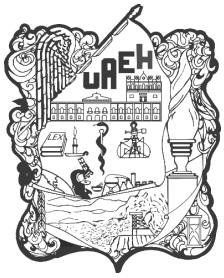


UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

ESCUELA SUPERIOR DE TLAHUELILPAN

LICENCIATURA EN INGENIERÍA DE SOFTWARE

TESIS

PREDICCIÓN DE ATAQUES CIBERNÉTICOS UTILIZANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO

Para obtener el título de

Licenciado en Ingeniería de Software

PRESENTA

Eric Jared Villeda Reyes

Director

Dr. Gabriel Sánchez Bautista

Comité tutorial

Mtra. Mónica García Munguía
Mtra. Mónica Cornejo Velázquez
Dr. Gabriel Sánchez Bautista
Dra. Silvia Patricia Ambrocio Cruz

Tlahuelilpan, Hidalgo., noviembre 2025

Universidad Autónoma del Estado de Hidalgo
Escuela Superior de Tlahuelilpan
Campus Tlahuelilpan

28 de noviembre de 2025

Asunto: Autorización de impresión formal.

M.C. MIGUEL ÁNGEL DE LA FUENTE LÓPEZ
Director de la Escuela Superior de Tlahuelilpan

Manifestamos a usted que se autoriza la impresión formal del trabajo de investigación del pasante ERIC JARED VILLEDA REYES, bajo la modalidad de tesis individual, cuyo título es: "PREDICCIÓN DE ATAQUES CIBERNÉTICOS UTILIZANDO TÉCNICAS DE APRENDIZAJE AUTOMÁTICO", debido a que reúne los requisitos de decoro académico a que obligan los reglamentos en vigor para ser discutidos por los miembros del jurado.

"AMOR, ORDEN Y PROGRESO"

Nombre de integrantes del jurado	Cargo	Firma
Mtra. Mónica García Munguía	Presidente	
Mtra. Mónica Cornejo Velázquez	Secretario	
Dr. Gabriel Sánchez Bautista	Vocal	
Dra. Silvia Patricia Ambrocio Cruz	Suplente	

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO
ESCUELA SUPERIOR DE TLAHUELILPAN

Ex-Hacienda de San Servando S/N, Col. Centro,
Tlahuelilpan, Hidalgo, México; C.P. 42780
Teléfono: 771 71 720 00 Ext. 50601 y 50603
esc_sup_tlahuelilpan@uaeh.edu.mx
uaeh.edu.mx

Agradecimientos

Quiero expresar mi más profundo agradecimiento a la institución por su apoyo a lo largo de mi formación académica, la cual ha sido fundamental para la realización de este trabajo.

Agradezco profundamente a mi asesor, Gabriel Sánchez Bautista, por su guía experta, paciencia y valiosos consejos que orientaron cada etapa de este proyecto y me ayudaron a superar los desafíos del proceso de investigación.

También agradezco al equipo de la carrera de Licenciatura en Ingeniería de Software de la Universidad por compartir su conocimiento y orientación durante el desarrollo de este trabajo.

Finalmente, agradezco a mi familia y amigos por su apoyo incondicional y aliento; su confianza y ánimo fueron un pilar fundamental para completar con éxito este proyecto.

Resumen

La creciente frecuencia e impacto de los ciberataques se ha convertido en un desafío crítico para la seguridad digital, especialmente en entornos donde la alfabetización digital es baja o la conciencia tecnológica es limitada. En estos casos, los usuarios son más susceptibles a prácticas maliciosas. Este trabajo se centra en la predicción de ataques cibernéticos utilizando técnicas de aprendizaje automático, con el objetivo de identificar patrones comunes en amenazas como el phishing, el ransomware y los ataques de denegación de servicio (DDoS).

Para este estudio, se recopilaron y procesaron conjuntos de datos tanto públicos como académicos. Se aplicaron modelos de machine learning, como CatBoost, Random Forest, SVM, K-Nearest Neighbors (KNN), XGBoost y Regresión Logística, para evaluar su rendimiento en la detección de ciberataques. Se llevaron a cabo pruebas comparativas para cada tipo de amenaza, utilizando métricas de precisión y capacidad predictiva.

Los resultados indican que CatBoost tuvo un rendimiento excepcional en la detección de phishing (0.99), mientras que Random Forest brilló en la identificación de ransomware (0.98) y KNN alcanzó un impresionante (0.99) en la detección de tráfico DDoS. Esto confirma la efectividad del aprendizaje automático en la lucha contra las amenazas. Entre los hallazgos más destacados, se encontraron patrones distintivos en URLs maliciosas, cambios en el tráfico de red y un uso inusual de protocolos, lo que subraya la necesidad de combinar técnicas automatizadas con estrategias preventivas.

Este estudio establece las bases para crear herramientas de ciberseguridad más sólidas y accesibles. Como parte del trabajo futuro, se propone desarrollar una plataforma web educativa que incluya un sistema predictivo para detectar phishing, con el objetivo de aumentar la concienciación y la protección contra ciberataques.

Abstract

The increasing frequency and impact of cyberattacks has become a critical challenge for digital security, especially in environments with low digital literacy or limited technological awareness. In such cases, users are more vulnerable to malicious activities. This study focuses on the prediction of cyberattacks using machine learning techniques, aiming to identify common patterns in threats such as phishing, ransomware, and denial-of-service (DDoS) attacks.

Public and academic datasets were collected and processed to evaluate the performance of machine learning models, including CatBoost, Random Forest, SVM, K-Nearest Neighbors (KNN), XGBoost, and Logistic Regression. Comparative tests were conducted for each type of threat, using accuracy and predictive capability metrics.

Results indicate that CatBoost achieved an outstanding performance in phishing detection 0.99, Random Forest excelled in ransomware identification 0.98, and KNN reached 0.99 in detecting DDoS traffic. These findings demonstrate the effectiveness of machine learning in combating cyber threats. Notable patterns were observed in malicious URLs, network traffic anomalies, and unusual protocol usage, highlighting the need to combine automated techniques with preventive strategies.

This study provides a foundation for the development of more robust and accessible cybersecurity tools. As future work, the creation of an educational web platform is proposed, incorporating a predictive system for phishing detection to enhance awareness and protection against cyberattacks.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Construcción del objeto de estudio

1.1. Introducción

La transformación digital y la creciente dependencia de las tecnologías de la información han incrementado significativamente la exposición de los sistemas informáticos a diversas amenazas cibernéticas. En este entorno, México ha experimentado un crecimiento en su demanda digital. Para 2024, más de 100 millones de personas, siendo el 83.1 % de la población de 6 años o más, ya utilizaban Internet, un aumento sustancial respecto a años anteriores [9].

A su vez, las organizaciones PyMEs han acelerado su proceso de digitalización, revelando que el 95 % plantea invertir en esta área y el 80 % de estas ya son parte, lo que indica una adaptación activa de tecnologías digitales como colaboración, marketing y ciberseguridad [10]. Además, datos del INEGI señalan que el 95.6 % de los negocios medianos cuenta con equipo de cómputo y el 91.9 % utiliza Internet en sus actividades diarias [11].

Entre los ataques más frecuentes y problemáticos se encuentra el phishing, el ransomware y los ataques distribuidos de denegación de servicio (DDoS), los cuales representan riesgos críticos tanto para usuarios individuales como para organizaciones. La velocidad, complejidad y volumen de estos ataques hacen cada vez más necesario el desarrollo de mecanismos predictivos que permitan anticipar su ocurrencia y fortalecer las defensas cibernéticas.

En este contexto, el uso de técnicas de aprendizaje automático (machine learning) ha cobrado relevancia como una estrategia eficaz para detectar patrones anómalos y realizar predicciones sobre comportamientos maliciosos en

tiempo real.

Enfocando en la predicción de tres tipos de ataques cibernéticos comunes: phishing, ransomware y DDoS, mediante la aplicación de técnicas de aprendizaje automático. Para lograrlo, se emplearon tres conjuntos de datos públicos y se implementaron diversos modelos de clasificación utilizando herramientas como Visual Studio Code y Python, junto con bibliotecas especializadas. El proceso incluyó etapas de preprocesamiento de datos, análisis exploratorio, entrenamiento, validación y predicción utilizando algoritmos como Random Forest, Support Vector Machines, Naive Bayes, XGBoost, Logistic Regression y k-Nearest Neighbors.

Los resultados obtenidos permitieron evaluar la eficacia de los modelos frente a los tres tipos de ataques analizados, utilizando métricas como: precisión, accuracy, recall (sensibilidad) y F1-score. Los valores alcanzados por estos indicadores reflejan el potencial del aprendizaje automático como herramienta eficaz para apoyar la detección temprana y la prevención de amenazas en el ámbito de la ciberseguridad.

1.2. Planteamiento del Problema

La problemática que se abordará en este estudio se enmarca en la falta de concienciación y preparación en ciberseguridad por parte de individuos con bajo nivel de alfabetización digital y organizaciones que están en proceso de adaptación tecnológica. Se ha observado que muchas de estas personas y entidades no son plenamente conscientes de la importancia crítica de la seguridad cibernética, ni entienden cómo protegerse adecuadamente de las amenazas cibernéticas en un mundo que se está digitalizando cada vez más. Esta falta de preparación y conocimiento nos convierte en objetivos vulnerables a ataques, los cuales pueden tener consecuencias altas, incluso en organizaciones de menor escala o con recursos limitados.

1.3. Justificación

La justificación de esta investigación se basa en una serie de razones sólidas y necesidades urgentes que deben abordarse:

1. Falta de datos y conciencia: no hay suficiente información o datos disponibles que sugiera que muestre la incidencia de ciberataques, además se ha experimentado un aumento significativo en los ciberataques en los últimos años, consolidándose como uno de los países más afectados por

esta problemática. Nueve de cada diez ciberataques podrían prevenirse con una adecuada educación y concientización [12].

2. Difusión del problema: la gente piensa que en la región no hay muchos ataques de ciberseguridad. Sin embargo, la falta de información precisa indica que el verdadero alcance de la amenaza es desconocido. Encuestas nacionales mencionan que el 25 % de las y los adolescentes de 12 a 17 años han vivido alguna forma de ciberataque o ciberacoso [13].

