SEDE
SANTO DOMINGO

UNIVERSIDAD DE LAS FUERZAS ARMADAS - ESPE

Trabajo de titulación, previo a la obtención del título de Ingeniería en Tecnologías de la Información

“Clasificación de ataques mediante técnicas de aprendizaje automático en Redes Definidas por Software ”

Autor: Castillo Camacho, Miguel Angel

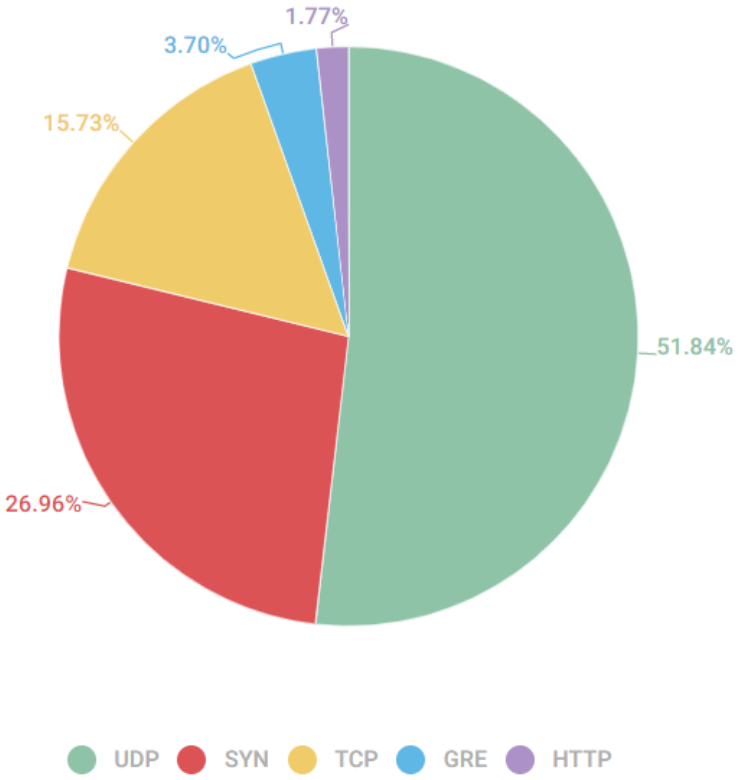
Tutor: Msc. Nuñez Agurto, Alberto Daniel Mgtr.

Agenda

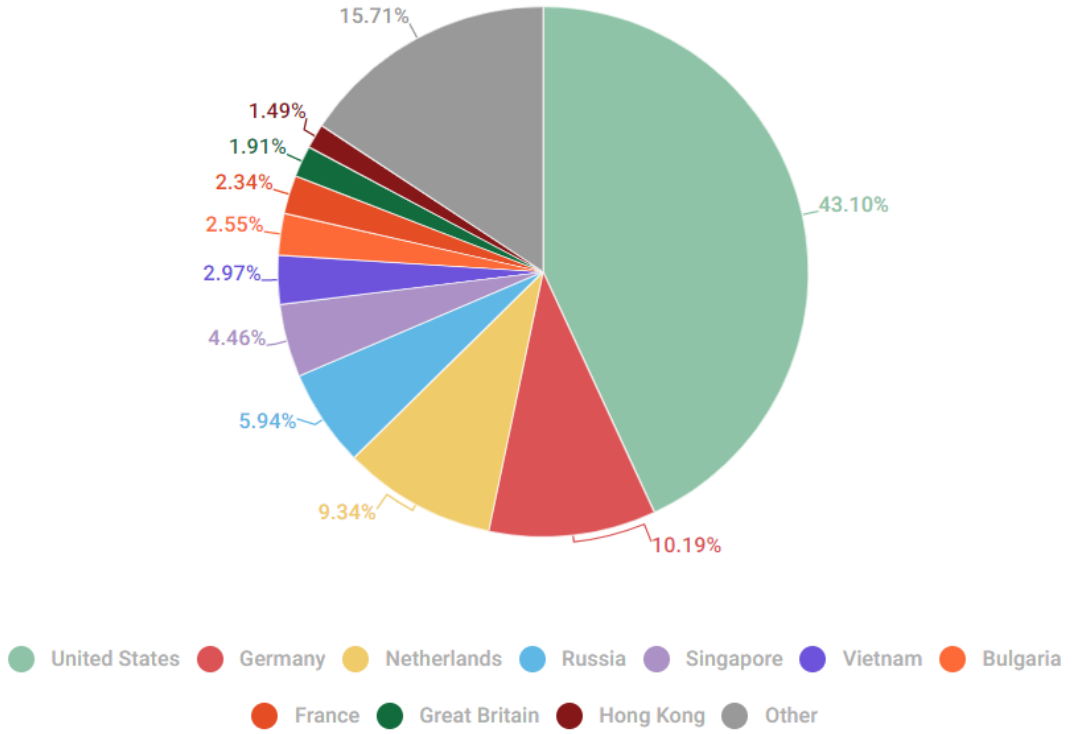
- ❑ Introducción
- ❑ Problemática
- ❑ Objetivos
- ❑ Marco Conceptual
- ❑ Revisión Sistemática de la Literatura
- ❑ Propuesta de solución
- ❑ Metodología
- ❑ Desarrollo
- ❑ Implementación
- ❑ Resultados
- ❑ Conclusiones
- ❑ Recomendaciones
- ❑ Trabajos Futuros

Introducción

Distribución de ataques DDoS.



