Clasificación del tráfico de red mediante técnicas de aprendizaje automático en Redes Definidas por Software

Ordóñez Ordóñez, Karla Xiomara y Pisco Quispe, Lesly Milena

Departamento de Ciencias de la Computación

Carrera de Tecnologías de la Información

Trabajo de Integración Curricular, previo a la obtención de título de Ingeniero/a en Tecnologías de la Información

Ing. Núñez Agurto, Alberto Daniel, Mgtr.

04 de septiembre de 2023

Reporte de verificación de contenido

Tesis_SDN_TC_ML.pdf

Scan details

Scan time:	Total Pages:	Total Words:
August 26th, 2023 at 20:36 UTC	61	15203

Plagiarism Detection

Types of plagiarism		Words
Identical	0.6%	96
Minor Changes	0.2%	34
Paraphrased	3.3%	496
Omitted Words	0%	0

AI Content Detection

Text coverage
- AI text
- Human text

Plagiarism Results: (26)

Quieres saber cuales son las fases de la metodologí... 0.8%

https://es.linkedin.com/pulse/quieres-saber-cuales-son-las-f...

Mauricio Mora Caballero

...

SDN: Todo lo que necesitas saber sobre la red defini... 0.7%

https://polaridad.es/sdn-todo-lo-que-necesitas-saber-sobre-l...

Saltar al contenido Cargando ahora ...

AntonioAquinoAldair.pdf?sequence=1&isAllowed=y 0.7%

https://cdigital.uv.mx/bitstream/handle/1944/50272/antonio...

ANTONIO AQUINO ALDAIR

UNIVERSIDAD VERACRUZANA FACULTAD DE ESTADÍSTICA E INFORMÁTICA

Proceso de minería de datos centrado en el usuario con base en la norma IS...

Firmado electrónicamente por:
ALBERTO DANIEL
NUNEZ AGURTO

- - - - - - - - - - - - - - - -

Núñez Agurto, Alberto Daniel

Director

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Certificación

Certifico que el trabajo de integración curricular: "**Clasificación del tráfico de red mediante técnicas de aprendizaje automático en Redes Definidas por Software**" fue realizado por las señoritas **Ordóñez Ordóñez, Karla Xiomara** y **Pisco Quispe, Lesly Milena**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Santo Domingo de los Tsáchilas, 04 de septiembre de 2023

Firmado electrónicamente por:
ALBERTO DANIEL
NUNEZ AGURTO

_____

Núñez Agurto, Alberto Daniel
C.C.: 1716572548

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Responsabilidad de Autoría

Nosotros, **Ordóñez Ordóñez, Karla Xiomara y Pisco Quispe, Lesly Milena** con cédulas de ciudadanía n° 2350064537 y 2351139551 declaramos que el contenido, ideas y criterios de trabajo de integración curricular: **Clasificación del tráfico de red mediante técnicas de aprendizaje automático en Redes Definidas por Software,** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando la citas bibliográficas.

Santo Domingo de los Tsáchilas, 04 de septiembre de 2023

Firma:

Ordóñez Ordóñez, Karla Xiomara
C.C.: 2350064537

Pisco Quispe, Lesly Milena
C.C.: 2351139551

ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Autorización de publicación

Nosotros, **Ordóñez Ordóñez, Karla Xiomara** y **Pisco Quispe, Lesly Milena** con cédulas de ciudadanía n° 2350064537 y 2351139551 autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Clasificación del tráfico de red mediante técnicas de aprendizaje automático en Redes Definidas por Software**, en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Santo Domingo de los Tsáchilas, 04 de septiembre de 2023

Firma:

Ordóñez Ordóñez, Karla Xiomara
C.C.: 2350064537

Pisco Quispe, Lesly Milena
C.C.: 2351139551

Dedicatoria

En primer lugar, dedico este tituló académico a Dios, quien fue mi fortaleza para desarrollarme profesionalmente.

A mis docentes, quienes con sus conocimientos me guiaron durante toda mi formación universitaria.

A Kevin Mejia que con su amor, paciencia y apoyo incondicional cada día me brindó las fuerzas y motivación necesaria para verme triunfar.

A mi hijo Matthew Mejia por ser mi fuente de inspiración, superación y ejemplo, por valorar y comprender el sacrificio del tiempo que tuve que dedicar para la obtención de este nuevo título académico.

A mi mamá Jeanette Ordóñez, que con su amor incondicional, sabiduría, perseverancia y sus frases de tu si puedes, ya falta poco, supo darme el valor y ganas de salir adelante.

A mis hermanos Adriana, Cristian, Daniela, Leidy y Josselyn por estar siempre pendiente de mis logros y que con su amor y consejos me ayudaron a crecer más con el objetivo de cumplir todos mis sueños.

Y por último, pero no menos importante me dedico mi trabajo a mí, para que quede como constancia de que los sueños y metas académicas se logran a pesar de las adversidades.

Ordóñez Ordóñez, Karla Xiomara

Dedico este proyecto de titulación y toda mi carrera universitaria a Dios y a mis padres, la Sra. Tania Quispe y el Sr. Raúl Pisco, por brindarme su amor, apoyo e inspiración para mantenerme firme en este recorrido. Gracias por creer siempre en mí y por proporcionarme la motivación para seguir adelante. Son padres ejemplares que han inculcado en mi corazón los mejores valores y principios.

A mis hermanos, Faviana, Raúl y Daniel, les dedico un agradecimiento especial por estar siempre a mi lado en los momentos más difíciles que tuve que atravesar, no solo durante mi desarrollo académico universitario, sino a lo largo de toda mi vida.

También, a mi novio, Edgar Mendoza, que con su amor, paciencia, apoyo y motivación nunca me permitieron rendirme. Gracias por ser parte de mis metas, sueños y logros. A mi gatito Tigger, que me acompañó durante todas las largas madrugadas de mi carrera, y siempre seguirá acompañándome desde el cielo.

Pisco Quispe, Lesly Minlena

Agradecimiento

Agradezco a Dios por brindarme la fuerza, la determinación y la sabiduría necesarias para alcanzar mis objetivos y completar mi carrera universitaria. Quiero agradecerme a mí misma, ya que con coraje y determinación logré alcanzar una meta más: obtener mi título universitario.

A Kevin, le agradezco enormemente por su apoyo incondicional día tras día. Él me ayudó a convertirme en lo que soy hoy, una profesional. También quiero expresar mi agradecimiento a mi hijo, quien ha sido mi fuente de inspiración para seguir adelante. Con su amor incondicional, me ha demostrado que puedo alcanzar mis sueños.

Mi gratitud se extiende a mis docentes, especialmente al Ing. Daniel Núñez, quien fue mi tutor en este estudio. Él estuvo siempre dispuesto a responder mis dudas y compartió su conocimiento, lo que fue fundamental para culminar este proceso con éxito. También deseo agradecer a la Universidad por brindarme la oportunidad de completar mi carrera universitaria y obtener mi título de Ingeniera.

No puedo dejar de mencionar a mi mamá y a mis hermanos, quienes han sido un pilar fundamental en mi vida. Ellos me acompañaron durante todo este proceso y me apoyaron incondicionalmente, en los momentos buenos y malos.

Quiero expresar mi gratitud a mi gran amiga y compañera de tesis, Lesly. Su compañía durante este viaje académico fue inestimable, y asumir este reto juntas nos llevó a completar una etapa importante en nuestras vidas.

Finalmente, agradezco a todas las personas que estuvieron presentes en este proceso y que, con un mensaje de aliento en los momentos en que más lo necesitaba, me ayudaron a lograr el éxito en la culminación de este proceso.

Ordóñez Ordóñez, Karla Xiomara

Agradezco a Dios ante todas las cosas y a mis padres por ser mi fortaleza, por

brindarme sabiduría y siempre bendecirme en este camino académico.

A la Universidad de las Fuerzas Armadas por darme la posibilidad de formarme profesionalmente en esta prestigiosa Institución y por la excelente acogida que he tenido. Expreso mi mas sincero agradecimiento a todos los docentes que han compartido sus conocimientos y sabiduría a lo largo de toda la carrera guiando mi crecimiento personal y profesional. No solo fueron educadores, sino también personas de las que siempre contamos con un apoyo incondicional.

A mi tutor de tesis, el Ing. Daniel Nuñez, por su calidad y el tiempo que ha dedicado para guiarnos en este proceso y permitirnos culminar con éxito este proyecto de titulación.

Pisco Quispe, Lesly Milena

Índice

Índice de fíguras

Índice de tablas

Resumen

En el contexto actual, la clasificación precisa de aplicaciones en el tráfico de red representa un desafío significativo para garantizar el funcionamiento óptimo y seguro de las redes. Este proyecto involucró un estudio exhaustivo sobre la clasificación de aplicaciones en el tráfico de red, adoptando un enfoque de granularidad fina en el conjunto de datos. Se extrajeron flujos de datos utilizando el software CICflowMeter, con dos ajustes de tiempo de espera de flujo diferentes: 120 segundos y 15 segundos. Con el objetivo de lograr una clasificación efectiva, se aplicó la metodología SEMMA y se emplearon cuatro algoritmos de aprendizaje supervisado: Máquina de Vectores de Soporte (Support Vector Machine, SVM), Árbol de Decisión (Decision Tree, DT), Bosque Aleatorio (Random Forest, RF) y K-Vecinos más Cercanos (K-Nearest Neighbors, KNN). Estos algoritmos se utilizaron con dos grupos de características diferentes: uno con 25 características y otro con 15 características. Los resultados obtenidos revelaron que el conjunto de datos con un tiempo de espera de flujo de 15 segundos y el grupo de 15 características lograron el mayor nivel de precisión, con resultados de entrenamiento de precisión en los algoritmos RF (99.99%), DT (99.89%), KNN (99.92%) y SVM (92.06%). Estos hallazgos destacan la notable efectividad del modelo RF en la clasificación de aplicaciones en el tráfico de red.

Palabras clave: Redes Definidas por Software, Aprendizaje Automático, Clasificación de aplicaciones.

Abstract

In the current context, accurate classification of applications in network traffic represents a significant challenge to ensure networks' optimal and secure operation. This project involved a comprehensive study on the classification of applications in network traffic, adopting a fine-grained approach to the data set. Data streams were extracted using CICflowMeter software, with two different flow timeout settings: 120 seconds and 15 seconds. The SEMMA methodology was applied to achieve effective classification, and four supervised learning algorithms were employed: Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and K-Nearest Neighbors (KNN). These algorithms were used with two different feature sets: one with 25 and the other with 15. The results obtained revealed that the dataset with a flow waiting time of 15 seconds and the 15-feature group achieved the highest level of accuracy, with accuracy training results in the RF (99.99%), DT (99.89%), KNN (99.92%) and SVM (92.06%) algorithms. These findings highlight the remarkable effectiveness of the RF model in classifying applications in network traffic.

Keywords: Software-Defined Networking, Machine Learning, Application Classification

Capítulo I: Introducción

Antecedentes

En la era digital actual, las redes de comunicación desempeñan un papel fundamental en la interconexión y el intercambio de información en todo el mundo. La creciente demanda de servicios en línea, la proliferación de dispositivos conectados y la continua evolución de las aplicaciones han dado lugar a un incremento significativo en la diversidad del tráfico de red. Esta complejidad plantea importantes desafíos para el monitoreo, la gestión y la optimización de las redes tradicionales (Belkadi et al., 2023). Por lo tanto, los proveedores de servicios de internet (ISP) y los administradores de redes deben adaptarse a estos cambios, empleando herramientas y técnicas adecuadas para garantizar una gestión de red eficiente.

En los últimos años, los investigadores han estado trabajando en el desarrollo de nuevas tecnologías y metodologías para mejorar la gestión y el rendimiento de las redes. Una de las innovaciones más prometedoras en este campo son las Redes Definidas por Software (SDN), que han sido consideradas como el futuro de las redes. Las SDN ofrecen una forma más simplificada y eficiente de gestionar y administrar las redes, permitiendo una mayor flexibilidad y adaptabilidad en la infraestructura de red (Silva, 2021). La principal ventaja de las SDN es su capacidad para separar el plano de datos del plano de control, lo que permite que los dispositivos de control se centren en la gestión de políticas, mientras que los dispositivos de datos se dedican exclusivamente al procesamiento y reenvío de paquetes (Almadani et al., 2021). Los controladores SDN pueden programar la red de manera centralizada, lo que simplifica la gestión y la configuración de la red, reduciendo el tiempo necesario para realizar cambios en la red (Castaneda Herrera et al., 2021). La programabilidad es otra característica clave de las SDN, lo que permite una mayor flexibilidad y adaptabilidad en la infraestructura de red. La identificación de los

distintos flujos de datos que atraviesan la red es esencial para aplicar políticas de Calidad de Servicio (Quality of Service, QoS), seguridad y optimización del rendimiento (Ganesan et al., 2021).

La clasificación precisa del tráfico se ha convertido en un desafío importante en la gestión de redes, ya que permite a los administradores de redes analizar y optimizar los recursos de la red con mayor precisión. Este desafío surge debido a la naturaleza dinámica de las nuevas aplicaciones de red. Existen tres técnicas de clasificación de tráfico destacadas: basada en puertos a nivel de TCP/IP (Transmission Control Protocol/Internet Protocol) (Dashevskiy & Luo, 2014), Inspección Profunda de Paquetes (DPI) , esta técnica de clasificación es adecuada para el tráfico P2P (Finsterbusch et al., 2013; Goli & Ambika, 2018) y el método de clasificación estadística (Belkadi et al., 2023). Sin embargo, las técnicas basada en puertos y DPI tienen limitaciones, ya que no pueden identificar tráfico de red cifrado.

En las últimas décadas, la Inteligencia Artificial (IA) se ha convertido en una herramienta crucial para analizar y clasificar el tráfico en las redes. Este proceso consiste en analizar los paquetes que se envían y reciben, y encontrar patrones que pertenezcan al mismo tipo de tráfico, para luego proceder a clasificarlos en categorías específicas. Los enfoques de IA se basan en estadísticas, y aquí es donde se aplica la técnica de estadística de flujo (Nguyen & Armitage, 2008). Esta técnica utiliza datos recopilados y aplica algoritmos de Aprendizaje Automático (Machine Learnig, ML) para la clasificación e identificación de los diferentes tipos de tráfico de red. Su funcionamiento radica en la observación de patrones y características transmitidas en los paquetes de red, tales como su tamaño, duración y frecuencia. Esto permite identificar inclusive tráfico de red cifrado al analizar la información contextual asociada a los paquetes de red (Fan & Liu, 2017; Tahaei et al., 2020). Por lo tanto, este método de clasificación es más precisa y eficiente para tráfico de red.

Planteamiento del problema

La clasificación de los flujos de tráfico por aplicación o de grano fino es crucial para mejorar la experiencia del usuario y aumentar el rendimiento de la red al asignar prioridad según los requisitos de QoS (Belkadi et al., 2023). Este desafío puede ser abordado mediante la integración de las SDN, que permiten una visión global de la red y la aplicación de métodos de aprendizaje automático.

La clasificación precisa del tráfico de red sigue siendo un problema crítico, por la creciente variedad de aplicaciones y servicios en línea ya que, dificulta aún más la tarea de clasificar el tráfico de manera confiable y eficiente. Si bien las SDN ofrecen ventajas significativas en términos de flexibilidad y control centralizado de la red, la clasificación precisa del tráfico dentro de un entorno SDN sigue siendo un desafío crítico. La gestión eficiente de una red SDN depende en gran medida de la capacidad de identificar y clasificar con precisión los diferentes flujos de datos que la atraviesan. Sin embargo, la implementación exitosa de estas técnicas en entornos SDN y la evaluación de su rendimiento en términos de precisión y eficiencia requiere una investigación más profunda.

La IA ha surgido como una solución prometedora para abordar este desafío. La aplicación de técnicas de ML, han demostrado ser efectiva para analizar y clasificar el tráfico de red en función de patrones y características identificables en los paquetes de datos. A pesar de los estudios previamente realizados sobre la clasificación de tráfico de grano fino en entornos SDN, aún existen desafíos por abordar, como la clasificación de nuevas clases de tráfico y la optimización de enfoques de clasificación y ajustes de hiperparámetros.

El problema central que se aborda en esta investigación se centra en la clasificación precisa y eficiente del tráfico de red en entornos SDN mediante el uso de técnicas de ML.

Este problema es de gran relevancia debido a su impacto en la gestión de redes, la seguridad cibernética y la optimización del rendimiento en el contexto de las redes modernas. Esto contribuirá significativamente a la mejora de la eficiencia y la capacidad de respuesta de las redes SDN en un entorno cada vez más complejo y dinámico.

Justificación del problema

La combinación de SDN e IA ofrece una solución eficaz para afrontar este reto. Las herramientas de IA se enfocan en aprovechar y analizar información útil, lo cual tiene un impacto significativo en la eficiencia y QoS de red. En particular, este estudio se enfoca en el uso del ML como una técnica precisa para la clasificación de estadísticas de flujos del tráfico que circula por la red. Por lo tanto, el objetivo principal de esta investigación es diseñar un modelo de clasificación de tráfico de red utilizando algoritmos de ML, específicamente enfocado en clasificar los flujos de aplicaciones con una clasificación de grano fino.

Esto permite discernir individualmente cada aplicación que circula por la red, facilitando un análisis más preciso y detallado de la composición del tráfico SDN. Para lograrlo, se llevará a cabo un proceso que consta de varias etapas. En primer lugar, se seleccionará cuidadosamente el conjunto de datos que se utilizará para el análisis y exploración. Esto incluirá la recopilación de datos relevantes y representativos del tráfico de la red SDN. A continuación, se aplicarán diversos algoritmos de aprendizaje automático, como algoritmos de ML, para determinar cuál de ellos proporciona los mejores resultados en términos de precisión y eficacia en la clasificación de los flujos de aplicaciones. Una vez que se haya desarrollado el modelo de clasificación, se llevará a cabo un exhaustivo análisis para evaluar su precisión en la clasificación de tráfico en una SDN. Los resultados de proyecto podrían conducir a nuevos métodos de clasificación de tráfico en SDN que puedan utilizarse para mejorar el rendimiento de las redes SDN.

Objetivos

Objetivo General

Definir un modelo de clasificación del tráfico de red mediante técnicas de aprendizaje automático en Redes Definidas por Software.

Objetivos Específicos

- Determinar los enfoques de clasificación de tráfico con técnicas de aprendizaje automático en SDN.
- Realizar la selección de métodos de clasificación, para detectar el tráfico de aplicaciones en un entorno fuera de línea.
- Evaluar los clasificadores seleccionados en un entorno de banco de pruebas de red.

Alcance

El presente proyecto, busca el análisis, desarrollo e implementación de un modelo de aprendizaje automático para clasificación de tráfico por aplicación, con el uso de técnicas ML, en un entorno de Redes Definidas por Software. El resultado de este proyecto, será el determinar que modelo de aprendizaje automático es el más adecuado para clasificar tráfico por aplicación.

Capítulo II: Marco teórico

Redes Definidas por Software

Las SDN representan un enfoque arquitectónico en la gestión de redes, enfocado en la programabilidad y la automatización (Figuerola, 2013). Estas redes separan el plano de control del plano de datos, lo que permite una distribución flexible del tráfico en la infraestructura de red. Los controladores basados en software brindan adaptabilidad a las necesidades de una organización, permitiendo gestionar redes virtualizadas sin depender de tecnologías de hardware integrales (Pradhan & Mathew, 2020; Ríos, 2016). Existen diferentes protocolos que permiten la comunicación entre los diferentes componentes de una SDN, los más comunes son OpenFlow, NETCONF, BGP, SNMP y REST API. Cada uno de ellos tiene su propio conjunto de funciones y características que los diferencia de los demás. Es importante recalcar que el protocolo OpenFlow es considerado el protocolo predominante para las SDN (Mondal et al., 2021). Por lo tanto, SDN reemplaza redes tradicionales al centralizar la configuración de toda la infraestructura de red, simplificando la gestión en lugar de configurar dispositivo por dispositivo."