3. Grupos vulnerables: Si ocurriera un gran número de ciberataques en la región, los más afectados podrían ser estudiantes, menores de edad e incluso propietarios o empleados de pequeñas y medianas empresas que carecen de conocimientos sobre seguridad cibernética y que están adoptando tecnologías digitales sin alguna preparación adecuada. En México el sector educativo se encuentra en el tercer lugar de eventos de ciberataques con más de 3 millones de ataques a sus sistemas [14]. Por otra parte, mas de la mitad de las PyMEs en América Latina reportan un aumento de ciberataques y un 20 % de estas reconoce no estar preparada para prevenirlos [15].

1.4. Objetivos de la investigación

Este trabajo consta de un objetivo general y cinco específicos que ayuden a desarrollar una plataforma que presente el resultado del análisis de datos.

1.4.1. Objetivo general

Evaluar y analizar ataques cibernéticos mediante técnicas de aprendizaje automático, con el fin de identificar patrones comunes de amenazas que afectan a estos sectores. Este análisis busca aportar conocimiento sobre las características y frecuencia de los ciberataques, así como sentar las bases para futuras estrategias de prevención y concienciación en materia de seguridad digital.

1.4.2. Objetivos específicos

Realizar una investigación de la actual conciencia en seguridad cibernética:

- Analizar estudios previos para determinar el nivel de conocimiento y comprensión de la seguridad cibernética en sectores vulnerables.

- Identificar las lagunas de conocimiento más importantes y las vulnerabilidades de seguridad cibernética de las áreas.

Recopilar información sobre los diferentes tipos de ciberataques:

- Investigar y recopilar información actualizada sobre los tipos de ciberataques más comunes, como phishing, ransomware y ataques de ingeniería social.
- Analizar casos de estudio y tendencias globales para comprender las amenazas cibernéticas actuales y futuras.
- Recopilar información sobre los tipos de ciberataques que han ocurrido en el pasado de la zona para identificar patrones y tendencias comunes. Esto ayudará a comprender las áreas de mayor vulnerabilidad y los escenarios más frecuentes de amenazas cibernéticas.

Aplicar técnicas de aprendizaje automático para el análisis de amenazas cibernéticas:

- Utilizar modelos de machine learning para procesar y clasificar información sobre incidentes de seguridad cibernética registrados.
- Identificar patrones y tendencias en los datos mediante algoritmos supervisados y no supervisados, con el fin de comprender mejor el comportamiento de los ataques.

1.5. Pregunta de investigación

¿Puede el uso de algoritmos de aprendizaje automático identificar patrones y correlaciones en los datos de incidentes cibernéticos que permitan identificar amenazas latentes?

1.6. Hipótesis

La utilización de algoritmos predictivos para la evaluación de datos sobre incidentes cibernéticos revelará patrones emergentes y posibles correlaciones, facilitando la identificación de amenazas latentes.

1.7. Propuesta de solución

Se propone una solución que combina el análisis avanzado de patrones de ataques cibernéticos con una plataforma informativa interactiva. Este enfoque permitirá identificar patrones y tendencias emergentes en los métodos de ataque cibernético, brindando una comprensión profunda de las tácticas utilizadas. En términos específicos, se abordarán los siguientes puntos clave.

Análisis detallado de patrones de ataque: Se llevará a cabo una exhaustiva recopilación de datos provenientes de diversas fuentes, incluyendo bases de datos de ataques cibernéticos y sitios web de seguridad informática. Estos datos se analizarán en busca de patrones y tendencias utilizando herramientas avanzadas de análisis de datos y aprendizaje automático.

Generación de modelos predictivos: Con base en el análisis realizado, se desarrollarán modelos predictivos capaces de anticipar posibles ataques y detectar comportamientos anómalos, contribuyendo así a la prevención y mitigación de riesgos en entornos digitales.

1.8. Metodología

La metodología propuesta está estructurada en dos etapas, cada una para tomar los aspectos específicos de la propuesta de solución.

1.8.1. Investigación del problema.

Se llevará a cabo una exhaustiva investigación con el fin de identificar con precisión cuál es el problema que afecta a la comunidad estudiantil.

1. Observación de la problemática: Se examinará la situación relacionada con los ciberataques que afecta a los sectores vulnerables.

2. Identificación del problema: Se detectarán las lagunas o deficiencias de información en el ámbito de los ciberataques.

3. Comparación de trabajos: Se llevará a cabo un estado del arte sobre la misma problemática identificada, seleccionando artículos o tesis relevantes para su comparación y obteniendo una visión general de las investigaciones previas realizadas.

1.8.2. Análisis detallado de patrones de ataque.

Se llevará a cabo una investigación para comprender los patrones y tendencias de los ataques cibernéticos.

1. Recopilación de datos: Se identificarán fuentes confiables de datos, incluyendo bases de datos de ataques cibernéticos y sitios web de seguridad informática. Posteriormente se extraerán y serán almacenados de forma segura para su análisis.

2. Tratamiento de los datos: Se llevará a cabo la limpieza de los datos recopilados previamente, así como la identificación de las variables más relevantes que serán utilizadas en el análisis posterior.

3. Análisis detallado: Se utilizarán técnicas de análisis de datos avanzado y aprendizaje automático para la identificación de patrones y tendencias en los datos recopilados.

1.9. Alcances y limitaciones

Los alcances de esta investigación son proporcionar un análisis detallado de los patrones de ataque cibernético. Este análisis permitirá una mejor comprensión de las tácticas que utilizan los ciberdelincuentes, siendo fundamental para desarrollar estrategias de defensa efectivas y contribuyendo a aumentar la conciencia sobre la importancia de la seguridad cibernética. Al proporcionar información y resultados accesibles, se espera que más personas comprendan la gravedad de las amenazas cibernéticas y tomen medidas para protegerse. Las variables que serán investigadas con mayor importancia son:

- Patrones y tendencias de ataques cibernéticos. Analizando métodos utilizados por los ciberdelincuentes para identificar patrones y tendencias emergentes.

- Concienciación en seguridad cibernética. Se evaluará el nivel de concienciación de la comunidad interesada y no interesada en la seguridad cibernética.

- Experiencias de ataque cibernético. Se investigará si as personas en la región han sido víctimas de ataques cibernéticos en el pasado, la frecuencia de estos y como han respondido a ellos.

- Uso de medios sociales. Se analizará como las personas suelen utilizar las redes sociales y otras plataformas informativas y si son conscientes de los riesgos asociados con la divulgación de información personal en estos entornos.

Las limitaciones en esta investigación pueden abarcar desde restricciones de datos, acceso limitado a recursos hasta cuestiones éticas. Estos factores pueden

tener un impacto erróneo en la validez, la generalización y la aplicabilidad de los resultados, así como la calidad y la escalabilidad.

- Limitaciones de recursos:
  - La calidad y disponibilidad de datos. Las fuentes de datos podrían tener limitaciones en cuanto a su veracidad y precisión, lo que puede afectar la validez de los patrones de ataque identificados.
  - Acceso a ciertas bases de datos o sitios web. El acceso podría ser restringido debido a cuestiones de seguridad o privacidad, lo que podría limitar la cantidad y la variedad de datos disponibles para el análisis.
- Participación de usuarios:
  - La participación de usuarios puede ser limitada por factores como la disponibilidad de tiempo, su nivel de interés y compromiso, así como su disposición a proporcionar información precisa y completa.
- Limitaciones de ética:
  - Cuando se abordan temas sensibles, como la privacidad, confidencialidad o consentimiento informado de los participantes, pueden surgir restricciones en la recopilación y el uso de ciertos datos, afectando la integridad y la aplicabilidad de los resultados obtenidos.

1.10. Organización del documento

El presente trabajo se estructura de la siguiente manera:

- Capitulo 2: Presenta los fundamentos conceptuales de la investigación. Se explican los algoritmos de aprendizaje automático, se analiza el panorama actual de los ciberataques con énfasis en sectores vulnerables, y se revisan trabajos previos relacionados en el estado del arte, comparando enfoques y resultados.
- Capitulo 3: Se describen las fuentes utilizadas de los conjuntos de datos. Luego, se explica de manera general la preparación realizada para garantizar su utilidad. Finalmente, se presenta un análisis exploratorio que incluye gráficas relevantes de los datos.

- Capitulo 4: Presenta los resultados obtenidos del entrenamiento de los modelos de aprendizaje automático aplicados a los diferentes tipos de ataques analizados: phishing, ransomware y DoS/DDoS. Para cada caso se incluyen comparativas de desempeño y representaciones gráficas. Asimismo, se destacan los hallazgos más relevantes en cuanto a patrones detectados en los datos y, finalmente, se muestran las predicciones realizadas con los datos de prueba para evaluar la capacidad real de los modelos.

- Capitulo 5: Presenta las conclusiones principales del estudio sobre la predicción de ciberataques mediante aprendizaje automático. Además, se incluye la propuesta para trabajo a futuro, enfocadas en la mejora de la prevención y concienciación sobre amenazas cibernéticas.

Capítulo 2

Marco teórico

2.1. Algoritmos de aprendizaje automático

Los algoritmos de aprendizaje automático (machine learning) son un conjunto de técnicas y modelos matemáticos que permiten a las computadoras aprender de datos, identificar patrones y hacer predicciones o decisiones sin ser explícitamente programadas para realizar tareas específicas. En lugar de ser programados con instrucciones estáticas, estos algoritmos utilizan datos para entrenarse y mejorar su rendimiento a medida que reciben más información [16]. Las aplicaciones de estos son extensas, abarcando cualquier área que nos podamos imaginar, alguno de los ejemplos son:

- Identificación de patrones.
- Reconocimiento de imágenes y videos.
- Recomendaciones personalizadas.
- Vehículos autónomos.
- Predicción de fallos.
- Procesamiento de lenguaje natural.

Además, estos algoritmos se clasifican en diferentes categorías con el propósito de manejar distintos tipos de datos.