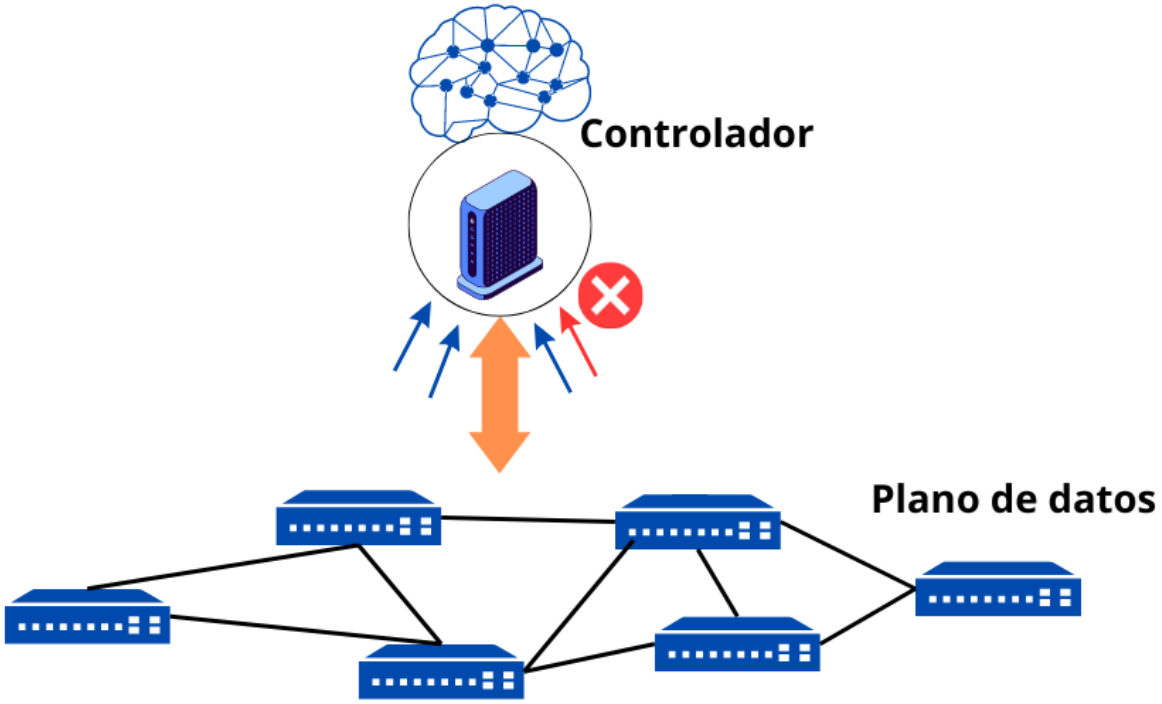
Ataques dirigidos a infraestructuras de redes llevados a cabo por Botnets.



Recuperado de Kaspersky (2022).

Problemática

- SDN presenta amenazas más graves que las redes tradicionales debido a las vulnerabilidades que presenta un componente centralizado de control.



Objetivos

Objetivo General

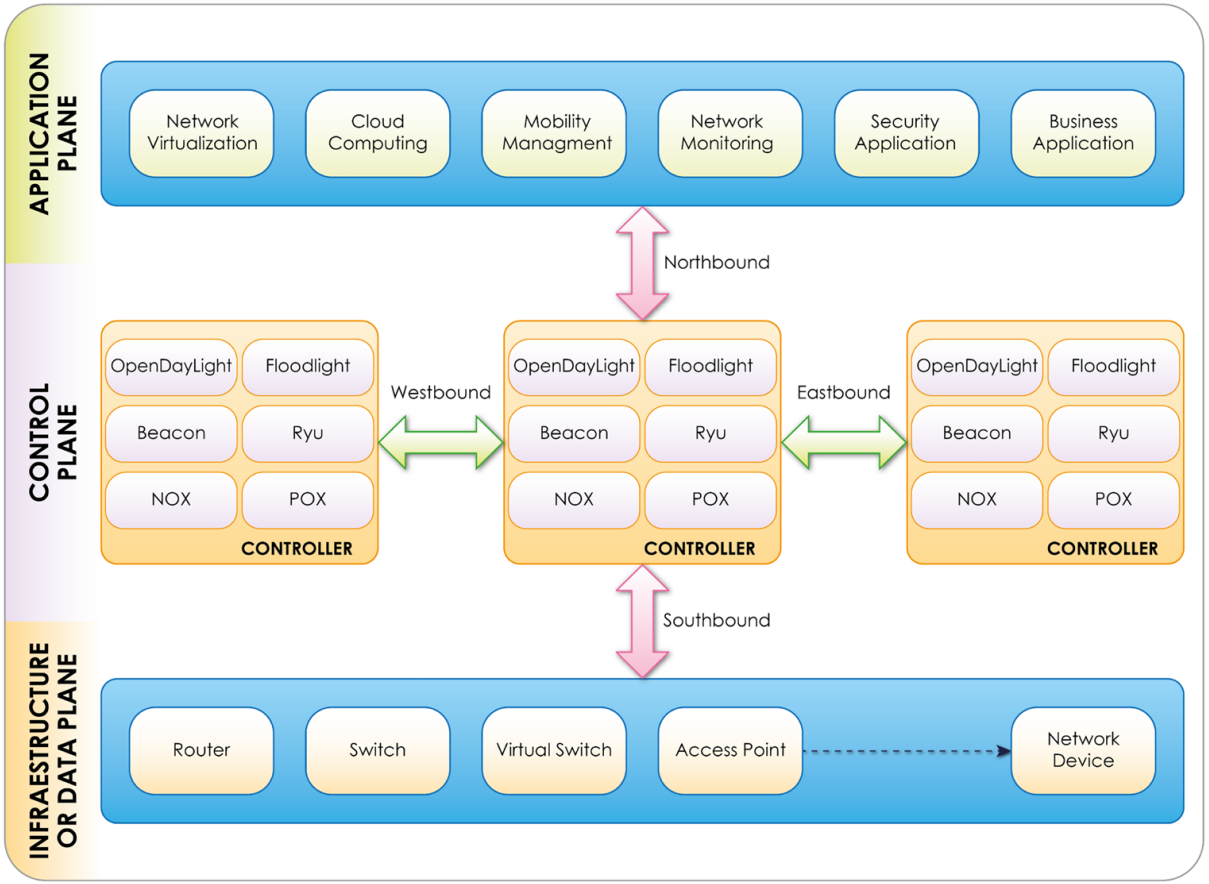
Definir un modelo de clasificación de ataques mediante el uso de técnicas de aprendizaje automático en Redes Definidas por Software.