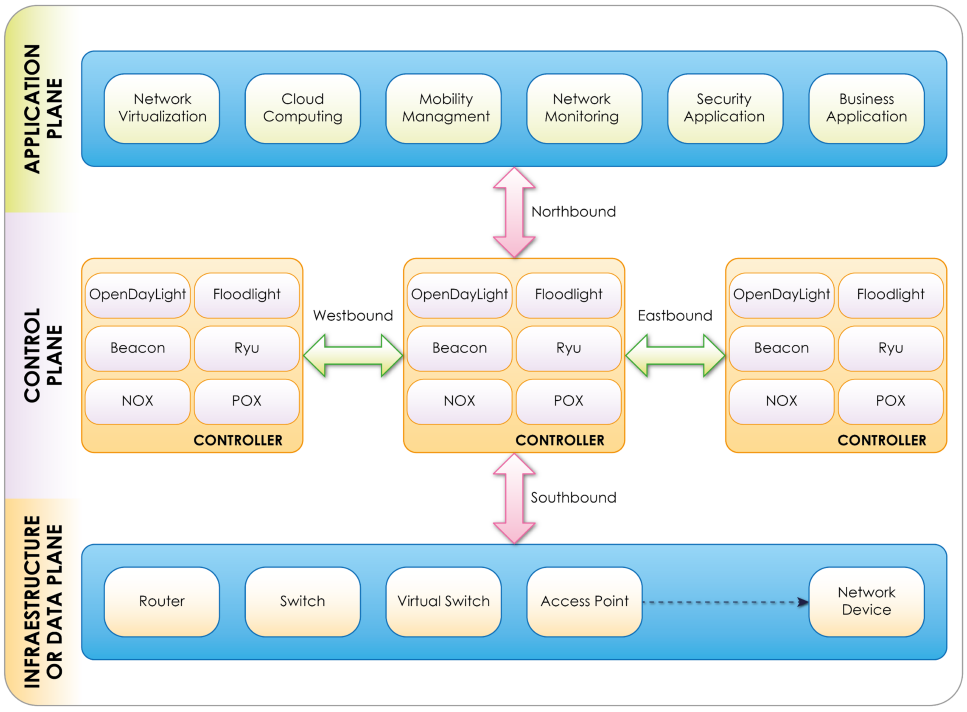
Arquitectura SDN

La arquitectura SDN proporciona mejoras significativas en la capacidad, rendimiento y administración general de las redes. Una característica clave de SDN es el desacoplamiento entre el plano de control y el plano de datos, lo que permite a los controladores de red trabajar solamente con un programa de control centralizado (Cox et al., 2017). Esta separación facilita el control de la red, proporcionando una mayor flexibilidad y agilidad en la gestión de la red, permitiendo una adaptación más rápida a los cambios. La arquitectura SDN como se muestra en la Figura 1consta de tres componentes principales, donde intercambian de manera recíproca la información, estos son la capa de infraestructura, capa de control y capa de aplicación.

En la capa de infraestructura se encuentran los dispositivos físicos de la red tales como, dispositivos de conectividad (enrutadores, conmutadores), lo cual cumplen la función de reenvío de datos guiándose por las instrucciones que reciben del controlador. La capa de control es el punto central donde lógicamente se controla la configuración de la red; siendo este el más importante, aquí se gestiona la comunicación de todos los dispositivos de red (Hu et al., 2014). Por otro lado, en la capa de aplicación están situadas las aplicaciones de usuarios que comunican sus requerimientos de red a la capa de control mediante una API que permite la intercomunicación entre aplicaciones externas con el controlador SDN, para encargarse de las necesidades y requisitos de red de la capa de aplicación (Nunes et al., 2014).

Figura 1

Arquitectura funcional de SDN.



Nota. Recuperado de Nunez-Agurto et al., 2022.

Controladores

Los controladores SDN permiten una gestión eficiente de la QoS de la red al aplicar políticas y reglas predefinidas basadas en características del tráfico. Esto permite a los encargados gestionar y controlar la funcionalidad de la red, permitiendo la programación y configuración centralizada de los dispositivos de red con el fin de redirigir, modificar o filtrar los flujos de datos según las políticas de la red (Xie et al., 2019). El controlador SDN es el componente clave de toda la arquitectura SDN, que controla principalmente los conmutadores SDN para administrar todo el flujo de datos (Gallo et al., 2016), (Hayes et al., 2018). Si bien el controlador SDN permite un control de flujo eficiente, no puede lograr una administración distribuida de QoS de extremo a extremo debido a la falta de control de los usuarios finales, así como a problemas de seguridad y privacidad (Wang et al., 2018)."

El controlador SDN se comunica con los dispositivos de red a través de protocolos de control como (OpenFlow) u otros definidos por la arquitectura SDN. Estas interfaces permiten al controlador enviar instrucciones a los dispositivos de red para configurar su comportamiento y establecer reglas de flujo (G & R, 2021). Es importante tener en cuenta que hay diferentes controladores SDN disponibles, tanto de código abierto como propietarios. En la Tabla 1 se presenta una comparación de controladores populares de código abierto utilizados en la arquitectura SDN (Ahmad & Mir, 2021). Cada controlador puede tener características y funcionalidades específicas, por lo que la elección del controlador dependerá de los requisitos y objetivos de la red SDN en particular.

Tráfico de Red

El tráfico de red se refiere al volumen de datos que circula a través de una red. Es el flujo de información generado y transmitido a través de un canal de comunicación a una velocidad determinada. En otras palabras, se trata de los paquetes de red que se envían y

Tabla 1

Controladores de código abierto para arquitectura SDN.

Controlador	Lenguaje de programación	Plataforma
NOX	C++	Linux
POX	Python	Linux, MacOS y Windows
Floodlight	Java	Linux, MacOS y Windows
Trema	Ruby y C	Linux
Opendaylight	Java	Linux, MacOS y Windows
Ryu	Python	Linux
ONOS	Java	Linux, MacOS y Windows
Beacon	Java	Linux, MacOS y Windows
Faucet	Python	Linux

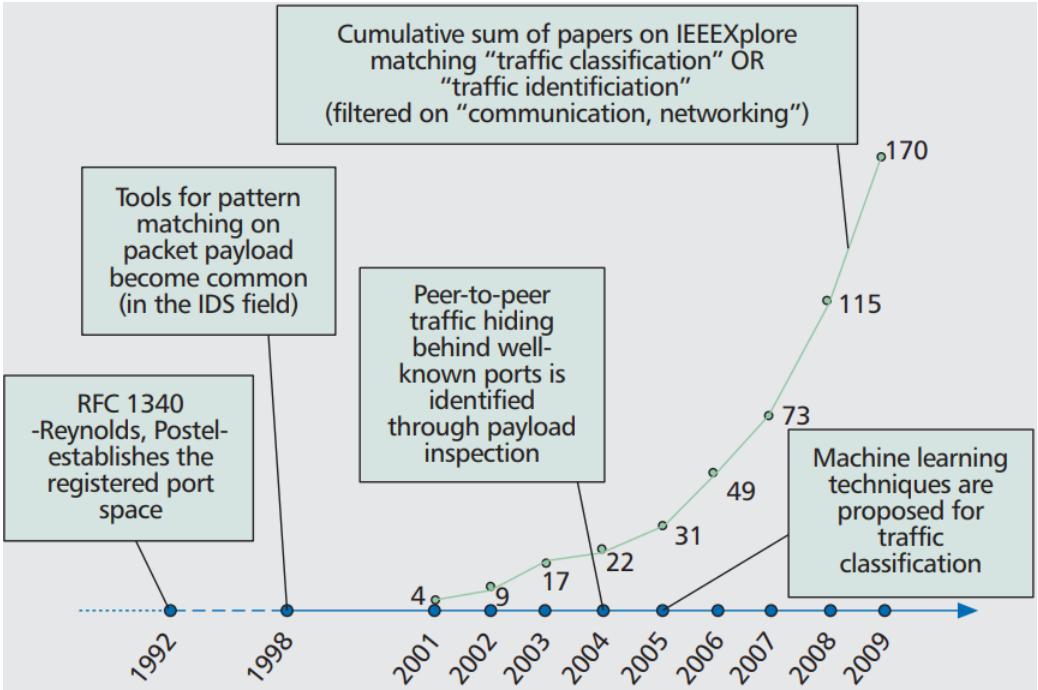
reciben en un momento específico. La gestión del tráfico de red es esencial para asegurar un óptimo rendimiento, desempeño y comportamiento de la red (Noormohammadpour & Raghavendra, 2017).

Métodos de Clasificación de Tráfico

Los métodos de clasificación de tráfico son técnicas utilizadas para analizar y categorizar los distintos flujos de datos que circulan en una red. Estos métodos permiten identificar el tipo de tráfico, como aplicaciones, protocolos o servicios específicos, y se utilizan para diversos propósitos, como la optimización de redes, la seguridad y la gestión del ancho de banda. En la actualidad, se dispone de diversos métodos para clasificar el tráfico de red. Entre estas técnicas se encuentran el enfoque basado en puertos, DPI y el de clasificación estadística. En la Figura 2 se muestra la evolución de la tecnología de clasificación del tráfico (Dainotti et al., 2012).

Figura 2

Evolución de los enfoques en la clasificación de tráfico.



Nota. Recuperado de Dainotti et al., 2012.

Clasificación basada en puertos: Es un método que utiliza números de puertos asociados a aplicaciones para identificar el tipo de tráfico, es decir, si un paquete utiliza el puerto 80, se puede inferir que está relacionado con el tráfico web; y si se detecta el uso del puerto 21, se puede clasificar como tráfico de FTP. Sin embargo, debido al crecimiento exponencial de de aplicaciones web y móviles, este enfoque se ha vuelto obsoleto y menos confiable. Esto se debe a que algunas aplicaciones pueden utilizar puertos no estándar o puertos dinámicos, lo que dificulta la clasificación precisa basada únicamente en el número de puerto, con el fin de enmascarar el tráfico y eludir la detección o filtrado de cortafuegos. Además, las técnicas de encriptación y ocultamiento de tráfico pueden evadir la clasificación basada en puertos (Dainotti et al., 2012). Por lo tanto, la identificación basada en puertos se ha vuelto una técnica de clasificación obsoleta.

DPI: Es un método altamente efectivo para clasificar el tráfico de red, ya que examina detalladamente el tipo, origen y destino de los datos. Se utiliza frecuentemente para detectar y bloquear tráfico no deseado, como virus o spam, gracias a su capacidad para analizar minuciosamente el contenido de los paquetes. Sin embargo, es importante destacar que la DPI requiere más tiempo de cálculo en comparación con otros métodos de clasificación debido a su análisis exhaustivo de cada paquete individual. Además, la DPI puede tener dificultades al tratar con datos encriptados, ya que el contenido protegido puede resultar inaccesible para el proceso de inspección profunda. A pesar de estas limitaciones, la inspección profunda de paquetes sigue siendo una técnica valiosa y ampliamente utilizada en la clasificación del tráfico de red (Finsterbusch et al., 2013).

Método de Clasificación Estadística: Es un método que se basa en la combinación de estadísticas de flujo y aprendizaje automático, lo cual ha posicionado a esta técnica como la más empleada en algunas organizaciones debido a su alta escalabilidad y eficacia. Mediante el análisis de datos y el aprendizaje automático, se logra identificar de manera precisa el tipo de tráfico de red que fluye a través de un dispositivo de red.

La principal fortaleza radica en el procesamiento eficiente de grandes volúmenes de datos y en la adaptación dinámica a cambios en los patrones de tráfico. Mediante algoritmos de aprendizaje automático, detecta características distintivas en flujos de datos para una precisa clasificación en tiempo real. Minimizando la dependencia de información específica de puertos, es resistente a técnicas de evasión de usuarios maliciosos. Así, la combinación de estadísticas de flujo y aprendizaje automático es altamente efectiva y confiable en la clasificación del tráfico de red en organizaciones actuales. (Belkadi et al., 2023).

Aprendizaje Automático

El aprendizaje autómaticao es una rama de la IA y una disciplina científica que cubre varias áreas de estudio como: matemáticas y estadística (Rouhiainen, 2018). En él se centra el desarrollo de algoritmos y modelos, los cuales permiten a los ordenadores aprender y mejorar automáticamente a través de la experiencia. En la actualidad se observa un crecimiento exponencial del aprendizaje en áreas como la minería de datos y a su vez el reconocimiento de patrones. De estas se pueden desarrollar varias tareas como la clasificación. A continuación, se presentan los diferentes tipos de aprendizajes automáticos (Belkadi et al., 2023).

Aprendizaje supervisado

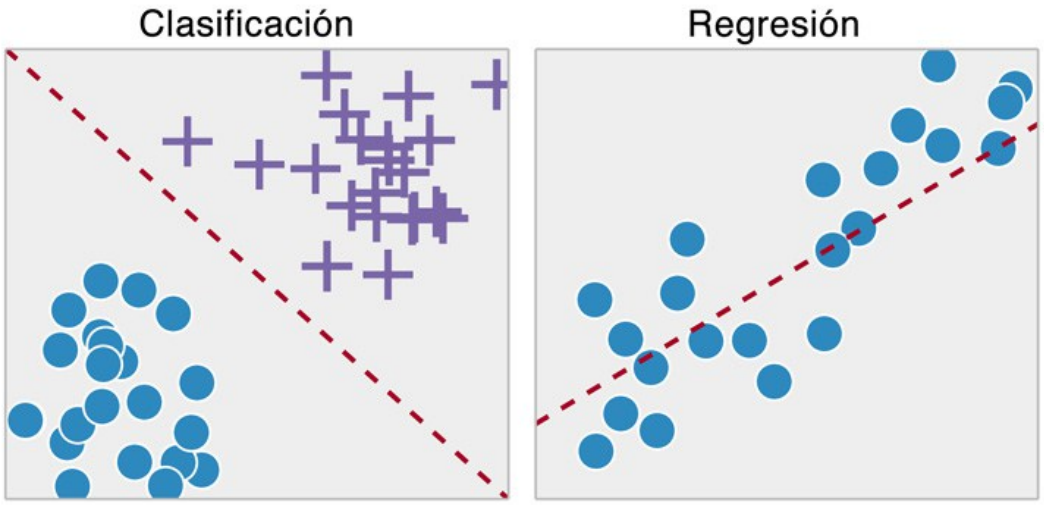
Este tipo de aprendizaje es el más utilizado y se enfoca en entrenar un conjunto de datos etiquetados, es decir que incluye tanto las entradas como las salidas esperadas. Se elige un modelo para que aprenda a realizar predicciones o a su vez a clasificar los nuevos datos que se vayan adhiriendo al conjunto de datos, basándose en la información de entrada/salida. Dentro del aprendizaje supervisado se tiene la clasificación y regresión. En la regresión se tiene que, una función de aprendizaje asigna los datos a una variable de valor real y se menciona que las salidas son valores continuos. La clasificación es un enfoque comúnmente utilizado en el aprendizaje automático donde los resultados se asignan a valores discretos (Rojas, 2020). En la Figura 3 se muestran diversos objetos o datos en un contexto de clasificación que se pueden categorizar mediante algoritmos; mientras que la regresión representa un solo tipo de datos, generalmente continuos.

Los modelos de aprendizaje supervisado se dividen en:

- ***Modelos de aprendizaje supervisado para clasificación:*** Regresión logística, DT, RF, SVM, Naive Bayes (NB), KNN, Redes neuronales artificiales (ANN), Gradient Boosting Machines (GBM), XGBoost, LightGBM, CatBoost, AdaBoost, Perceptrón

Figura 3

Aprendizaje Supervisado.



Nota. Recuperado de Epicalsoft, 2018.

multicapa (MLP), Análisis discriminante lineal (LDA), Análisis discriminante cuadrático (QDA) (Aouedi et al., 2022).

Existen diferentes métodos de clasificación, sin embargo, uno de los métodos más comunes de clasificación es el etiquetado, que puede ser binario, multiclase o multietiquetado, implicando distintos niveles de complejidad en la asignación de etiquetas a los datos (Andersen & Hansson, 2023).

Clasificación Binaria: Se refiere a un escenario en el que solo se consideran dos clases posibles. Por ejemplo, en la clasificación de tráfico de red, se puede clasificar el tráfico como normal o ataque. Se trata de una distinción simple entre dos categorías.

Clasificación Multiclase: Implica que la muestra de entrada puede ser clasificada en una sola clase dentro de un grupo de clases más amplio. Por ejemplo, al clasificar

el tráfico de red, se podrían tener clases como juego, streaming y chat, y la muestra de tráfico se asignaría a una de estas tres clases. Este enfoque permite la clasificación en múltiples categorías, y la cantidad de clases puede ampliarse dependiendo del conjunto de datos utilizado.

Clasificación Multi-etiqueta: Permite que una muestra de entrada se clasifique en más de una clase dentro del grupo de clases. Siguiendo con el ejemplo del tráfico de red, se podría clasificar el tráfico como Skype y al mismo tiempo asignar una etiqueta adicional para indicar el tipo de tráfico, como video. Esto permite asignar múltiples etiquetas a una muestra para una clasificación más precisa y detallada.

- ***Modelos de aprendizaje supervisado para regresión:*** Regresión lineal (RL), Regresión de vecinos más cercanos (KNN), Regresión polinómica, Máquinas de vectores de soporte para regresión (SVR), Árboles de decisión para regresión, Bosques aleatorios para regresión, Gradient Boosting Machines (GBM) para regresión, XGBoost para regresión, LightGBM para regresión, CatBoost para regresión, Regresión elástica, Redes neuronales artificiales (ANN) para regresión, Regresión por mínimos cuadrados parciales (PLS), Regresión de mínimos cuadrados parciales (PLSR), Regresión de mínimos cuadrados parciales por kernel (KPLS), Regresión de mínimos cuadrados parciales generalizada (GPLS) (Aouedi et al., 2022).

Aprendizaje no supervisado

El aprendizaje no supervisado, entrena el modelo con datos de entrada sin información previa sobre las salidas esperadas, es decir, datos sin etiquetar. El modelo aprende a encontrar patrones y estructuras en los datos de entrada y puede agruparlos en categorías o clasificarlos en función de su similitud (Aouedi et al., 2022). Con ello también se menciona que las instancias dentro del mismo clúster tienen una mayor

similitud en comparación con las instancias en otros clústeres. Esto se visualiza en la Figura 4 , que muestra se forma gráfica el aprendizaje no supervisado. Ejemplos de algoritmos de aprendizaje no supervisado incluyen el agrupamiento y las mezclas gaussianas. Los modelos de aprendizaje no supervisado son:

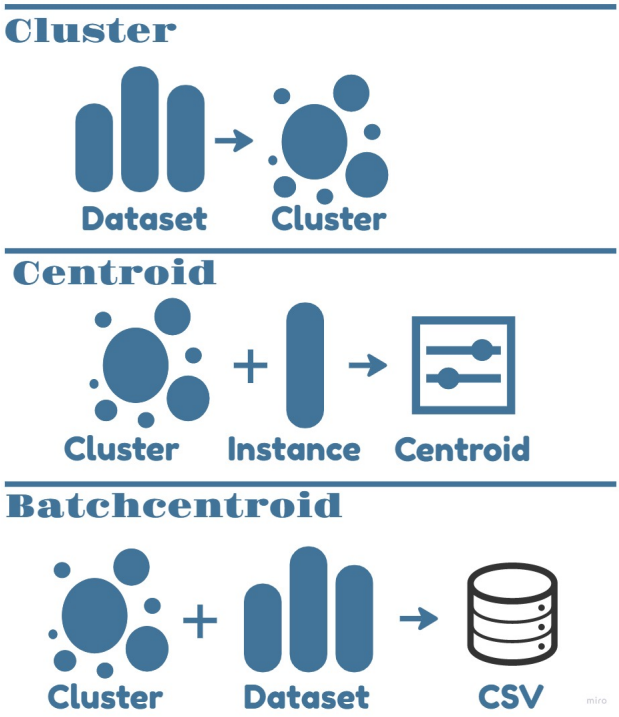
- ***Agrupamiento (Clustering):*** K-Means, Mean Shift, DBSCAN, Agrupamiento jerárquico, Agrupamiento espectral.
- ***Reducción de dimensionalidad:*** Análisis de componentes principales (PCA), Análisis discriminante lineal (LDA), T-SNE (t-Distributed Stochastic Neighbor Embedding), Factorización matricial no negativa (NMF)
- ***Asociación:*** Reglas de asociación (Apriori), FP-Growth
- ***Detección de anomalías:*** Detección de valores atípicos (Outlier detection), Detección de cambios (Change detection), One-Class SVM
- ***Aprendizaje no supervisado profundo:*** Autoencoders, Restricted Boltzmann Machines (RBMs), Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs)
- ***Agrupamiento de texto:*** Latent Dirichlet Allocation (LDA), Non-negative Matrix Factorization (NMF), Hierarchical Dirichlet Process (HDP)

Aprendizaje por refuerzo RL

El aprendizaje por refuerzo, tiene por idea principal la inspiración en los sistemas de aprendizaje biológico. El modelo aprende a tomar decisiones en un entorno específico a través de la experiencia y recibe retroalimentación en tiempo real sobre sus decisiones para mejorar su rendimiento. Es una técnica que permite a un agente aprender su comportamiento interactuando con su entorno y se basa en tres elementos clave:

Figura 4

Aprendizaje no Supervisado.



Nota. Recuperado de BigML, 2014.

observaciones, recompensas y acciones (Aouedi et al., 2022). En otras palabras, el agente de software realiza observaciones y ejecuta acciones dentro de un entorno, recibiendo recompensas a cambio. La Figura 5 ilustra gráficamente estos procesos. A continuación, presentaremos los modelos basados en aprendizaje por refuerzo:

Modelos de aprendizaje por refuerzo: Q-Learning, SARSA (State-Action-Reward-State-Action), DQN (Deep Q-Network), A2C (Advantage Actor-Critic), PPO (Proximal Policy Optimization), DDPG (Deep Deterministic Policy Gradient), TRPO (Trust Region Policy Optimization).

Figura 5

Aprendizaje Por Refuerzo.

Nota. Recuperado de Epicalsoft, 2018.

Metodologías para ciencias de datos

Las metodologías para ciencias de datos son enfoques estructurados que guían la ejecución eficaz de proyectos, es decir, describen pasos, tareas y actividades en el proceso de desarrollo de un proyecto . Al seguir una metodología, se organiza y gestiona el flujo de trabajo. A continuación, se presentan algunas de estas metodologías, cada una con sus etapas correspondientes. (Huber et al., 2019).

