UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA DE COMPUTACIÓN

OPTIMIZACIÓN DE UNA RED LAN DESPUÉS DE UN ATAQUE DDOS DETECTADO CON TÉCNICAS DE INTELIGENCIA ARTIFICIAL

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Ciencias de la Computación

AUTOR: JORGE SANTIAGO VIZCAINO TAIPE
TUTOR: DIEGO FERNANDO VALLEJO HUANGA

Quito - Ecuador
2022

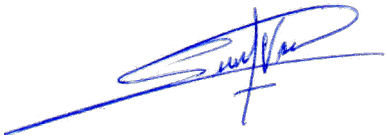
CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, Jorge Santiago Vizcaino Taipe, con documento de identificación N° 1720218575, manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 14 de marzo del año 2022

Atentamente,



.....
Jorge Santiago Vizcaino Taipe
1720218575

CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA

Yo, Jorge Santiago Vizcaino Taipe, con documento de identificación N° 1720218575, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Artículo Académico: "Optimización De Una Red Lan Después De Un Ataque Ddos Detectado Con Técnicas De Inteligencia Artificial", el cual ha sido desarrollado para optar por el título de: ingeniero en ciencias de la computación, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, 14 de marzo del año 2022

Atentamente,

.....
Jorge Santiago Vizcaino Taipe
1720218575

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo Diego Fernando Vallejo Huanga con documento de identificación N° 1720162708, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: OPTIMIZACIÓN DE UNA RED LAN DESPUÉS DE UN ATAQUE DDOS DETECTADO CON TÉCNICAS DE INTELIGENCIA ARTIFICIAL, realizado por Jorge Santiago Vizcaino Taipe con documento de identificación N° 1720218575, obteniendo como resultado final el trabajo de titulación bajo la opción Artículo Académico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 14 de marzo del año 2022

Atentamente,

.....
Ing. Diego Fernando Vallejo Huanga, MSc
1720162708

Optimización de una Red LAN después de un Ataque DDoS detectado con Técnicas de Inteligencia Artificial.

1st Santiago Vizcaino-Taipe
jvizcainot@est.ups.edu.ec

2rd Diego Vallejo-Huanga
dvallejoh@ups.edu.ec

***Resumen*—El ataque de Denegación de Servicios Distribuidos (DDoS) es uno de los ciberataques más peligrosos en el Internet, ya que puede afectar a cualquier servidor en cualquier tipo de red, causando problemas de conectividad e incluso la pérdida total de los servicios. El desarrollo del *Machine Learning* ha permitido resolver problemas computacionales de seguridad y es frecuentemente utilizado para la defensa contra ciberataques. En este artículo se propone la construcción de una topología de red, en la cual se aplican varios ataques de DDoS, que posteriormente serán detectados por tres algoritmos de clasificación de *Machine Learning*. A partir de la recolección de datos circulantes en la red se obtuvo un *dataset*, con muestras de tráfico normal y paquetes de tipo malicioso, sobre el cual se realizaron las pruebas experimentales. En la tarea de clasificación, el algoritmo de aprendizaje supervisado con el mejor rendimiento fue *Random Forest*, con un *accuracy* del 100%. Finalmente, cuando se ha detectado que la red se encuentra bajo un ataque DDoS, se aplica el algoritmo de optimización de Dijkstra para encontrar una ruta alternativa que permita mitigar la sobresaturación de la red. Se plantearon dos escenarios, el primero que analiza la ruta óptima en una red bajo ataque y otra sin afectación. Los resultados muestran que la red se reconfigura para evitar las rutas donde se aplicó la detección de ataques DDoS.**

***Palabras Clave*—Ciberseguridad, Detección de Tráfico Anómalo, Aprendizaje de Máquina, Random Forest, Máquina de Soporte Vectorial, Regresión Logística, Algoritmo de Dijkstra.**

***Abstract*—The Distributed Denial of Service (DDoS) attack is one of the most dangerous cyberattacks on the Internet, so can affect any server on any type of network, causing connectivity problems and even total loss of services. Machine learning can solve computational security problems and is frequently used to defend against cyber attacks. This article proposes the construction of a network topology where several DDoS attacks were applied, which will be detected by three Machine Learning classification algorithms. A dataset was generated from the collection of packets circulating in the network with samples of normal traffic and malicious packets, on which the experimental tests were carried out. In the classification task, the best performing supervised learning algorithm was Random Forest, with a scoring accuracy of 100%. Finally, upon detecting a DDoS attack on the network, Dijkstra's optimization algorithm is applied to find an alternative route to mitigate network oversaturation. Two scenarios were proposed, the first analyzes the optimal route in an attacked network and the second without attacks. The results show reconfiguration in the network to avoid routes where DDoS attack detection was applied.**

***Keywords*—Cybersecurity, Malicious Traffic Detection, Machine Learning, Random Forest, Support Vector Machine, Logistic Regression, Dijkstra Algorithm**

I. Introducción

Un ataque de denegación de servicio distribuido (DDoS, por sus siglas en ingles) es un ciberataque que usa varios dispositivos conectados a Internet que han sido infectados previamente para su uso malicioso. Estos dispositivos mandan millones de peticiones por segundo hacia un objetivo víctima, el cual queda deshabilitado y por lo tanto el servicio que presta se ve denegado, causando pérdidas millonarias en las organizaciones que son víctimas de estos ataques [1]. El *Machine Learning* (ML) es una subdisciplina de la Computación, que ha tenido un gran desarrollo en las últimas décadas y ha desarrollado una cantidad enorme de algoritmos para resolver problemas de diferente índole [2].