2.1.1. Algoritmos supervisados

Algoritmo de aprendizaje automático en el que se entrena un modelo utilizando un conjunto de datos que contiene tanto las entradas (características) como las salidas (etiquetas o valores deseados). El objetivo del algoritmo es aprender a mapear las entradas a salidas correctas para hacer predicciones sobre nuevos datos [17]. Sus características principales son:

- Los datos incluyen las características (entradas) como las etiquetas o valores asociados (salidas).
- El objetivo es minimizar la diferencia entre las predicciones del modelo y las respuestas correctas (etiquetas).
- Es comúnmente utilizado en tareas de clasificación (cuando las salidas son categóricas).
- La calidad del modelo se evalúa usando métricas como precisión, Recall, F1-score (para clasificación) o el error cuadrático medio (para regresión).

En la siguiente imagen se presenta un diagrama del flujo de los modelos de aprendizaje supervisado, donde se visualiza el tratamiento de los datos divididos para obtener un resultado.

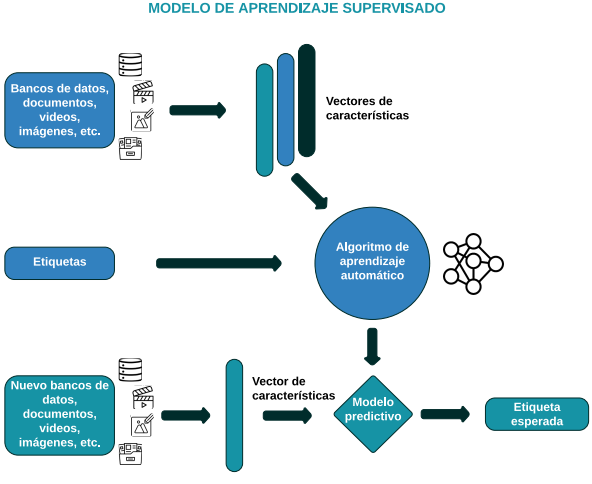


Figura 2.1: Diagrama de flujo del aprendizaje supervisado.

Estos tipos de algoritmos tienen la ventaja principal de ser altamente efectivos cuando se dispone de un conjunto de datos etiquetado de calidad, ya que su objetivo es aprender sobre una relación directa entre las entradas y las salidas, lo que puede resultar en modelos precisos y fáciles de interpretar. Además, estos modelos permiten evaluar su rendimiento mediante métricas claras. Sin embargo, una de sus desventajas es que requieren una gran cantidad de datos etiquetados, lo que puede ser costoso y laborioso de obtener. Además, los modelos supervisados pueden sufrir de sobreajuste (overfitting) si no se gestionan adecuadamente, lo que significa que se ajustan demasiado a los datos de entrenamiento y tienen un rendimiento deficiente en nuevos datos no vistos. Ejemplos de algoritmos supervisados:

Árboles de decisión (Decision Trees)

Los Árboles de Decisión son un algoritmo tanto para clasificación como de regresión. Su funcionamiento se basa en una serie de preguntas binarias sobre las características de los datos, que van dividiendo el conjunto en subgrupos más pequeños en cada nodo del árbol. Cada rama representa un posible resultado derivado de una condición, mientras que las hojas corresponden a las predicciones finales, como se muestra en la siguiente imagen [18].

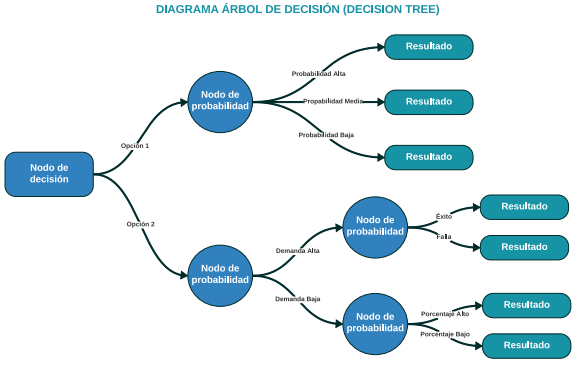


Figura 2.2: Diagrama de un Árbol de decisión.

Los Árboles de decisión tienen características como:

- División jerárquica: Los árboles de decisión dividen iterativamente el conjunto de datos en subconjuntos que proporciona la mayor 'pureza' en las

clases.

- Criterios de división: Los criterios más comunes para dividir los nodos son la 'Gini impurity' y la 'ganancia de información' (usada en la entropía).
- Interpretabilidad: Es fácil de interpretar, ya que las decisiones siguen una estructura lógica y son visualmente comprensibles.

La siguiente tabla resume las ventajas y desventajas del algoritmo.

Ventajas	**Desventajas**
Visualización intuitiva	Falta de precisión en comparación con otros modelos
No requiere normalización	Sesgo de características dominantes
Poca influencia de los valores faltantes	Limitado a divisiones rectas
Robusto frente a datos irrelevantes	Crecimiento descontrolado

Tabla 2.1: Ventajas y desventajas de los árboles de decisión.

Bosque Aleatorio (Random Forest)

El Bosque Aleatorio (Random Forest) es un algoritmo que consiste en una combinación de múltiples árboles de decisión como se muestra en la Figura 2.3. En lugar de construir un solo árbol, el bosque aleatorio construye muchos árboles de decisión y promedia sus resultados (en el caso de la regresión) o elige el voto mayoritario (en el caso de la clasificación). La idea principal es reducir el riesgo de sobre ajuste al promediar los resultados de muchos árboles que fueron entrenados en diferentes subconjuntos de datos.

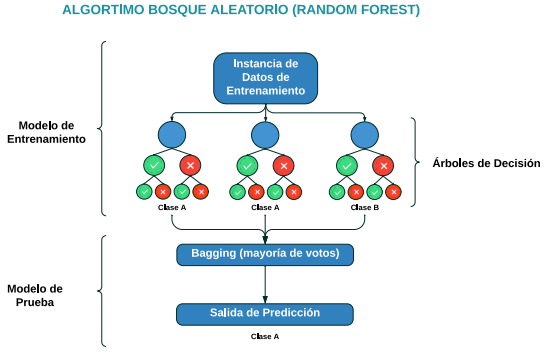


Figura 2.3: Diagrama de un Bosque aleatorio.

Los Bosques aleatorios tienen características como:

- Bagging (Bootstrap Aggregating): Cada árbol se entrena en un subconjunto aleatorio de los datos originales, mejorando la generalización del modelo.
- Selección aleatoria de características: Además de seleccionar subconjuntos de datos, cada árbol se entrena usando una selección aleatoria de características, lo que reduce la correlación entre los árboles.
- Mejora de precisión: Al combinar varios árboles de decisión, el bosque aleatorio tiende a ser más preciso y menos propenso al sobreajuste que un árbol de decisión individual.

La siguiente tabla resume las ventajas y desventajas del algoritmo.

Ventajas	**Desventajas**
Resistente a los datos faltantes	Mayor uso de memoria
Reduce la varianza	Rendimiento disminuye en tiempo real
Identificación de características importantes	No es efectivo en datos dispersos o de alta dimensionalidad
Escalabilidad	Tendencia a ser un 'caja negra'

Tabla 2.2: Ventajas y desventajas de los Bosques aleatorios.

Maquina de Vectores de Soporte (SVM, por sus siglas en inglés: Support Vector Machine)

La Máquina de Vectores de Soporte es un algoritmo de aprendizaje supervisado utilizado para clasificación y regresión. Su objetivo principal es encontrar un hiperplano óptimo que separe las distintas clases de datos con el mayor margen posible. En problemas no lineales, SVM puede aplicar técnicas de transformación de datos mediante funciones kernel, permitiendo proyectar los datos a espacios de mayor dimensión donde sean separables linealmente. Este enfoque reduce el riesgo de sobreajuste y mejora la capacidad de generalización del modelo [19].

La siguiente figura describe cada cómo el algoritmo traza un hiperplano que separa los datos en distintas clases, maximizando la distancia entre dicho hiperplano y los puntos más cercanos.

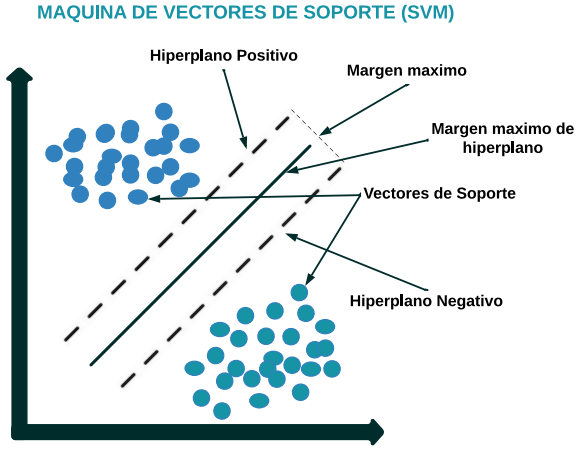


Figura 2.4: Diagrama Maquina de Vectores de Soporte.

Las Máquinas de Vectores de Soporte tienen características como:

- Maximización del margen: SVM busca el hiperplano que maximiza la distancia entre las clases, lo que mejora la capacidad de generalización del modelo al separar los datos con el mayor margen posible.
- Uso de kernel: SVM puede aplicar funciones kernel para transformar datos no lineales en espacios de mayor dimensión, permitiendo encontrar un hiperplano lineal que separa las clases incluso en casos complejos.

- Reducción del sobreajuste: SVM tiene un buen control sobre el sobreajuste, ya que busca un margen lo más grande posible sin ajustar excesivamente a los datos de entrenamiento.

La siguiente tabla resume las ventajas y desventajas del algoritmo.

Ventajas	**Desventajas**
Alta precisión en clasificación	Requiere mucho tiempo de entrenamiento
Reducción del sobreajuste	No funciona bien con ruido en los datos
Eficaz en espacios de alta dimensión	Lento en grandes volúmenes de datos
Versatilidad con funciones kernel	Ajuste de parámetros complejo

Tabla 2.3: Ventajas y desventajas de las Maquinas de Vectores de Soporte.

CatBoost

CatBoost es un algoritmo de aprendizaje automático basado en el método de Gradient Boosting sobre árboles de decisión. Está diseñado para manejar datos categóricos de forma eficiente, sin necesidad de realizar una codificación manual compleja. CatBoost utiliza una técnica innovadora llamada Ordered Boosting que reduce el sobreajuste y mejora la generalización del modelo. Es especialmente útil en tareas de clasificación y regresión con datos estructurados, y se destaca por su robustez, velocidad y alta precisión en competiciones de ciencia de datos [20].

