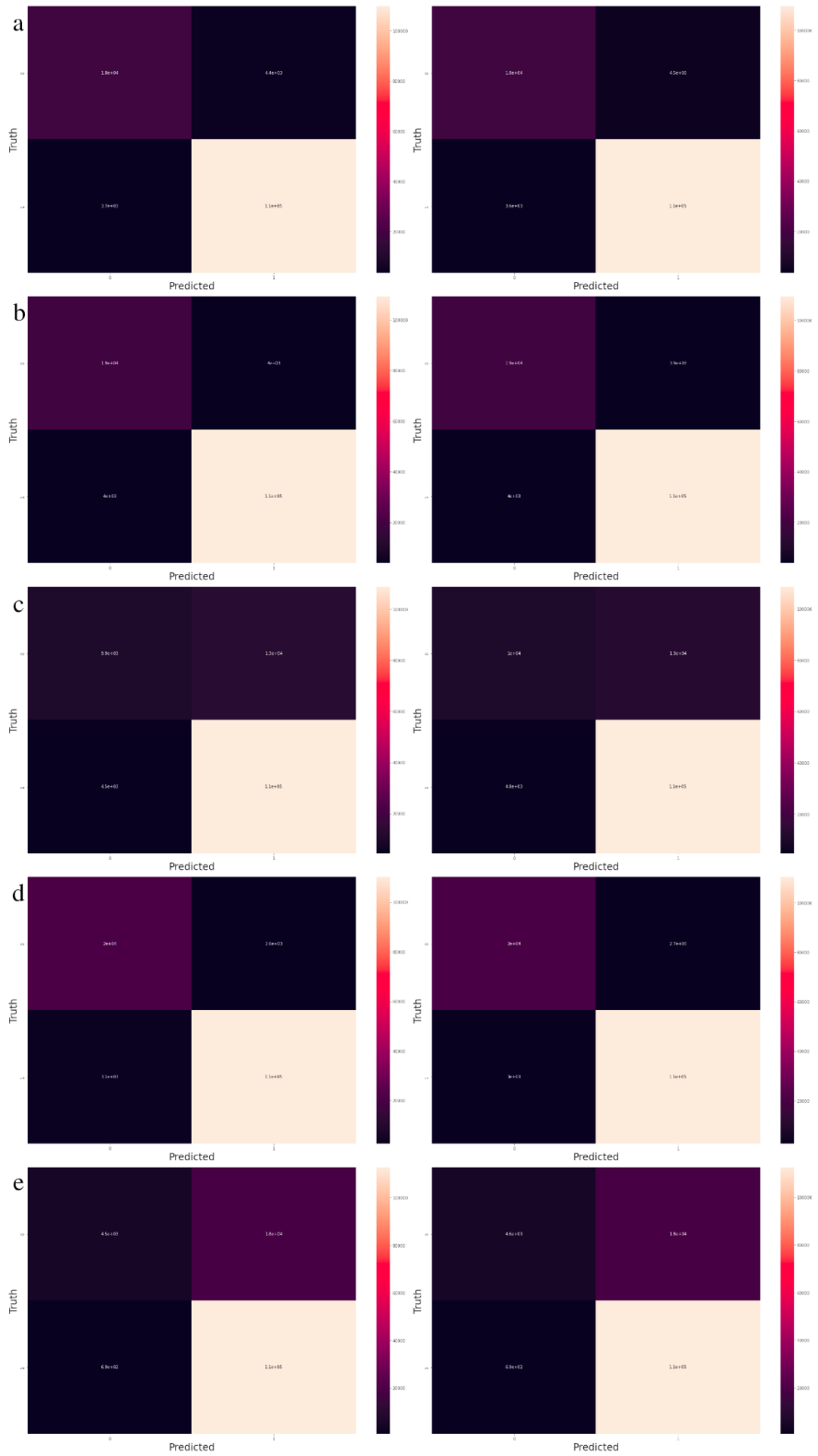

Figura 12. Matrices de confusión para ISOT. a) *AdaBoost.* b) *K Nearest Neighbor.* c) *Linear Discriminant.* d) *Random Forest.* e) *Support Vector Machine.*