SEMMA

SEMMA (Sample Explore Modify Model Asses) es un enfoque ampliamente utilizado en el análisis de datos que se refiere al proceso para llevar a cabo proyectos de minería de datos. Este enfoque se centra en acceder directamente a los datos a analizar y aplicar técnicas de minería de datos sin tener en cuenta los objetivos específicos de un negocio en particular. SEMMA consta de un conjunto de etapas secuenciales para el

desarrollo de modelos predictivos, lo que lo hace un proceso fácil de comprender y seguir. Estas etapas se diseñan para guiar de manera sistemática el proceso de explotación de la información y facilitar el desarrollo exitoso de proyectos de análisis de datos. En la Figura 6, se pueden observar sus fases y a continuación se describen cada una de ellas (Azevedo & Santos, 2008).

- Muestreo **(Sample)**: es fundamental trabajar con un conjunto de datos inicial lo suficientemente grande para extraer una muestra representativa que se pueda manipular eficazmente. La selección de un conjunto de datos adecuado es esencial, ya que debe contener la información necesaria para realizar un análisis significativo y capturar la variabilidad de los datos, garantizando la validez de los resultados obtenidos.

- Exploración **(Explore)**: la exploración de datos consiste en examinar minuciosamente los datos en busca de información relevante. Esto se logra a través de la visualización de datos para identificar patrones, tendencias y relaciones significativas. La exploración de datos proporciona una comprensión más profunda de la información obtenida y tiene como objetivo descubrir ideas y generar una comprensión inicial de los datos antes de avanzar a las etapas posteriores del análisis.

- Modificación **(Modify)**: en esta fase, se lleva a cabo la transformación de los datos, que incluye tareas como la limpieza de datos, la detección y el manejo de valores atípicos, la gestión de datos faltantes y la realización de transformaciones en las variables. Además, puede implicar la selección de variables relevantes según sea necesario para el análisis. El propósito de esta fase es asegurar que los datos estén en un formato apropiado y óptimo para el posterior desarrollo y evaluación de modelos.

- Modelado **(Model)**: en esta fase utiliza los datos preparados en la etapa anterior para entrenar y validar modelos. Estos modelos pueden ser algoritmos de clasificación, regresión u otras técnicas, dependiendo del objetivo del análisis. Se construyen modelos predictivos mediante técnicas de aprendizaje automático, donde el objetivo es que el algoritmo busque automáticamente patrones y relaciones en los datos para realizar predicciones o clasificaciones basadas en estos modelos. El objetivo final es obtener un modelo preciso que pueda utilizarse para predecir resultados esperados en datos futuros.

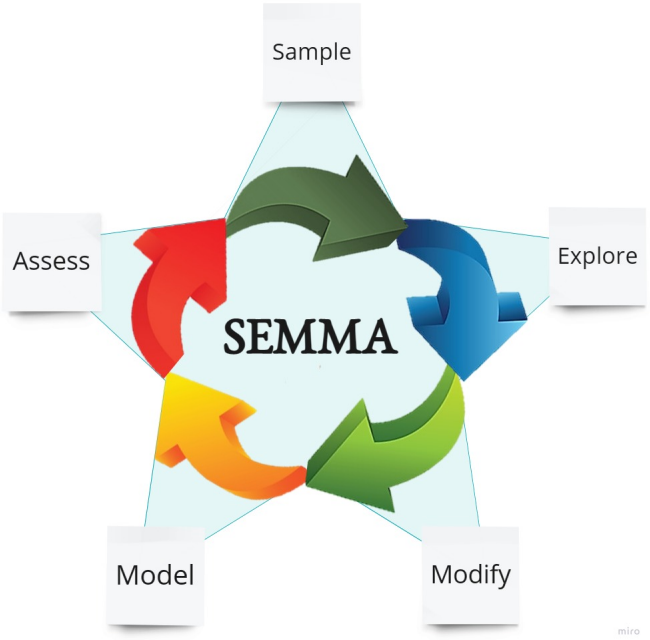
- Evaluación **(Assess)**: se validan los modelos construidos analizando las métricas de rendimiento, como la precisión, sensibilidad o el error cuadrático medio, para determinar la calidad y utilidad del modelo aplicado. Se evalúa el funcionamiento del modelo en términos de su capacidad predictiva y que cumpla con los objetivos establecidos. Con ello se determina si el modelo es lo suficientemente sólido y confiable para ser utilizado en la toma de decisiones o en aplicaciones prácticas. La evaluación también puede incluir la realización de pruebas adicionales para verificar su estabilidad y fiabilidad.

CRISP-DM

CRISP-DM (Cross Industry Standard Process for Data Mining) es una metodología ampliamente utilizada en el campo de la ciencia de datos y la minería de datos (Wirth & Hipp, 2000). Esta metodología abarca las fases de un proyecto, sus tareas correspondientes y sus resultados (Huber et al., 2019). Es importante destacar que CRISP-DM es un enfoque iterativo, lo que significa que es posible retroceder a fases anteriores según sea necesario, en función de los resultados y descubrimientos obtenidos durante el proceso. Esto permite un enfoque flexible y adaptable a medida que avanza el proyecto de ciencia de datos (Wirth & Hipp, 2000), (Castaneda Herrera et al., 2021).

Figura 6

Fases de la metodología SEMMA.



Nota. Recuperado de SAS Institute Documentation, 2023.

CRISP-DM consta de seis fases interconectadas que guían el desarrollo de un proyecto de ciencia de datos, como se puede observar en la Figura 7. A continuación, se detallan estas fases.

- **Comprensión empresarial:** denominada fase inicial, se dice que trabaja en la comprensión del problema empresarial y los objetivos del proyecto. En el cual se definen los requisitos, se establecen los criterios de éxito y se determina cómo los resultados de la ciencia de datos, pueden contribuir a la resolución del problema.

- **Comprensión de los datos:** se busca identificar, recopilar, comprender los conjuntos de datos que pueden ayudar con el objetivo de la empresa. En esta fase se recopila datos iniciales que sean necesarios para su análisis. Posteriormente se

examinan los datos desde su formato, numero de registros, identificación de relaciones entre datos y su verificación de calidad.

- **Preparación de los datos:** en esta fase, se realizan tareas de limpieza, transformación y manipulación de datos para prepararlos adecuadamente para el modelado. Esto incluye la selección de las características más relevantes, el tratamiento de valores nulos o atípicos, la normalización de los datos y otros procesos necesarios.

- **Modelado:** en esta etapa, se seleccionan y aplican técnicas de modelado adecuadas para resolver el problema en cuestión. Se construyen modelos predictivos o descriptivos utilizando algoritmos de ML, estadísticas u otras técnicas relevantes. Se ajustan los modelos y se evalúa su desempeño.

- **Evaluación:** en esta fase, se evalúan los modelos desarrollados durante la etapa de modelado. Se verifica su precisión, se examinan los resultados y se realizan pruebas de validación para garantizar que los modelos sean sólidos y útiles. Se toma en cuenta la retroalimentación del cliente o del equipo para mejorar los modelos si es necesario.

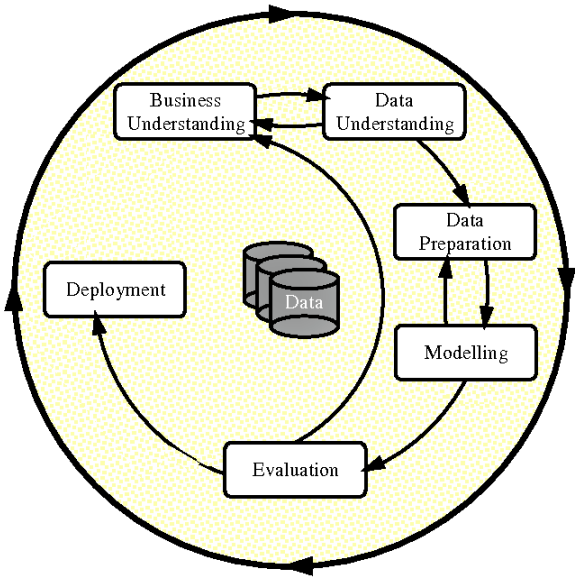
- **Despliegue:** en esta última fase, se implementan los modelos en el entorno operativo y se lleva a cabo la integración con las aplicaciones o sistemas existentes. Se documenta el proceso y se presenta el informe final con los resultados obtenidos. Además, se establecen medidas de seguimiento y se planifica el mantenimiento continuo de los modelos.

KDD

La metodología basada en el Descubrimiento de Conocimientos en Bases de Datos (KDD, por sus siglas en inglés, Knowledge Discovery in Databases) es un proceso

Figura 7

Fases de la metodología CRISP-DM.



Nota. Recuperado de Wirth y Hipp, 2000.

iterativo y cíclico utilizado en la minería de datos para descubrir conocimiento valioso a partir de grandes conjuntos de datos. Esta metodología sigue un proceso sistemático que comienza por comprender el problema en cuestión y luego procesa los datos disponibles. El proceso de KDD involucra varias etapas en las que se puede obtener información valiosa que ayuda a comprender mejor los datos y tomar decisiones fundamentadas (Fayyad et al., 1996). En la Figura 8, se mencionan estas fases, y a continuación, se describen en detalle.

- **Selección:** en esta fase se lleva a cabo la identificación de los conjuntos de datos relevantes para el proceso de descubrimiento de conocimiento. Se establecen los criterios necesarios para determinar qué conjuntos de datos deben ser incluidos en el análisis. Su selección de ser cuidadosa ya que, es fundamental para garantizar que se disponga de la información necesaria y relevante de manera efectiva.

- **Preprocesamiento y limpieza:** se lleva a cabo la limpieza y transformación de los datos con el objetivo de prepararlos para el análisis. Se realiza la eliminación de datos irrelevantes, así como el manejo de valores faltantes, erróneos, nulos, entre otros para asegurar la integridad de los datos. El preprocesamiento de los datos es una etapa crítica que garantiza la calidad y la adaptación de los datos ayudando a mejorar la precisión y la eficacia de los resultados obtenidos.

- **Transformación y reducción:** se preparan los datos de manera que sean más convenientes para las etapas posteriores del proceso de minería de datos. Se aplican diversas técnicas con el fin de transformar los datos en una forma más apropiada. Se puede realizar técnicas de selección de características para identificar y mantener únicamente las variables más relevantes y significativas para el análisis, lo cual ayuda a reducir la complejidad mejorando la eficiencia del proceso.

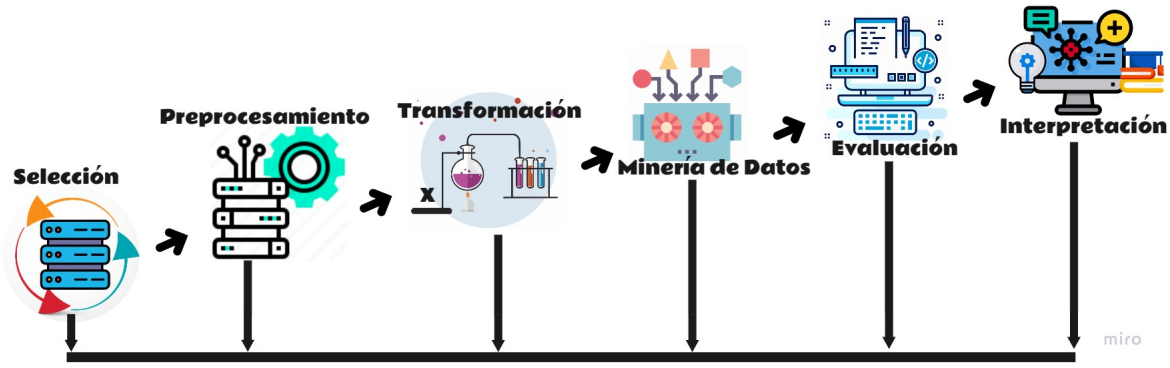
- **Mineria de datos:** se lleva a cabo el análisis y la exploración de los datos utilizando diversas técnicas de minería de datos. Se procede a aplicar algoritmos de aprendizaje automático, donde se descubre patrones o relaciones en los datos que puedan proporcionar conocimiento y comprensión. Esta técnica revela información significativa que ayuda a la toma de decisiones extrayendo el valor de los datos analizados. Se enfoca en explorar y analizar los datos con el fin de extraer conocimiento útil y significativo.

- **Evaluación:** se analiza la calidad empleando diversas métricas de rendimiento como la exactitud, precisión, sensibilidad, especificidad u otras métricas relevantes. Se determina la eficacia de los modelos evaluados en términos de capacidad para proporcionar información útil. Es fundamental para asegurar la calidad y la validez de los resultados obtenidos.

- **Interpretación:** se realiza la deducción de los resultados obtenidos y se presentan de manera clara y comprensible, de modo que puedan ser utilizados de manera efectiva en la toma de decisiones. (Fayyad et al., 1996).

Figura 8

Fases de la metodología KDD.



Nota. Recuperado de Fayyad et al., 1996.

Comparativa de las metodologías para ciencias de datos

Dentro de las tres metodologías antes mencionadas, se establecen los criterios por los cuales se diferencian entre su enfoque, sus fases y énfasis. Las fases de la metodología KDD y SEMMA tienen una equivalencia entre ellas con relación a la de CRISP-DM y se puede deducir que esta última metodología es más completa y puede ser más efectiva en su implementación. A continuación, en la Tabla 2 se muestran sus respectivas diferencias entre estas tres metodologías (Azevedo & Santos, 2008).

Tabla 2

Resumen comparativo de las metodologías KDD, CRISP-DM y SEMMA.

Metodologías	Enfoque	Fases	Énfasis
KDD	Se centra en el descubrimiento de conocimiento útil a partir de grandes conjuntos de datos.	Selección de datos, preprocesamiento, transformación, minería de datos y evaluación.	Su énfasis se centra en la etapa de minería de datos y el descubrimiento de patrones interesantes y útiles.
CRISP-DM	Se enfoca en el desarrollo de proyectos de minería de datos y ciencia de datos de manera estructurada y sistemática	Comprensión empresarial, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue.	Se enfatiza en la comprensión del problema y los datos antes de avanzar hacia las etapas de modelado y evaluación.
SEMMA	Se centra en el desarrollo de modelos predictivos y descriptivos precisos	Muestra, exploración, modificación, modelado y evaluación.	Su énfasis es en la modificación de los datos para mejorar la calidad y la precisión de los modelos.

Capítulo III: Materiales y métodos

Metodología de Desarrollo

El desarrollo del presente proyecto se ha llevado a cabo utilizando la metodología SEMMA. La elección de esta metodología se alinea perfectamente con nuestro objetivo principal, que es definir un modelo de clasificación del tráfico en Redes Definidas por Software mediante técnicas de aprendizaje automático. El enfoque estructurado de SEMMA nos permitirá abordar este objetivo de manera efectiva y eficiente. Esta metodología consta de una serie de etapas secuenciales, a saber: muestra, exploración, modificación, modelado y evaluación, que guían el proceso de análisis de datos. En la siguiente sección, se presenta el desarrollo del proyecto basado en esta metodología.

Fase de muestreo

Los conjuntos de datos son importantes en esta fase, por lo tanto, para obtener un conjunto de datos se tomó como referencia una muestras de tráfico real, con el fin de analizar y comprender mejor el comportamiento de la red, identificar patrones, tendencias, entre otros. Para ello se realizó una investigación en las bases científicas tales como: Scopus, IEEE y WebOfScience. La Tabla 3 muestra una comparación de los artículos que contenían los conjuntos de datos con tráfico real, es decir, que se generaron los flujos de tráfico.

Tabla 3

Comparación de los conjuntos de datos implementados en diferentes artículos.

Referencia	Nombre	Clases	Tamaño	Formato
(Castaneda Herrera et al., 2021)	Datavideo1 y Datavideo2 (2021)	3	7.8GB	PCAP
(Draper-Gil. et al., 2016)	ISCX-VPN-NonVPN (2016)	14	28GB	PCAP
(Naas & Fesl, 2023)	Encrypted VPN conjunto de datos (2022)	5	18GB	JSON
(Moore et al., 2005)	Moore (2005)	10	No publicado	ARFF

Selección del conjunto de datos: El conjunto de datos con el que se va a trabajar es *ISCX-VPN-NoVPN*. Este ofrece una amplia variedad de tipos de tráfico, lo que permite abordar diferentes aspectos y escenarios. Sus archivos están en formato PCAP, lo que indica que los datos no están procesados. Cabe recalcar que los datos no se encuentran en un solo archivo, sino que están divididos en cinco archivos zip. De ellos, tres archivos son NonVPN y dos archivos son VPN, con un tamaño total de 28 GB de datos en bruto. Se puede acceder a la descarga en la página oficial.

Origen del conjunto de datos: El conjunto de datos se generó a partir de tráfico real encriptado, garantizando la autenticidad de los flujos capturados y ofreciendo un enfoque sólido para clasificación de flujo de tráfico. Además, los autores utilizaron herramientas conocidas para capturar dichos flujos de tráfico (Draper-Gil. et al., 2016).

La tabla 4 presenta una lista completa de los tipos de tráfico y aplicaciones

incluidas en el conjunto de datos, considerando sesiones regulares a través de VPN (Draper-Gil. et al., 2016). Esta diversidad de categorías contribuye a la amplitud y la representatividad del conjunto de datos, lo que facilita un análisis más completo y robusto de los diferentes tipos de tráfico.

Tabla 4

Lista de protocolos y aplicaciones capturadas.

Tráfico	Contenido
Navegación web	Firefox y Chrome
Correo electrónico	SMPTS, POP3S e IMAPS
Chat	ICQ, AIM, Skype, Facebook y Hangouts
Streaming	Vimeo y Youtube
Transferencia de archivos	Skype, FTPS y SFTP mediante Filezilla y un servicio externo
VoIP	Llamadas de voz de Facebook, Skype y Hangouts (1h de duración)
P2P	Torrent y Transmisión (Bittorrent)

Extracción de archivos PCAP: Los archivos PCAP proporcionados por el conjunto de datos fueron procesados para extraer los flujos de red. Para la extracción de los archivos en formato PCAP, se empleó la herramienta CICFlowMeter, escrita en el lenguaje de programación Java, la cual fue desarrollada por el laboratorio Canadian Institute for Cybersecurity (CIC) en la Universidad de Nuevo Brunswick, Canadá (Lashkari et al., 2017). Esta herramienta tiene la capacidad de extraer

características basadas en el tiempo de los flujos de red, lo cual es crucial para calcular estadísticas relacionadas con el tiempo. Los flujos generados son calculados en modo bidireccional, lo que significa que se capturan y analizan los paquetes de red que se transmiten en ambas direcciones entre dos puntos de comunicación. Esto implica tener en cuenta tanto los paquetes que se envían desde una fuente a un destino como los paquetes que se envían desde el destino hacia la fuente.

Al calcular los flujos en modo bidireccional, se obtiene una visión completa y más precisa de la comunicación y el intercambio de datos entre los dispositivos de red (Draper-Gil. et al., 2016). Por lo tanto, esta es una herramienta especializada que permite analizar de manera eficiente los flujos de red y obtener datos estructurados en un formato fácilmente manipulable para su posterior análisis o utilización. La salida de CICFlowMeter, basada en la actividad de tiempo, se configuró con tiempos de espera de flujo de 15 y 120 segundos para su evaluación, lo que proporcionó más de 80 características estadísticas guardadas en un formato de archivo CSV. Esta información detallada y estructurada facilita el análisis y la comprensión de los flujos de red capturados.

Fase de exploración

En esta fase, se lleva a cabo el proceso de comprensión y análisis de los datos extraídos en los archivos CSV. Se realiza una serie de tareas para explorar los datos y obtener una visión general de su estructura, contenido, características y etiquetas.

Inspección inicial: Se realizó el análisis visual y exploratoria de los tipos de datos y los valores almacenados en cada característica. Este análisis resulta fundamental para lograr una comprensión más intensa de los datos y, a partir de ello, determinar qué características son relevantes para las etapas subsiguientes (Gutiérrez Galeano et al., 2022).

La comprensión de las características del conjunto de datos es fundamental para obtener una visión completa y significativa de la información. Las características del conjunto de datos otorgan detalles importantes sobre las variables incluidas, como su tipo, escala, distribución y relación entre sí. Al comprender estas características, se obtiene una idea clara de la naturaleza de los datos, cabe recalcar que, todas las características están en formato numérico. En la tabla 5 se identifican las características que posee el conjunto de datos.

Tabla 5

Lista de características del conjunto de datos.

Característica	Descripción
Src IP	Dirección IP de origen
Src Port	Puerto de origen
Dst IP	Dirección IP de destino
Dst Port	Puerto de destino
Protol	Protocolo
Flow Duration	Duración del flujo en µs
Tot Fwd Pkts	Total de paquetes enviados del origen al destino
Tot Bwd Pkts	Total de paquetes enviados del destino al origen
TotLen Fwd Pkts	Total de tamaño del paquete enviado del origen al destino
TotLen Bwd Pkts	Total de tamaño del paquete enviado del destino al origen
Fwd Pkt Len Max	Tamaño máximo del paquete enviado del origen al destino
Fwd Pkt Len Min	Tamaño mínimo del paquete enviado del origen al destino
Fwd Pkt Len Mean	Tamaño medio del paquete enviado del origen al destino
Fwd Pkt Len Std	Desviación estándar del paquete enviado del origen al destino

Continúa en la siguiente página...