Objetivos Específicos

- Determinar los enfoques de clasificación de ataques y los trabajos relacionados en el ámbito de seguridad en SDN.
- Realizar la selección de algoritmos de aprendizaje automático y el conjunto de datos apropiado para entrenar el modelo de clasificación de ataques en SDN.
- Desarrollar la implementación del modelo de clasificación de ataques en un escenario emulado.
- Evaluar el modelo de clasificación de ataques.

Marco Conceptual

Arquitectura de Red Definida por Software



Recuperado de Núñez (2022).

Marco Conceptual

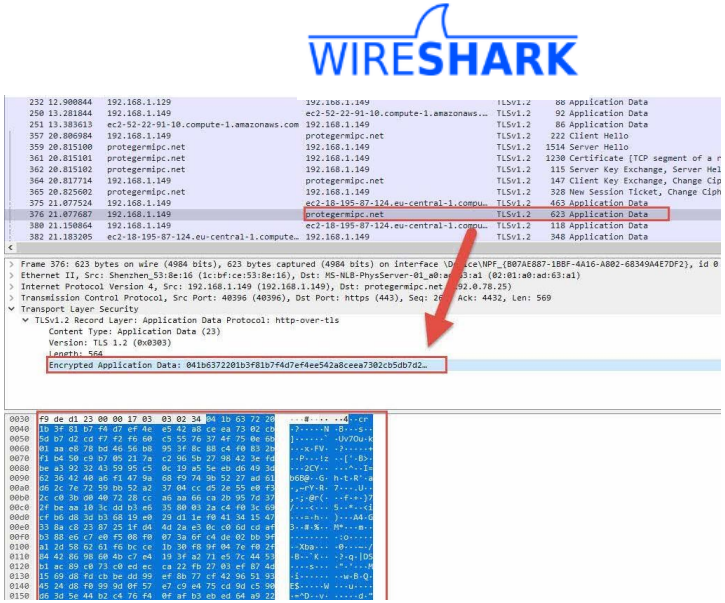
Controladores open source

CONTROLADOR	LENGUAJE DE PROGRAMACIÓN	PLATAFORMA
NOX	C++	LINUX
POX	PYTHON	LINUX, MAC OS Y WINDOWS
FLOODLIGHT	JAVA	LINUX, MAC OS Y WINDOWS
OPENDAYLIGHT	JAVA	LINUX, MAC OS Y WINDOWS
RYU	PYTHON	LINUX
TREMA	RUBY Y C	LINUX
ONOS	JAVA	LINUX, MAC OS Y WINDOWS
BEACON	JAVA	LINUX, MAC OS Y WINDOWS

Marco Conceptual

Métodos de clasificación de tráfico

1. Analizar los paquetes activos de la red y clasificarlos según el puerto de las aplicaciones

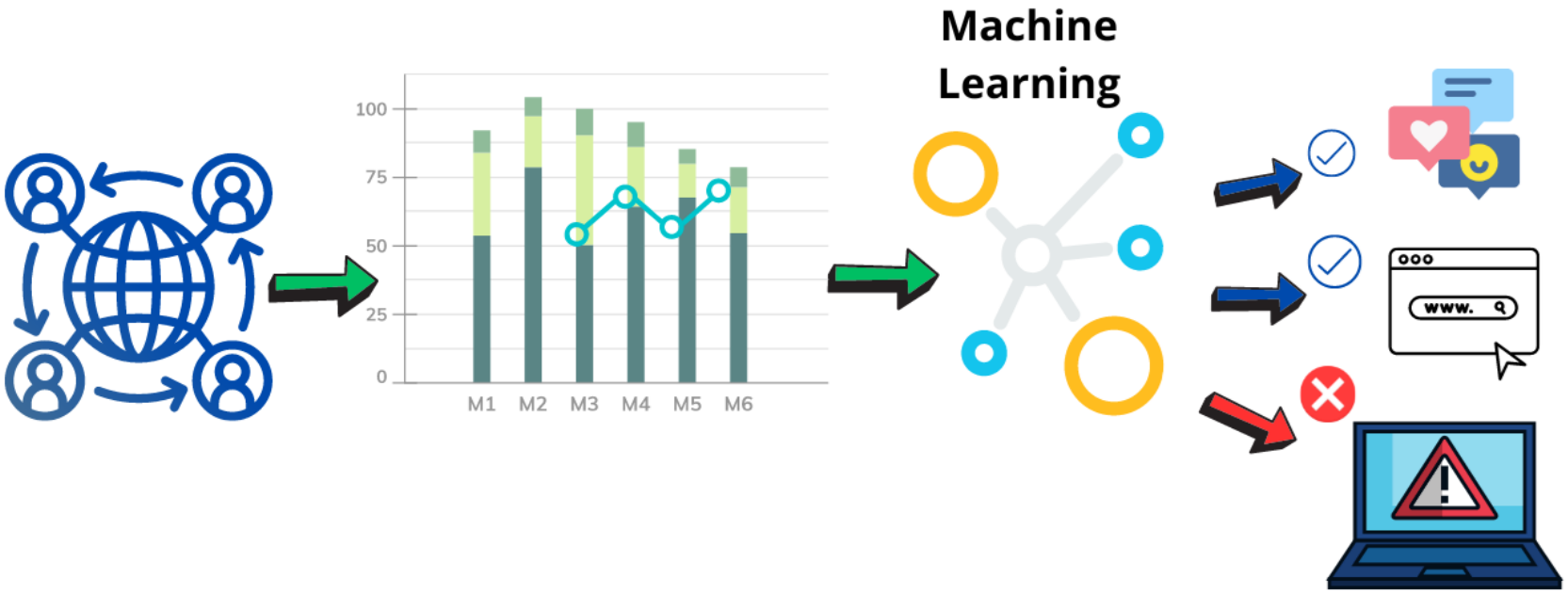


2. Inspección profunda de paquetes, es decir, la carga útil de los paquetes.

Marco Conceptual

Métodos de clasificación de tráfico

3. Aprendizaje automático usando datos estadísticos del tráfico de red.



Revisión Sistemática de la Literatura

RQ1. ¿Cuáles son los modelos de Machine Learning utilizados en la clasificación de ataques en SDN?