4) TRAbID

La base de datos TRAbID está conformada por tráfico normal y ataques tipo *probing*, es decir tráfico que recopila la información sin consentimiento del propietario. En la **tabla XV** se presentan los resultados de la clasificación. Se observa que el extractor modificado presenta una mejoría cuando utiliza los algoritmos de *AdaBoost*, *Linear Discriminant* y *Random Forest* y se observa un ligero decremento cuando se utilizan los algoritmos de *K Nearest Neighbor* y *Support Vector Machine*, lo cual puede ser consecuencia de los conjuntos de datos utilizado para entrenar y evaluar el modelo. Por último, se destaca la mejoría superior a 2 puntos porcentuales de los algoritmos *AdaBoost* y *Random Forest*. Por ende, el mejor método de clasificación de acuerdo con los resultados obtenidos es *Random Forest*.

TABLA XV
COMPARACIÓN DE LA PRECISIÓN CON LAS 2 VERSIONES DEL EXTRACTOR UTILIZANDO DIFERENTES ALGORITMOS DE CLASIFICACIÓN DE TRÁFICO EN LA BASE DE DATOS TRABID

Algoritmos	**Extractor Original**	**Extractor Modificado**
	Precisión (%)	**Precisión (%)**
AdaBoost (AB)	77.60	79.84
K Nearest Neighbor (KNN)	71.93	71.81
Linear Discriminant (LDA)	72.54	73.30
Random Forest (RM)	77.91	80.44
Support Vector Machine (SVM)	71.67	71.45

En la **tabla XVI** se profundizan los resultados de la clasificación. Para el extractor modificado se tiene una disminución de falsos positivos en la clasificación de ataques y en 4 de los 5 algoritmos exceptuando el método de *Linear Discriminant* se observa una mejoría o como una constante en la precisión para la clasificación de los ataques. De igual forma, se obtienen mejores valores para la sintonía y el puntaje F1 en 4 de 5 algoritmos lo que permite mejorar el

modelo al aumentar la cantidad de muestras requeridas para entrenar y evaluar el modelo. Con respecto al tráfico normal en todos los 5 algoritmos se mantiene o se aumenta la precisión en esta categoría de clasificación para la nueva versión del extractor.

TABLA XVI

MEDIDAS DE RENDIMIENTO EN LA COMPARACIÓN DE LOS EXTRACTORES PARA LA BASE DE DATOS TRABID

Algoritmos		**Extractor Original**			**Extractor Modificado**		
		Precisión (%)	**Sintonía (%)**	**F1 (%)**	**Precisión (%)**	**Sintonía (%)**	**F1 (%)**
AB	Normal	70	83	76	74	83	78
	Ataque	85	74	79	86	78	81
KNN	Normal	65	75	70	65	76	70
	Ataque	79	69	74	79	69	74
LDA	Normal	62	94	75	63	92	75
	Ataque	93	56	70	91	59	72
RF	Normal	71	83	76	74	85	79
	Ataque	86	74	79	87	77	82
SVM	Normal	61	91	73	61	90	73
	Ataque	89	58	70	89	57	70

Nota: *AB - AdaBoost, KNN - K Nearest Neighbor, LDA - Linear Discriminant, RF - Random Forest, SVM - Support Vector Machine*

En la **figura 13** se presentan las matrices de confusión de los algoritmos implementados. En la columna izquierda para el extractor original y en la derecha para el extractor modificado. Se puede visualizar que los mejores métodos para clasificar ataques son los algoritmos de *AdaBoost*, *K Nearest Neighbor* y *Random Forest*. También se observan mejoras considerables con el método de *Random Forest* para clasificar tanto tráfico normal como anómalo.