Tabla 5 – continuación de la página anterior

Característica	Descripción
Bwd Pkt Len Max	Tamaño máximo del paquete enviado del destino al origen
Bwd Pkt Len Min	Tamaño mínimo del paquete enviado del destino al origen
Bwd Pkt Len Mean	Tamaño medio del paquete enviado del destino al origen
Bwd Pkt Len Std	Desviación estándar del paquete enviado del destino al origen
Flow Byts/s	Bytes del flujo por segundo
Flow Pkts/s	Paquetes del flujo por segundo
Flow IAT Mean	Tiempo medio entre dos paquetes enviados en el mismo flujo
Flow IAT Std	Desviación estándar entre dos paquetes enviados en el mismo flujo
Flow IAT Max	Tiempo máximo entre dos paquetes enviados en el mismo flujo
Flow IAT Min	Tiempo mínimo entre dos paquetes enviados en el mismo flujo
Fwd IAT Tot	Total de tiempo entre dos paquetes enviados del origen al destino
Fwd IAT Mean	Tiempo medio de dos paquetes enviados del origen al destino
Fwd IAT Std	Desviación estándar de dos paquetes enviados del origen al destino
Fwd IAT Max	Tiempo máximo de dos paquetes enviados del origen al destino
Fwd IAT Min	Tiempo mínimo de dos paquetes enviados del origen al destino
Bwd IAT Tot	Total de tiempo entre dos paquetes enviados del destino al origen
Bwd IAT Mean	Tiempo medio de dos paquetes enviados del destino al origen

Continúa en la siguiente página...

Tabla 5 – continuación de la página anterior

Característica	Descripción
Bwd IAT Std	Desviación estándar de dos paquetes enviados del destino al origen
Bwd IAT Max	Tiempo máximo de dos paquetes enviados del destino al origen
Bwd IAT Min	Tiempo mínimo de dos paquetes enviados del destino al origen
Fwd PSH Flags	Veces que la bandera PSH fue establecida en los paquetes enviados del origen al destino
Bwd PSH Flags	Veces que la bandera PSH fue establecida en los paquetes enviados del destino al origen
Fwd URG Flags	Veces que la bandera URG fue establecida en los paquetes enviados del origen al destino
Bwd URG Flags	Veces que la bandera URG fue establecida en los paquetes enviados del destino al origen
Fwd Header Len	Total de bytes utilizados para la cabecera para paquetes enviados del origen al destino
Bwd Header Len	Total de bytes utilizados para la cabecera para paquetes enviados del destino al origen
Fwd Pkts/s	Paquetes enviados por segundo del origen al destino
Bwd Pkts/s	Paquetes enviados por segundo del destino al origen
Pkt Len Min	Longitud mínima de un paquete
Pkt Len Max	Longitud máxima de un paquete
Pkt Len Mean	Longitud media de un paquete
Pkt Len Std	Desviación estándar media de un paquete

Continúa en la siguiente página...

Tabla 5 – continuación de la página anterior

Característica	Descripción
Pkt Len Var	Varianza de la longitud de un paquete
FIN Flag Cnt	Paquetes con la bandera establecida FIN
SYN Flag Cnt	Paquetes con la bandera establecida SYN
RST Flag Cnt	Paquetes con la bandera establecida RST
PSH Flag Cnt	Paquetes con la bandera establecida PUSH
ACK Flag Cnt	Paquetes con la bandera establecida ACK
URG Flag Cnt	Paquetes con la bandera establecida URG
CWE Flag Count	Paquetes con la bandera establecida CWR
ECE Flag Cnt	Paquetes con la bandera establecida ECE
Down/Up Ratio	Ratio de carga y descarga
Pkt Size Avg	Tamaño promedio de un paquete
Fwd Seg Size Avg	Tamaño promedio de los paquetes enviados del origen al destino
Bwd Seg Size Avg	Tamaño promedio de los paquetes enviados del destino al origen
Fwd Byts/b Avg	Tasa promedio de bytes del volumen de paquetes enviados del origen al destino
Fwd Pkts/b Avg	Tasa promedio de paquetes de los que han sido enviados del origen al destino
Fwd Blk Rate Avg	Tasa promedio del volumen de paquetes enviados del origen al destino

Continúa en la siguiente página...

Tabla 5 – continuación de la página anterior

Característica	Descripción
Bwd Byts/b Avg	Tasa promedio de bytes del volumen de paquetes enviados del destino al origen
Bwd Pkts/b Avg	Tasa promedio de paquetes de los que han sido enviados del destino al origen
Bwd Blk Rate Avg	Tasa promedio del volumen de paquetes enviados del destino al origen
Subflow Fwd Pkts	Promedio de paquetes en un subflujo en paquetes enviados del origen al destino
Subflow Fwd Byts	Promedio de bytes en un subflujo en paquetes enviados del origen al destino
Subflow Bwd Pkts	Promedio de paquetes en un subflujo en paquetes enviados del destino al origen
Subflow Bwd Byts	Promedio de bytes en un subflujo en paquetes enviados del destino al origen
Init Fwd Win Byts	Total de bytes enviados en una ventana inicial en los paquetes enviados del origen al destino
Init Bwd Win Byts	Total de bytes enviados en una ventana inicial en los paquetes enviados del destino al origen
Fwd Act Data Pkts	Cantidad de paquetes con al menos 1byte de carga útil TCP de datos en paquetes enviados del origen al destino
Fwd Seg Size Min	Tamaño mínimo de segmento observado en paquetes enviados del origen al destino

Continúa en la siguiente página...

Tabla 5 – continuación de la página anterior

Característica	Descripción
Active Mean	Tiempo medio en el que el flujo estuvo activo
Active Std	Desviación estándar del tiempo en el que el flujo estuvo activo
Active Max	Tiempo máximo en el que el flujo estuvo activo
Active Min	Tiempo mínimo en el que el flujo estuvo activo
Idle Mean	Tiempo medio en el que el flujo estuvo inactivo
Idle Std	Desviación estándar del tiempo en el que el flujo estuvo inactivo
Idle Max	Tiempo máximo en el que el flujo estuvo inactivo
Idle Min	Tiempo mínimo en el que el flujo estuvo inactivo

Es importante comprender las etiquetas asociadas a la variable objetivo, ya que nos permitirá interpretar el significado de cada una de ellas dentro del conjunto de datos. Las etiquetas proporcionan información descriptiva y contextual sobre qué representa cada variable, con esto se puede realizar un análisis más preciso y relevante al identificar patrones, tendencias, y tomar decisiones informadas basadas en la interpretación de los datos.

Granularidad en las etiquetas: El principal enfoque del autor del conjunto de datos es clasificar los tipos de tráfico de forma general, como navegación web, correo electrónico, chat, streaming, transferencia de archivos, VoIP y P2P. Sin embargo, tras realizar una investigación y análisis exhaustivos, se ha determinado que es posible aplicar un procesamiento de granularidad alta con el fin de analizar los datos en un nivel de detalle más fino. Esto implica que las etiquetas se vuelvan más específicas y detalladas, lo que permite descubrir patrones y detalles más precisos (Chen et al., 2018). Debido a esto, se logró recopilar un total de 23 aplicaciones que se utilizaron para capturar los flujos de

tráfico de red. En la Tabla 6 se pueden visualizar las aplicaciones que surgieron como resultado de la implementación del enfoque de granularidad.

Tabla 6

Lista de etiquetas del conjunto de datos.

Aplicación	Tipo	Instancias 120s	Instancias 15s
AIM chat	Texto en tiempo real	596	1033
Email Client	Texto	5640	6976
Facebook Audio	Voz sobre IP	54191	62544
Facebook Chat	Texto en tiempo real	1885	2285
Facebook video	Video en tiempo real	633	1226
FTPS	Archivo de transferencia	1501	2045
Gmail chat	Texto en tiempo real	615	1053
Hangouts Audio	Voz sobre IP	61899	72397
Hangouts Chat	Texto en tiempo real	3509	4033
Hangouts Video	Video en tiempo real	1921	2882
ICQ	Texto en tiempo real	601	1131
Netflix	Video	1059	2084
SCP	Archivo de transferencia	8046	9546
SFTP	Archivo de transferencia	303	449
Skype Audio	Voz sobre IP	24037	27756
Skype Chat	Texto en tiempo real	4883	6703
Skype File	Archivo de transferencia	36792	42640
Skype Video	Video en tiempo real	1062	1594
Spotify	Música	898	1498

Continúa en la siguiente página...

Tabla 6 – continuación de la página anterior

Aplicación	Tipo	Instancias 120s	Instancias 15s
BitTorrent	P2P	709	777
Vimeo	Video	1038	1752
VoIP Buster	Voz sobre IP	4408	6217
YouTube	Video	1845	2962

Fase de modificación

En esta fase, se preparó y transformó el conjunto de datos, realizando cambios o modificaciones necesarias en los datos para disponerlos adecuadamente antes de construir y entrenar un modelo. Estas modificaciones son necesarias para garantizar la calidad de los datos antes de proceder a la etapa de modelado.

Limpieza de datos: El proceso de limpieza de datos es fundamental en el análisis de datos, ya que busca garantizar la calidad y la integridad de los datos utilizados. En esta etapa, se tratan los valores faltantes, se eliminan datos duplicados o inconsistentes, se manejan valores atípicos y se corrigen errores presentes en los datos. Durante el análisis realizado, se llevó a cabo una minuciosa verificación de la integridad de los datos y se constató la ausencia tanto de valores faltantes como de duplicados. Sin embargo, se identificaron errores de valores infinitos en nueve registros, los cuales fueron eliminados de la muestra. Esto se debió a que, a pesar de aplicarse el método de escalamiento, se encontró que dichos valores eran excesivamente altos y podrían distorsionar los resultados.

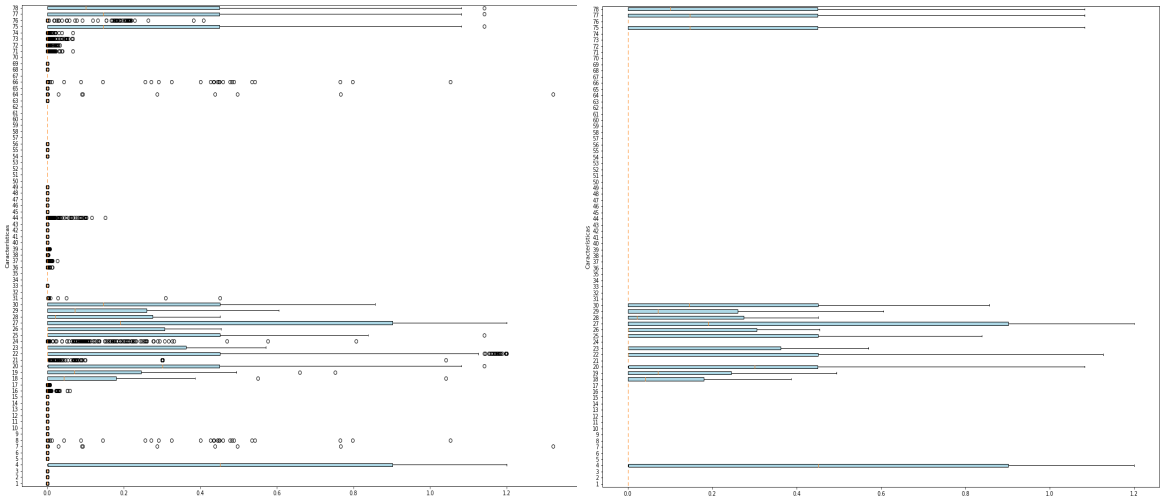
Valores Atípicos: En este estudio se encontró una amplia dispersión de valores atípicos. A menudo, los valores atípicos pueden tener un impacto negativo en el análisis de los modelos estadísticos, ya que pueden distorsionar los resultados y llevar a conclusiones incorrectas. Para abordar este problema, se optó por utilizar el método Tukey

(Benjamini & Braun, 2002), este se basa en la diferencia mínima significativa (DMS) para determinar si existen diferencias significativas entre los grupos, en él se optó por la técnica de imputación media. Este proceso se aplicó para cada uno de los archivos correspondientes a las aplicaciones o clases. Al realizar este proceso en cada grupo, se logra un control de valores atípicos más específico y adaptado a las características de cada clase, lo que conduce a conclusiones más precisas y decisiones más informadas. En la Figura 9 se puede observar el control de los valores atípicos en la aplicación Netflix.

Figura 9

Control de valores atípicos en la aplicación Netflix

(a) *Con outliers* (b) *Sin outliers*

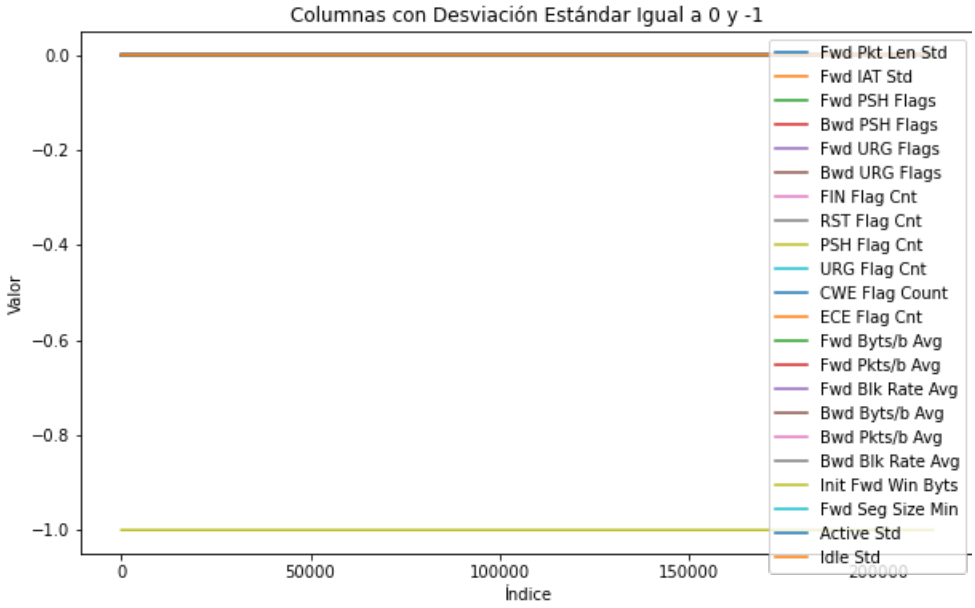


Desviación Estándar: El análisis de la desviación estándar se lleva a cabo en un conjunto de datos para evaluar la dispersión o variabilidad de los valores en relación con la media. Proporciona información sobre cuánto se alejan los valores individuales de la media del conjunto de datos. Durante este análisis, se identificaron columnas con varianza nula y varianza constante. En la Figura 10 se muestra la lista de columnas que serán eliminadas. Este proceso de eliminación contribuye a reducir la dimensionalidad y

complejidad del conjunto de datos, evitando la introducción de ruido innecesario o información redundante en el modelo de aprendizaje automático. Inicialmente se tenían 79 características y con este análisis se han eliminado 22, quedando con un total de 57 características.

Figura 10

Desviación Estándar.



Transformación de datos: Para mejorar la distribución de las variables objetivo y cumplir con los requisitos de los modelos, se procede a realizar una transformación de las etiquetas de la variable objetivo de letras a números enteros. Esta transformación numérica permite que los modelos trabajen de manera más efectiva al utilizar valores numéricos en lugar de categorías de letras. Además, facilita el procesamiento de los datos y la aplicación de técnicas de aprendizaje automático para la clasificación.

Balanceo: El balanceo de clases se utiliza para abordar el problema de desequilibrio en un conjunto de datos. En este caso, el conjunto de datos tuvo un desbalance de clases, lo que puede generar problemas significativos en el rendimiento de

los modelos. Para lograr un equilibrio en la cantidad de ejemplos en casi todas las clases, se aplicó una técnica ampliamente utilizada conocida como submuestreo. Esta estrategia implicó igualar la cantidad de ejemplos en cada clase, es decir, eliminando instancias de las clases mayoritarias con el objetivo de asegurar que cada clase tuviera aproximadamente 3500 ejemplos. Por otro lado, también fueron eliminadas las clases minoritarias que contaban con menos de mil datos (ver la Tabla 6).

Como resultado de este proceso en la Figura 11a se puede observar que se redujeron el número de 23 a 17 clases de 23 para el conjunto de datos con tiempo de espera de flujo en 120s y en la Figura 11b se redujeron el número de clases de 23 a 21 para el conjunto de datos con tiempo de espera de flujo en 15s. Esta estrategia busca asegurar que cada clase tenga una presencia significativa en el conjunto de datos y permitir un aprendizaje más efectivo por parte del modelo, garantizando que el modelo pueda aprender de manera justa y precisa, evitando sesgos o distorsiones en los resultados (Batista et al., 2004).

Selección de características: Se eligen las características más relevantes o informativas de un conjunto de datos para construir el modelo de aprendizaje automático. Esto se hace con el objetivo de mejorar la precisión del modelo, reducir la dimensionalidad de los datos y mejorar la interpretación de los resultados. Para ello se toma en cuenta el análisis de la correlación Pearson, y la ingeniería de características.

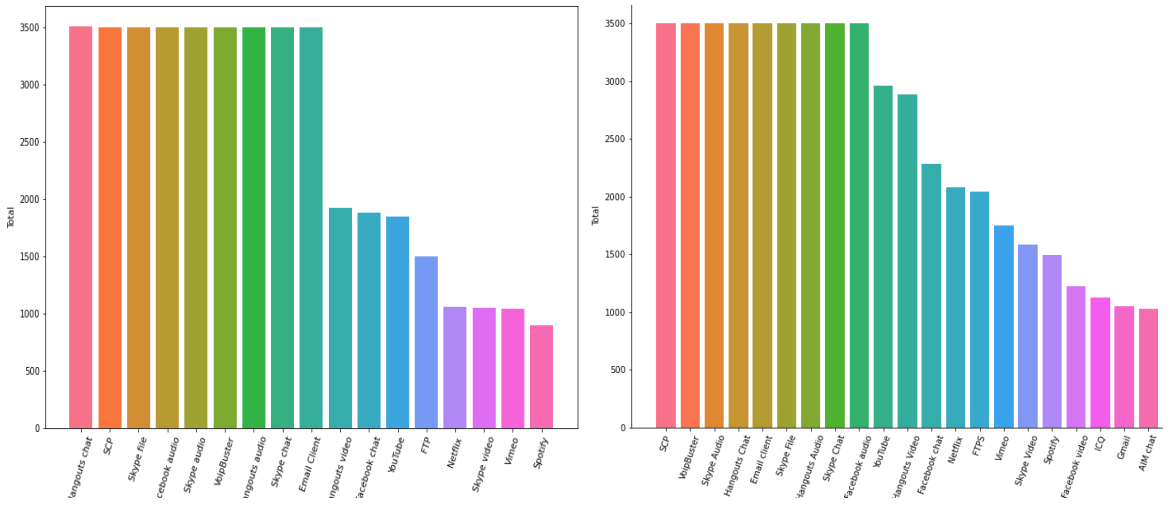
Correlación Pearson: La correlación de Pearson se utiliza como una herramienta en el análisis estadístico y exploratorio de datos, para comprender la relación entre las características y la variable objetivo (Choi et al., 2010). Si existe una correlación fuerte, esto podría indicar que la característica es informativa y podría ser relevante para el modelo.

En la función creada para la búsqueda de características correlacionadas, se consideró un umbral de 0.95. Esto significa que las características con una correlación del

Figura 11

Cantidad de datos por Aplicación

(a) *Para características extraídas con 120s* (b) *Para características extraídas con 15s*



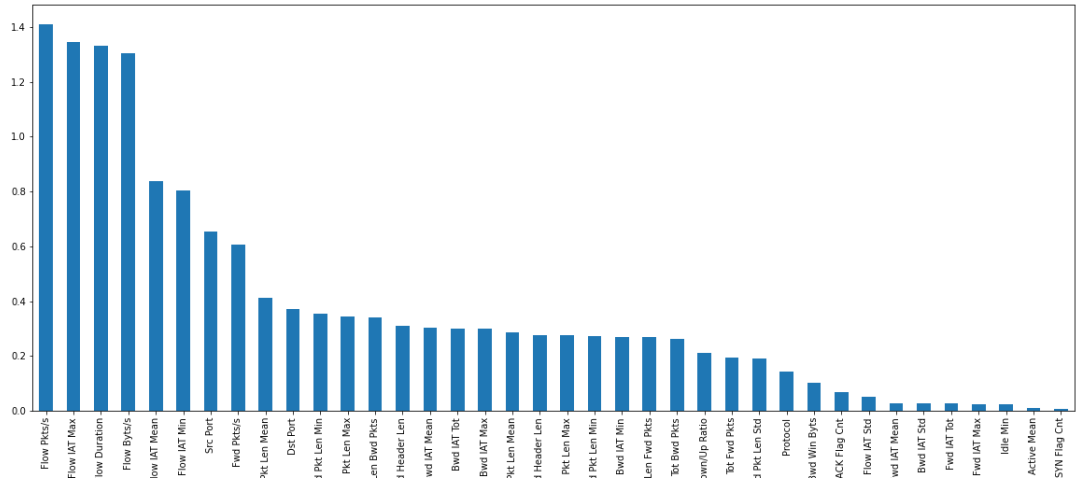
95% o más no tienen ningún efecto en la salida, por lo que se procede a eliminarlas. Esto se debe a que presentan alta redundancia y pueden afectar la interpretación o la estabilidad del modelo. Durante este análisis, se identificaron 18 columnas irrelevantes para el modelo, las cuales fueron eliminadas. Por lo tanto, se tiene un total de 38 características junto con la variable objetivo.