Técnica	EP1	EP2	EP3	EP4	EP5	EP6	EP7	EP8	EP9	EP10	EP11	EP12	EP13	EP14
Decision Tree					■	■	■		■			■	■	
Random Forest				■	■	■	■							■
Support Vector Machine	■		■	■	■	■	■		■					■
KNN					■		■	■	■					■
Naive Bayes			■		■		■	■			■	■		■
XG Boost		■			■	■								

Revisión Sistemática de la Literatura

RQ2. ¿Cuáles son las características utilizadas por las soluciones actuales en la clasificación de ataques en SDN?

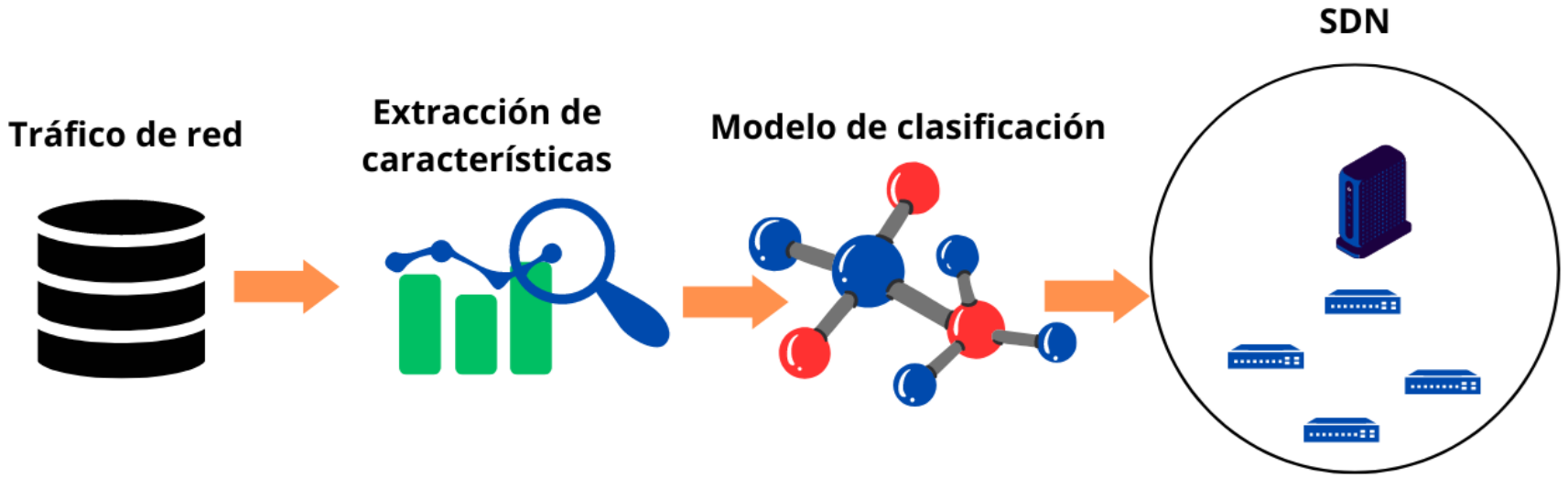
	EP1	EP2	EP3	EP4	EP5	EP6	EP7	EP8	EP9	EP10	EP11	EP12	EP13	EP14
Características		20	11	16		5		5	14		14	15		
Grupo de características			2											
Balanceo de Clases														

Revisión Sistemática de la Literatura

RQ3. ¿Cuáles son los conjuntos de datos disponibles para entrenar modelos de clasificación de ataques en SDN?

Dataset	Tipo de tráfico
NSL-KDD	DoS, Probe, R2L y U2R
CTU-13	Botnet
ISOT	DoS, Botnet, Ransomware
DDoS attacks SDN Dataset	DDoS
UNB-ISCX	DoS, DDoS, Ataques de Fuerza Bruta.
CSE-CIC-IDS2018	DoS, DDoS, Botnet, Heartbleed, Fuerza bruta y ataques web.

Propuesta de solución



Metodología

1. Investigación Científica del Diseño (Design Science Research, DSR)
2. Descubrimiento de conocimiento en bases de datos (Knowledge Discovery in Databases, KDD).
3. Análisis exploratorio de datos (Exploratory Data Analysis, EDA)