A nivel mundial los ciberataques se producen con una frecuencia de uno cada 39 segundos y los datos muestran que existe una falta de conocimientos básicos de seguridad informática en los usuarios de los sistemas [3]. Un empleado de una empresa puede desconocer por completo el hecho de que él o su sistema corren el riesgo de ser atacados. En forma genérica los empleados desconocen la premisa de que cuanto más dependen de la tecnología, más vulnerables son a un ataque [4].

Los datos comerciales de corporaciones, entidades financieras, hospitales y gobiernos se almacenan en computadoras o en servidores almacenados en la nube. El auge del Internet y las tecnologías de comunicación, han propiciado el aumento de transmisión de datos por la red con lo cual se ven expuestos a los peligros que existen en ella. Por lo tanto, es fundamental proteger la información y los datos de los ciberataques para mantener los datos con un mayor nivel de seguridad [5].

En los países de América Latina y el Caribe, la magnitud de los ciberataques pasó desapercibido en diversas organizaciones. Debido a la pandemia provocada por el COVID-19 y el aumento de la actividad digital que se ha generado en la región, se puso de manifiesto las vulnerabilidades del espacio digital en América Latina y el Caribe [6].

Según el Índice Global de Ciberseguridad (IGC), de la Unión Internacional de Telecomunicaciones (UIT) de julio del 2021, las cinco naciones más preparadas para contrarrestar un ciberataque en el mundo son: Singapur, Estados Unidos, Malasia, Omán y Estonia. Estos índices que publica la UIT

se construyen según aspectos legales, técnicos, organizativos, capacidad de creación y cooperación, de los 193 países miembros. El compromiso con la ciberseguridad en América Latina está por debajo del 0.70%. Para los casos de Brasil y Uruguay este índice oscila entre el 0.60% y 0.70%, Colombia 0.58% y para Ecuador entre el 0.35% y 0.47% [7] [8].

El presente proyecto de investigación tiene por objetivo generar un modelo que detecte el tráfico anómalo para un ataque DDoS y a-posteriori permita un proceso de optimización de la red mediante técnicas de ML.

A. Trabajos Relacionados

La seguridad en la red es uno de los factores mas importantes a considerar en tópicos relacionados con Internet. La detección de ataques DDoS durante mucho tiempo fue una tarea onerosa, debido a la ingente cantidad de datos que circulan por la red. Con el desarrollo de ML se han generado varios algoritmos de detección, mitigación y defensa, que han sido capaces de combatir los ataques DDoS [9]. Varios autores han desarrollado técnicas para la identificación de ataques DDoS usando algoritmos de ML, sin embargo, algunas soluciones no contemplan los efectos adversos del ataque y la posibilidad de revertirlos.

Peneti et al. [10] usaron el *dataset* CICIDS2017 para construir un sistema inteligente de detección que alerte a los administradores de red cuando existan peticiones maliciosas. Con la premisa de que las técnicas de ML hacen uso eficiente de los datos almacenados, su investigación se enfocó en construir un sistema que pueda estar preparado para detectar nuevos tipos de ataques DDoS. Su objetivo principal fue clasificar un ataque malicioso cualquiera, utilizando tres técnicas de ML: *Random Forest*, *Multilayer Perceptron Model* (MLP) y *AdaBoost*. Los experimentos arrojaron un rendimiento en *F1 Score* de 1, 0.98 y 1, respectivamente, para cada algoritmo.

El Sistema de Detección de Intrusión en la Red (NIDS, por sus siglas en inglés) creado por Das et al. [11] pretende detectar ataques DDoS, dentro y fuera de la red. Su propuesta combina diferentes clasificadores en un solo conjunto heterogéneo, donde cada clasificador puede etiquetar diferentes tipos de intrusiones para conseguir una defensa robusta. El Sistema de Detección de Intrusiones (IDS, por sus siglas en inglés) fue diseñado usando un conjunto de aprendizaje, con varios algoritmos de ML, combinándolos en un sistema unificado. El IDS fue capaz de clasificar 125679 instancias correctamente con un 99.1% de *accuracy*.

Fouladi et al. [12], utilizaron el clasificador *Naive Bayes* (NB) con dos frecuencias basadas en la Transformada Discreta de Fourier (DFT, por sus siglas en inglés) y la Transformada Discreta de Wavelet (DWT, por sus siglas en inglés) para discriminar el tráfico normal del malicioso. Este trabajo considera que todas las características del *dataset* son independientes entre ellas. El conjunto de datos utilizado para el experimento tiene 1936 registros normales y 2649 registros maliciosos. El estudio concluyó que el clasificador NB es el más rápido y fácil de implementar. El *accuracy* para la tarea de clasificación fue de 85.32%, mientras que usando el algoritmo NB junto con DFT + DWT fue de 95.93%.

Kiruthika et al. [13] proponen un Sistema de Monitoreo en Línea (OMS, por sus siglas en inglés), compuesto por un algoritmo de Conteo de Saltos, HCF, junto con una Máquina de Soporte Vectorial (SVM). El sistema fue entrenado durante varios meses, donde se observó a los usuarios dentro de la red, con el fin de construir el perfil de cada uno. Sus variaciones en el número de saltos y sus estadísticas son guardadas para crear un promedio para esa dirección IP. El *dataset* de entrenamiento consta de 3000 instancias de las cuales 2000 son ataques y 1000 corresponden a tráfico normal. El algoritmo SVM fue entrenado con la dirección IP de origen y su respectivo promedio de número de saltos, el *accuracy* obtenido con este enfoque fue de 98.99%.