Ingeniería de características: Para este proceso se tomo en cuenta la función de información mutua ya que calcula la información mutua entre cada característica y la variable objetivo en un problema de clasificación. Esta función se basa en el concepto de entropía y la teoría de la información. Cuanto mayor sea el valor de información mutua, mayor será la dependencia o relevancia de la característica con respecto a la variable objetivo (Najafabadi et al., 2015). En el resultado los valores no salen en orden descendente o ascendente, por lo que, usando funciones de Python se los imprime de forma descendente para observar las características mas importantes. Este resultado

también es graficado para para visualizar de mejor forma las características más importantes. En la Figura 12 se puede observar la importancia de las características de forma descendente.

Figura 12

Distribución de puntuación para la selección de características.



Para la selección de características se dio uso de la técnica SelectBest (George, 2017) enfocada en seleccionar las mejores características capaces de contribuir a la predicción o clasificación del modelo. Esta técnica se basa en la utilización de pruebas estadísticas para evaluar la relación entre cada característica y la variable objetivo. En la selección final de características, se ha considerado un umbral de valor de 0.30 para la información mutua. Como resultado, se obtuvieron dos grupos de características: el primer grupo con 15 características y el segundo grupo con 25 características (ver la Tabla7), estos dos grupos serán implementados en los tiempo de espera con 120 y 15 segundos. El propósito de esta evaluación es investigar si el número de características influye en los resultados de precisión. Analizar la variación de precisión entre estos dos grupos nos permitirá comprender mejor la importancia relativa de cada conjunto de características en el rendimiento del modelo.

Tabla 7

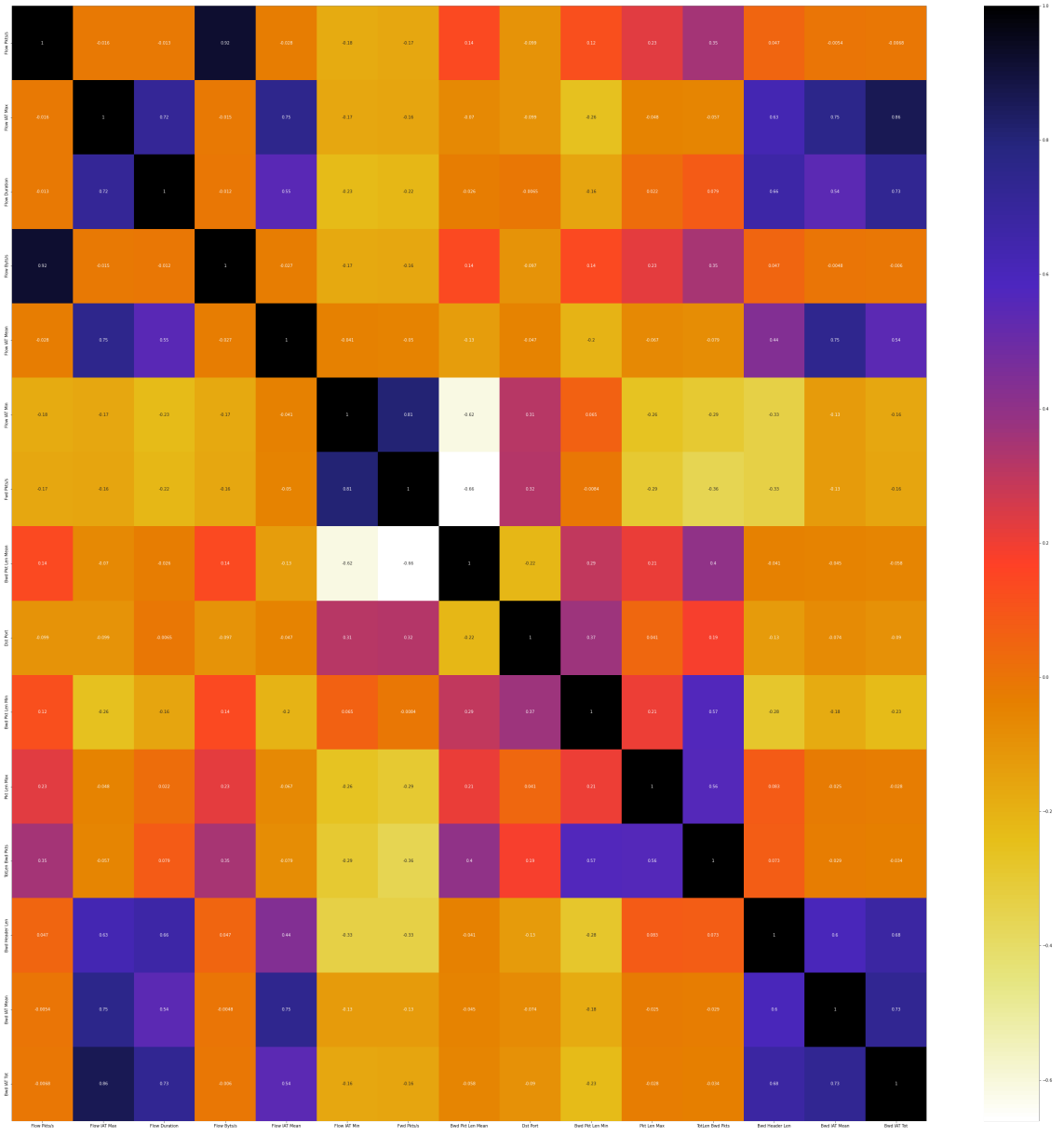
Grupos de características

Grupo	Características
CG1-120s y CG1-15s	Flow Pkts/s, Flow IAT Max, Flow Duration, Flow Byts/s, Flow IAT Mean, Flow IAT Min, Fwd Pkts/s, Bwd Pkt Len Mean, Dst Port, Bwd Pkt Len Min, Pkt Len Max, TotLen Bwd Pkts, Bwd Header Len, Bwd IAT Mean
CG2-120s y CG2-15s	Flow Pkts/s, Flow IAT Max, Flow Duration, Flow Byts/s, Flow IAT Mean, Flow IAT Min, Fwd Pkts/s, Bwd Pkt Len Mean, Dst Port, Bwd Pkt Len Min, Pkt Len Max, TotLen Bwd Pkts, Bwd Header Len, Bwd IAT Mean, Bwd IAT Max, Bwd IAT Tot, Fwd Pkt Len Mean, Fwd Header Len, Fwd Pkt Len Max, Fwd Pkt Len Min, Bwd IAT Min, TotLen Fwd Pkts, Tot Bwd Pkts, Down/Up Ratio, Tot Fwd Pkts

En la Figura 13 se puede observar la matriz de correlación que representa las relaciones lineales entre las características finales del primer grupo. Esta visualización es útil para analizar la fuerza y dirección de la relación entre cada par de variables. La matriz de correlación proporciona información importante sobre posibles dependencias o patrones entre las características, lo que nos ayuda a comprender mejor la estructura de los datos y su relevancia para los modelos.

Figura 13

Matriz de correlación del primer grupo de características.



Fase de modelado

Esta fase se centra en la construcción y entrenamiento de modelos de aprendizaje automático utilizando los datos previamente preparados en la fase de modificación.

Durante esta etapa, se realizó una investigación exhaustiva basada en artículos relevantes que aplican algoritmos ML. Se realizaron exploraciones donde, el enfoque principal fue determinar los modelos más relevantes y ampliamente aplicados en el contexto específico de la clasificación de aplicaciones. Para el proceso de selección como se puede observar en la Tabla 8, se tuvieron en cuenta múltiples factores, entre ellos la eficacia en la clasificación, el número total de características y el número total de clases. Estos criterios permitieron evaluar y comparar detenidamente las opciones de modelado disponibles.

Tabla 8

Clasificación de Tráfico por tipo de Aplicación basado en Aprendizaje Automático.

Ref.	Técnica ML	Caracteríticas	Etiquetas	Precisión
(Abdulazzaq & Demirci, 2020)	k-NN, SVM, DT, RF, DNN y CNN	13	3	kNN:98% RF:97% CNN:95% DNN:94% DT:88% SVM:86%
(Fan & Liu, 2017)	SVM y K-Means	30	8	SVM:98.7% K-M:88%

Continúa en la siguiente página...

Tabla 8 – continuación de la página anterior

Ref.	Técnica ML	Caracteríticas	Etiquetas	Precisión
(Zaki & Chin, 2019)	DT, k-NN, NB y SVM	Seleccionadas por el algoritmo	12	DT:99.39% kNN:98.34% NB:96.71% SVM:94.65%
(Xu et al., 2018)	DNN, SVM, k-NN y DT	9	4	DNN:88% DT:85% SVM:80% kNN:79%
(Owusu & Nayak, 2020)	k-NN, RF y DT	6	4	DT:87.2% RF: 85.1% kNN: 79.5%
(Mondal et al., 2021)	k-NN, RF y DT	7	10	kNN:97.14% RF:96.69% DT:95.8%
(Su et al., 2023)	ResNet Mejorado	Seleccionadas por el algoritmo	16	ResNet:99.93%

Selección de los algoritmos de modelado: Al analizar la variedad de modelos aplicados en cada artículo, se puede inferir que los modelos más utilizados para la

clasificación en tipos de aplicaciones son SVM, DT, RF y KNN. Estos modelos han demostrado ser efectivos y ampliamente adoptados en el campo de la clasificación de aplicaciones debido a su capacidad para manejar conjuntos de datos complejos y generar resultados precisos en términos de categorización y etiquetado de aplicaciones.

Hardware y Software: Para la implementación de los modelos en la ejecución experimental, se utilizó el servidor RIG, el cual tenía las siguientes especificaciones: un procesador AMD Ryzen Threadripper 2920X de 12 núcleos, una capacidad de almacenamiento interno de 500 GB en un SSD Crucial M.2 NVMe, 64 GB de RAM Crucial Ballistix DDR4-3000 y una tarjeta aceleradora GPU Phantom Gaming X Radeon VII de 16 GB. El experimento se llevó a cabo en el sistema operativo Ubuntu 18.04 de 64 bits, utilizando Python 3.5.2 en Jupyter Notebook 6.0.2 y las bibliotecas NumPy, pandas y scikit-learn.

Dado que el modelo SVM puede tener un tiempo de ejecución lento, se decidió aprovechar el Servidor de la Universidad para realizar los cálculos y obtener resultados más eficientes. Esta estrategia permitió acelerar el proceso de entrenamiento y predicción, optimizando el rendimiento del modelo SVM en términos de tiempo de ejecución.

Hiperparámetros definidos: Una vez que los modelos han sido definidos, se procede a dividir el conjunto de datos en un 70% para entrenamiento y un 30% para prueba. A continuación, se utiliza la técnica de búsqueda en cuadrícula (Grid Search) con la ayuda de la librería de python GridSearchCV para determinar los hiperparámetros óptimos. Esta técnica evalúa exhaustivamente diferentes combinaciones de hiperparámetros y selecciona aquellas que producen el mejor rendimiento del modelo mediante validación cruzada. En la Tabla9 se presentan los mejores hiperparámetros encontrados para cada modelo, los cuales han sido seleccionados en base a su rendimiento durante la búsqueda en cuadrícula. Estos hiperparámetros optimizados son fundamentales para obtener un modelo con el mejor desempeño posible en la clasificación de los datos.

Tabla 9

Hiperparámetros para los modelos clasificadores

Modelo	Hiperparámetro	Valor
Decision Tree	criterion	entropy
	max_depth	None
	min_impurity_decrease	0.0
	min_samples_leaf	1
	min_samples_split	2
	ccp_alpha	0.0001
	splitter	best
Random Forest	n_estimators	10
	criterion	entropy
	max_depth	None
	max_features	Auto
	min_samples_split	2
	min_samples_leaf	1
	oob_score	True
	random_state	42
SVM	C	10000
	class_weight	None
	coef0	-2
	decision_function_shape	ovr
	gamma	0.1
	kernel	rbf

Continúa en la siguiente página...

Tabla 9 – Continúa

Modelo	Hiperparámetro	Valor
	max_iter	10000
	tol	0.01
	probability	True
KNN	algorithm	auto
	leaf_size	10
	metric	manhattan
	metric_params	None
	n_jobs	-1
	n_neighbors	3
	p	1
	weights	distance

Capítulo IV: Resultados

Fase de evaluación

Se evalúan los cuatro modelos construidos durante la etapa de modelado, con el propósito de medir su calidad y el rendimiento. Se realiza una evaluación exhaustiva de los resultados obtenidos en la clasificación por tipo de aplicación, considerando los dos grupos de características previamente establecidos.

Métricas de evaluación

Durante la evaluación de un modelo de clasificación, se emplean múltiples métricas para medir su desempeño y evaluar su capacidad predictiva. Estas métricas proporcionan información valiosa sobre el rendimiento y la precisión de cada modelo. Al analizar estas métricas, podemos obtener una comprensión más completa de cómo se comporta cada modelo y cómo se ajusta a los datos (Castelli et al., 2018).

Exactitud (Accuracy): Indica qué tan exacto es el modelo al predecir los resultados correctos. Se calcula dividiendo el número de predicciones correctas entre el total de predicciones realizadas.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precisión (Precision): Indica qué tan preciso es el modelo al identificar los casos positivos. Se calcula dividiendo el número de instancias positivas correctamente clasificadas entre el total de instancias clasificadas como positivas.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Sensibilidad o Exhaustividad (Recall): Indica qué proporción de los casos positivos se han identificado correctamente. Se calcula dividiendo el número de instancias positivas correctamente clasificadas entre el total de instancias reales que son positivas.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

F1-Score: Busca obtener un rendimiento equilibrado en la clasificación de los casos positivos y negativos. Es una medida combinada de precisión y recall que proporciona un equilibrio entre ambas métricas. Esta métrica fue útil ya que existe un desequilibrio en las clases.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

Comparación de los modelos: En esta sección se comparan las métricas de evaluación de los modelos desarrollados con el fin de seleccionar el modelo con mejor rendimiento y tomar decisiones sobre cuál algoritmo es más efectivo.

En la Tabla10 se presentan los resultados obtenidos para cada conjunto de características para el tiempo de espera de flujo en 120 y 15 segundos. Al analizar la métrica de exactitud (accuracy), se destaca que los modelos DT y RF alcanzaron las puntuaciones más altas en CG1-120s, con un 87.23% y 87.65%, respectivamente, lo que indica un excelente rendimiento en la clasificación de los datos. Por otro lado, para CG2-120s, los modelos RF y DT fueron los que mostraron la mejor predicción, con un 87.34% y 87.23% de exactitud, respectivamente. Estos modelos demostraron ser más efectivos en la tarea de predicción en comparación con los otros modelos evaluados.

Las métricas de evaluación para CG1-15s y CG2-15s demuestran un alto rendimiento en sus resultados, lo que concuerda con el autor (Draper-Gil. et al., 2016) al afirmar que los modelos tienen un mejor desempeño cuando el tiempo de espera de flujo es de 15 segundos. En tres modelos, los resultados superan el 98.63%, excepto para SVM, donde la precisión es un poco más del 92%.

Tabla 10

Resultados de las métricas de evaluación para tiempo de espera de flujo en 120s y 15s para cada grupo de características

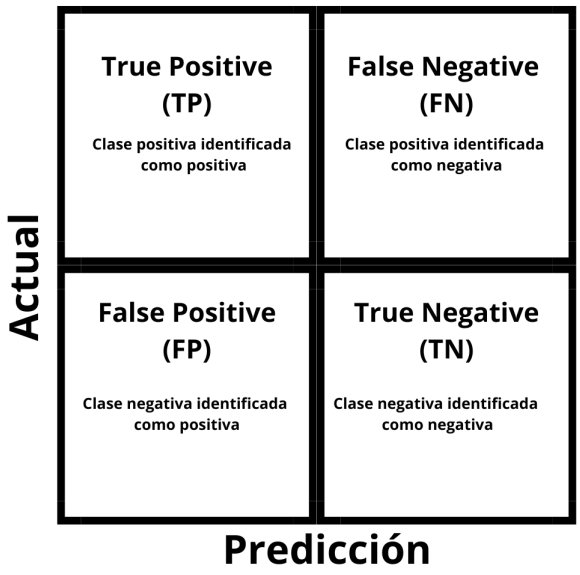
Modelo	Accuracy	Precision	Recall	F1-Score
CG1-120s				
DT	87.37%	90.63%	90.57%	90.56%
RF	87.65%	91.01%	90.99%	90.99%
SVM	60.66%	81.19%	70.85%	72.38%
KNN	82.37%	87.05%	86.61%	86.78%
CG2-120s				
DT	87.23%	90.74%	90.76%	90.73%
RF	87.34%	90.92%	90.90%	90.90%
SVM	61.52%	82.67%	72.12%	74.84%
KNN	82.50%	87.43%	87.05%	87.18%
CG1-15s				
DT	99.89%	99.83%	99.84%	99.84%
RF	99.99%	99.99%	99.98%	99.98%
SVM	92.06%	94.09%	92.24%	92.45%
KNN	98.54%	97.74%	97.33%	9752%
CG2-15s				
DT	99.86%	99.81%	99.74%	99.77%
RF	99.97%	99.96%	99.95%	99.95%
SVM	93.00%	94.59%	93.34%	93.41%
KNN	98.66%	97.89%	97.63%	97.75%

Matriz de confusión

Es una herramienta utilizada en problemas de clasificación para evaluar el rendimiento de un modelo en función de las predicciones realizadas. Tiene un diseño de filas y columnas que representan las clases reales y las clases predichas, respectivamente. Para una clasificación multiclase, la matriz tendrá una dimensión *NxN*, donde *N* es el número de clases (Castelli et al., 2018). La Figura 14 presenta algunas terminologías para su interpretación.

Figura 14

Matriz de confusión.



El resultado de los cuatro modelos ha permitido generar cuatro matrices de confusión correspondientes a cada grupo de características CG1-120s, CG2-120s, CG1-15s y CG2-15s basado en el tiempo de espera de flujo en 120 y 15s. A través del análisis de estas matrices, se puede evaluar la capacidad de los modelos para clasificar correctamente las instancias en cada clase y detectar posibles errores en las predicciones. Con ello, proporcionan una visión detallada del desempeño de los modelos al mostrar la

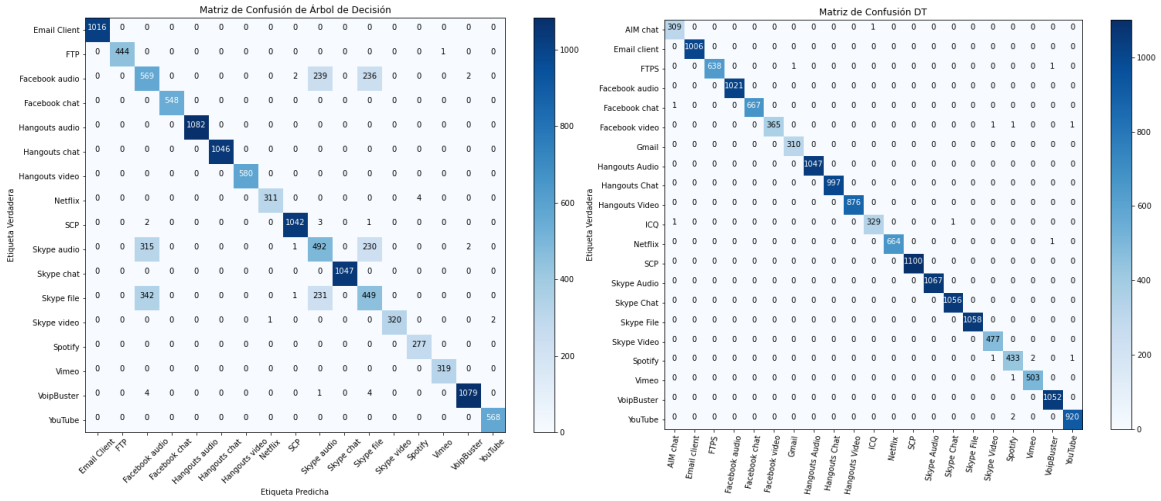
distribución de las predicciones realizadas en comparación con las clases reales.

En las Figuras 15a y 15b se presenta una comparativa de los resultados de la matriz de confusión del modelo DT para CG1-120s y CG1-15s, respectivamente. En la Figura 15a, se observa que hay pocas clases que varían en la predicción entre CG1-120s y CG1-15s. Por otro lado, en la Figura 15b, las predicciones muestran un mejor rendimiento en CG1-15s. Esto es relevante considerando la cantidad de datos asignados a cada clase.