El siguiente diagrama muestra cómo el modelo combina múltiples árboles, reduciendo errores en cada iteración y logrando una clasificación más precisa. Esta visualización permite comprender cómo CatBoost maneja variables categóricas y mejora el rendimiento en comparación con métodos tradicionales.

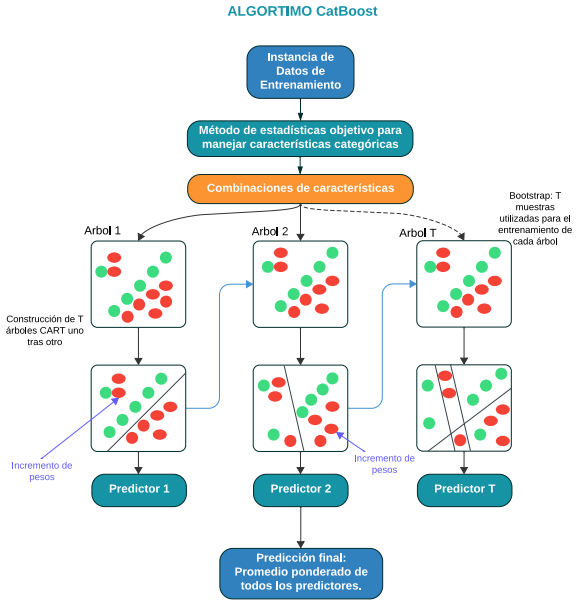


Figura 2.5: Diagrama CatBoost.

CatBoost tiene características como:

- Alto rendimiento y precisión: Basado en árboles de decisión que destaca por su capacidad para generar modelos altamente precisos, incluso con pocos ajustes manuales.
- Manejo eficiente de variables categóricas: A diferencia de otros algoritmos, CatBoost puede procesar directamente variables categóricas sin necesidad de codificarlas previamente, gracias a su enfoque de estadísticas objetivo.
- Reducción del sobreajuste: Utiliza una técnica llamada Ordered Boosting que evita el uso de datos futuros durante el entrenamiento, lo que reduce significativamente el sobreajuste y mejora la generalización del modelo.
- Construcción secuencial de árboles: Entrena múltiples árboles CART de forma secuencial, donde cada árbol corrige los errores del anterior, utilizando un enfoque de promedio ponderado de predictores para la predicción final.

- Buen rendimiento con conjuntos de datos heterogéneos: Funciona especialmente bien con datos tabulares, que pueden contener una mezcla de variables numéricas y categóricas, y ofrece resultados competitivos sin necesidad de mucha ingeniería de características.

La siguiente tabla resume las ventajas y desventajas del algoritmo.

Ventajas	**Desventajas**
Alto rendimiento y precisión	Mayor complejidad computacional
Manejo automático de variables categóricas	Interpretabilidad limitada frente a modelos más simples
Menor riesgo de sobreajuste (Ordered Boosting)	Puede requerir ajuste de hiperparámetros para máximo rendimiento
Eficaz con datos tabulares mixtos	Entrenamiento más lento comparado con modelos lineales

Tabla 2.4: Ventajas y desventajas de CatBoost.

XGBoost, por sus siglas en inglés: Extreme Gradient Boosting

XGBoost es un algoritmo de clasificación y regresión que utiliza el enfoque de Boosting. Este algoritmo combina múltiples modelos débiles, generalmente árboles de decisión, para crear un modelo más robusto y preciso. En XGBoost, cada nuevo árbol intenta corregir los errores cometidos por los árboles anteriores, lo que mejora progresivamente la predicción del modelo. Es eficaz en problemas complejos y con grandes volúmenes de datos y a menudo se utiliza en competencias de machine learning debido a su rapidez en el entrenamiento y su habilidad para prevenir el sobreajuste [21].

La figura muestra cómo el modelo construye árboles de decisión de manera secuencial, donde cada nuevo árbol corrige los errores cometidos por los anteriores. Este enfoque permite obtener predicciones más precisas y robustas, siendo uno de los algoritmos más utilizados en tareas de clasificación y detección de anomalías.

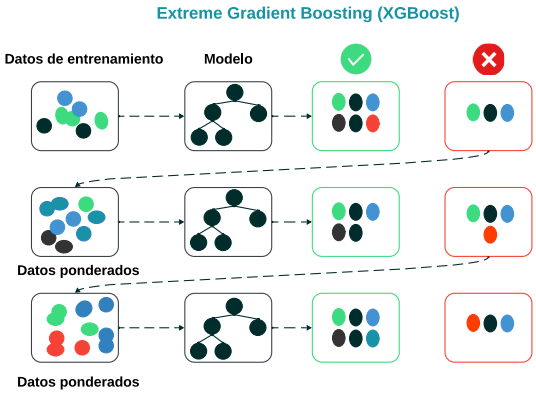


Figura 2.6: Diagrama XGBoost.

XGBoost tienen características como:

- Eficiencia y rapidez: XGBoost es extremadamente rápido en el entrenamiento y la predicción debido a su implementación optimizada. Esto lo hace adecuado para grandes volúmenes de datos y problemas complejos.
- Regularización para evitar el sobreajuste: XGBoost incluye términos de regularización L1 y L2 para controlar el sobreajuste del modelo, mejorando su capacidad de generalización.
- Manejo de datos faltantes: XGBoost maneja de forma automática los datos faltantes durante el entrenamiento, asignando rutas óptimas para mejorar la precisión del modelo.
- Flexibilidad con funciones de pérdida: Puede trabajar con diferentes funciones de pérdida según el problema, lo que permite ajustarlo tanto para tareas de clasificación como de regresión.

La siguiente tabla resume las ventajas y desventajas del algoritmo.

Ventajas	Desventajas
Alta eficiencia y rapidez	Complejidad de configuración
Manejo de sobreajuste	Sensibilidad a datos desbalanceados
Capacidad para detectar interacciones no lineales	Requiere recursos computacionales
Flexibilidad	Dificil interpretación

Tabla 2.5: Ventajas y desventajas de XGBoost.

Regresión Logística (Logistic Regression)

La Regresión Logística es un método de clasificación utilizado para predecir la probabilidad de pertenencia a una clase binaria (0 o 1). Aunque originalmente fue diseñada para problemas de clasificación binaria, también se puede extender a problemas multiclase. Este algoritmo modela la relación entre las características independientes y la variable objetivo utilizando la función sigmoide, lo que permite obtener una probabilidad de pertenencia a una clase específica, en la siguiente figura se representa y facilita la comprensión del proceso de decisión del modelo [22].

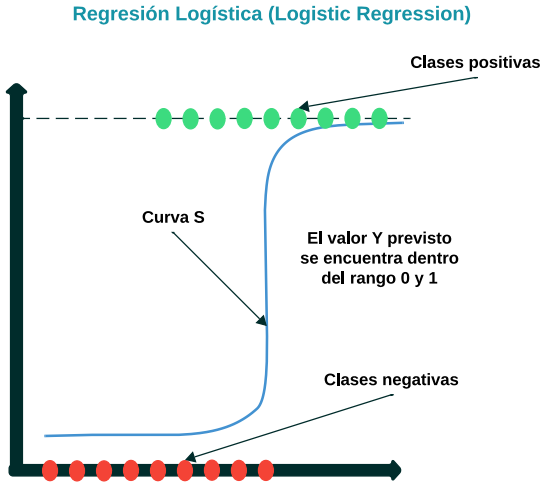


Figura 2.7: Diagrama Regresión Logística.

Regresión Logística tienen características como:

- Interpretabilidad sencilla: El modelo proporciona coeficientes que son fáciles de interpretar y entender, lo que permite analizar el impacto de cada característica en la predicción.
- Probabilidad de predicción: Ofrece como resultado una probabilidad de pertenencia a una clase específica, lo que facilita la toma de decisiones basada en umbrales.
- Eficiencia en conjuntos de datos pequeños: Funciona bien con conjuntos de datos relativamente pequeños y linealmente separables.
- Extensiones para problemas complejos: Puede ampliarse a problemas multiclase con técnicas como la regresión logística multinomial o one-vs-all.

La siguiente tabla resume las ventajas y desventajas del algoritmo.

Ventajas	Desventajas
Simplicidad y rapidez	Limitación lineal
Robusta ante ruido	Sensibilidad a datos desbalanceados
Base para métodos complejos	Poca capacidad para problemas complejos
Probabilidad directa	Dependencia de la multicolinealidad

Tabla 2.6: Ventajas y desventajas de Regresión Logística.

2.1.2. Algoritmos no supervisados

El aprendizaje no supervisado es un tipo de algoritmo de aprendizaje automático que se utiliza cuando los datos no tienen etiquetas o valores de salida predefinidos. El objetivo del algoritmo es encontrar patrones o estructuras subyacentes en los datos, como grupos o relaciones entre las características [23]. Sus características principales son:

- Los datos de entrenamiento no tienen etiquetas, es decir, solo reconocen las características de los datos, pero no se sabe a que categoría o valor pertenecen.
- El algoritmo intenta identificar estructuras o patrones inherentes en los datos, como agrupamiento o reducción de dimensionalidad.
- Comúnmente agrupa datos en grupos o clústeres basados en similitudes, también reduce el numero de características en un conjunto de datos mientras se manteniente la mayor parte de la información.

La siguiente figura muestra un diagrama del flujo de los modelos de aprendizaje no supervisado. Se observa que el tratamiento de los datos es similar al de los modelos supervisados, aunque en este caso las etiquetas no se utilizan, y el algoritmo identifica patrones o estructuras de manera autónoma.

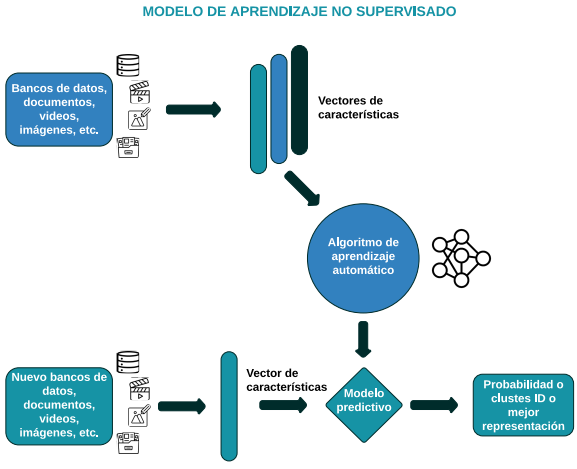


Figura 2.8: Diagrama de flujo del aprendizaje no supervisado.