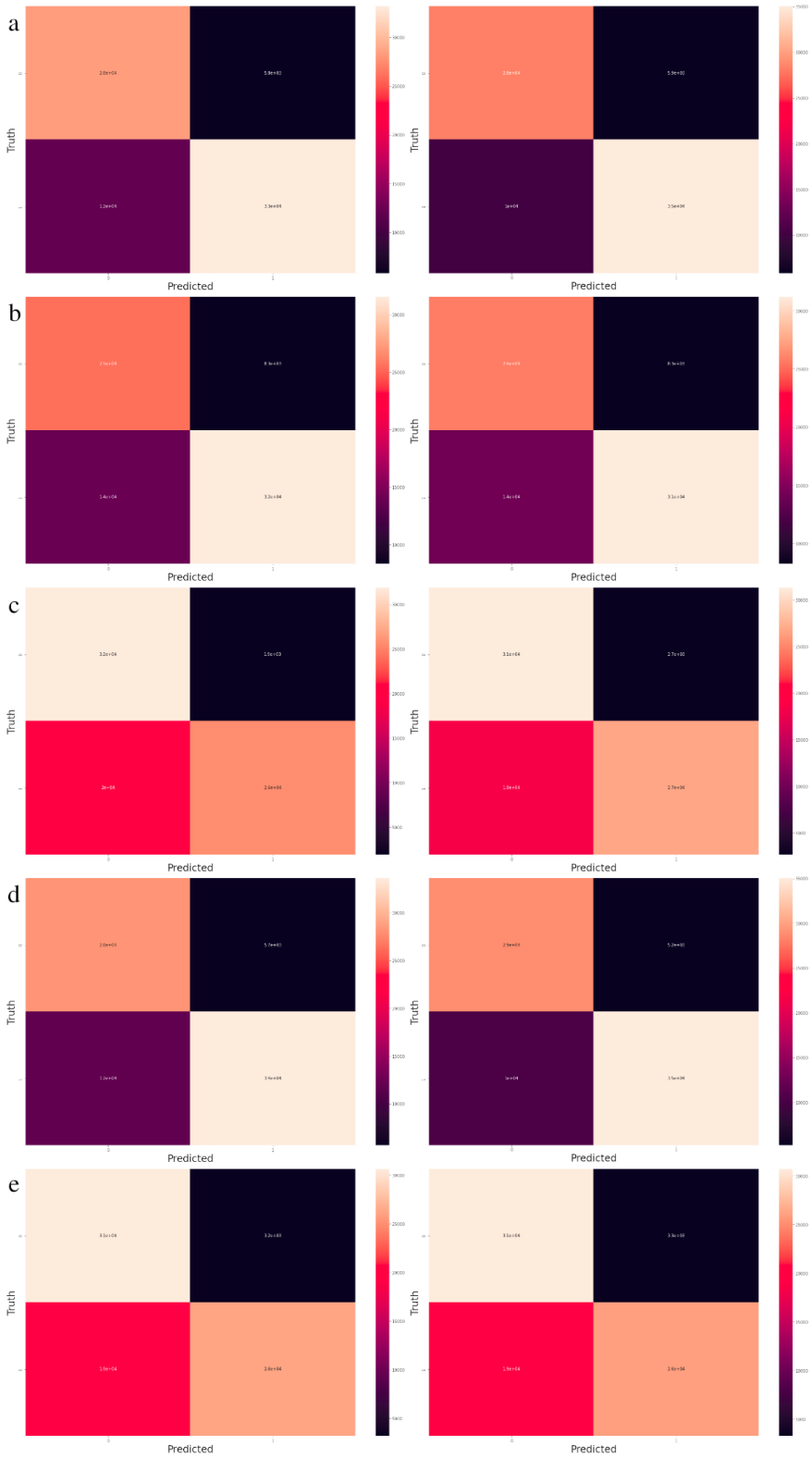

Figura 13. Matrices de confusión para TRABID. a) *AdaBoost.* b) *K Nearest Neighbor.* c) *Linear Discriminant.* d) *Random Forest.* e) *Support Vector Machine.*

5) TRAbID DDoS

Un segundo conjunto de datos perteneciente a la base de datos TRAbID se centra en los ataques DDoS, ataques de denegación de servicios donde un usuario legítimo no posee acceso a determinados servicios. En la **tabla XVII** se presentan los resultados de clasificar ataques tipo DDoS respecto al tráfico normal. En 4 de los 5 algoritmos exceptuando *Linear Discriminant* hay una mejoría en la precisión de la clasificación con el extractor que se modificó. En la evaluación con esta base de datos se tiene que el mejor algoritmo para clasificar es *Random Forest*.