Figura 15

Matriz de confusión del modelo DT

(a) *CG1-120s* (b) *CG1-15s*



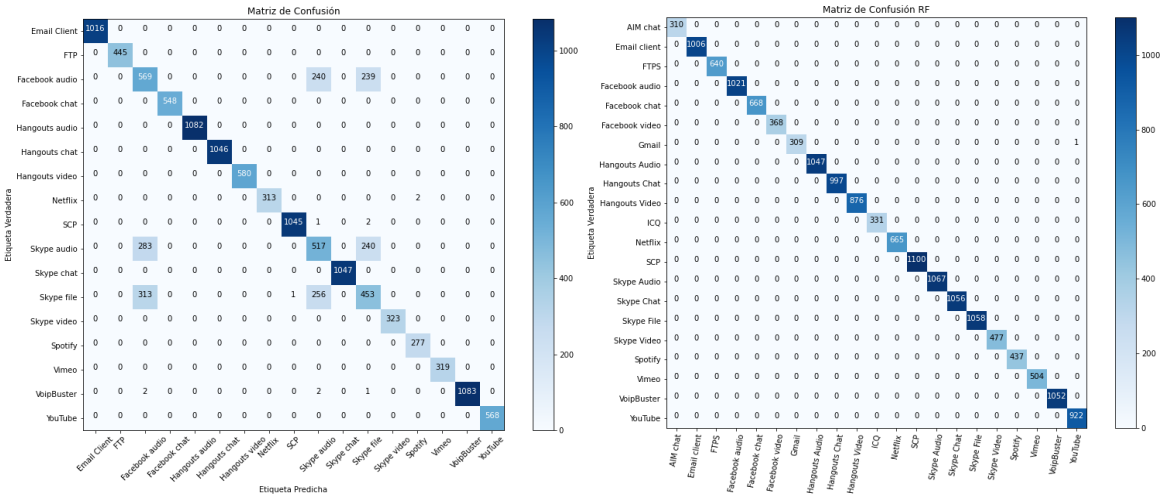
En las Figuras 16a y 16b se presentan los resultados de la matriz de confusión del modelo de RF para CG1-120s y CG1-15s, respectivamente. En la Figura 16a, se puede apreciar que los valores predichos por el modelo RF en CG1-120s son similares a los obtenidos por el modelo de DT en el mismo conjunto de datos. Por otro lado, en la Figura 16b, los resultados de RF para CG1-15s muestran un buen rendimiento en todas las clases.

En las Figuras 17a y 17b se muestran los resultados de la matriz de confusión del modelo de SVM para CG1-120s y CG1-15s, respectivamente. En la Figura 17a, se

Figura 16

Matriz de confusión del modelo RF

(a) *CG1-120s* (b) *CG1-15s*



observa que las predicciones no son buenas, ya que la clase Facebook audio se confunde con las clases Skype audio y Skype file. El enfoque de esta clase se basa en que solo se predijeron correctamente dos valores. Como resultado, este modelo no demostró ser eficiente para la clasificación de aplicaciones en CG1-120s. Por otro lado, en la Figura 17b se obtuvieron mejores resultados en las predicciones al aplicar el modelo en CG1-15s, lo que indica una mejora en el rendimiento del modelo.

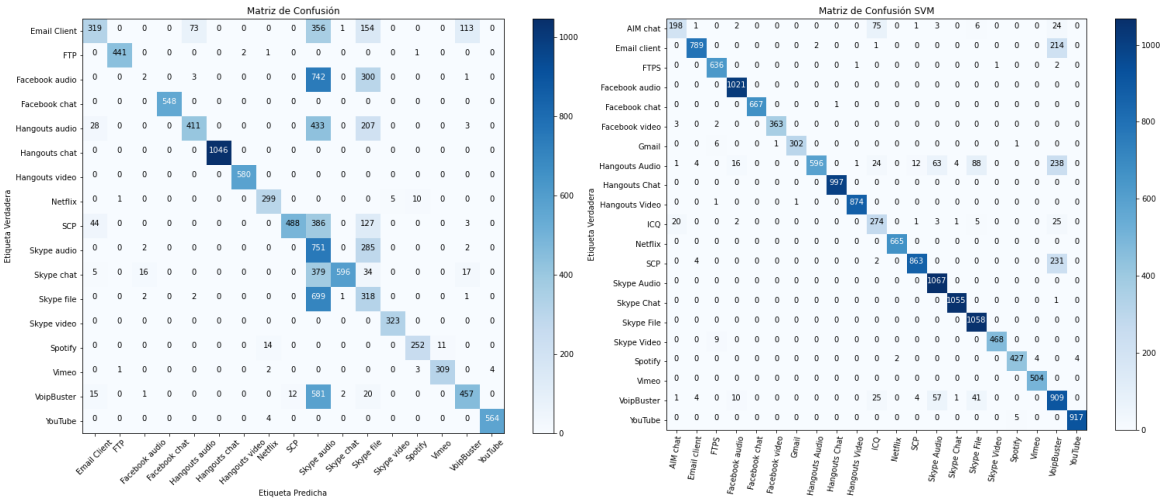
En las Figuras 18a y 18b se presentan los resultados de la matriz de confusión del modelo de KNN para CG1-120s y CG1-15s, respectivamente. Se puede observar que las predicciones son altas para todas las clases, lo que indica que este modelo resultó ser más eficiente para la clasificación de aplicaciones tanto en CG1-120s como en CG1-15s.

De la misma manera ocurre para CG2 tanto como para CG2-120s. y CG2-15s. En la Figura 19a, se muestra la matriz de confusión del modelo DT, que permite comparar los resultados con la Figura 19b. Claramente, se observa una diferencia en las predicciones de

Figura 17

Matriz de confusión del modelo SVM

(a) *CG1-120s* (b) *CG1-15s*



clasificación para cada variable objetivo. En la Figura 19a, se pueden identificar predicciones erróneas que afectan la precisión del modelo DT para CG2-120s. Por otro lado, en la Figura 19b, las predicciones son casi perfectas, lo que arroja un resultado de predicción satisfactorio para CG2-15s.

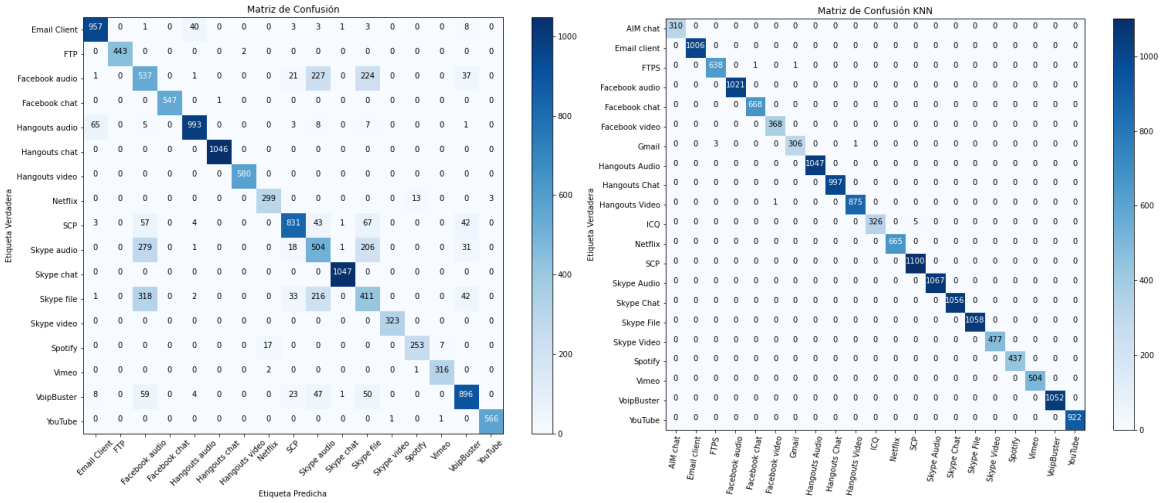
En la Figura 20a la matriz de confusión para el modelo RF en CG2-120s las predicciones en su clasificación son notorias a comparación con el modelo RF para CG2-15s (ver la Figura 20b), esta muestra altos niveles de precisión en todas las clases, lo que indica que este modelo demostró una eficiencia notable en la clasificación de aplicaciones.

En la Figura 21a y Figura 21b, se muestran los resultados de las matrices de confusión para el modelo SVM para CG2-120 y CG2-15s respectivamente. Se puede observar que las predicciones no son óptimas en CG2-120s, ya que el modelo solo clasifica correctamente pocas clases (ver Figura 21a). Sin embargo, en la Figura 21b con

Figura 18

Matriz de confusión del modelo KNN

(a) *CG1-120s* (b) *CG1-15s*



CG2-15s las predicciones mejoraron notoriamente, lo que indica que este modelo resultó más eficiente para la clasificación de aplicaciones.

En la Figura 22a y 22b, se muestra el resultado de la matriz de confusión del modelo de KNN para el conjunto de características CG2-120s y CG2-15s respectivamente. Se puede observar en la Figura 22a que las predicciones varían de manera errónea en todas las clases, lo que resulta en un rendimiento más bajo del modelo CG2-120s. Por otro lado en la Figura 22b para CG2-15s las predicciones mejoran radicalmente en la clasificación de aplicaciones.

Figura 19

Matriz de confusión del modelo DT

(a) *CG2-120s* (b) *CG2-15s*

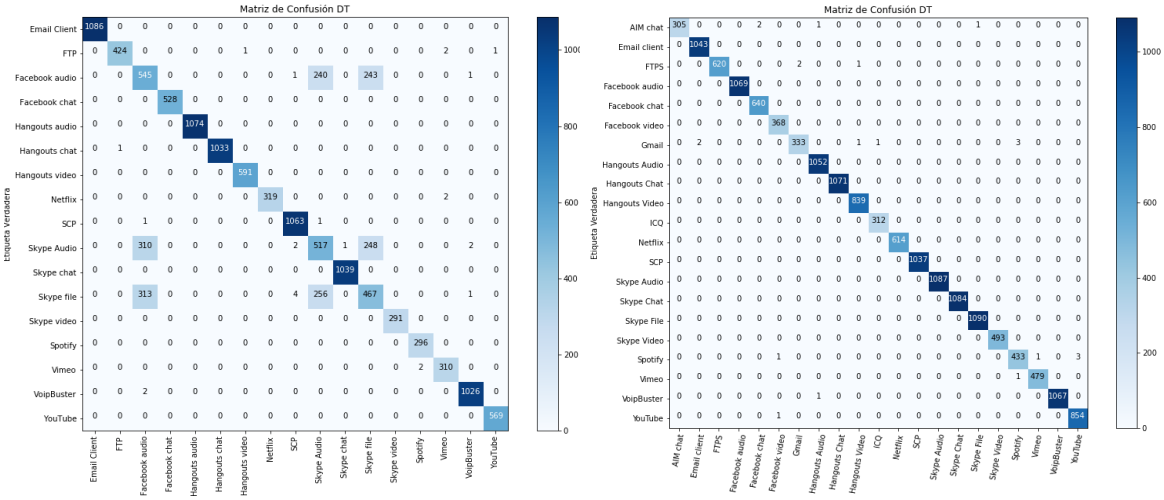


Figura 20

Matriz de confusión del modelo RF

(a) *CG2-120s* (b) *CG2-15s*

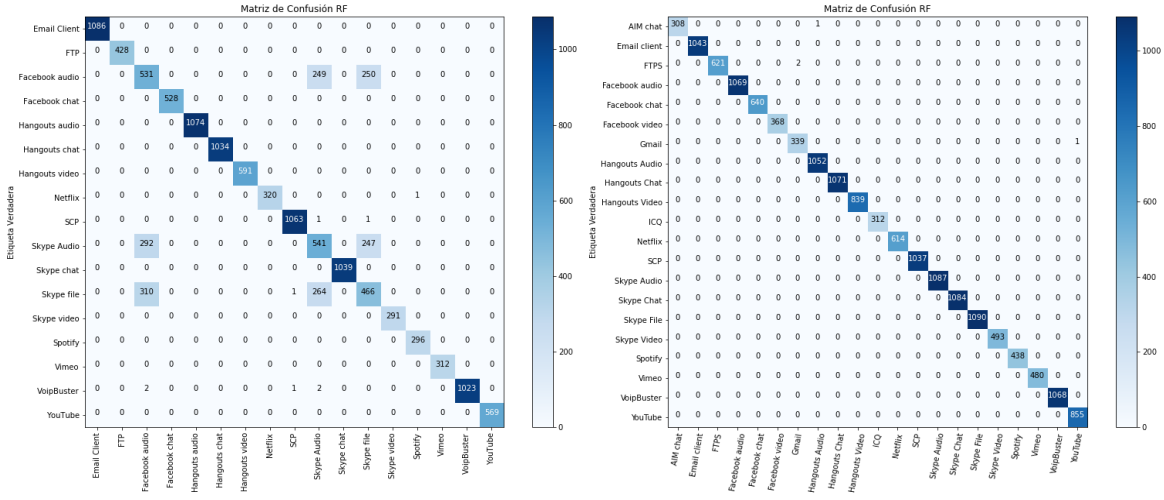


Figura 21

Matriz de confusión del modelo SVM

(a) *CG2-120s* (b) *CG2-15s*

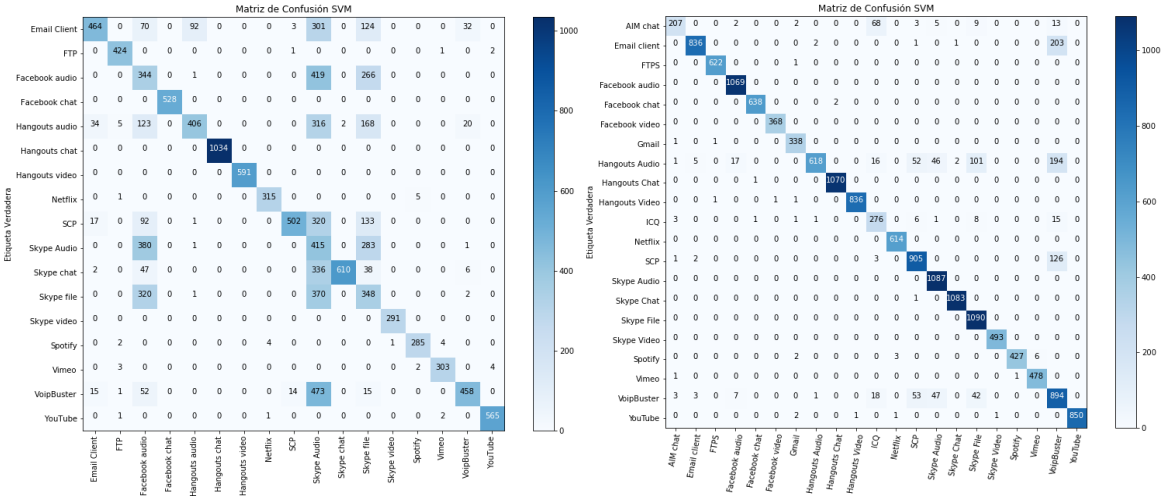
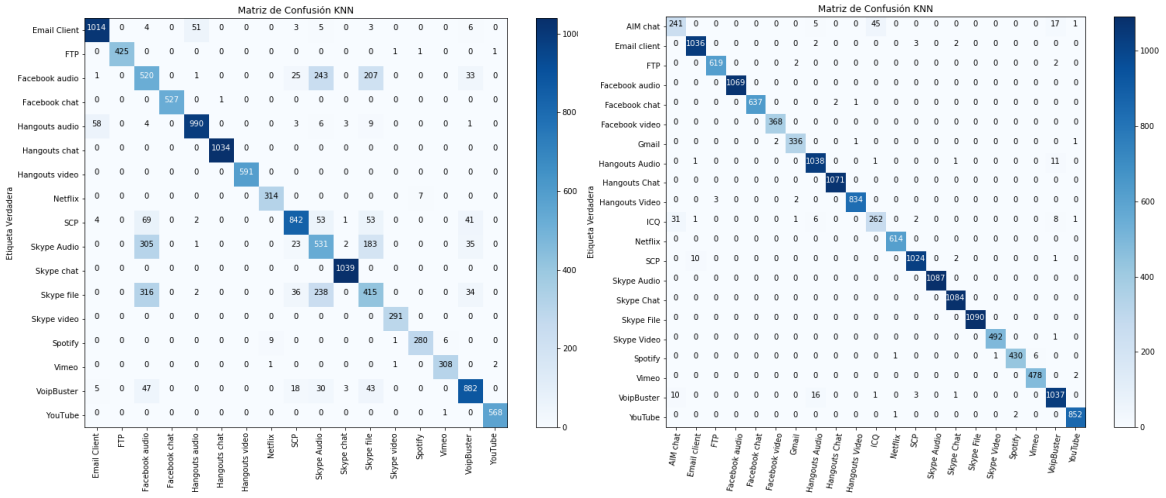


Figura 22

Matriz de confusión del modelo KNN

(a) *CG2-120s* (b) *CG2-15s*



Validación del modelo

La validación de los modelos de aprendizaje automático ayuda a determinar si el modelo es capaz de hacer predicciones precisas y confiables en situaciones reales. Para validar los modelos entrenados, se emplearon dos técnicas: la validación cruzada y la curva ROC.

Validación cruzada

Es una técnica utilizada para evaluar el rendimiento de un modelo de aprendizaje automático de manera más robusta y confiable. En lugar de utilizar solo un conjunto de entrenamiento y un conjunto de prueba, la validación cruzada divide los datos en múltiples subconjuntos conocidos como *folds*, realiza múltiples iteraciones de entrenamiento y evaluación (Diana & Tommasi, 2002). En la Tabla 11 se puede observar una comparativa de la validación cruzada con 10 iteraciones en todos los modelos de clasificación para CG1-120s, CG1-15s, CG2-120s y CG2-15s. Con estos resultados se puede constatar que el modelo RF tiene mejores resultados con 87.89% en la validación cruzada para CG1-120s y también para CG1-120s con 88% pero aumentando su precisión. Por otro lado, el modelo que sobresale en CG1-15s y CG2-15s sigue siendo RF destacando una notable estabilidad con más del 99.9% de precisión en la clasificación en todas las validaciones.

Tabla 11

Resultados de la validación cruzada

Modelo	Accuracy	Precision	Recall	F1-Score
		CG1-120s		
DT	87.49%	90.98%	90.97%	90.96%
RF	87.89%	91.33%	91.31%	91.31%
	Siguiente...			

Tabla 11 – Continúa

Modelo	Accuracy	Precision	Recall	F1-Score
SVM	55%	81%	45%	49%
KNN	83.80%	88.21%	87.88%	88.01%
CG2-120s				
DT	87%	91%	91%	91%
RF	88%	91%	91%	91%
SVM	56%	55%	54%	54%
KNN	82%	85%	84%	84%
CG1-15s				
DT	99.88%	99.82%	99.82%	99.82%
RF	99.99%	99.98%	99.98%	99.98%
SVM	92.11%	93.72%	92.41%	92.44%
KNN	98.63%	97.81%	97.46%	97.62%
CG2-15s				
DT	99.91%	99.86%	99.84%	99.85%
RF	99.98%	99.97%	99.97%	99.97%
SVM	91.92%	93.01%	92.47%	91.92%
KNN	98.74%	98.01%	97.68%	97.83%

Curva ROC

La curva ROC es una representación gráfica que muestra el rendimiento de un modelo de clasificación en función de diferentes umbrales de decisión. Es una herramienta comúnmente utilizada para evaluar y comparar la capacidad de discriminación de los modelos. El rendimiento de discriminación del modelo se cuantifica mediante el área bajo

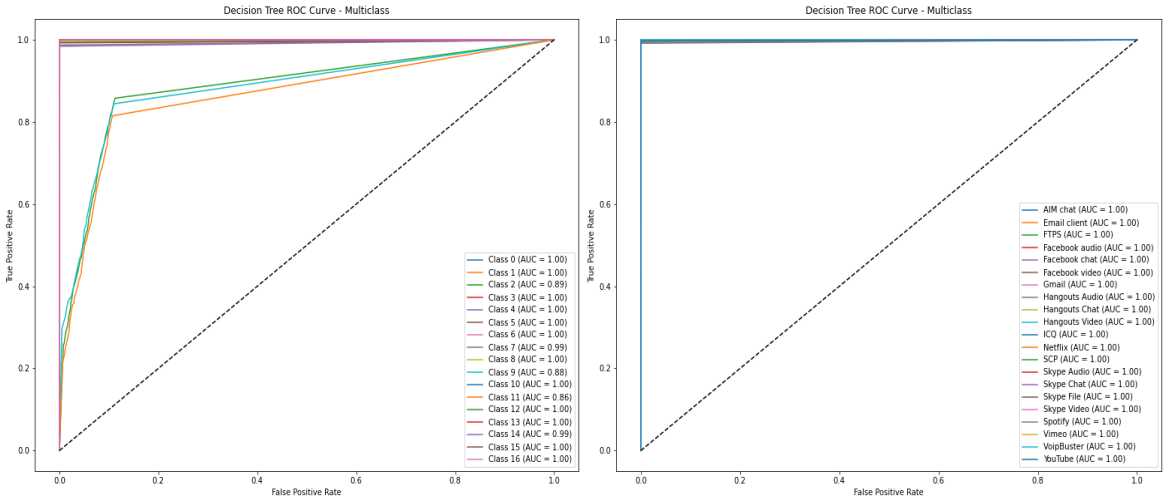
la curva ROC (AUC-ROC). Un AUC-ROC cercano a 1 indica una alta capacidad de discriminación, mientras que un valor cercano a 0.5 indica un rendimiento similar al azar y una discriminación limitada. Al gráficar la curva ROC el eje X representa los Falsos Positivos (FPR) y el eje Y representa verdaderos Positivos (TPR) (Mandrekar, 2010).