Estos tipos de algoritmos ofrecen la ventaja de no necesitar datos etiquetados, lo que los hace útiles cuando no se dispone de información etiquetada o cuando es difícil obtenerla. Esto permite explorar patrones y estructuras ocultas en grandes volúmenes de datos, como en el caso del clustering o la reducción de dimensionalidad. Sin embargo, su principal desventaja es que la evaluación de los resultados puede ser más difícil, ya que no hay una 'respuesta correcta' para comparar. Además, los modelos generados pueden ser más complejos de interpretar y es posible que los patrones descubiertos no siempre tengan relevancia práctica o sean difíciles de validar. Ejemplos de algoritmos no supervisados:

K-means

K-means es uno de los algoritmos más conocidos y sencillos utilizado para clustering. Su objetivo es dividir un conjunto de datos en K clústeres, donde cada punto de datos pertenece al clúster con el centroide más cercano. El algoritmo funciona de manera iterativa, ajustando los centroides de los clústeres hasta que la posición de los puntos dentro de los clústeres ya no cambia significativamente [24]. En la siguiente se puede visualizar como detecta y segmentan los datos.

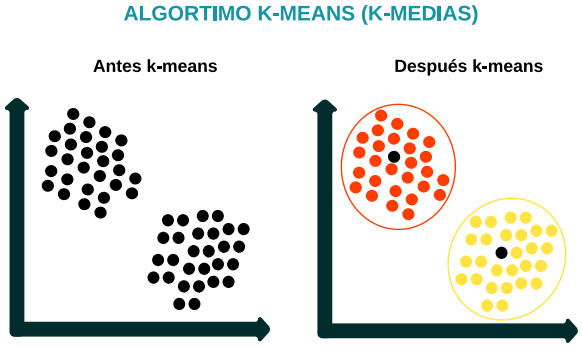


Figura 2.9: Gráfico de K-means.

K-means tiene características como:

- Centroides: El algoritmo asigna cada punto de datos a uno de los K clústeres basándose en la distancia entre el punto y el centroide del clúster.
- Iterativo: El proceso se repite hasta que los centroides no cambian más o hasta alcanzar un número máximo de iteraciones.
- Distancia Euclidiana: K-means generalmente utiliza la distancia euclidiana para calcular la similitud entre los puntos de datos y los centroides.

K-means cuenta con 4 importantes pasos:

1. Inicialización: Se seleccionan aleatoriamente K centroides en el espacio de los datos.
2. Asignación: Cada punto de datos se asigna al centroide más cercano, formando K clústeres.
3. Actualización: Los centroides se recalculan como el promedio de los puntos dentro de cada clúster.
4. Repetición: Se repiten los pasos de asignación y actualización hasta que los centroides no cambian o el número máximo de iteraciones se alcanza.

La siguiente tabla resume las ventajas y desventajas del algoritmo.

Ventajas	Desventajas
Simplicidad	Necesita predefinir K
Escalabilidad	Sensibilidad a la inicialización
Velocidad	Forma de los clústeres
Flexibilidad	Sensibilidad a los outliers

Tabla 2.7: Ventajas y desventajas de los K-means.

Vecinos más cercanos (KNN o K-Nearest Neighbors)

Es un algoritmo de aprendizaje supervisado que se basa en la similitud entre los puntos de datos. No construye un modelo explícito durante una fase de entrenamiento, sino que toma decisiones directamente a partir de los datos almacenados. Para predecir la clase de un nuevo punto, el algoritmo busca los K vecinos más cercanos (medidos mediante una métrica de distancia, como la distancia euclidiana) y asigna la clase más común, en tareas de clasificación y el valor promedio, en tareas de regresión [25].

En la figura se muestra cómo el algoritmo clasifica una nueva observación considerando las etiquetas de sus k vecinos más cercanos, determinando la clase mayoritaria.

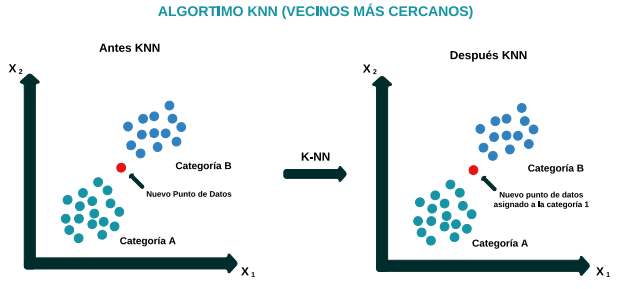


Figura 2.10: Gráfico de KNN.

KNN tiene características como:

- Simplicidad: Es uno de los algoritmos más sencillos de implementar, ya que no requiere de un modelo complicado o un proceso de entrenamiento largo.
- Clasificación basada en proximidad: KNN asigna una etiqueta a un punto basándose en las etiquetas de sus vecinos más cercanos. En problemas de regresión, calcula la media o mediana de los valores de los vecinos.
- Memoria intensiva: KNN es un algoritmo lazy learning, lo que significa que no entrena un modelo previamente; en cambio, almacena todo el conjunto de entrenamiento y realiza cálculos en el momento de la predicción.

KNN cuenta con 4 importantes pasos:

1. Definir K: Elige el número de vecinos más cercanos (K) a considerar para la clasificación o regresión.
2. Calcular la distancia: Para un nuevo punto de datos, calcula la distancia a cada uno de los puntos en el conjunto de entrenamiento utilizando una métrica de distancia (por ejemplo: distancia Euclidiana, Manhattan, etc.).
3. Identificar los K vecinos más cercanos: Selecciona los K puntos del conjunto de entrenamiento más cercanos al punto nuevo.
4. Asignar clase (clasificación) o predecir valor (regresión): En problemas de clasificación, asigna la clase más común entre los K vecinos cercanos. En regresión, devuelve el valor promedio de los vecinos.

La siguiente tabla resume las ventajas y desventajas del algoritmo.

Ventajas	**Desventajas**
Facil de implementar	Costo computacional
No requiere entrenamiento	Memoria intensiva
Capaz de manejar datos multicategóricos	Sensibilidad al valor de K
Flexible	Afectado por la escala de los datos

Tabla 2.8: Ventajas y desventajas de KNN.

2.2. Ciberataques y digitalización: enfoque a los sectores vulnerables

Antes de proceder a la selección y preparación de datos, es importante contextualizar los tipos de ciberataques más comunes y realizados en los sectores mas vulnerables, tanto a nivel global como localmente. Este análisis proporciona una base para comprender el impacto de dichos ataques y su relevancia en el área.

La población con bajo nivel de concienciación en este tema son un blanco fácil para los criminales cibernéticos, ya que, según información publicada por El País, uno de cada cinco delitos en internet se comete a través de dispositivos personales o móviles y se calcula que aproximadamente el 80 % tienen su origen en fallos humanos [26]. En situaciones donde la población carece de conocimiento de seguridad digital, la exposición al malware aumenta de manera exponencial.

A nivel mundial los sectores con menor preparación digital como las pequeñas y medianas empresas (PyMEs) y las personas con baja alfabetización tecnológica se han convertido en blancos recurrentes de la ciberdelincuencia, cada año aumenta significativamente de acuerdo a la creciente digitalización en estos entornos que se ha presentado desde aquella pandemia del 2019. De acuerdo con un informe de Forbes, más del 40 % de los ciberataques están dirigidos a PyMEs, y solo el 14 % de estas se considera preparada para enfrentarlos [27].

Estas amenazas son altamente graves y con consecuencias como:

- Interrupción de operaciones comerciales o de servicios.
- Pérdida o secuestro de información sensible.
- Filtración de datos financieros o personales.
- Alteración de registros.
- Suplantación de identidad.

Desde aquella pandemia en 2019 se ha presentado un gran aumento en los años posteriores sobre los ciberataques, la siguiente figura muestra que el sector educativo fue el mas afectado, recibiendo en 2021 una media de 1605 ataques por organización cada semana, suponiendo el crecimiento del 75 % respecto al 2020. Siguiendo el área gubernamental/militar, que tuvo 1.136 asaltos por semana (47 % de subida), y la industria de las comunicaciones, con 1.079 por organización (51 % de incremento) [1].

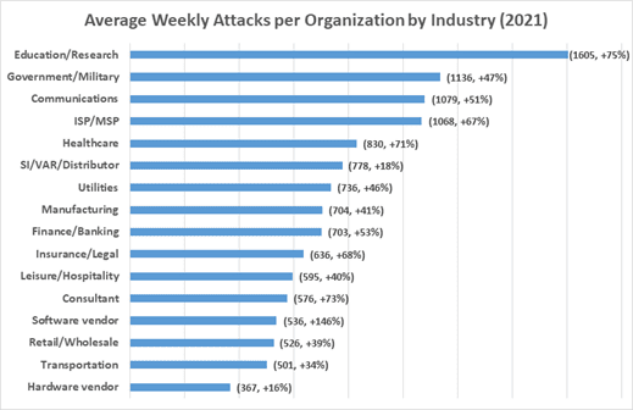


Figura 2.11: Promedio de ataques semanales por organización e industria [1].

De igual forma, en la figura siguiente se representa el aumento registrado durante cada trimestre del año 2021.

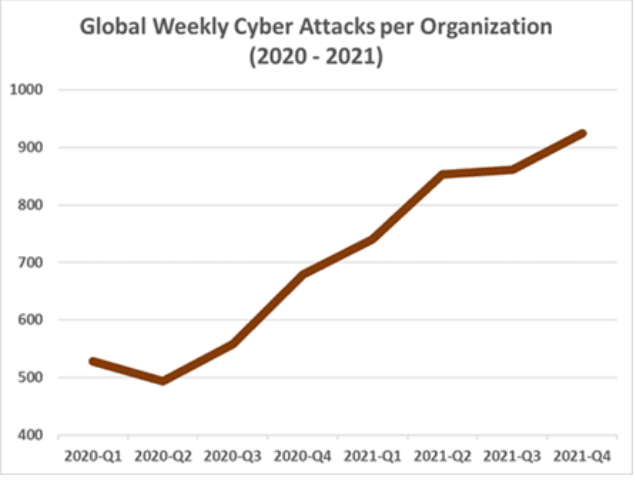


Figura 2.12: Aumento de ciberataques semanales globales por organización [1].