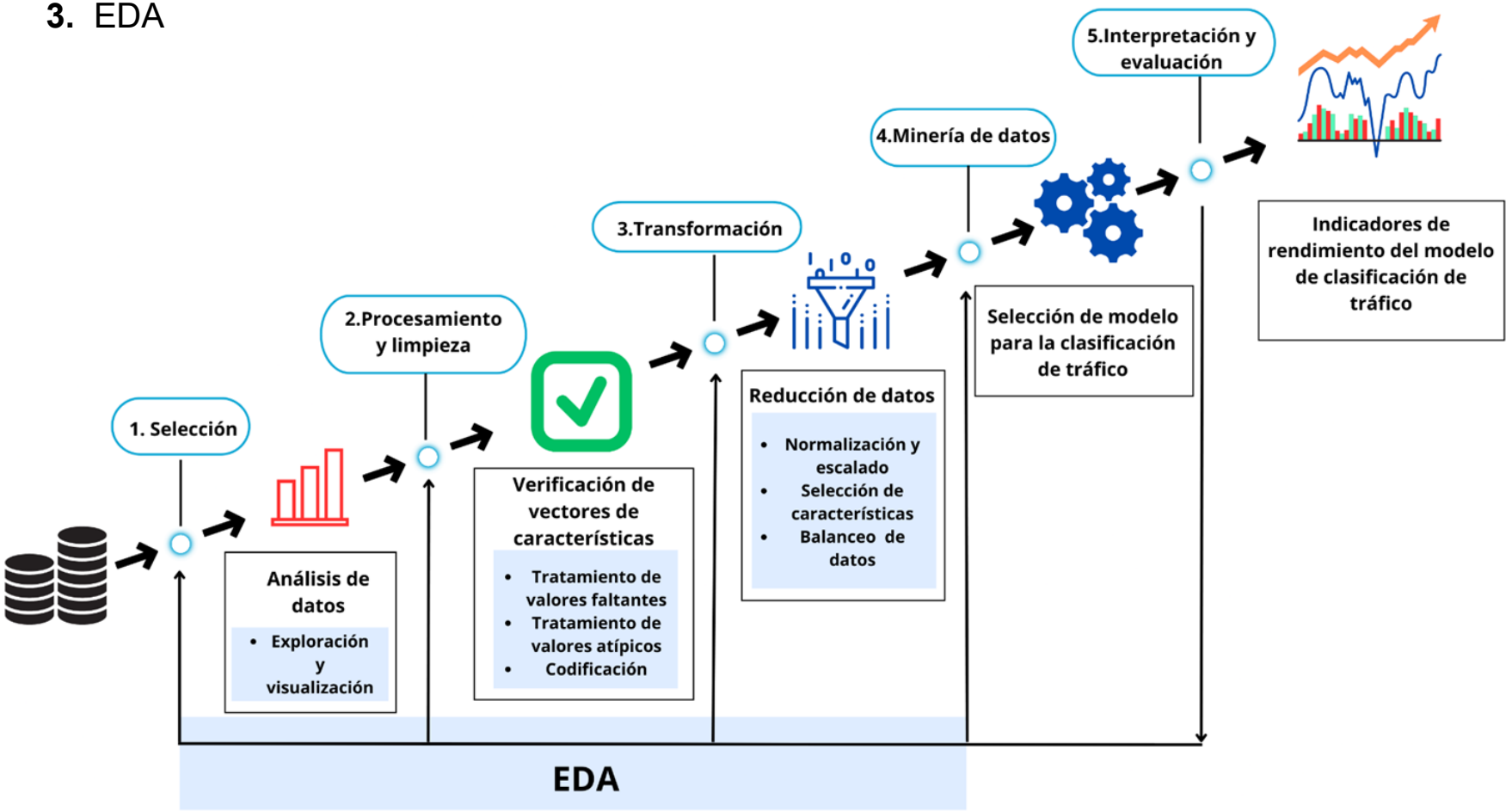
Metodología

1. Investigación Científica del Diseño

1. Identificación del problema	Ausencia de aplicaciones de seguridad en redes SDN debe detectar e identificar a tiempo los flujos malignos de la red.
2. Establecimiento de objetivos	Definir un modelo de clasificación de ataques usando técnicas de aprendizaje automático en Redes Definidas por Software
3. Diseño y desarrollo de la propuesta	• Desarrollo del modelo de clasificación • Desarrollo de la arquitectura de implementación
4. Implementación	Despliegue del modelo de clasificación en un entorno SDN emulado.
5. Evaluación	Examinar el modelo de clasificación en un entorno de pruebas.
6. Resultados	Ponderaciones alcanzadas por el modelo de clasificación de tráfico.

Metodología

2. KDD
3. EDA



Desarrollo: origen de los datos

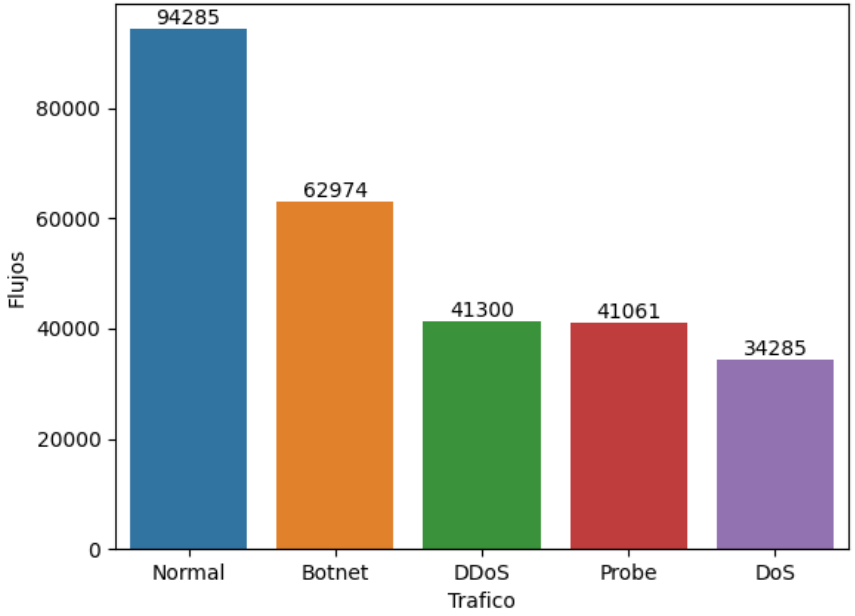
Extracción de características.

InSDN			
Tráfico	**Cantidad**	**CICFlowMeter**	**Argus**
Normal	68424	53784	111719
DoS	52471	33842	35030
DDoS	48413	398643	3352394
Probe	36372	41980	49452
Botnet	164	92	193

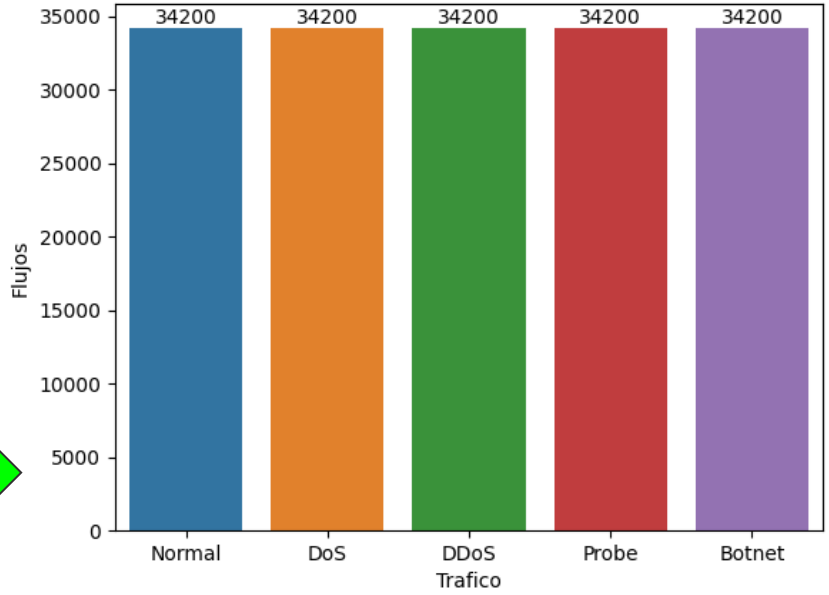
CTU-13			
Tráfico	**Cantidad**	**CICFlowMeter**	**Argus**
Botnet	2753290	132168	192168

Desarrollo: limpieza de datos

Cantidad de datos sin valores faltantes.