En las Figuras 23a y 23b se observa el rendimiento del modelo DT para CG1-120s y CG1-15s, respectivamente. En estas figuras se examinan las clases con alta y baja discriminación. Como resultado, se puede apreciar que en todas las clases se obtiene una alta capacidad de discriminación, lo que indica que el modelo DT es efectivo para clasificar correctamente las aplicaciones en ambos conjuntos de características.

Figura 23

Cuva ROC del modelo DT

(a) *CG1-120s* (b) *CG1-15s*



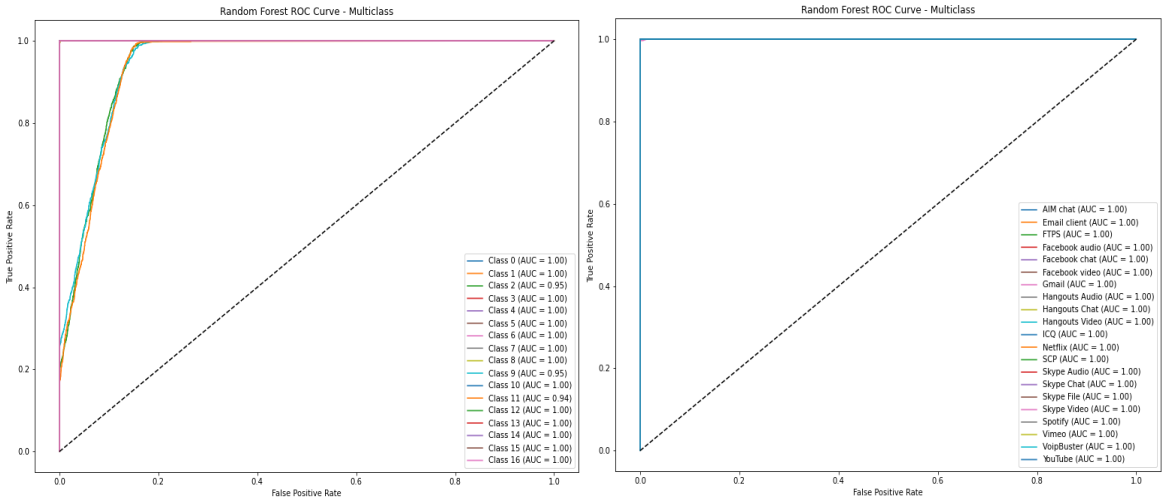
En las Figuras 24a y 24b se observa el rendimiento del modelo RF para CG1-120s y CG1-15s, respectivamente. En estas figuras se examinan las clases con alta y baja discriminación. Como resultado, en todas las clases se obtiene una alta capacidad de discriminación. Es importante destacar que los resultados de AUC no bajan del 95% en el

caso de CG1-120s, mientras que para CG1-15s los resultados alcanzan el 100%. Esto indica que el modelo RF es altamente efectivo en la clasificación de aplicaciones en ambos conjuntos de características, con una alta precisión en CG1-15s.

Figura 24

Cuva ROC del modelo RF

(a) *CG1-120s* (b) *CG1-15s*



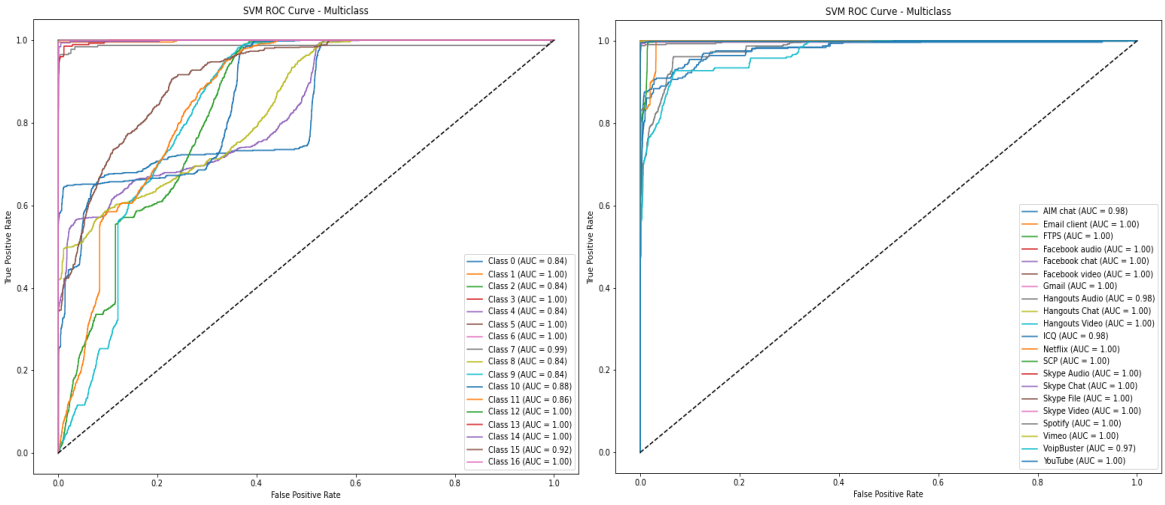
En las Figuras 25a y 25b se observa el rendimiento del modelo SVM para CG1-120s y CG1-15s, respectivamente. En estas figuras se examinan las clases con alta y baja discriminación. Los resultados de las clases varían, pero en general se obtiene una alta capacidad de discriminación en ambas configuraciones. Es importante destacar que los resultados de AUC en CG1-120s superan el 84%, lo que indica un buen rendimiento del modelo en la clasificación de aplicaciones. Por otro lado, en CG1-15s, los resultados de AUC alcanzan más del 98%, lo que muestra una alta precisión en la clasificación de aplicaciones en este conjunto de características.

En las Figuras 26a y 26b se observa el rendimiento del modelo KNN para CG1-120s y CG1-15s, respectivamente. En estas figuras se examinan las clases con alta y

Figura 25

Cuva ROC del modelo SVM

(a) *CG1-120s* (b) *CG1-15s*



baja discriminación. Los resultados de AUC en CG1-120s son mayores al 78%, lo que indica un rendimiento moderado del modelo en la clasificación de aplicaciones. Por otro lado, en CG1-15s, los resultados de AUC empiezan desde el 93%, demostrando una mejora en la clasificación de aplicaciones para este conjunto de características en función al tiempo de espera del flujo.

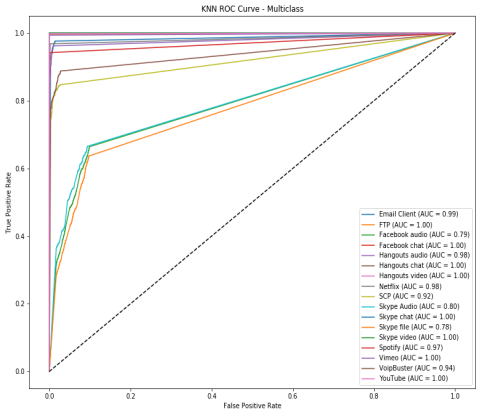
Por otro lado, en la Figura 27a la curva ROC DT para CG2-120s algunas clases, como Facebook audio, Skype Audio y Skype file, tienen un AUC ligeramente menor, indicando una menor precisión en su clasificación. Sin embargo en la Figura 27b para CG2-15s todas las clases tiene su valor igual 100% orientando a una alta capacidad de discriminación en la clasificación de aplicaciones.

En la Figura 28a la curva ROC RF para CG2-120s, las clases Facebook audio, Skype Audio y Skype file se obtuvo una capacidad discriminatoria es cercana al 100% a comparación con la Figura 28b para CG2-15s donde, se obtuvo en todas las clases el

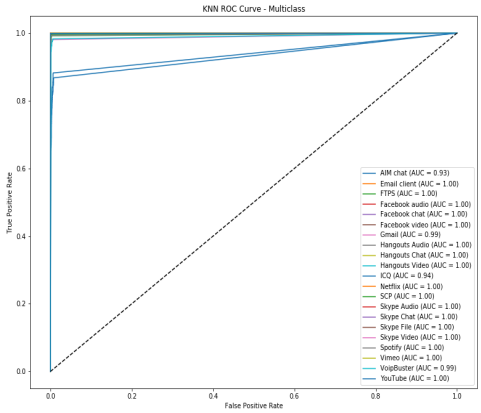
Figura 26

Cuva ROC del modelo KNN

(a) *CG1-120s*



(b) *CG1-15s*



100% de capacidad de discriminación.

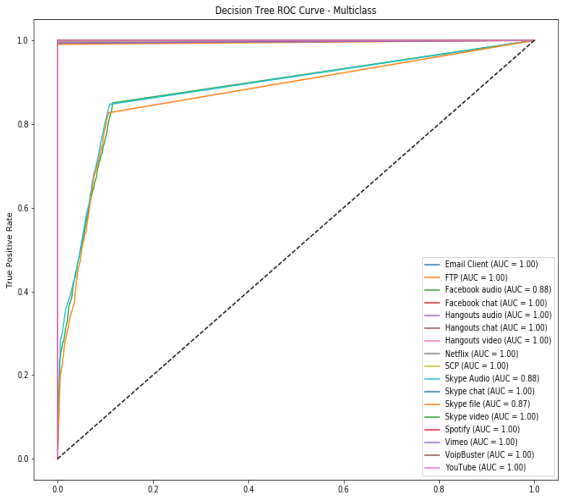
En la Figura 29a la curva ROC de SVM para CG2-120s, muestra un rendimiento variado en la capacidad de discriminación de la mayoría de las aplicaciones con menos AUC del 100%. Por otro lado, en la Figura 29b para CG2-15s mejora significativamente su capacidad discriminatoria aumentando a la mayoría de clases a un 100%.

Para finalizar en la Figura 30a la curva ROC KNN para CG2-120s la capacidad discriminatoria varia desde el 78% al 100%. Y en la Figura 30b para CG2-15s la capacidad discriminatoria esta entre el 95% al 100% en las aplicaciones.

Figura 27

Cuva ROC del modelo DT

(a) *CG2-120s*



(b) *CG2-15s*

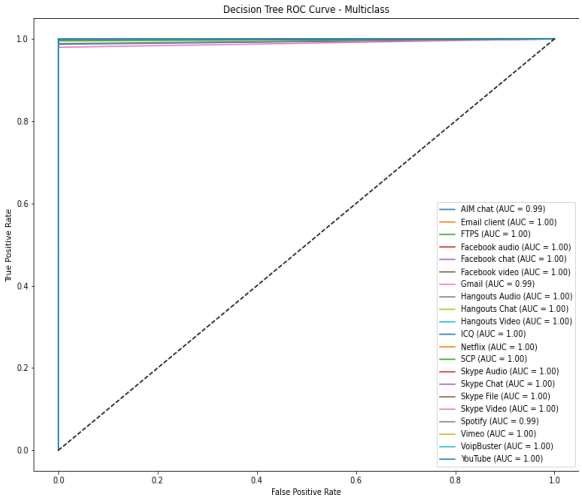
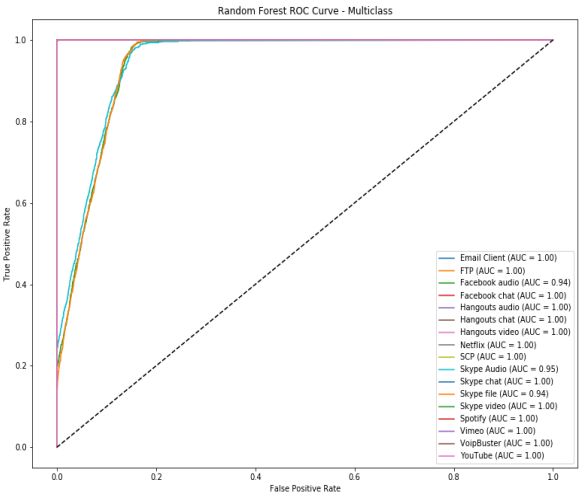


Figura 28

Cuva ROC del modelo RF

(a) *CG2-120s*



(b) *CG2-15s*

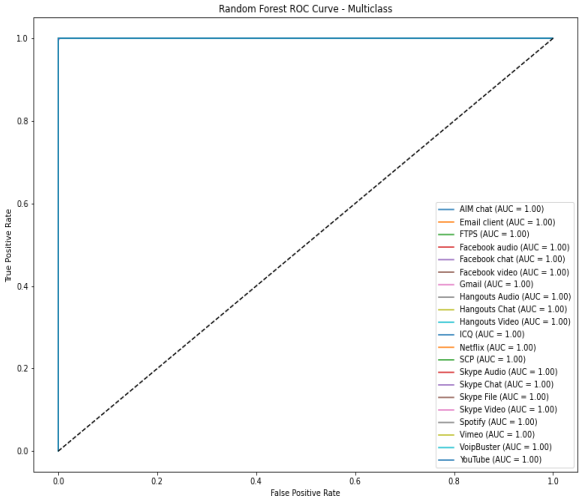
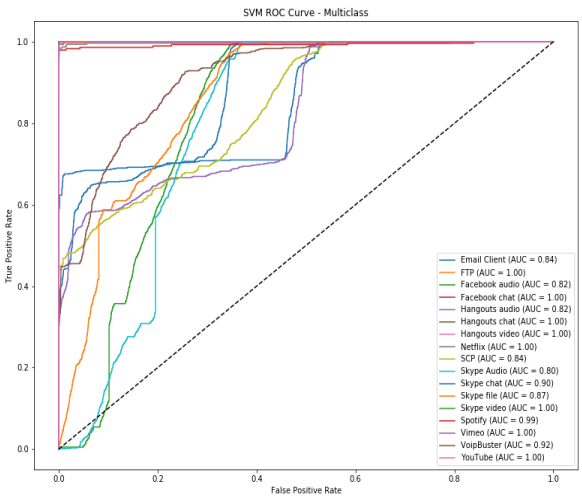


Figura 29

Cuva ROC del modelo SVM

(a) *CG2-120s*



(b) *CG2-15s*

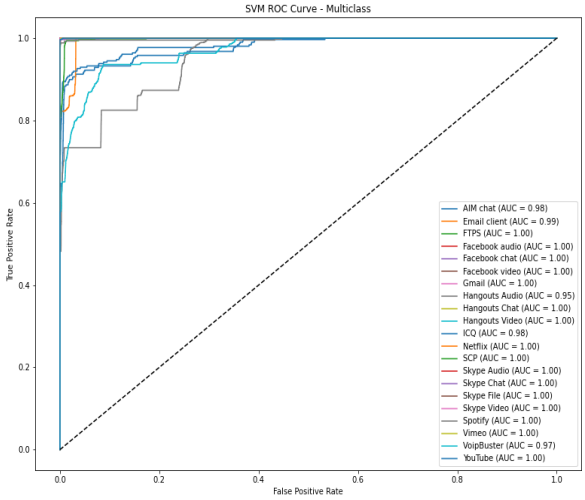
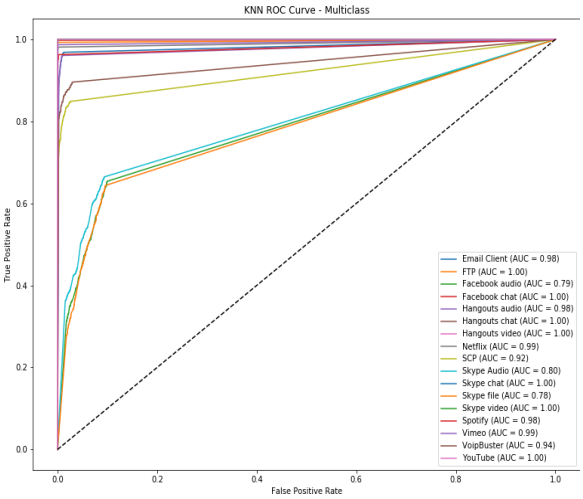


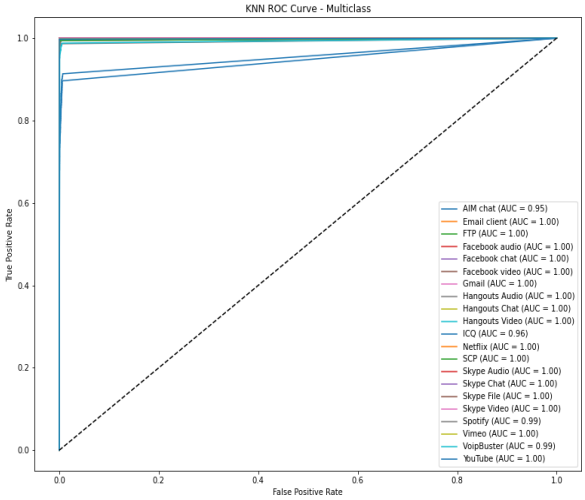
Figura 30

Cuva ROC del modelo KNN

(a) *CG2-120s*



(b) *CG2-15s*



Implementación del modelo en SDN

En esta sección, se implementan en una SDN los dos modelos de ML pre-entrenados con la mayor precisión: RF y DT. El propósito es anticipar el tráfico que fluye a través del controlador. En este contexto, se ha realizado una investigación exhaustiva para determinar el tipo de controlador a utilizar, y tras considerar sus características y la facilidad de uso, se concluyó que RYU es la opción más adecuada.

RYU es un controlador de código abierto que facilita la creación y gestión de SDN, este proporciona una plataforma adaptable para el desarrollo de aplicaciones de control de red que utilicen el protocolo OpenFlow (controla las acciones de reenvío de conmutadores). Permite crear aplicaciones que controlen el comportamiento de los conmutadores de red a través de la programación de reglas de flujo en función del tráfico de red u otros eventos (Asadollahi et al., 2018). En el controlador RYU puede implementarse varias funciones de red como enrutamiento, políticas de seguridad y optimización de tráfico. Su plataforma es basada en el lenguaje de programación Python lo que permite a los usuarios el desarrollo de aplicaciones personalizadas.

Por consiguiente, se creó un archivo cuya estructura se asemeja a los archivos por defecto en RYU. En este nuevo archivo, se importan las librerías y módulos necesarios para el funcionamiento correcto del controlador. Es importante resaltar que, en la verificación del tráfico, se ha desarrollado un método para predecir el comportamiento a partir de un archivo en formato PCAP. En este proceso, se extraen las características y se llevan a cabo los cálculos correspondientes para cada una de las 15 características previamente utilizadas en el entrenamiento de los modelos. Estas características incluyen tasas de flujo, duración de flujo, tamaños de paquetes, puertos de destino, entre otras.

En entornos reales los datos tienden a ser desnormalizados, se ha aplicado la normalización a todas las características extraídas y calculadas. Luego, se procede con la

predicción de los paquetes. Utilizando un decorador de eventos, se invoca el archivo PCAP que se desea analizar, y se crea un diccionario con las clases que se pretenden predecir. A partir de aquí, se realizan las predicciones pertinentes. Además, se lleva un registro del conteo de paquetes predichos por cada clase y se calcula su exactitud correspondiente. Esta información proporciona una idea clara del rendimiento de los modelos en términos de predicciones precisas. El algoritmo 1 muestra el funcionamiento del controlador y el modelo.

En la Figura 31a se presentan los resultados al predecir el tráfico contenido en un archivo PCAP correspondiente a datos AIM chat. Los datos contenidos en el PCAP se mantienen en su estado original, y a pesar de haber sido sujetos a un proceso de estandarización, aún pueden contener ruido, inconsistencias o valores atípicos. Estos elementos podrían tener un impacto adverso en el rendimiento del modelo. Sin embargo, el modelo DT logra realizar predicciones coherentes con las aplicaciones que se esperaba predecir. Por ejemplo, identifica 453 instancias de Facebook chat y 295 instancias de Hangouts chat. Por otro lado, en la Figura 31b, se aprecia que el modelo de RF no logra predecir adecuadamente el tráfico correspondiente a AIM chat, ya que realiza cuatro predicciones adicionales a esta categoría.