Por otro lado, el estudio de [14] se enfoca en Redes Definidas por Software (SDN, por sus siglas en inglés), dado que varias organizaciones y negocios están desplegando su infraestructura en ambientes digitales, que atraen a ciberdelincuentes. Las SDN son las redes que más se están desarrollando en la actualidad y por esta razón el volumen de los ataques de DDoS sobre redes SDN, desde el 2016 hasta la fecha se han incrementado. Este trabajo utiliza el *dataset* CICIDS2017 y determina que las variables más latentes para el algoritmo predictivo de ML son: i) La longitud total de los paquetes *backward*, ii) el tamaño medio de los paquetes, iii) la longitud total de los paquetes *forward*, iv) los paquetes *backward* por segundo y v) la duración del flujo.

En este sentido, la revisión de la literatura científica y trabajos relacionados muestran que la mayoría no consideran, en el proceso de detección de un ataque DDoS, varios aspectos de las redes de comunicación en un sistema complejo. Estos sistemas al ejecutar un cambio, pueden tardar varios días, para que los cambios se vean reflejados en el flujo de paquetes [15]. ML proporciona técnicas eficientes para adaptarse al comportamiento de la red, en el presente estudio se implementó una arquitectura que proporciona seguimiento y análisis de datos, lo que es fundamental para poder realizar la optimización automática de la red.

II. Materiales y Métodos

Para la propuesta de optimización de una red LAN después de un ataque DDoS, detectado con técnicas de Inteligencia Artificial (IA), se desarrolló un modelo enfocado en la detección de anomalías. La arquitectura propuesta es el resultado de un proceso continuo de diseño, instalación, configuración, pruebas de conexión, rendimiento de máquinas virtuales y software.

A. Generación del Dataset

La generación del conjunto de datos tiene como objetivo la recopilación de tráfico normal y anómalo que servirán para la construcción de los modelos de ML. Se trabajó sobre una topología de red LAN simulada por *software*, en donde se aplicaron herramientas automatizadas para el envió de los ataques DDoS. Este *dataset* es de dominio público, desarrollado

explícitamente con fines educativos y puede ser consumido desde https://github.com/santiagovizcaino/ddos_optimization_project.

1) Topología de red: Para la adquisición de datos de tráfico normal y anómalo se desplegó una red LAN que cuenta con *routers* y varios dispositivos finales que apuntan hacia un servidor que brinda el servicio HTTP. El tráfico que circula dentro de la red, será usado para la construcción del *dataset*. La implementación de la topología física, que se muestra en la Figura 1, fue implementada en el emulador de entornos de red GNS3.

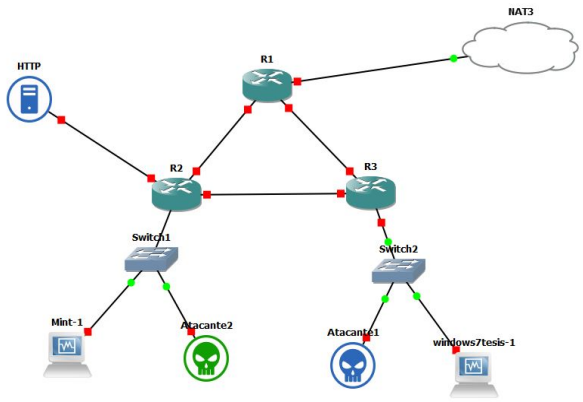


Fig. 1: Diagrama de topología física de la red LAN para la generación del *dataset*

GNS3 es un programa de código abierto que permite ejecutar, emular, configurar y probar virtualmente topologías pequeñas. Está compuesta por una interfaz gráfica y una máquina virtual [16].

Una de las características que posee GNS3 es la posibilidad de conectar máquinas virtuales, generadas en *Virtual Box* con la topología de red. Esto último permite tener un ambiente de pruebas más cercano a la realidad. En la Tabla I se resumen las características técnicas, operativas y funcionales de las máquinas virtuales que forman parte de la red LAN.

Sistema Operativo	Distribución	Nombre
Linux	Centos 7	HTTP
Linux	Mint	Contabilidad
Linux	Kali 2020	Atacante 1
Linux	Kali 2021	Atacante 2
Windows	7	Ventas

TABLA I: Características de las máquinas virtuales de la red LAN

2) Recolección de datos: Las características de flujo de tráfico en la red LAN, son monitoreadas y capturadas mediante la herramienta de software *Wireshark*, que es un analizador de protocolos, creado por Gerald Combs en 1998. Esta herramienta permite visualizar los componentes de la red en tiempo real y su comportamiento [17].

Para la extracción de las características más significativas que poseen los paquetes, tanto de tráfico malicioso como normal, se tomó como referencia los atributos usados en investigaciones relacionadas con ML y ataques DDoS. En la Tabla II se observan los artículos científicos y las variables usadas en cada investigación para la clasificación de ataques DDoS.

Artículo	Variable
[18]	• Número de secuencia del paquete • Promedio del tamaño del paquete • Variación del tiempo de llegada del paquete • Variación del tamaño del paquete • Número de bytes • Velocidad del paquete
[19]	• Número de paquetes • Promedio del tamaño del paquete • Variación del intervalo de tiempo • Variación del tamaño del paquete • Número de bytes • Tasa de paquetes • Tasa de bits
[20]	• Tamaño del paquete • Delta time • Protocolo
[21]	• Tiempo medio de llegada entre paquetes • Probabilidad de ocurrencia de una IP por 15 segundos • Respuesta a registros de recursos • Registros de recursos de autoridad • Registros de recursos adicionales • Tamaño mínimo de paquete • Tamaño de paquete medio • Tamaño máximo de paquete
[18]	• Tiempo de llegada del paquete • IP de origen • IP de destino • Protocolo • Tamaño del paquete
[9]	• Promedio del tamaño del paquete • Número de paquetes • Variación del intervalo de tiempo • Variación del tamaño del paquete • Número de bytes • Tasa de paquetes • Tasa de bits
[22]	• Uso promedio de CPU • Promedio del tamaño de paquetes • Número de conexiones TCP

TABLA II: Variables características en la literatura científica relacionada con ataques DDoS

Para la captura de tráfico, bajo la topología propuesta, se monitoreo la red por un lapso de dos horas en cuatro días consecutivos. Los dos primeros días, se recolectó tráfico de la red cuando no se había generado ningún ataque y los siguientes dos días se recolectó tráfico en el momento que la red se encontraba bajo ataques DDoS. Los registros obtenidos durante los días de recolección fueron almacenados en archivos *pcap* y se encuentran disponibles en el repositorio de Github: https://github.com/santiagovizcaino/ddos_optimization_project.