En un reciente reporte de Check Point Research sobre las tendencias de ciberataques en el sector de la educación y la investigación ha sufrido el mayor número de ciberataques en la primera mitad de 2023, con una diferencia asombrosa con respecto a otros sectores como se recalca en la siguiente figura.

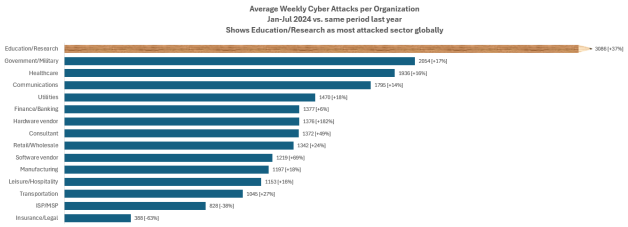


Figura 2.13: Promedio de ataques cibernéticos semanales por organización [2].

Cada semana, el número medio de ciberataques es de 2.256 por organización de Educación/Investigación a nivel mundial. En concreto, en Europa las amenazas a este sector se han visto incrementadas en un 11 % con respecto a 2022. [28]. Y en el segundo trimestre de 2024, se destaca que aumento aun mas este año, reportando un promedio de 3086 ataques por organización en cada semana [29].

2.2.1. Situación en México

México actualmente no se encuentra en una situación segura para prevenir ataques de los cibercriminales, tan solo en Latinoamérica el sector tuvo una media de 2721 ataques a la semana por cada institución y 3507 ataques en México lo que esta representando en el año 2024 un aumento del 22 % [30]. México actualmente se encuentra en el top 4 en promedio de ataques por organización como se muestra en la siguiente tabla.

País	Promedio de ataques por organización	Cambio interanual
India	6874	+97 %
Reino Unido	4793	+36 %
Italia	4730	+40 %
México	3507	+22 %
Portugal	3042	+66 %
Alemania	2041	+77 %
Estados Unidos	1667	+38 %

Tabla 2.9: Ataques promedio a la semana en el sector educativo [2].

Este incremento demuestra la necesidad crítica de medidas de ciberseguridad sólidas y una mayor concienciación entre organizaciones y usuarios con limitada preparación digital. De acuerdo con Miguel Hernández, director de

Check Point México, la situación de vulnerabilidad en el país se refleja en que las organizaciones mexicanas reciben en promedio 3,048 ataques por semana, una cifra considerablemente superior al promedio global de 1,891 ataques [31].

2.2.2. Ataques con más frecuencia

Entre los ataques más frecuentes en sectores con baja alfabetización tecnológica, pequeñas, medianas empresas y como ya lo vimos en sectores educativos y de gobierno, se destaca el phishing, ransomware, ataques DoS (Denegación de Servicio), ataques DDoS (Denegación de Servicio Distribuido) y el compromiso de correos electrónicos. Estos comparten el objetivo de comprometer la seguridad de los sistemas para obtener beneficios a costa de las vulnerabilidades humanas y tecnológicas [32].

La alta frecuencia de estos ataques se debe a factores como:

- Falta de capacitación en el tema.
- Limitada implementación de medidas de seguridad.
- Gran cantidad de información confidencial que se gestiona.
- Dependencia creciente de plataformas digitales.

En la siguiente tabla se describen los ataques mencionados junto con la frecuencia que se han realizado.

Tipo de ataque	Descripción	Frecuencia
Ransomware	Secuestra sistemas y solicita un rescate para liberarlos.	Representó 3 de cada 5 ataques registrados en 2019 y en el 2023 el 80 % de las instituciones fueron víctimas y esperando un aumento más del 72 % [33].
Phishing	Ataques por correo o sitios falsos que buscan obtener información confidencial.	México registró 285,400 intentos de ataques de ransomware entre junio 2023 y julio 2024, con promedio de 781 diarios y un aumento del 165 % respecto al año anterior [34].
Ataques DoS y DDoS	Sobrecarga individual y en conjunto de sistemas para interrumpir servicios en línea.	Amenaza común en instituciones educativas, frecuentemente dirigida contra sistemas críticos, en el último año han aumentado un 84 % registrando más de 3000 ataques semanales [2].
Compromiso de correo (BEC)	Suplantación de identidad empresarial por correo para acceder a sistemas o fondos.	Estudio de brechas indica que más del 50 % de incidentes incluyen factor humano, muchos a través de suplantación de correo empresarial dentro del patrón de ingeniería social [35].

Tabla 2.10: Comparación de estadísticas de ataques.

Ransomware

El ransomware es un tipo de malware que cifra los datos de un sistema, bloqueando el acceso a ellos. Los atacantes exigen un rescate a las víctimas, usualmente en criptomonedas, para proporcionar la clave de descifrado, en la mayoría de las ocasiones no liberan los datos como lo prometen. Existen diferentes variantes, cada uno con características especificas:

- Crypto-ransomware: Cifrar archivos y exig 8 e un pago para desbloquearlos.
- Locker-ransomware: Bloquea el acceso al sistema operativo sin cifrar archivos.
- Scareware: Falsas alertas de virus que engañan a los usuarios para que paguen.
- Doxware: Amenazan con filtrar información confidencial si no se pagan el rescate.

El caso del ataque al Grupo Bimbo por parte del grupo Medusa en 2024. Los atacantes cifraron archivos y exfiltraron bases de datos, exigiendo un rescate de 6.5 millones de dólares; publicaron datos financieros, correos y documentos internos como prueba [36].

El método de propagación de este ataque no se hizo público pero los informes técnicos señalan:

1. El grupo Medusa suele acceder a las redes mediante credenciales comprometidas, aprovechando vulnerabilidades no parcheadas o también utilizan ingenieria social.
2. Dentro de la red, deplegaron su ransomware que cifra archivos y añade la extensión '.MEDUSA'.
3. Por ultimo amenazan con filtrar la información si el rescate no es pagado, aplicando la técnica de doble extorsión.

Respecto a la contención del ataque, Grupo Bimbo respondió activando sus protocolos de ciberseguridad, que incluyeron aislamiento de sistemas, investigación forense digital y restauración a partir de respaldos seguros.

Phishing

El phishing es una técnica de ciberataque en la que los atacantes se hacen pasar por entidades legítimas, como bancos, empresas o personas conocidas, para engañar a sus víctimas y obtener información confidencial, como contraseñas o datos financieros. Existen diferentes tipos de phishing, entre los más comunes se encuentran:

- Spear Phishing: Son ataques dirigidos a individuos específicos, generalmente dentro de empresas o instituciones, utilizando información personalizada para hacer mas creíble el engaño.
- Smishing: Se envían mensajes de texto con enlaces maliciosos o solicitudes falsas para que la víctima comparta información.
- Pharming: Los atacantes manipulan el DNS para redirigir a los usuarios a sitios web falsos sin que lo noten y al ingresar sus credenciales en el sitio falso, la información es robada.
- Vishing: Llamadas telefónicas en las que los atacantes se hacen pasar por empleados de empresas legítimas para obtener datos sensibles.

Entre los años 2013 y 2015, Google y Facebook fueron víctimas de un ataque de phishing altamente sofisticado, en el que un ciberdelincuente lituano, Evaldas Rimasauskas, logró estafar a ambas compañías por un monto superior a $100 millones de dólares. El ciberdelincuente orquestó el fraude haciéndose pasar por Quanta Computer, un proveedor real de hardware con sede en Taiwán que trabajaba con ambas empresas y Mediante correos electrónicos fraudulentos, logró engañar a empleados de Google y Facebook para que realizaran transferencias millonarias a cuentas bancarias controladas por él [37].

El método de este ataque fue el siguiente:

1. Creación de una empresa falsa. Rimasauskas registró una empresa en Letonia con el mismo nombre que Quanta Computer, imitando la identidad del proveedor legítimo.
2. Envió de correos electrónicos fraudulentos. Utilizó spear phishing para enviar facturas falsas a empleados de Google y Facebook. Los correos parecían legítimos, con logos, firmas y terminología profesional similares a las del proveedor real.
3. Manipulación de pagos. Los empleados, creyendo que se trataba de pagos legítimos, autorizaron transferencias millonarias a cuentas bancarias en Letonia, Chipre, Eslovaquia, Lituania y Hong Kong. El dinero era transferido inmediatamente a otras cuentas para dificultar su rastreo.

Este ataque fue mitigado después de que mas de 100 millones de dólares fueron desviados a cuentas fraudulentas, ya que mediante una investigación Rimasauskas fue arrestado en Lituania y extraditado a Estados Unidos, en 2019 se declaro culpable de fraude electrónico, lavado de dinero y robo de identidad, siendo condenado a solo 5 años de prisión. En cuanto al dinero Facebook y Google trabajaron con las autoridades para rastrear las transacciones logrando recuperar parte del dinero robado.

DoS y DDoS

Un ataque de Denegación de Servicio (DoS) consiste en saturar un servidor, red y aplicación con tráfico malicioso o solicitudes que consumen recursos, interrumpiendo su funcionamiento y haciendo inaccesibles los servicios para los usuarios legítimos. Estos ataques se originan desde una única fuente, lo que facilita su identificación y mitigación.

A diferencia de los ataques de Denegación de Servicio Distribuido (DDoS), los cuales suelen aprovechar una botnet o una red de dispositivos comprometidos y controlados por un atacante para generar tráfico excesivo desde múltiples ubicaciones, dificultando así su detección y mitigación.