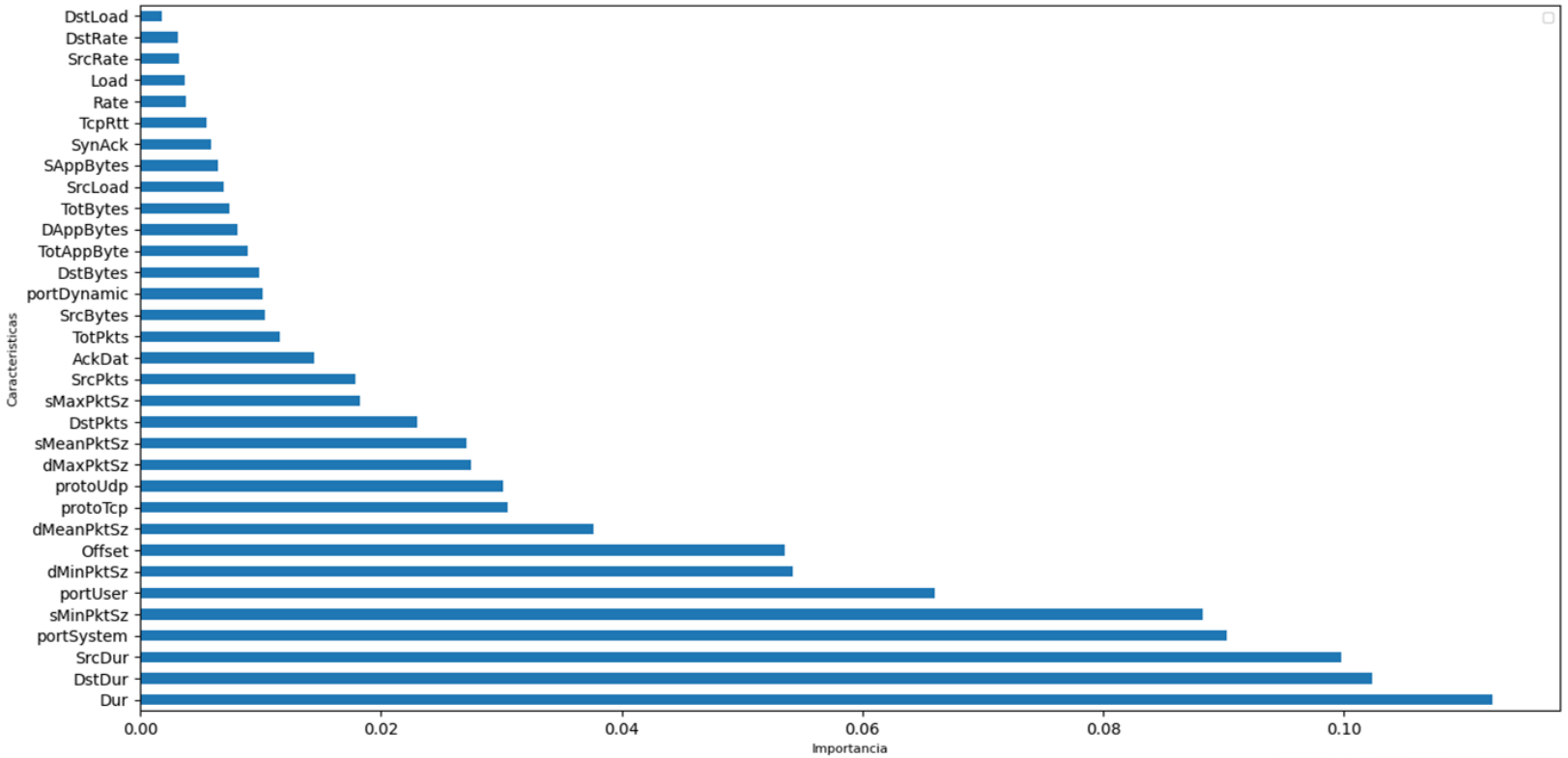
Cantidad de datos equilibrados



Desarrollo: selección de características

1. Matriz personalizada para seleccionar características.
2. Select K-Best
3. Matriz de Correlación
4. Análisis de Componentes Principales

Ponderación de características:



Desarrollo: selección de características

Definición de conjuntos de características.

GRUPO	NÚMERO	CARACTERÍSTICAS
FG1	9	Dur, Proto, Dport, TotPkts, SrcPkts, DstPkts, TotBytes, SrcBytes, DstBytes
FG2	12	Dur, Proto, Dport, TotPkts, SrcPkts, DstPkts, TotBytes, SrcBytes, DstBytes, Rate, SrcRate, DstRate
FG3	20	Dur, SrcDur, DstDur, Proto, Dport, TotPkts, SrcPkts, DstPkts, TotBytes, SrcBytes, DstBytes, Rate, SrcRate, DstRate, sMeanPktSz, dMeanPktSz, sMaxPktSz, dMaxPktSz, sMinPktSz, dMinPktSz
FG4	14	Dur, SrcDur, DstDur, Proto, Dport, TotPkts, SrcPkts, DstPkts, TotBytes, SrcBytes, DstBytes, Rate, SrcRate, DstRate

Desarrollo: selección de modelos

Análisis de selección de modelos:

MODELO ML	VENTAJA	DESVENTAJA	EJECUCIÓN
NAIVE BAYES	IMPLEMENTACIÓN FÁCIL	POCOS PARÁMETROS	MEDIA
RANDOM FOREST	ALTO RENDIMIENTO	SOBREAJUSTE	MEDIA
REGRESIÓN LOGÍSTICA	SALIDA VARIADA	SUPONE	RÁPIDO
SVM	EFICIENTE	RESULTADOS BAJOS	LENTO
DECISION TREE	ALTO RENDIMIENTO	SOBREAJUSTE	RÁPIDO
KNN	DEDUCCIÓN RÁPIDA	ENCONTRAR N VECINOS	MEDIA