Desde los ficheros *pcap* se generaron dos archivos *CSV*. Un archivo *CSV* (A) donde se encuentran los registros creados a

partir de tráfico normal y otro (B) en el que se encuentran registros con tráfico malicioso. Los archivos A y B contienen cada uno 50 instancias seleccionadas de forma aleatoria.

3) Generación de tráfico malicioso: Desde los dispositivos finales, usados para la generación de ataques, se envió tráfico malicioso hacia el servidor HTTP, usando tres herramientas: i) *Hping3* que es un software capaz de enviar paquetes ICMP, UDP, TCP y maneja fragmentación y tamaño aleatorio del paquete [23]. ii) *Low Orbit Ion Cannon* (LOIC) que es una aplicación de código abierto desarrollada por "Praetox Technologies" utilizada para pruebas de estrés de red y para generación de ataques DDoS. iii) *TCP SYN flood*, herramienta que se encuentra dentro del paquete de *Pentesting Metasploit* de Kali Linux y que utiliza el protocolo de conexión TCP de tres vías. La herramienta envía mensajes *synchronize* (SYN) al servidor sin esperar el mensaje de respuesta *acknowledge* (ACK) que establece la conexión entre el *host* y el servidor. Por lo tanto, el envío de paquetes por parte del atacante es continuo y sin interrupciones, lo que permite causar saturación en la red [24].

4) Generación de tráfico normal: Para la construcción del conjunto de datos se empleó un entorno simulado, con el objetivo de tener control total sobre la red LAN. De esta manera se puede monitorear los períodos de tiempo en donde el *host* está siendo atacado y capturar diferentes variables que corresponden a características de tráfico.

Las variables obtenidas durante el monitoreo de la red LAN, y que son compartidas tanto por el tráfico malicioso como por el tráfico normal se describen en la Tabla III.

Variable	**Descripción**
id	Número identificador del registro
time	Describe el tiempo en el cual el paquete llegó a destino.
source	Dirección IP de donde se origina el paquete.
destination	Dirección IP a donde se dirige el paquete.
protocol	Normativas que siguen los paquetes para establecer comunicación entre origen y destino
length	Longitud del paquete medido en bytes.
timedelta	El valor delta se calcula encontrando la diferencia de tiempo entre dos paquetes consecutivos que fluyen en una red.
srcport	Es el número de puerto desde el cual salió el paquete.
dstport	Es el número de puerto a donde debe llegar el paquete.
ack	Número aleatorio de confirmación

TABLA III: Descripción de las variables del conjunto de datos

B. Pre-procesamiento de los datos

Una vez que se han seleccionado las características más discriminantes entre tráfico normal y tráfico malicioso, se utilizó pre-procesamiento de los datos para el análisis y transformación del *dataset*.

De todas las variables capturadas en el conjunto de datos, algunas fueron descartadas porque no son latentes para el modelo de clasificación de ML o porque simplemente son identificadores de instancias. Las variables descartadas son *id*, *source*, *destination*, *protocol*, *srport* y *ack*. Entonces, en los archivos *CSV* A y B se eliminan las características descartadas y se añade una columna de etiquetas *type* con el propósito de identificar cada registro como malicioso o normal según corresponda. Así, el conjunto de datos final contiene cinco variables, cuatro variables cuantitativas independientes (*time*, *timedelta*, *srcport*, *length*) y una variable cualitativa dependiente (*type*), que el algoritmo de ML clasificará.

Finalmente, se concatenaron los archivos A y B en uno solo (C), el resultado fue un dataset balanceado que consta de 100 instancias, con sus correspondientes etiquetas.

C. Configuración de los Algoritmos de ML

Los algoritmos de ML fueron entrenados a partir del *dataset* que tiene igual cardinalidad en las clases. El *dataset* al ser balanceado, evita que los algoritmos de ML favorezcan a una clase. El objetivo del algoritmo de ML es predecir la variable dependiente *type* (y) en función de las variables independientes: *time* (x_1), *length* (x_2), *timedelta* (x_3), *srcport* (x_4). Dado que la variable dependiente (y) es dicotómica los algoritmos de ML realizarán una tarea de clasificación, sobre el tráfico de red, determinando si un paquete es de tipo malicioso o normal.

La variable dependiente *type* posee únicamente dos estados, ergo, se realizó un proceso de codificación usando el método de variables ficticias. La codificación se efectuó con $n-1$ variables ficticias, donde n es la cantidad de estados que posee la variable cualitativa. La variable ficticia, solo puede tomar los valores booleanos 0 y 1, para que no exista penalizaciones a ningún estado. Así, las instancias cuyo valor original era "normal" se codificaron como un 1 y las instancias con etiqueta "ddos", fueron codificadas como 0.

Para todos los algoritmos de ML se realizó validación cruzada aleatoria, dividiendo el *dataset* en secciones de *test* y *training* con porcentajes de 30% y 70%, respectivamente. Se implementaron tres algoritmos de ML para la predicción de la variable dependiente *type*: i) Regresion Logística, ii) Random Forest y iii) Máquina de Soporte Vectorial.