Estos ataques explotan vulnerabilidades en los sistemas o simplemente agotan recursos como el ancho de banda, la memoria o la capacidad de procesamiento. Además, existen variantes de estos ataques como:

- **Ataques Volumétricos**: Buscan consumir el ancho de banda disponible, sobrecargando la red con tráfico masivo.
  - **UDP Flood**: Envía una gran cantidad de paquetes UDP a puertos aleatorios del servidor, agotando sus recursos.
  - **ICMP Flood (Ping Flood)**: Inunda el sistema con solicitudes ICMP (ping), saturando la capacidad de respuesta.
  - **DNS Amplification**: Usa servidores DNS abiertos para amplificar el tráfico malicioso dirigido a la víctima, generando un impacto mayor con menos recursos.
- **Ataques de Protocolo**: Apuntan a vulnerabilidades en la infraestructura de red, afectando el manejo de conexiones.
  - **SYN Flood**: Envía solicitudes SYN masivas sin completar la conexión, agotando la tabla de conexiones del servidor.
  - **ACK Flood**: Sobrecarga la tabla de estado de las conexiones TCP enviando paquetes ACK sin una secuencia válida.

    - **RST Flood**: Envía paquetes TCP con la bandera RST activada, forzando el cierre de conexiones activas.

- **Ataques a la Capa de Aplicación**: Buscan interrumpir servicios específicos explotando las solicitudes legítimas del usuario.
    - **HTTP Flood**: Simula múltiples solicitudes legítimas a un servidor web, consumiendo sus recursos.
    - **Slowloris**: Mantiene abiertas muchas conexiones HTTP sin completarlas, bloqueando nuevas solicitudes.
    - **DNS Query Flood**: Envía un número excesivo de consultas DNS para sobrecargar el servidor y degradar su rendimiento.

Un caso para este ataque fue para el medio digital Revista Espejo, con sede en Culiacán, Sinaloa, fue blanco de múltiples ataques de denegación de servicio distribuidos (DDoS), particularmente agresivos en marzo de 2021. Estos ataques afectaron gravemente su capacidad de operación en línea y visibilizaron una tendencia preocupante en México hacia la censura digital mediante ciberataques [38].

El método utilizado fue el siguiente:

1. Adquisición de una red distribuida de dispositivos comprometidos localizados en diferentes países.

2. Utilizando técnicas automatizadas para saturar los recursos de un servidor enviaron al rededor de 136 millones de solicitudes.

La revista trabajó de la mano con sus proveedores de hosting para implementar medidas de mitigación como el filtrado de IPs maliciosas, uso de redes de distribución de contenido (CDN) y configuraciones de defensa en capa de aplicación. También migraron su infraestructura a servicios con mayor tolerancia a tráfico anómalo y reforzaron su seguridad en la capa DNS.

La siguiente tabla resume los cuatro tipos de ataques, indicando su método de propagación, forma de ataque y objetivos principales.

Tipo de ataque	Propagación	Ataque	Objetivo
Ransomware	Correos maliciosos, enlaces fraudulentos o vulnerabilidades.	Cifrado: Bloquea archivos con algoritmos avanzados. Rescate: Exige pago en criptomonedas.	Pérdida de datos, interrupción de servicios y altos costos.
Phishing	Crea correos o sitios web falsos imitando entidades confiables.	Engaño: Envía enlaces o archivos maliciosos con pretextos urgentes. Recopilación: La víctima ingresa datos en formularios falsos o ejecuta malware.	Los atacantes acceden a cuentas, roban dinero o realizan otros ataques.
Ataques DoS	Crean una botnet mediante malware.	Ataque: Envían solicitudes masivas para saturar al objetivo. Interrupción: Sobrecargan el servidor o red, impidiendo el acceso legítimo.	Datos financieros, hacktivismo, competencia desleal o malicia.
Ataques DDoS	Crean una botnet mediante malware.	Ataque: Envían solicitudes masivas para saturar al objetivo. Interrupción: Sobrecargan el servidor o red, impidiendo el acceso legítimo.	Datos financieros, hacktivismo, competencia desleal o malicia.

Tabla 2.11: Comparación de características de los ciberataques.

2.3. Estado del arte

Los ciberataques representan una amenaza constante en el mundo actual, afectando tanto a individuos como a organizaciones. La ciberseguridad se ha convertido en un tema crucial y de amplia investigación, con el objetivo de desarrollar métodos avanzados para la detección de amenazas. En artículos existentes, el aprendizaje automático (Machine Learning) ha emergido como una herramienta poderosa para la detección de ciberataques, empleando diversos algoritmos y técnicas que analizan datos en busca de anomalías que puedan indicar la presencia de ataques. Cada artículo examina diferentes enfoques y áreas tecnológicas, proporcionando una visión amplia y variada sobre cómo enfrentarlos. A continuación, se presentan algunas de las investigaciones más relevantes y sus hallazgos en el campo de la detección de ciberataques. [39] Francisco. J. Fernández Rosique, en su artículo Estudio experimental de ciberataques a través de códigos QR, Universidad Politécnica de Cartagena, 2022. Señala que el objetivo es analizar de forma experimental qué tipos de ciberataques podrían realizarse utilizando como arma los códigos QR. Abordando tanto el proceso de creación como el de lectura. Se analizaron las posibles vulnerabilidades asociadas a la respuesta inmediata que caracteriza a los códigos QR y se demostró que el uso malintencionado de los códigos QR podría representar una alta amenaza, especialmente debido a su imposible legibilidad para el usuario antes de ser escaneado. Sin embargo, varias de las vulnerabilidades identificadas requerían acciones de confirmación por parte del usuario, como escanear el código en una aplicación específica, aceptar la conexión a un punto de acceso o realizar una llamada intencionada a un número marcado. Enfocándonos en otra área como el Internet de las Cosas (IoT) [40] Alsamiri, J. M., & Alsubhi, K, con su artículo Internet of Things Cyber Attacks Detection using Machine Learning, King Abdulaziz University, 2019. Se centraron en la detección de ciberataques en estas redes, utilizando métodos de aprendizaje

automático con el conjunto de datos Bot-IoT para evaluar diferentes algoritmos de detección. Empleando CICFlowMeter (herramienta para la captura y el análisis de flujos de red) para extraer características basadas en flujos a partir de trazas de tráfico en bruto, generando 84 características de tráfico de red del conjunto de datos. Los algoritmos aplicados, como Naive Bayes con 0.77, QDA con 0.86, Random Forest con 0.97, ID3 con 0.97, AdaBoost con 0.97, MLP con 0.83 y K Nearest Neighbours con 0.99, destacaron con su efectividad, pero especialmente Random Forest, ID3 y AdaBoost, que obtuvieron altas tasas de rendimiento. En el mismo enfoque, pero en el área de la medicina [3] Saheed, Y. K., & Arowolo, M. O. con el artículo Efficient cyber-attack detection on the internet of medical things-smart environment based on deep recurrent neural network and machine learning algorithms, 2021. Desarrollaron una detección eficiente de ciberataques en el Internet de las Cosas Médicas (IoMT) utilizando una arquitectura de entorno IoMT-Smart como se muestra en la siguiente Figura 2.14.

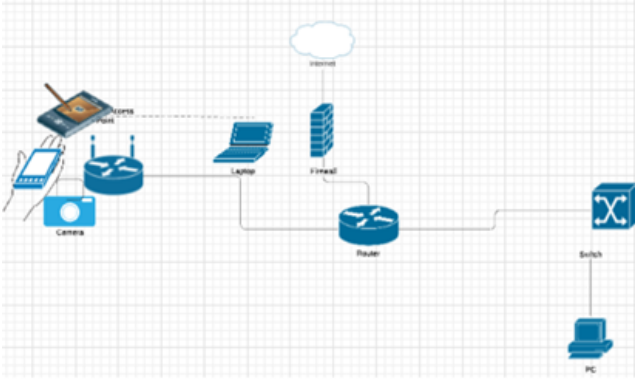


Figura 2.14: Arquitectura del entorno IoMT-Smart propuesta [3].

Además de una Red Neuronal Recurrente Profunda (DRNN) y varios modelos de aprendizaje supervisado (SML) como Random Forest optimizado con PSO (PSO-RF) con 99.76, Árbol de Decisión optimizado con PSO (PSO-DT) con 99.58, K-Nearest Neighbors (KNN) optimizado con PSO (PSO-KNN) con 98.90 y Regresión Logística optimizada con PSO (PSO-RC) con 97.61 de precisión y con el modelo PSO-RF se mejoró la precisión de detección, alcanzando una precisión del 99.76 %. Este enfoque no solo mejora la precisión de la detección, sino que también aborda la eficiencia computacional requerida para aplicaciones en tiempo real en entornos IoT. El IoT se extiende a redes alámbricas como inalámbricas, en el siguiente caso de [4] Khaista Rahman, Muhammad

Adnan Aziz, Nighat Usman, Tayybah Kiren, Tanweer A. Cheema, Hina Shoukat, Tarandeep K. Bhatia, Asrin Abdollahi y Ahthasham Sajid, en su artículo Cognitive Lightweight Logistic Regression-Based IDS for IoT-Enabled FANET to Detect Cyberattacks, 2023. Investigaron la integración de redes inalámbricas 5G con redes FANET (Flying Ad-Hoc Networks) como un nuevo concepto para mejorar la cobertura y reducir el retardo en la comunicación esto especialmente UAVs (vehículos aéreos no tripulados). Destacó la importancia de asegurar las FANETs contra ciberataques, que pueden interrumpir la conectividad entre los nodos y comprometer la comunicación, especialmente ante amenazas como ataques de datos falsos y DoS/DDoS. Se propuso un modelo de sistema que incluye un enfoque cognitivo ligero basado en regresión logística para la detección dinámica de ataques, utilizando algoritmos de aprendizaje automático como Decision Trees (DT) con 49.17 %, Random Forest (RF) con 71.59 %, XGBoost con 49.54 %, AdaBoost con 28.39 %, Bagging con 44.70 % y regresión logística con la precisión más alta de 82.54 %.

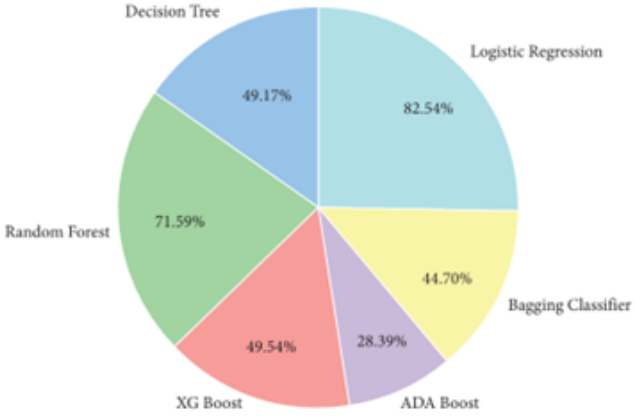


Figura 2.15: Análisis del rendimiento de clasificadores de aprendizaje automático (DT, RF, XGBoost, AdaBoost, Bagging y regresión logística) [4].