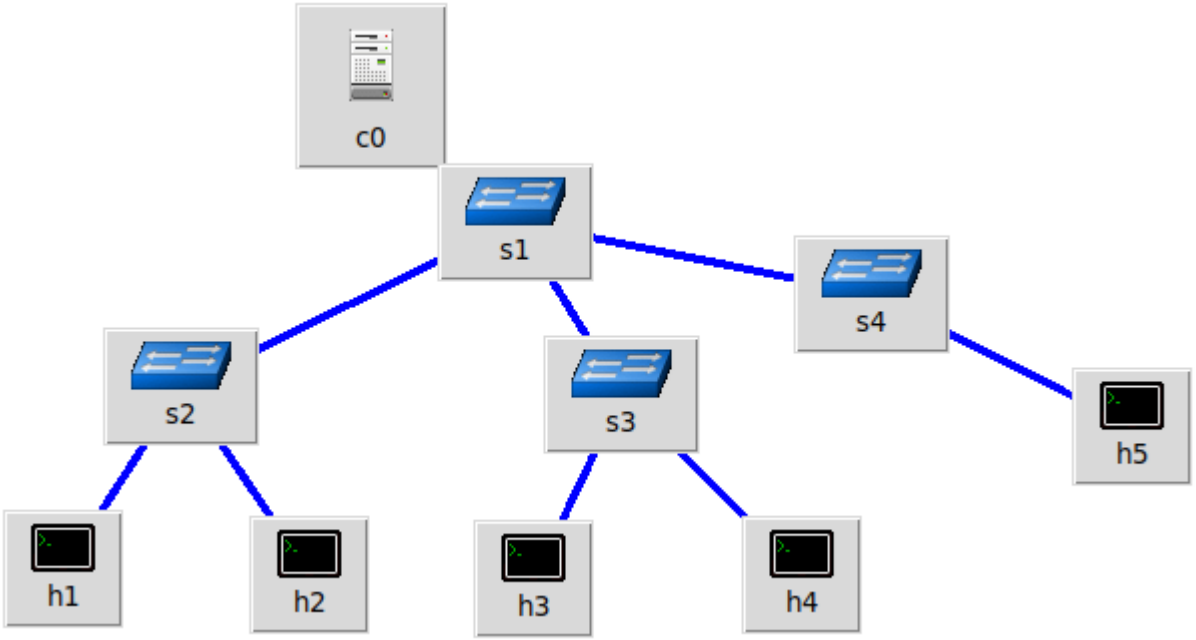
Desarrollo: entrenamiento del modelo

Resultados de hiperparámetros por modelo de clasificación:

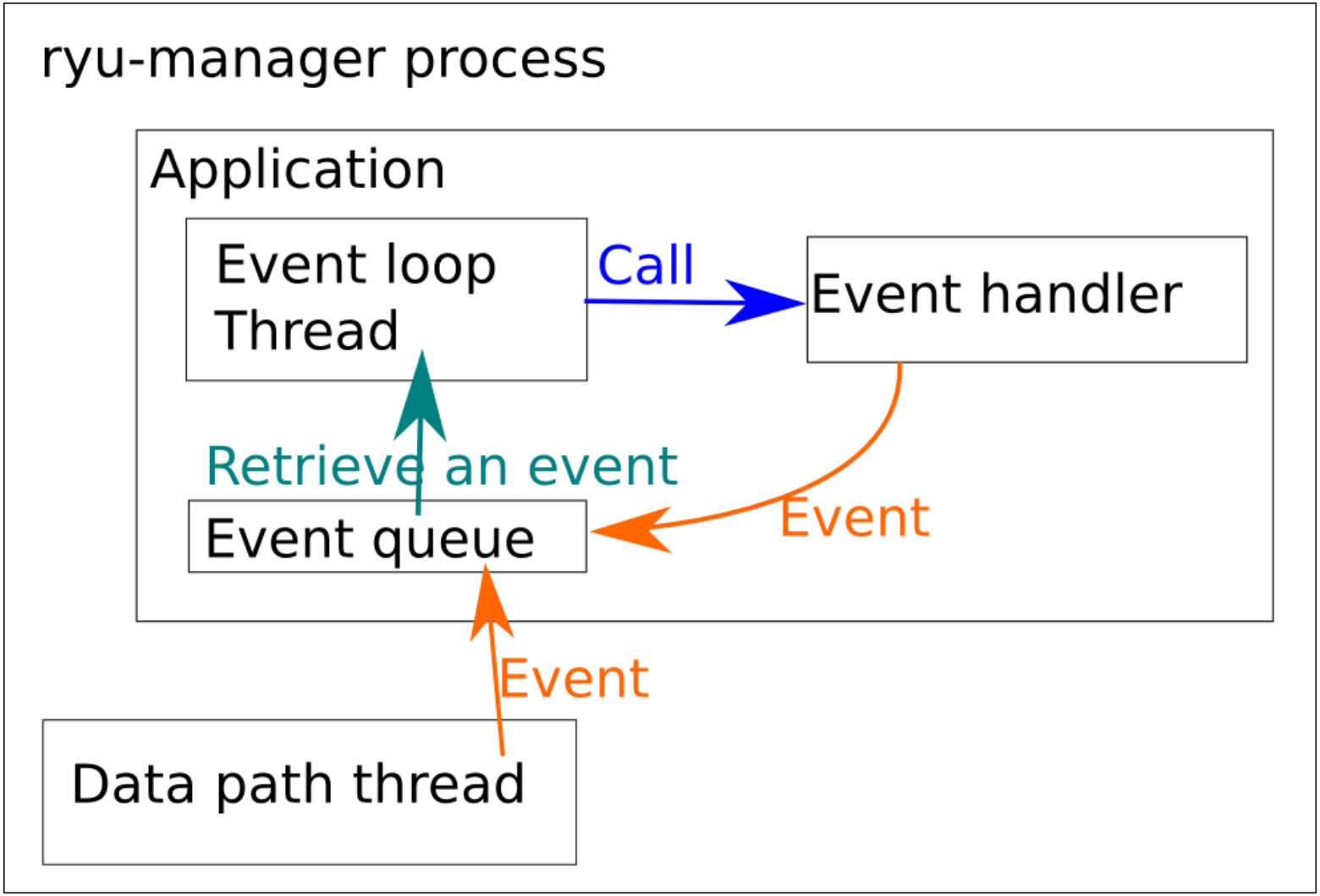
MODELO	HYPERPARÁMETRO	VALOR
DECISION TREE	CRITERION MAX_DEPTH MIN_SAMPLES_SPLIT MIN_SAMPLES_LEAF CCP_ALPHA SPLITTER	ENTROPY 12 2 1 0.0001 BEST
RANDOM FOREST	N_ESTIMATORS CRITERION MAX_DEPTH MIN_SAMPLES_SPLIT MIN_SAMPLES_LEAF CCP_ALPHA	10 ENTROPY 12 5 2 0.0001
SVM	C KERNEL GAMMA DECISION_FUCTION_SHAPE	1000 RBF SCALE OVO