1) Regresión Logística (RL): El algoritmo de RL es un modelo lineal para problemas de clasificación de una o más variables independientes y una variable dependiente [25]. El modelo utiliza una función sigmoide (Ec. 1) que restringe la salida en un intervalo de 0 y 1.

$$S(h) = \frac{1}{1+e^{-h}} \tag{1}$$

Esta salida es representada por la probabilidad p de que el evento suceda con éxito (1) o fracaso (0) Ec. (2) [26].

$$\widehat{y} = p(y = 1 \mid x_1, x_2, x_3, x_4) \tag{2}$$

Donde, x_1, x_2, x_3, x_4 son las variables independientes que se utilizaron para predecir la variable *type* y.

Entonces, el valor esperado de la variable y, que permite clasificar tráfico anómalo o normal, se define por la Ec. (3).

$$E(y) = \frac{e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\beta_3 x_3+\beta_4 x_4}}{1+e^{\beta_0+\beta_1 x_1+\beta_2 x_2+\beta_3 x_3+\beta_4 x_4}} \quad (3)$$

Ya que el modelo de clasificación es binomial y las salidas de la ecuación estimada de RL son probabilidades, $p \in [0, 1]$, es necesario discretizar la variable de salida a los dos posibles estados que puede tener la variable *type*. Para esto se usó un umbral de 0.5 para tomar la decisión de clase.

2) Random Forest (RF): El algoritmo RF es un clasificador de conjunto, que crea un bosque compuesto por varios árboles de decisión. Estos están formados por dos tipos de nodos: padres e hijos. El *root-nodo* es el nodo principal, de donde se dividen los demás nodos. Los *leaf-nodes*, se caracterizan por no tener divisiones y son los nodos finales en los que se almacena la decisión a la que ha llegado ese árbol en particular. La salida final de RF se basa en el voto mayoritario entre todo el conjunto de árboles [20].

El funcionamiento del algoritmo está dividido en dos etapas. Para el caso de estudio, en la primera etapa, el algoritmo genera 10 árboles de decisión y en la segunda etapa, para cada árbol se analiza el cumplimiento de la regla de clasificación dada al subconjunto aleatorio de variables predictoras. Como se muestra en la Figura 2, la característica *time* es la más relevante para el algoritmo y es el punto de partida para formar los árboles de decisión. La salida de este nodo evalúa el cumplimiento de la regla, si esta se cumple, el segundo punto de discriminación está dado por la variable *srcport*, la salida de este punto determina la clase a la cual pertenece la variable dependiente y.

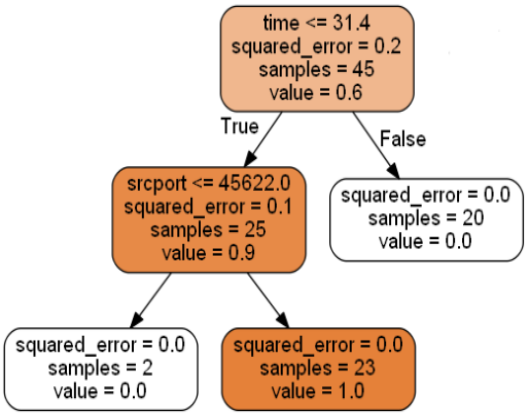


Fig. 2: Ramificación del árbol de decisión del algoritmo *Random Forest*

3) Máquina de Soporte Vectorial (SVM): La SVM es un técnica de ML que permite clasificar instancias, mediante la separación de clases con un hiperplano que maximiza las fronteras entre ellas y que utiliza una función de *kernel* para optimizar su proceso [27]. Los vectores de soporte permiten generan las fronteras de clases y determinan el límite de decisión [28]. En el caso de estudio, el modelo realiza la clasificación entre instancias de la clase normal y DDoS, el hiperplano se lo construye computacionalmente con un kernel RBF de la Ec. (4).

$$K(x_i, x_j) = \mathrm{e}\left(\frac{\|x_j - x_i\|^2}{\sigma^2}\right) \quad (4)$$

Donde σ, es el atributo que da la flexibilidad al modelo y tiene un valor de 1.0. La salida del algoritmo está dada por la distancia Euclidiana entre x_i y x_j. La clasificación de las instancias toma el valor de 0 para la clase anómalo y 1 para la normal.

D. Métricas de Evaluación de Rendimiento de los Algoritmos de ML

Los algoritmos de ML son entrenados con la partición de datos de *training* y evaluados en la partición de *test*. Entonces, mediante métricas de evaluación de ML, se procede a determinar el rendimiento de cada algoritmo con una matriz de confusión.

Se seleccionaron cuatro métricas para medir el rendimiento de los algoritmos de ML, estas son: i) *Accuracy* (Acc) que cuantifica todas las instancias clasificadas correctamente; ii) *Recall* (Re) que representa el porcentaje de casos positivos detectados por el modelo y mide los *True Positive* predichos correctamente de todas las instancias positivas reales [29]; iii) *Precision* (Pr) que cuantifica los *True Positive* predichos correctamente de todas las instancias positivas predichas; iV) *F1-score* (F1) que es el promedio armónico de la *Precision* y *Recall* [30]. Todas las métricas tienen un valor normalizado entre 0 y 1, siendo 0 el peor caso y 1 el mejor.

E. Algoritmo de Optimización de la Red LAN

Después de que un ataque ha sido detectado, por el algoritmo de ML, se realiza el proceso de optimización de la ruta para encontrar un nuevo enrutamiento, que disminuya la saturación de la red.

Dijkstra es un algoritmo que, mediante la representación en forma de grafo de una red, está diseñado para encontrar la ruta más corta desde un nodo origen a un nodo cualquiera. Este algoritmo toma en consideración los pesos asociados a los arcos del grafo, para encontrar la ruta con el mínimo coste. Además, tiene complejidad computacional $O(n^2)$, donde n es el número de nodos que forman el grafo [31].