TABLA XVII
COMPARACIÓN DE LA PRECISIÓN CON LAS 2 VERSIONES DEL EXTRACTOR UTILIZANDO DIFERENTES ALGORITMOS DE CLASIFICACIÓN DE TRÁFICO EN LA BASE DE DATOS TRABID DDOS

Algoritmos	**Extractor Original**	**Extractor Modificado**
	Precisión (%)	**Precisión (%)**
AdaBoost (AB)	77.80	80.23
K Nearest Neighbor (KNN)	72.26	73.60
Linear Discriminant (LDA)	72.80	71.95
Random Forest (RM)	77.90	80.75
Support Vector Machine (SVM)	71.86	71.87

Cuando se hace un análisis más profundo de los resultados utilizando la **tabla XVIII**, se observa que los algoritmos con estimadores, es decir, *AdaBoost* y *Random Forest*, presentan un incremento en la precisión para clasificar los ataques y esto se debe a que al agregar características en el extractor modificado se tienen nuevas líneas de decisión lo que permite caracterizar de mejor manera el tráfico. También se tiene un decremento en la cantidad de falsos positivos en los ataques con el extractor modificado. En cuanto a las medidas de sintonía y el puntaje F1 se tiene un incremento para el tráfico anómalo, lo cual indica que con más datos para entrenar y evaluar el modelo se puede tener mejores resultados de precisión.

TABLA XVIII

MEDIDAS DE RENDIMIENTO EN LA COMPARACIÓN DE LOS EXTRACTORES PARA LA BASE DE DATOS TRABID DDOS

Algoritmos		**Extractor Original**			**Extractor Modificado**		
		Precisión (%)	**Sintonía (%)**	**F1 (%)**	**Precisión (%)**	**Sintonía (%)**	**F1 (%)**
AB	Normal	70	83	76	74	84	78
	Ataque	85	74	79	86	78	82
KNN	Normal	65	76	70	65	75	70
	Ataque	79	70	74	79	70	74
LDA	Normal	62	94	75	63	91	75
	Ataque	93	57	70	90	60	72
RF	Normal	71	83	76	74	84	79
	Ataque	85	74	79	87	78	82
SVM	Normal	62	90	73	62	91	74
	Ataque	89	58	70	89	58	70

Nota: *AB - AdaBoost, KNN - K Nearest Neighbor, LDA - Linear Discriminant, RF - Random Forest, SVM - Support Vector Machine*

En la **figura 14** se presentan las matrices de confusión para los diferentes algoritmos de clasificación, en la columna de la izquierda se tiene el extractor original y a la derecha el extractor modificado. De esta gráfica se tiene una disminución de los falsos positivos cuando se clasifican los ataques con las características agregadas al extractor que se modificó. También se visualiza un incremento en la detección correcta de ataques con 4 de los 5 algoritmos a excepción de *K Nearest Neighbor*. Por lo tanto, se puede observar que hay una mejoría significativa con el extractor modificado cuando se hace la detección de ataques de tipo DDoS.