Algoritmo 1 Clasificación de flujos

```
procedure CLASIFICACIÓN EN EL CONTROLADOR
Require: Cargar modelo ML
    procedure GESTIÓN PARA CARGAR EL MODELO PRE-ENTRENADO
Require: Metodo para predecir el trafico a partir de un archivo pcap
        LeerPCAP ← PCAP
        for packet in LeerPCAP : do
            Calcular_Caracteristicas ← LeerPCAP
            Agregar_a_Lista ← Calcular_Caracteristicas
        end for
        Guardar_Dataframe ← Agregar_a_Lista
        Normalizar_Datos ← Dataframe
        Predecir ← Normalizar_Datos
Require: Decorador para manejar eventos de paquetes entrantes
        Cargar ← PCAP
        Predicciones ← MetodoPredecir
        for i, prediction in enumerate(Predicciones) do
            predicted_class ← prediction
            if predicted_class in class_counts then
                class_counts[predicted_class]+ = 1
            end if
        end for
        for class_name, count in class_counts.items do
            Imprimirconteoclases
        end for
        Imprimir Total de Ejemplos, Predicciones Correctas y Exactitud del Modelo
    end procedure
end procedure
```

Figura 31

Clasificación del tráfico de tipo AIM chat

(a) *Modelo DT*

```
Package 1236 - Prediction: AIM chat (Class 0)
Package 1237 - Prediction: Facebook chat (Class 4)
Package 1238 - Prediction: Facebook chat (Class 4)
Package 1239 - Prediction: AIM chat (Class 0)
Package 1240 - Prediction: Facebook chat (Class 4)
Package 1241 - Prediction: AIM chat (Class 0)
Package 1242 - Prediction: Facebook chat (Class 4)
Package 1243 - Prediction: AIM chat (Class 0)
Predicted class count:
AIM chat: 495 times
Facebook chat: 453 times
Hangouts Chat: 295 times
Total number of samples: 1243
Correct predictions: 495
Model accuracy: 39.82%
```

(b) *Modelo RF*

```
Package 1237 - Prediction: Facebook video (Class 5)
Package 1238 - Prediction: Facebook video (Class 5)
Package 1239 - Prediction: Facebook video (Class 5)
Package 1240 - Prediction: Facebook video (Class 5)
Package 1241 - Prediction: Facebook video (Class 5)
Package 1242 - Prediction: Facebook video (Class 5)
Package 1243 - Prediction: Facebook video (Class 5)
Predicted class count:
Facebook video: 888 times
Vimeo: 25 times
VoipBuster: 11 times
AIM chat: 35 times
Facebook chat: 284 times
Total number of samples: 1243
Correct predictions: 35
Model accuracy: 2.82%
```

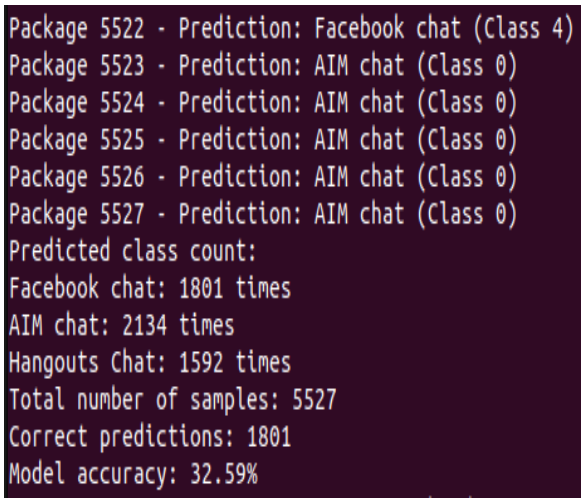
En la Figura 32a se presentan los resultados de la predicción del tráfico contenido en un archivo PCAP correspondiente a datos de chat de Facebook. Similar a los datos de AIM chat, estos también se mantienen en su estado original. Los resultados de las predicciones con el modelo DT presentan coherencia con las clases que se esperaban predecir. Estos resultados son mejores que los observados en la Figura 32b, aunque en este ejemplo, el modelo RF mejoró en las predicciones en comparación con el ejemplo de AIM chat.

En la Tabla 12 se pueden apreciar de manera más clara los resultados de las predicciones de las clases. En este caso, se han incluido más aplicaciones.

Figura 32

Clasificación del tráfico de tipo Facebook chat

(a) *Modelo DT*



(b) *Modelo RF*

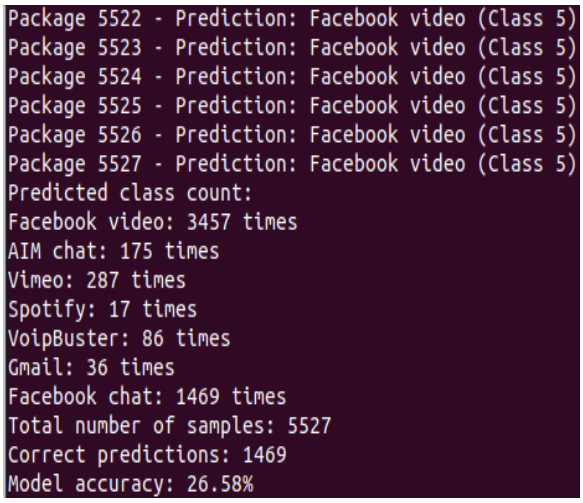


Tabla 12

Resultados de la implementación de los modelos DT y RF en SDN.

Clase	Exactitud	Clases Predichas DT	Clases Predichas RF	Ejemplos
AIM chat	DT: 39.82% RF: 2.82%	AIM Chat:495, Facebook chat:453, Hangouts Chat:295	Facebook video:888, Vimeo:25, VoipBuster:11, AIM chat:35, Facebook chat: 284	1243

Continúa en la siguiente página...

Tabla 12 – continuación de la página anterior

Clase	Exactitud	Clases Predichas DT	Clases Predichas RF	Ejemplos
Facebook chat	DT: 32.59% RF: 26.58%	Facebook chat: 1801, AIM chat: 2134, Hangouts Chat: 1592	AIM chat: 175, Vimeo: 287, Spotify: 17, VoipBuster: 86, Gmail: 36, Facebook chat: 1469	5527
Hangouts Chat	DT: 16.28% RF: 0.00%	AIM chat: 2772, Facebook chat: 1979, Hangouts Chat: 924,	Facebook video: 4090, AIM chat: 500, Vimeo: 161, VoipBuster: 16, Gmail: 114, Facebook chat: 793, FTPS: 1	5675

Capítulo V: Conclusiones, recomendaciones y trabajo futuro

Conclusiones

El presente proyecto permitió explorar y definir enfoques para la clasificación de tráfico en un entorno SDN mediante el uso de técnicas de aprendizaje automático. Estos enfoques fueron guiados por la metodología SEMMA (Muestreo, Exploración, Modificación, Modelado y Evaluación), que proporcionó una estructura coherente para desarrollar modelos precisos y obtener resultados confiables. Durante este proceso, se evaluaron cuatro de los modelos de aprendizaje supervisado más utilizados en la clasificación de tráfico de red: DT, RF, SVM y KNN.

Cada uno de los modelos realizó una clasificación en grano fino utilizando el conjunto de datos VPN-NonVPN donde se extrajeron sus flujos en un tiempo de espera de 15 y 120 segundos, con ello se obtuvo dos conjuntos de datos con 53037 y 42710 flujos o registros respectivamente. Se llevo a cabo el proceso de ingeniería de características para obtener un mayor impacto en el rendimiento del modelo. Se consideraron dos grupos de distintos como 15 y 25 características relevantes y significativas. Esta estrategia fue fundamental para mejorar la calidad y relevancia de los datos, y simplificar la complejidad del modelo mejorando su capacidad predicativa. Al culminar la evaluación se determinó que el proceso de extracción de flujos obtuvo un mejor rendimiento con un en tiempo de espera de 15 segundos y utilizando el grupo de 15 características. Este conjunto de datos proporcionó 21 clases con diferentes tipos de aplicaciones. Los resultados de la precisión (accuracy) obtenidos para cada modelo fueron: DT obtuvo un 99.89%, RF alcanzó el 99.99%, SVM logró un 92.06% y KNN presentó un 98.63%.

Los modelos con mayor precisión fueron RF con 99.99% y DT 99.89%, estos modelos fueron implementados en una SDN utilizando el controlador RYU. En este controlador se evaluaron algunos archivos PCAP de cada aplicación que se obtuvieron

inicialmente del conjunto de datos VPN-NonVPN. Para esto se requirió un buen rendimiento computacional y debido al peso de los PCAPs se delimito la evaluación de pocas aplicaciones. Específicamente, las aplicaciones AIM Chat, Facebook Chat y Hangouts Chat fueron analizadas rigurosamente en ambos modelos. Los resultados obtenidos arrojaron que AIM chat registro un porcentaje de precisión del 39.82% para DT y un 2.82% para RF, mientras que Facebook Chat mostró un 32.59% para DT y un 26.58% para RF y por último Hangouts Chat obtuvo un 16.28% para DT y un 0.00% para RF. Esta evaluación demostró que la precisión varía según la aplicación y el modelo, por lo que es necesario explorar profundamente en las razones que afectan en los resultados.

Recomendaciones

La complejidad y la alta importancia de la clasificación de aplicaciones en el tráfico de red utilizando modelos de aprendizaje automático, es esencial adoptar un enfoque estratégico en el desarrollo de estos modelos. Un paso fundamental implica realizar un preprocesamiento exhaustivo de los datos. Esto incluye la detección metódica de valores atípicos en cada clase de aplicaciones por separado, con el propósito de identificar patrones inusuales que puedan influir en los resultados del modelo.

El ajuste de hiperparámetros es una etapa determinante para la construcción de los modelos, ya que esto afecta directamente en el rendimiento y evaluación de estos. Se debe realizar una exploración de los diferentes valores relevantes para cada hiperparámetro enfocandosé en ajustar aquellos que tienen mayor impacto en los distintos algoritmos. Una técnica puede ser la búsqueda en cuadrilla donde se determinen que valores son más aptos para mejorar el rendimiento de los modelos.

La precisión de la clasificación de los modelos es muy importante, por lo tanto, se recomienda llevar a cabo pruebas adicionales donde se utilicen otros conjuntos de datos. Esto permitirá verificar la confiabilidad y firmeza de los resultados obtenidos, asegurando

que la precisión obtenida no sea puntual en un conjunto de datos en particular.

La eficiencia del proceso de evaluación de lo modelos en el controlador RYU, se puede mejorar realizando la evaluación en una máquina física, donde los recursos son más potentes en lugar de una maquina virtual ya que estas reducen los recursos computacionales en cuestión de memoria RAM y CPU.

Trabajo futuro

Aplicar los cuatro modelos de aprendizaje automático desarrollados en un entorno real, con el propósito de observar su comportamiento frente a datos que están en constante variación y calcular el tiempo que se requiere para generar predicciones. En la implementación, se han considerado los modelos DT y RF; por lo tanto, para un análisis más completo, sería necesario evaluar también el rendimiento de los modelos SVM y KNN.

Para mejorar los resultados en la clasificación de tráfico en una SDN, se puede considerar la posibilidad de utilizar controladores como OpenDaylight y ONOS, ya que ofrecen ventajas superiores, funcionalidades más avanzadas y, lo más importante, contienen más características.

Una posible evolución en la clasificación de tráfico en entornos SDN sería la aplicación de técnicas de aprendizaje profundo, dado que estas poseen una mayor capacidad de aprendizaje y evitan la necesidad de realizar ingeniería de características. No obstante, es importante considerar la disponibilidad de recursos computacionales para procesar diversos flujos de tráfico de manera simultánea.

Referencias

Abdulazzaq, S., & Demirci, M. (2020). A Deep Learning Based System for Traffic Engineering in Software Defined Networks. *International Journal of Intelligent Systems and Applications in Engineering*, *8*(4), 206-213.

Ahmad, S., & Mir, A. H. (2021). Scalability, Consistency, Reliability and Security in SDN Controllers: A Survey of Diverse SDN Controllers. *JOURNAL OF NETWORK AND SYSTEMS MANAGEMENT*, *29*(1). https://doi.org/10.1007/s10922-020-09575-4

Almadani, B., Beg, A., & Mahmoud, A. (2021). DSF: A distributed sdn control plane framework for the east/west interface. *IEEE Access*, *9*, 26735-26754.

Andersen, M., & Hansson, A. (2023). Supervised Learning. En *Optimization for Learning and Control* (pp. 297-326). https://doi.org/10.1002/9781119809180.ch10

Aouedi, O., Piamrat, K., & Parrein, B. (2022). Intelligent traffic management in next-generation networks. *Future internet*, *14*(2), 44. https://doi.org/10.3390/fi14020044

Asadollahi, S., Goswami, B., & Sameer, M. (2018). Ryu controller's scalability experiment on software defined networks. *2018 IEEE international conference on current trends in advanced computing (ICCTAC)*, 1-5.

Azevedo, A., & Santos, M. F. (2008). KDD, SEMMA and CRISP-DM: a parallel overview. *IADS-DM*.

Batista, G. E., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, *6*(1), 20-29.

Belkadi, O., Vulpe, A., Laaziz, Y., & Halunga, S. (2023). ML-Based Traffic Classification in an SDN-Enabled Cloud Environment. *Electronics*, *12*(2). https://doi.org/10.3390/electronics12020269

Benjamini, Y., & Braun, H. (2002). John W. Tukey's contributions to multiple comparisons. *Annals of Statistics*, 1576-1594.

BigML. (2014). Seminario web sobre la versión de BigML Spring 2014 - Clustering [Accessed: 2023-5-12].

Castaneda Herrera, L. M., Campo Munoz, W. Y., & Duque-Torres, A. (2021). Video Streaming Service Identification on Software-Defined Networking. *International Journal of Computers, Communications & Control*, *16*(5).

Castelli, M., Vanneschi, L., & Largo, Á. R. (2018). Supervised learning: classification. *por Ranganathan, S., M. Grisbskov, K. Nakai y C. Schönbach*, *1*, 342-349.

Chen, Z., Ding, R., Chin, T.-W., & Marculescu, D. (2018). Understanding the impact of label granularity on cnn-based image classification. *2018 IEEE international conference on data mining workshops (ICDMW)*, 895-904.

Choi, J., Peters, M., & Mueller, R. O. (2010). Correlational analysis of ordinal data: from Pearson'sr to Bayesian polychoric correlation. *Asia Pacific education review*, *11*, 459-466.

Cox, J., Jr., Chuang, J., Donvan, S., Ivey, J., Clarx, R. J., Riley, G., & Owen, H. L., III. (2017). Advancing Software-Defined Networks: A Survey. *IEEE ACCESS*, *5*, 25487-25526. https://doi.org/10.1109/ACCESS.2017.2762291

Dainotti, A., Pescape, A., & Claffy, K. C. (2012). Issues and future directions in traffic classification. *IEEE Network*, *26*(1), 35-40. https://doi.org/10.1109/MNET.2012.6135854

Dashevskiy, M., & Luo, Z. (2014). Network traffic classification and demand prediction. *Conformal Prediction for Reliable Machine Learning; Balasubramanian, VN, Ho, S.-S., Vovk, V., Eds*, 231-259.

Diana, G., & Tommasi, C. (2002). Cross-validation methods in principal component analysis: a comparison. *Statistical Methods and Applications*, *11*, 71-82.

Draper-Gil., G., Lashkari., A. H., Mamun., M. S. I., & A. Ghorbani., A. Characterization of Encrypted and VPN Traffic using Time-related Features. En: En *Proceedings of the 2nd International Conference on Information Systems Security and Privacy - ICISSP*. INSTICC. SciTePress, 2016, 407-414. ISBN: 978-989-758-167-0. https://doi.org/10.5220/0005740704070414.

Epicalsoft. (2018). Azure Machine Learning Tipos de problemas en Machine Learning [Accessed: 2023-5-12].

Fan, Z., & Liu, R. (2017). Investigation of machine learning based network traffic classification. *2017 International Symposium on Wireless Communication Systems (ISWCS)*, 1-6.

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, *17*(3), 37-37.

Figuerola, N. (2013). SDN–Redes definidas por Software. *Línea Dispon. En Httpsarticulosit Files Wordpress Com201310sdn Pdf*.

Finsterbusch, M., Richter, C., Rocha, E., Muller, J.-A., & Hanssgen, K. (2013). A survey of payload-based traffic classification approaches. *IEEE Communications Surveys & Tutorials*, *16*(2), 1135-1156.

G, R., & R, M. (2021). Prediction Based Dynamic Controller Placement in SDN. *EAI Endorsed Transactions on Scalable Information Systems*, *8*(32). https://doi.org/10.4108/eai.27-4-2021.169420

Gallo, P., Kosek-Szott, K., Szott, S., & Tinnirello, I. (2016). SDN@home: A method for controlling future wireless home networks. *IEEE Communications Magazine*, *54*(5), 123-131. https://doi.org/10.1109/MCOM.2016.7470946

Ganesan, E., Hwang, I.-S., Liem, A., & Ab Rahman, M. S. (2021). SDN-Enabled FiWi-IoT Smart Environment Network Traffic Classification Using Supervised ML Models. *Photonics*, *8*, 201. https://doi.org/10.3390/photonics8060201

George, B. (2017). A study of the effect of random projection and other dimensionality reduction techniques on different classification methods. *Baselius Res*, *18*, 201769.

Goli, Y. D., & Ambika, R. (2018). Network traffic classification techniques-a review. *2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS)*, 219-222.

Gutiérrez Galeano, L. J., et al. (2022). Predicción de ciberataques mediante el empleo de algoritmos deep learning.

Hayes, M., Ng, B., Pekar, A., & Seah, W. K. G. (2018). Scalable Architecture for SDN Traffic Classification. *IEEE Systems Journal*, *12*(4), 3203-3214. https://doi.org/10.1109/JSYST.2017.2690259

Hu, F., Hao, Q., & Bao, K. (2014). A survey on software-defined network and openflow: From concept to implementation. *IEEE Communications Surveys & Tutorials*, *16*(4), 2181-2206.

Huber, S., Wiemer, H., Schneider, D., & Ihlenfeldt, S. (2019). DMME: Data mining methodology for engineering applications – a holistic extension to the CRISP-DM model [12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy]. *Procedia CIRP*, *79*, 403-408. https://doi.org/https://doi.org/10.1016/j.procir.2019.02.106

Lashkari, A. H., Zang, Y., Owhuo, G., Mamun, M., & Gil, G. (2017). CICFlowMeter. *GitHub.[vid. 2021-08-10]. Dostupné z: https://github. com/ahlashkari/CICFlowMeter/blob/master/ReadMe. txt*.

Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, *5*(9), 1315-1316.

Mondal, P. K., Aguirre Sanchez, L. P., Benedetto, E., Shen, Y., & Guo, M. (2021). A dynamic network traffic classifier using supervised ML for a Docker-based SDN network. *Connection Science*, *33*(3), 693-718.

Moore, A., Zuev, D., & Crogan, M. (2005). *Discriminators for use in flow-based classification* (inf. téc.).

Naas, M., & Fesl, J. (2023). A novel dataset for encrypted virtual private network traffic analysis. *Data in Brief*, *47*, 108945.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of big data*, *2*(1), 1-21.

Nguyen, T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials*, *10*(4), 56-76. https://doi.org/10.1109/SURV.2008.080406

Noormohammadpour, M., & Raghavendra, C. S. (2017). Datacenter traffic control: Understanding techniques and tradeoffs. *IEEE Communications Surveys & Tutorials*, *20*(2), 1492-1525.

Nunes, B. A. A., Mendonca, M., Nguyen, X.-N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications surveys & tutorials*, *16*(3), 1617-1634.

Nunez-Agurto, D., Fuertes, W., Marrone, L., & Macas, M. (2022). Machine Learning-Based Traffic Classification in Software-Defined Networking: A

Systematic Literature Review, Challenges, and Future Research Directions. *IAENG International Journal of Computer Science*, *49*(4).

Owusu, A. I., & Nayak, A. (2020). An intelligent traffic classification in sdn-iot: A machine learning approach. *2020 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 1-6.

Pradhan, A., & Mathew, R. (2020). Solutions to Vulnerabilities and Threats in Software Defined Networking (SDN) [Third International Conference on Computing and Network Communications (CoCoNet'19)]. *Procedia Computer Science*, *171*, 2581-2589. https://doi.org/https://doi.org/10.1016/j.procs.2020.04.280

Ríos, R. A. (2016). Conceptualización de SDN y NFV. *Maskay*, *6*(1), 29-34.

Rojas, E. M. (2020). Machine Learning: análisis de lenguajes de programación y herramientas para desarrollo. *Revista Ibérica de Sistemas e Tecnologias de Informação*, (E28), 586-599.

Rouhiainen, L. (2018). Inteligencia artificial. *Madrid: Alienta Editorial*.

SAS Institute Documentation. (2023). The acronym SEMMA – sample, explore, modify, model, assess – Introduction to SEMMA. [Accessed: 2023-5-12].

Silva, J. (2021). Tecnología de red definida por software para el aprendizaje en grupos de investigación y educación. *Revista Innova Educación*, *3*(3), 85-96.

Su, C., Liu, Y., & Xie, X. (2023). Fine-grained Traffic Classification Based on Improved Residual Convolutional Network in Software Defined Networks. *IEEE Latin America Transactions*, *21*(4), 565-572.

Tahaei, H., Afifi, F., Asemi, A., Zaki, F., & Anuar, N. B. (2020). The rise of traffic classification in IoT networks: A survey. *Journal of Network and Computer Applications*, *154*, 102538.

Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*.

Xie, J., Yu, F. R., Huang, T., Xie, R., Liu, J., Wang, C., & Liu, Y. (2019). A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges. *IEEE Communications Surveys & Tutorials*, *21*(1), 393-430. https://doi.org/10.1109/COMST.2018.2866942

Xu, J., Wang, J., Qi, Q., Sun, H., & He, B. (2018). Deep neural networks for application awareness in SDN-based network. *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, 1-6.

Zaki, F. A. M., & Chin, T. S. (2019). FWFS: Selecting robust features towards reliable and stable traffic classifier in SDN. *IEEE Access*, *7*, 166011-166020.