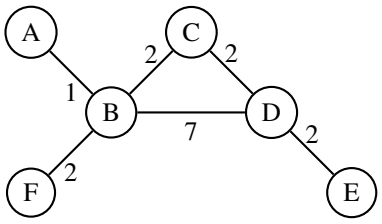


Fig. 3: Representación de la topología mediante grafo

Para aplicar el algoritmo de Dijkstra, se representó la topología mediante el grafo de la Figura 3. El objetivo es encontrar la ruta que tenga el coste mínimo entre los nodos de origen, que pueden ser (E) o (F) y el de destino (A). Los costes de los arcos están dados por la métrica *ancho de banda*

que es un valor asignado para indicar el peso de los enlaces dentro de la red.

Los valores de *ancho de banda* fueron asignados a los arcos dependiendo el tipo de puerto de enlace en el que están conectados. El coste de 1 fue asignado a los enlaces de tipo *Gigabit Ethernet* (GB), el coste 2 fue para los *Fast Ethernet* (FE) y en los *Ethernet* (E) se utilizó un coste de 7. Estas asignaciones son de tipo heurísticas ya que sus costes fueron determinados en función a la velocidad de transferencia de datos que cada enlace maneja, considerando que GB maneja velocidades de hasta 1000 Mbps, FE velocidades de hasta 100 Mbps y E tiene una velocidad de 10 Mbps.

El algoritmo de Dijkstra utiliza los costes de las rutas resumidas en la Tabla IV, entre todos los nodos que conforman el grafo.

	A	B	C	D	E	F
A	0	1	0	0	0	0
B	1	0	2	7	0	2
C	0	2	0	2	0	0
D	0	7	2	0	2	0
E	0	0	0	2	0	0
F	0	2	0	0	0	0

TABLA IV: Matriz de distancias

F. Recursos Computacionales para el Test Bed

Los experimentos se ejecutaron en un computador personal con procesador Intel Core i7-8750H, con sistema operativo Windows 10. Los *scripts* de desarrollo, para el *benchmark* de algoritmos se encuentran disponibles en el repositorio de Github: https://github.com/santiagovizcaino/ddos_optimization_project.

III. Resultados y Experimentos

Para cada uno de los algoritmos de *Machine Learning*, se obtuvo una matriz de confusión de los resultados, evaluada en la partición de datos de prueba. La matriz de confusión permite visualizar el desempeño que tuvieron los algoritmos al momento de realizar la tarea de clasificación. La clase positiva es nomenclada con la etiqueta 1 y representa al tráfico normal y la clase negativa es nomenclada con la etiqueta 0 y representa al tráfico en un ataque DDoS.

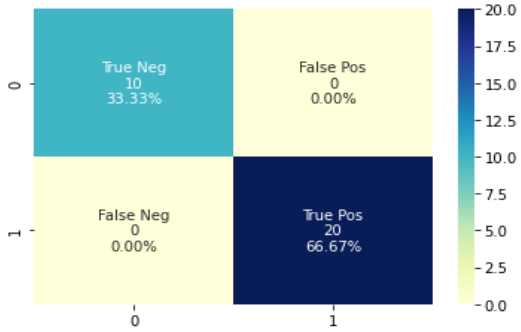


Fig. 4: Matriz de confusión Random Forest

La Figura 4 muestra los resultados del algoritmo de RF. Este algoritmo no presenta errores en la clasificación de las 100 instancias y tuvo un 0% tanto de falsos positivos como falsos negativos. En este sentido, todas las métricas de rendimiento del algoritmo supervisado son del 100%.

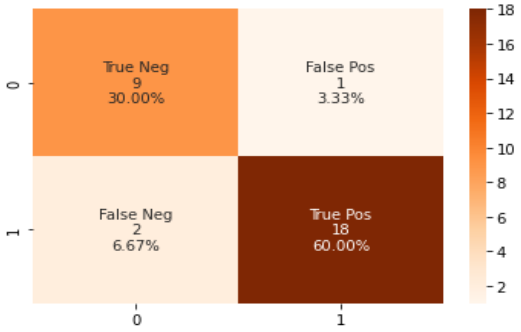


Fig. 5: Matriz de confusión Regresión Logística

Por otro lado, la Figura 5 muestra la matriz de confusión para el algoritmo de RL. En este caso se obtuvo el 10% de predicciones erróneas, i.e., que 10 observaciones fueron clasificadas de forma incorrecta. La tasa de falsos positivos fue de 3.33% y de falsos negativos del 6.67%. Esto último implica que el algoritmo tiende a clasificar de forma predominante a la clase anómala. El *accuracy* del algoritmo fue del 90%.

Finalmente, para el tercer algoritmo, en la matriz de confusión de la Figura 6 se muestra los resultados de clasificación de la SVM. Este algoritmo falló clasificando dos instancias anómalas como normales. El modelo en forma genérica tiene un rendimiento evaluado en términos de balance entre Pr y Re del 95%, medido por el *F1-score*.

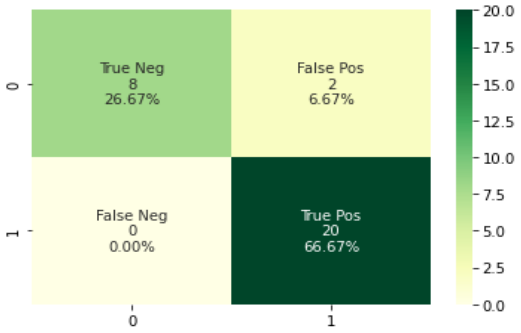


Fig. 6: Matriz de confusión SVM

En la Tabla V se detallan todas las métricas que evalúan el rendimiento de cada algoritmo en el proceso de clasificación entre tráfico normal y anómalo. De estos resultados se concluye que el mejor clasificador para la construcción del modelo de predicción es RF, seguido por SVM y RL.