Implementación

- Topología de red SDN



Controlador Ryu



Recuperado de Ryubook 1.0 documentation (2023).

Funcionamiento del modelo de clasificación

Algoritmo 1 Clasificación de flujos

```
1: procedure CONEXION CONTROLADOR Y APLICACION DE MODELO DE CLASIFICACION
Require: IP del servidor de aplicacion
2:     procedure GESTIÓN DE ESTADÍSTICAS DE FLUJO
Require: Controlador envia solicitud de estadisticas de flujo al Switch
3:         Controlador ← Mensaje
4:         while Mensaje = TCP or UDP do
5:             ListaMensajes ← Mensaje
6:             if len(ListaMensajes)%2 == 0 then
7:                 for Caracteritica in ListaMensajes do
8:                     if Duracion ≤ 20s then
9:                         Flujo1 ← Caracteristica
10:                        Flujo2 ← Caracteristica + 1
11:                        FlujoBidireccional ← Flujo1 + Flujo2
12:                        POST ← FlujoBidireccional
13:                        Modelo ← POST
14:                        Respuesta ← Modelo
15:                        Controlador ← Respuesta
16:                    end if
17:                end for
18:            end if
19:        end while
20:    end procedure
21: end procedure
```

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Resultados

Resultados de entrenamiento y validación del modelo.

Modelo	Accuracy de Entrenamiento	Accuracy de Validación
9 Características		
DT	99.34%	99.28%
RF	99.31%	99.31%
SVM	96.83%	96.83%
12 Características		
DT	99.28%	99.25%
RF	99.36%	99.30%
SVM	96.91%	97.09%

Modelo	Accuracy de Entrenamiento	Accuracy de Validación
14 Características		
DT	99.80%	99.76%
RF	99.76%	99.75%
SVM	99.49%	99.50%
20 Características		
DT	99.25%	99.25%
RF	99.32%	99.25%
SVM	97.06%	97.06%

Resultados

Resultados de la implementación del modelo de clasificación con captura de tráfico en directo.

Características	DT	RF	SVM
9	20.02%	21.45%	27.33%
12	24.14%	23.97%	43.28%

Conclusiones

- Se desarrollaron tres modelos de aprendizaje automático para la clasificación de tráfico normal, DoS, DDoS, probe y botnet. Las técnicas utilizadas fueron DT, RF y SVM, entrenadas con 136800 muestras. El conjunto de datos utilizado para entrenar los modelos resultó de la combinación entre InSDN y CTU-13, para el tratamiento de los datos se aplicó las metodologías KDD y EDA.

- Según el SLR existen diversas ideas para desarrollar modelos que puedan clasificar ataques de tráfico en SDN; muchos de estos estudios se centran en la detección de tráfico, es decir, en su mayoría manejan dos tipos de tráfico: normal y de ataque. Por lo general, el tráfico de ataque se centra únicamente en DDoS. Además, los conjuntos de datos en su mayoría no son de origen SDN.

Conclusiones

- Para la implementación del modelo de clasificación de ataques, se utilizó Mininet, Ryu y Flask. En Mininet se desarrolló la topología, la cual representa el plano de datos. En cambio, Ryu es el controlador que permite gestionar el tráfico de la red. Por último, Flask permite desplegar el modelo de clasificación de ataques y escenifica el plano de aplicación.

- Los resultados encontrados durante el entrenamiento de los modelos DT, RF y SVM fueron del 99.76%, 99.31% y 99.50% de precisión, respectivamente. Estos hallazgos son obtenidos a partir de la validación del modelo y no del resultado del entrenamiento, como se realiza en algunos trabajos relacionados. Además, se implementó el modelo en un entorno de emulación, y se obtuvo 43.28%, 24.14% y 23.97% para los modelos SVM, DT y RF, respectivamente.

Recomendaciones

- Analizar los datos de origen y utilizar metodologías interactivas para el tratamiento de los datos.

- Se recomienda conocer qué tipos de datos aceptan los algoritmos y si estos son sensibles al ruido de los datos, sobreajuste o desajuste.

- Para el despliegue de un modelo de clasificación de tráfico, se recomienda conocer las características que ofrece el controlador, así como también, los módulos, clases, eventos y APIs. Dado que para el modelo con cualquier tipo de aprendizaje, la aplicación en el controlador es un requisito importante, porque desde allí se extraen las características y el funcionamiento de toda la red.

Trabajos Futuros

- Para mejorar los resultados del modelo de clasificación y realizar su implementación en una red SDN, se podría aumentar el conjunto de datos. Además, se debería analizar la existencia de nuevas características en distintas versiones de controladores open source.

- Una de las posibles mejoras para trabajos futuros es la aplicación de técnicas de aprendizaje profundo para entrenar el modelo de clasificación de ataques, lo que permitiría detectar patrones en los datos de forma más eficiente y precisa. El uso de estas técnicas podría mejorar significativamente el nivel de predicción del modelo.

Muchas gracias