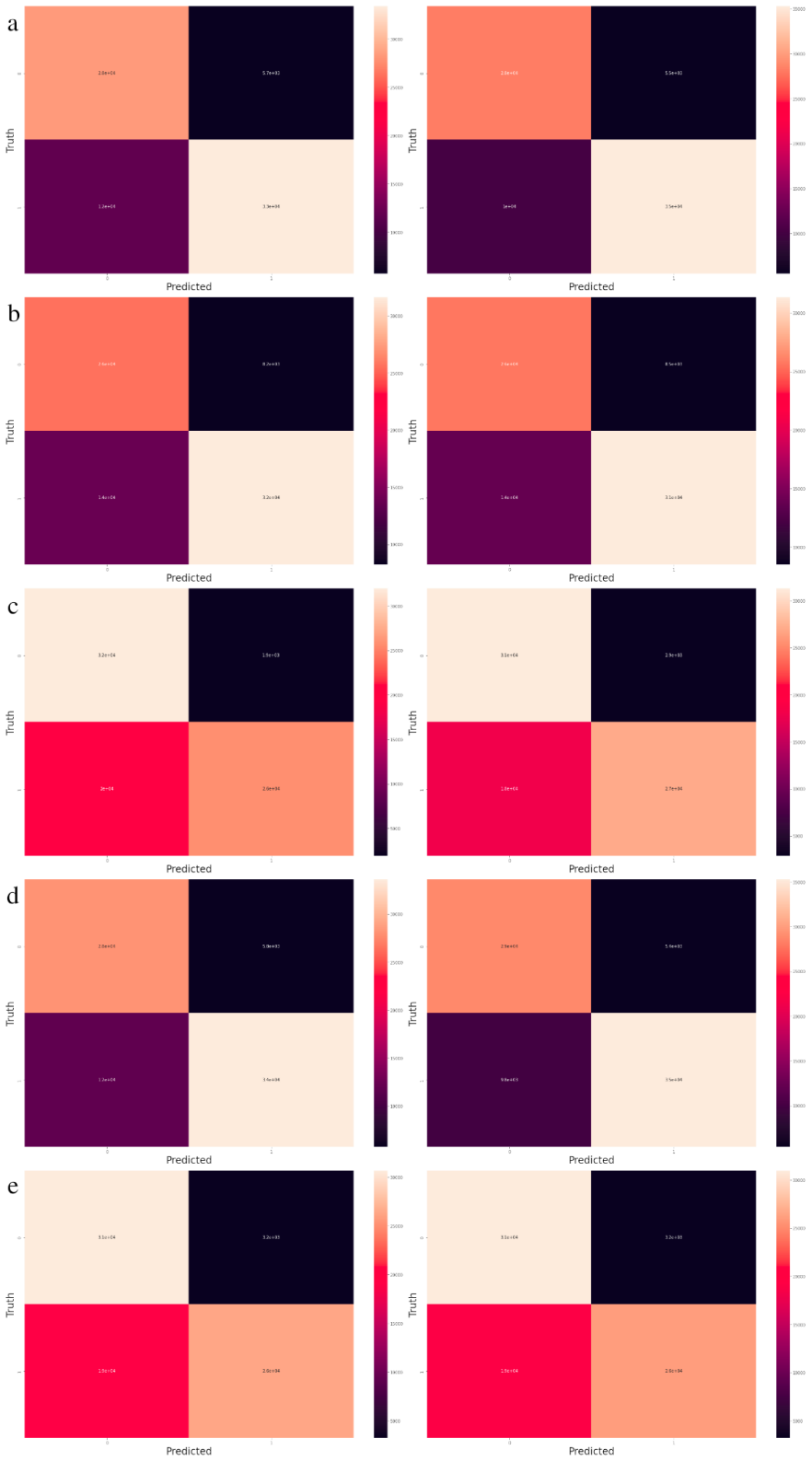

Figura 14. Matrices de confusión para TRABID DDoS. a) *AdaBoost.* b) *K Nearest Neighbor.* c) *Linear Discriminant.* d) *Random Forest.* e) *Support Vector Machine.*

VI. CONCLUSIONES

En este trabajo se presentó el estudio y el procedimiento necesario para modificar un extractor de tráfico de red con el objetivo de evaluar su rendimiento en la clasificación del tráfico normal y anómalo. El proceso está distribuido en 5 etapas: la selección de las bases de tráfico con las que se harán pruebas de funcionamiento, la selección del extractor a modificar, determinar qué cambios se van a realizar, implementar la modificación y por último realizar las pruebas de clasificación de tráfico.