Para probar la optimización del tráfico de red después de un ataque DDoS se configuraron dos escenarios en donde circulan diferentes tipos de tráfico por la red LAN.

1) Escenario I: En este escenario se utilizó el nodo (E) como *host* de origen para la comunicación hacia el servidor,

Algoritmo	Acc (%)	Pr (%)	Re(%)	F1 (%)
RF	1	1	1	1
RL	0.9	0.94	0.9	0.92
SVM	0.93	0.90	1	0.95

TABLA V: Métricas de Rendimiento de Algoritmos de ML

representado por el nodo (A). Durante todo el proceso de transferencia de datos entre origen y destino no hubo presencia de ataques DDoS. Por lo tanto, existen dos tramos por los cuales puede fluir la comunicación. El primer trayecto es entre los nodos: $(A-B)$, $(B-C)$, $(C-D)$ y $(D-E)$ y cuyos costes son de: 1, 2, 2 y 2, respectivamente, con un coste total para la ruta de 7. Por otro lado, la ruta formada por los nodos $(A-B)$, $(B-D)$ y $(D-E)$ tiene un coste total de 10, que esta calculado por la suma de sus tramos 1, 7 y 2, respectivamente. Bajo este escenario, el algoritmo de Dijkstra optimiza el tráfico en la red enviando los paquetes de comunicación por la ruta cuyo coste es de 7. En la Figura 7 se visualiza la aplicación de algoritmo de Dijkstra sobre la red LAN.

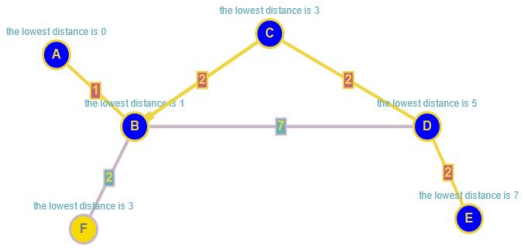


Fig. 7: Optimización de ruta con tráfico normal en la red

2) Escenario II: Para este escenario, se trabajó en el momento que la red se encontraba bajo ataques DDoS. Tras la detección de tráfico malicioso, por parte de los algoritmos de ML. Se redistribuye el flujo de red, penalizando los tramos comprendidos entre los nodos $(B-C)$ y $(C-D)$ con coste de 11 para cada uno. Esta métrica se obtiene debido a la disminución de velocidad de comunicación entre origen y destino, resultado de la saturación de red causado por los ataques DDoS. Al replicar la comunicación entre el nodo de origen (E) y el nodo de destino (A), la ruta, con coste de 7 por la cual transitaban los paquetes es ahora penalizada y tiene un coste de 25. Motivo por el cual, el algoritmo de Dijkstra recalcula los costes de red y arroja como resultado que la ruta óptima para la transferencia de paquetes está dada por los nodos $(A-B)$ $(B-D)$ y $(D-E)$. En la Figura 8 se muestra el grafo de la red y se visualiza como el algoritmo de Dijkstra evita los tramos penalizados.

IV. Conclusiones y Limitaciones

En el presente artículo se presentó una arquitectura de red sobre la cual se realizaron ataques de denegación de servicio distribuidos con el propósito de detectarlos por medio de

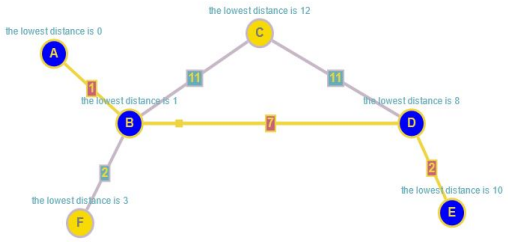


Fig. 8: Optimización de ruta con tráfico malicioso en la red bajo un ataque DDoS

algoritmos de ML. Los resultados mostraron que el mejor algoritmo para la clasificación de paquetes que circulan en la red es RF. La propuesta metodológica de este artículo también incluye la optimización de la red frente a un ataque DDoS, con el uso del algoritmo de Dijkstra, logrando obtener un camino emergente cuando la infraestructura de red se encuentra bajo un ataque de DDoS.

La detección de ataques DDoS, con algoritmos de ML, usando las variables contenidas dentro de la cabecera de los paquetes de red fue eficiente teniendo la mejor clasificación con una exactitud del 100%. Estos resultados no son determinantes debido a que las pruebas y experimentos se hicieron dentro de un ambiente controlado, con una topología de red pequeña y monitoreada en un corto lapso de tiempo. Sin embargo, los resultados muestran que la metodología planteada podría ser extrapolada hacia ambientes de producción con tráficos de red empresariales.

References

[1] K. Sonar and H. Upadhyay, "A survey: Ddos attack on internet of things," *International Journal of Engineering Research and Development*, vol. 10, no. 11, pp. 58–63, 2014.

[2] A. Pramod, H. S. Naicker, and A. K. Tyagi, "Machine learning and deep learning: Open issues and future research directions for the next 10 years," *Computational Analysis and Deep Learning for Medical Care: Principles, Methods, and Applications*, pp. 463–490, 2021.

[3] M. Cukier, "Study: Hackers attack every 39 seconds," *A. James Clark School of Engineering, University of Maryland*, 2007.

[4] R. Adlakha, S. Sharma, A. Rawat, and K. Sharma, "Cyber Security Goal's, Issue's, Categorization Data Breaches," in *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019*. Institute of Electrical and Electronics Engineers Inc., feb 2019, pp. 397–402.