Durante el proceso de selección de la base de datos se destaca la importancia de una escogencia de estas ya que es imperativo contar con una gran variedad de tráfico que contenga tráfico normal como diferentes tipos de ataques y de esta forma agregar robustez a los sistemas de clasificación. En referencia al estudio de extractores, se encontraron diferentes alcances en el tráfico capturado y para propósitos de este trabajo, el extractor debe ser orientado a la clasificación, por lo tanto, se escogió el extractor CICFlowMeter para su modificación. Este extractor captura tráfico en vivo y mediante lectura de archivos pcap y como salida genera flujos de tráfico que son almacenados en un archivo csv. Se escogió este extractor para su modificación por su versatilidad y estructuración, la cual permite agregar y capturar nuevas propiedades del tráfico mediante la implementación de funciones. Se agregaron 29 características para extraer y se evidencia que estas cambian los resultados de clasificación ya que se tienen nuevas maneras de caracterizar y discriminar el tráfico normal con respecto a los ataques informáticos.

Además, de acuerdo con los resultados obtenidos y el análisis realizado la modificación del extractor cambia los valores de precisión en la clasificación. Las características agregadas durante el proceso de modificación del extractor permiten incrementar el nivel de precisión para detectar ataques y se destaca una mejoría con los algoritmos que utilizan estimadores, ya que al agregar nuevas características se tienen nuevas formas para tomar decisiones que clasifican el tráfico. En general se tiene una mejoría para clasificar el tráfico con el uso de los algoritmos *Random Forest Classifier, K Nearest Neighbor* y *Linear Discriminant Analysis*. Asimismo, las

nuevas características permiten clasificar de mejor manera los ataques de tipo DDoS y *probing* dado que algunas de las características se centran en la frecuencia y tamaño de los paquetes en la transmisión. Por lo tanto, las propiedades del tráfico a capturar juegan un papel importante en el proceso de detección de tráfico anómalo. Cuando se agregan nuevas características incrementa la capacidad de los sistemas de mejorar la clasificación de tráfico y estas nuevas propiedades deben ser agregadas teniendo en cuenta el tipo de tráfico que se va a capturar y su relación con los ataques a detectar con el objetivo de incrementar el grado de precisión en la clasificación.

VII. REFERENCIAS BIBLIOGRÁFICAS

[1] Posgrados Ibero, México, "¿Qué es la seguridad informática?". [En línea]. 2020. Disponible en: https://blog.posgrados.ibero.mx/seguridad-informatica/ [Accedido 14 Dic. 2021]

[2] M. Ring, S. Wunderlich, D. Scheuring, D. Landes and A Hotho, "A Survey of Network-based Intrusion Detection Data Sets". En Computers & Security, Vol. 86, pp. 147-167, 2019, DOI: 10.1016/j.cose.2019.06.005

[3] R. Alshamy and M. Ghurab, "A review of big data in network intrusion detection system: Challenges, approaches, datasets, and tools", En International Journal of Computer Sciences and Engineering, Vol. 8, Issue 7, 2020.

[4] R. Boutaba, M. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano and M. Caicedo, "A comprehensive survey on machine learning for networking: evolution, applications, and research opportunities", En Journal of Internet Services and Applications, 2018, DOI: https://doi.org/10.1186/s13174-018-0087-2

[5] G. Sourek and F. Zelezny, "Efficient extraction of network event types from NetFlows", Security and Communication Networks, Vol. 2019, pp. 1-18, Article ID 8954914, 2019, DOI: https://doi.org/10.1155/2019/8954914

[6] Cisco IOS, "NetFlow Version 9 Flow-Record Format". [En línea]. 2019. Disponible en: https://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.html [Accedido 14 Dic. 2021]

[7] C. Garcia-Cordero, E. Vasilomanolakis, A Wainakh, M. Mühlhäuser and S. Nadjm-Tehrani, "On generating network traffic datasets with synthetic attacks for intrusion detection", En ACM Transactions on Privacy and Security Vol. 8, pp. 1-39, 2021, DOI: https://doi.org/10.1145/3424155

[8] S. Abt and H. Baier, "Are we missing labels? A study of the availability of ground-truth in network security research". Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security, 2014, DOI: 10.1109/BADGERS.2014.11

[9] L. Mor, "What is computer security?". Simplilearn, March 10, 2021. [En línea]. Disponible en: https://www.simplilearn.com/what-is-computer-security-article [Accedido 14 Dic. 2021]

[10] Huawei. "Differences between IDS and IPS". [En línea]. 2018. Disponible en: https://forum.huawei.com/enterprise/en/differences-between-ids-and-ips/thread/484497-867 [Accedido 14 Dic. 2021]

[11] S. Anwar, J. Zain, M. Zolkipli, Z. Inayat, S. Khan, B. Anthony and V. Chang, "From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions". En algorithms, Vol. 10, no. 2, pp. 39, 2017, DOI: https://doi.org/10.3390/a10020039

[12] R. Jindal, and A. Anwar, "Emerging trends of recently published datasets for intrusion detection systems (IDS): A survey. Cornell University". [En línea]. 2021. Disponible en: arXiv:2110.00773

[13] J. Brownlee, "Supervised and unsupervised machine learning algorithms". Machine Learning Mastery. March 16, 2016. [En línea]. 2016. Disponible en: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ [Accedido 14 Dic. 2021]

[14] Scikit Learn. "An AdaBoost Classifier". [En línea]. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html [Accedido 14 Dic. 2021]

[15] Scikit Learn. "K-nearest neighbors classifier". [En línea]. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html [Accedido 14 Dic. 2021]

[16] Scikit Learn. "Linear Discriminant Analysis". [En línea]. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html [Accedido 14 Dic. 2021]

[17] Scikit Learn. "A Random Forest Classifier". [En línea]. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html [Accedido 14 Dic. 2021]

[18] Scikit Learn. "C Support Vector Classification". [En línea]. Disponible en: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html [Accedido 14 Dic. 2021]

[19] Canadian Institute of Cybersecurity. "CIC-Bell-DNS-EXF-2021 Dataset". [En línea]. 2019. Disponible en: https://www.unb.ca/cic/datasets/dns-exf-2021.html [Accedido 14 Dic. 2021]

[20] S. Mahdavifar, A. Hanafy Salem, P. Victor, M. Garzon, A. H. Razavi, N. Hellberg and A. Habibi Lashkari, "Lightweight hybrid data exfiltration using DNS based on machine learning", The 11th IEEE International Conference on Communication and Network Security (ICCNS), Dec. 3-5, 2021, Beijing Jiaotong University, Weihai, China.

[21] "The CTU-13 Dataset: A labeled Dataset with botnet, normal and background traffic". [En línea]. Disponible en: https://mcfp.felk.cvut.cz/publicDatasets/ [Accedido 14 Dic. 2021]

[22] S. Garcia, M. Grill, J. Stiborek and A. Zunino, "An empirical comparison of botnet detection methods", En Computers and Security Journal, Elsevier, Vol. 45, pp. 100-123,

2014, DOI: http://dx.doi.org/10.1016/j.cose.2014.05.011

[23] University of Victoria. "Botnet and ransomware detection datasets". [En línea]. 2010. Disponible en: https://www.uvic.ca/ecs/ece/isot/datasets/botnet-ransomware/index.php [Accedido 14 Dic. 2021]

[24] A. Alenazi, I. Traore, K. Ganame and I. Woungang, "Holistic Model for HTTP Botnet Detection Based on DNS Traffic Analysis", In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, Vol. 10618. Springer, Cham, 2017.