[5] K. Sathya, J. Premalatha, and S. Suwathika, "Reinforcing Cyber World Security with Deep Learning Approaches," *Proceedings of the 2020 IEEE International Conference on Communication and Signal Processing, ICCSP 2020*, pp. 766–769, jul 2020.

[6] R. Mariano Díaz, "Cybersecurity in The Time of COVID-19 and The Transition to Cyberimmunity," dec 2020. [Online]. Available: https://repositorio.cepal.org/handle/11362/46511

[7] S. M. T. Toapanta, R. P. R. Pesantes, and L. E. M. Gallegos, "Impact of cybersecurity applied to IoT in public organizations in Latin America," *Proceedings of the World Conference on Smart Trends in Systems, Security and Sustainability, WS4 2020*, pp. 154–161, jul 2020.

[8] A. C. Ochoa Marcillo *et al.*, "Desafíos globales del cibercrimen: caso ecuador período 2014–2019," Master's thesis, Quito, EC: Universidad Andina Simón Bolívar, Sede Ecuador, 2021.

[9] R. R. Rejimol Robinson and C. Thomas, "Ranking of machine learning algorithms based on the performance in classifying ddos attacks," in *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, 2015, pp. 185–190.

[10] S. Peneti and H. E, "Ddos attack identification using machine learning techniques," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, 2021, pp. 1–5.

[11] S. Das, A. M. Mahfouz, D. Venugopal, and S. Shiva, "DDoS Intrusion Detection Through Machine Learning Ensemble; DDoS Intrusion Detection Through Machine Learning Ensemble," 2019.

[12] R. F. Fouladi, C. E. Kayatas, and E. Anarim, "Frequency based ddos attack detection approach using naive bayes classification," in *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, 2016, pp. 104–107.

[13] B. Kiruthika Devi, G. Preetha, G. Selvaram, and S. Mercy Shalinie, "An impact analysis: Real time ddos attack detection and mitigation using machine learning," in *2014 International Conference on Recent Trends in Information Technology*, 2014, pp. 1–7.

[14] A. D. Lopez, A. P. Mohan, S. Nair, A. Lopez, and A. Mohan, "Network Traffic Behavioral Analytics for Detection of DDoS Attacks," Tech. Rep. 1, 2019. [Online]. Available: https://scholar.smu.edu/datasciencereview/vol2/iss1/14

[15] D. Rafique and L. Velasco, "Machine learning for network automation: Overview, architecture, and applications [Invited Tutorial]," *Journal of Optical Communications and Networking*, vol. 10, no. 10, pp. D126–D143, oct 2018.

[16] GNS3 Staff, "Getting Started with GNS3 — GNS3 Documentation," 2020. [Online]. Available: https://docs.gns3.com/docs/

[17] W. Foundation, "Wireshark Â· Go deep." 2016. [Online]. Available: https://www.wireshark.org/#learnWShttps://www.wireshark.org/

[18] D. Peraković, M. Periša, I. Cvitić, and S. Husnjak, "Artificial neuron network implementation in detection and classification of ddos traffic," in *2016 24th Telecommunications Forum (TELFOR)*, 2016, pp. 1–4.

[19] C.-J. Hsieh and T.-Y. Chan, "Detection ddos attacks based on neural-network using apache spark," in *2016 International Conference on Applied System Innovation (ICASI)*, 2016, pp. 1–4.

[20] S. Shanmuga Priya, M. Sivaram, D. Yuvaraj, and A. Jayanthiladevi, "Machine Learning based DDOS Detection," *2020 International Conference on Emerging Smart Computing and Informatics, ESCI 2020*, pp. 234–237, mar 2020.

[21] K. J. T. Meitei, Irom Lalit, "Detection of ddos dns amplification attack using classification algorithm," in *Proceedings of the International Conference on Informatics and Analytics*, ser. ICIA-16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2980258.2980431

[22] T. Zhao, D. C.-T. Lo, and K. Qian, "A neural-network based ddos detection system using hadoop and hbase," in *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, 2015, pp. 1326–1331.

[23] Kali.org, "rainbowcrack — Kali Linux Tools," 2021. [Online]. Available: https://www.kali.org/tools/hping3/

[24] Imperva, "TCP SYN Flood." [Online]. Available: https://www.rapid7.com/db/modules/auxiliary/dos/tcp/synflood/ https://www.imperva.com/learn/ddos/syn-flood/

[25] X. Zou, Y. Hu, Z. Tian, and K. Shen, "Logistic regression model optimization and case analysis," in *2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT)*, 2019, pp. 135–139.

[26] Z. Li, H. Zhang, H. Shahriar, D. Lo, K. Qian, M. Whitman, and F. Wu, "Denial of service (dos) attack detection: Performance comparison of supervised machine learning algorithms," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2020, pp. 469–474.

[27] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, V. Vapnik *et al.*, "Support vector regression machines," *Advances in neural information processing systems*, vol. 9, pp. 155–161, 1997.

[28] M. De Cock, R. Dowsley, C. Horst, R. Katti, A. C. A. Nascimento, W.-S. Poon, and S. Truex, "Efficient and private scoring of decision trees, support vector machines and logistic regression models based on pre-computation," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 2, pp. 217–230, 2019.

[29] W. Nazih, Y. Hifny, W. S. Elkilani, H. Dhahri, and T. Abdelkader, "Countering ddos attacks in sip based voip networks using recurrent neural networks," *Sensors*, vol. 20, no. 20, p. 5875, 2020.

[30] A. Fernández, S. García, M. Galar, R. C. Prati, B. Krawczyk, and F. Herrera, *Learning from imbalanced data sets*. Springer, 2018, vol. 10.

[31] Y. Zhao and X. Zhang, "New media identity authentication and traffic optimization in 5g network," in *2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2017, pp. 1331–1334.