[25] Security & Privacy Laboratory, SecPLab. "Trabid Datasets". [En línea]. 2017. Disponible en: https://secplab.ppgia.pucpr.br/?q=trabid [Accedido 14 Dic. 2021]

[26] E. K. Viegas, A. O Santin and L. S. Oliveira, "Toward a reliable anomaly-based intrusion detection in real-world environments", En Computer Networks. Vol. 127, pp. 200-216, 2017, DOI: 10.1016/j.comnet.2017.08.013

[27] Canadian Institute for Cybersecurity. "Applications: CICFlowMeter (formerly ISCXFlowMeter)". [En línea]. 2016. Disponible en: https://www.unb.ca/cic/research/applications.html [Accedido 14 Dic. 2021]

[28] A. Habibi Lashkari, "CICFlowMeter Version 4.0.", GitHub. [En línea]. 2021. Disponible en: https://github.com/ahlashkari/CICFlowMeter [Accedido 14 Dic. 2021]

[29] B. Yamansavascilar, "Network-Traffic-Classification --- Feature-Extraction". GitHub. [En línea]. 2017. Disponible en: https://github.com/Anamort/Network-Traffic-Classification---Feature-Extraction. [Accedido 14 Dic. 2021]

[30] R. Bikmukhamedov, "ML-based network traffic classifier". GitHub. [En línea]. 2021. Disponible en: https://github.com/RadionBik/ML-based-network-traffic-classifier. [Accedido 14 Dic. 2021]

[31] CISCO. "Joy". GitHub. [En línea]. 2019. Disponible en: https://github.com/cisco/joy.

[Accedido 14 Dic. 2021]

[32] Y. Mirsky, "Kitsune.py feature extractor". GitHub. [En línea]. 2020. Disponible en: https://github.com/ymirsky/Kitsune-py/blob/master/FeatureExtractor.py [Accedido 14 Dic. 2021]

[33] Wand Network Research Group. "Libprotoident". GitHub. [En línea]. 2021. Disponible en: https://github.com/wanduow/libprotoident [Accedido 14 Dic. 2021]

[34] Cybersecurity and Infrastructure Security Agency. "Malcom". GitHub. [En línea]. 2021. Disponible en: https://github.com/cisagov/Malcolm [Accedido 14 Dic. 2021]

[35] Nfstream. "Nfstream: Flexible network data analysis framework". GitHub. [En línea]. 2021. Disponible en: https://github.com/nfstream/nfstream [Accedido 14 Dic. 2021]

[36] Nfstream. "Nfstream: Getting started". [En línea]. 2021. Disponible en: https://www.nfstream.org/docs/ [Accedido 14 Dic. 2021]

[37] Nprint. "Nprint". GitHub. [En línea]. 2021. Disponible en: https://github.com/nprint/nprint [Accedido 14 Dic. 2021]

[38] L. Ward, "OpenFPC". GitHub. [En línea]. 2021. Disponible en: https://github.com/leonward/OpenFPC [Accedido 14 Dic. 2021]

[39] D. De Sensi, "Peafowl". GitHub. [En línea]. 2021. Disponible en: https://github.com/DanieleDeSensi/peafowl [Accedido 14 Dic. 2021]

[40] G. Draper-Gil, A. Habibi Lashkari, M. Mamun and A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features". Proceedings of the 2nd International Conference on Information Systems Security and Privacy - ICISSP. SBN 978-989-758-167-0; ISSN 2184-4356. pp. 407-414, 2016, DOI: http://dx.doi.org/10.5220/0005740704070414

[41] A. Habibi Lashkari, G. Draper-Gil, M. Mamun and A. Ghorbani, "Characterization of Tor traffic using time based features". In Proceedings of the 2nd International Conference on Information System Security and Privacy - ICISSP. ISBN 978-989-758-209-7; ISSN

2184-4356, pp. 253-262, 2017, DOI: 10.5220/0006105602530262

[42] T. Auld, W. Moore and S. F. Gull, “Bayesian neural networks for internet traffic classification”. IEEE Transactions on Neural Networks. Vol. 18, no. (1), pp. 223-239, 2017, DOI: 10.1109/TNN.2006.883010

[43] N. Mustafa and J Slay, “The significance features of the UNSW-NB15 and the KDD99 data sets for network intrusion detection systems”. En 2015 4th International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security (BADGERS), 2015, DOI: https://doi.org/10.1109/BADGERS.2015.014