Esta investigación se basó en el conjunto de datos UNSW-NB15 para entrenar y probar los clasificadores de aprendizaje automático. Ahora con el modelo de aprendizaje profundo de Multilayer Perceptron (DMLP) [41] Panda, M., Abd Allah, A. M., & Hassanien, A. E., con el artículo Developing an efficient feature engineering and machine learning model for detecting IoT-botnet cyber-attacks, 2021. Investigaron la detección eficiente de ataques cibernéticos en sistemas de Internet de las Cosas (IoT), especialmente a través de la detección y mitigación de botnets. Utilizando el conjunto de datos UNSW-NB15

que fue diseñado específicamente para IoT y botnets, que es ruidoso y desbalanceado. Las técnicas que se aplicaron para obtener un conjunto de datos representativo con subconjuntos óptimos de características fueron K-Medoid y búsqueda de dispersión (scatter search). Y para la detección se utilizaron los algoritmos de aprendizaje automático de JChaid, A2DE y HGC más dos métodos de aprendizaje profundo como DMLP y CNN. Los resultados del análisis experimental mostraron que el clasificador DMLP basado en búsqueda de dispersión superó a los otros modelos en términos de precisión y eficiencia computacional. Logró una tasa de detección del 100 % con precisión, recall y F1-score macro-averaged del 100 %, junto con tiempos de entrenamiento y prueba muy bajos. [5] Delplace, A., Hermoso, S., & Anandita, K., Cyber-attack detection thanks to machine learning algorithms, Universidad Politécnica de Cartagena, 2020. También se centraron en la detección y clasificación de tráfico malicioso en redes con solo el énfasis en la detección de botnets. Realizando un análisis exhaustivo de datos a partir de conjuntos de datos NetFlow, resulto en la extracción de 22 características principales. Estas características fueron sometidas a un proceso de selección para comparar su eficacia y después evaluar cinco algoritmos diferentes de aprendizaje automático siendo Random Forest logrando detectar más del 95 % de los botnets en 8 de 13 escenarios y más del 55 % en los conjuntos de datos más difíciles.

Botnet	Dataset		Training			Test		
	Size	Botnets	Precision	Recall	f_1 Score	Precision	Recall	f_1 Score
Neris - Scenario 1	2 226 720	1.28 ‰	1.0	1.0	1.0	1.0	0.95	0.975
Neris - Scenario 2	1 431 539	1.45 ‰	1.0	1.0	1.0	1.0	0.98	0.99
Rbot - Scenario 3	2 024 053	4.99 ‰	1.0	0.99	0.99	1.0	0.96	0.98
Rbot - Scenario 4	470 663	2.36 ‰	1.0	0.90	0.95	1.0	0.69	0.82
Virut - Scenario 5	63 643	3.46 ‰	1.0	1.0	1.0	1.0	0.25	0.4
DonBot - Scenario 6	220 863	5.57 ‰	1.0	1.0	1.0	1.0	0.9	0.95
Sogou - Scenario 7	50 629	1.38 ‰	1.0	1.0	1.0	1.0	0.25	0.4
Murlo - Scenario 8	1 643 574	6.80 ‰	1.0	1.0	1.0	1.0	0.94	0.97
Neris - Scenario 9	1 168 424	12.51 ‰	1.0	0.99	1.0	1.0	0.94	0.97
Rbot - Scenario 10	559 194	9.67 ‰	1.0	0.98	0.99	1.0	0.90	0.95
Rbot - Scenario 11	61 551	2.76 ‰	1.0	1.0	1.0	0.5	0.33	0.4
NSIS.ay - Scenario 12	156 790	10.20 ‰	1.0	0.82	0.90	0.92	0.41	0.56
Virut - Scenario 13	1 294 025	7.57 ‰	1.0	1.0	1.0	1.0	0.96	0.98

Figura 2.16: Resultados resumidos usando Radom Forest Classifier [5].

Si cambiamos de enfoque [42] Ajmal, M. A., Imran, M., Raza, M. A., & Raza, A., en el artículo Cyber Threats Prediction Model using Advanced Data Science Approaches. 2022. Realizaron un modelo de predicción basado en Data Science y Machine Learning para detectar y predecir ataques de tipo DDoS, donde utilizó el conjunto de datos CICDDOS2019 y se aplicaron diferentes modelos de aprendizaje automático, como Decision Tree, Random Forest, SVM y

Naïve Bayes. Se concluyó que un modelo basado en Data Science y Machine Learning es más apropiado y exitoso en ciberseguridad, especialmente en la predicción de ciberataques, en comparación con enfoques tradicionales. Random Forest fue el algoritmo más preciso, con una precisión del 99,87 %, seguido del árbol de decisión con un 99,84 %, el SVM con un 66,04 % y el Naive Bayes con un 87,70 %. Los ciber ataques también existen en el área de las redes eléctricas inteligentes o mejor conocidas como Smart Grids, estas son sistemas de distribución eléctrica modernizados que integran tecnología digital avanzada para mejorar la eficiencia, la confiabilidad y la seguridad de las redes eléctricas tradicionales. Estas redes utilizan tecnología de comunicación bidireccional entre consumidores, generadores y operadores de la red para optimizar la gestión del suministro eléctrico. [43] Karimipour, H., Dehghantanha, A., Parizi, R. M., Choo, K. K. R., & Leung, H., con el artículo A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids, 2019. Investigaron el desarrollo de un sistema de detección de anomalías no supervisado, basado en la correlación estadística entre mediciones. El objetivo fue diseñar un motor de detección de anomalías escalable adecuado para redes inteligentes a gran escala, capaz de diferenciar entre fallas reales, perturbaciones y ciberataques inteligentes. Se aplicó la extracción de características utilizando el filtrado dinámico simbólico (SDF) para reducir la carga computacional, y se realizaron simulaciones en sistemas IEEE de 39, 118 y 2848 buses, logrando una precisión del 99 % con una tasa de verdaderos positivos del 98 % y una tasa de falsos positivos de menos del 2 %. Ahora si comparamos con otro artículo sobre investigación de Smart Grids [6] Almalaq, A., Albadran, S., & Mohamed, M. A., en su artículo Deep machine learning model-based cyber-attacks detection in smart power systems, 2022. Propusieron el uso de PCA para la selección de características, mejorando significativamente la detección de ciberataques en redes eléctricas inteligentes con una precisión del 93.87 % y también se probaron algoritmos como KNN 91.34 %, SVM con 90.02 %, GBDT no especifica, XGBoost no especifica y CNN no especifica la precisión. Los datos utilizados contienen 128 características registradas utilizando PMUs (Unidades de Medición de Fasores) y alarmas de relés y registros Snort.

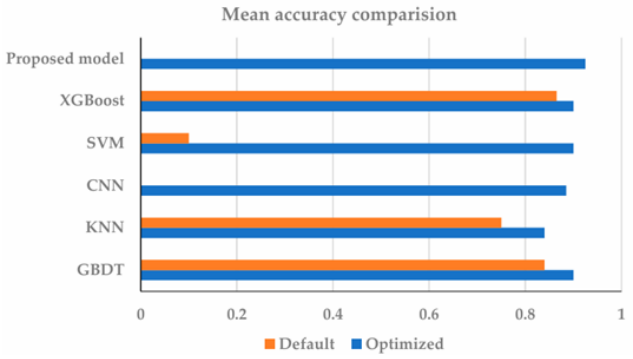


Figura 2.17: Comparación de precisión [6].

Para finalizar también las redes informáticas frecuentemente son otra puerta para los ciber ataques y [44] Ozalp, A. N., & Albayrak, Z., con su artículo Detecting cyber attacks with high-frequency features using machine learning algorithms, 2022. Investigaron la detección de ciberataques con características de alta frecuencia utilizando algoritmos de aprendizaje automático, se utilizaron características del conjunto de datos NSL-KDD para determinar las frecuencias y evaluar la efectividad en la detección de ciberataques mediante algoritmos como Random Forest, J48, Naive Bayes y Multi-Layer Perceptron (MLP). Random Forest proporcionó un alto rendimiento en términos de clasificación y precisión, con un 99.76 %.

Algoritmo	Áreas tecnológicas	Precisión
Random Forest	Redes IoT, Redes Informáticas, IoMT y Smart Grids	99.76 % en IoMT 99.87 % en DDoS 97 % en IoT 99 % en Smart Grids
ID3 y AdaBoost	Redes IoT	97 %
AdaBoost	Redes IoT y FANETs	28.39 % en FANETs 97 % en IoT
XGBoost	FANETs	49.54 % en FANETs
SVM (Support Vector Machine)	DDoS y Smart Grids	66.04 % en DDoS 90.02 % en Smart Grids
Naive Bayes	Redes IoT y DDoS	87.70 % en DDoS 0.77 % en IoT
K-Nearest Neighbors (KNN)	IoMT y Smart Grids	99 % en IoT 98.90 % en IoMT 91.34 % en Smart Grids
Decision Trees	FANETs y DDoS	49.17 % en FANETs 99.84 % en DDoS
Multilayer Perceptron (MLP)	Redes Informáticas y IoMT	99.76 % en IoT 83 % en IoT
Deep Recurrent Neural Network (DRNN)	IoMT	99.76 % en IoMT

Tabla 2.12: Análisis comparativo de algoritmos de aprendizaje automático, resaltando su precisión y rendimiento en diversos ámbitos tecnológicos.

En esta tabla se muestra un panorama más detallado de las investigaciones, destacando las aplicaciones de diferentes algoritmos de aprendizaje automático en diversos contextos tecnológicos. Cada enfoque tiene sus fortalezas y debilidades. Los algoritmos como Random Forest y KNN demostraron alta precisión en varios escenarios, mientras que otros como SVM y AdaBoost muestran un rendimiento variable dependiendo del área tecnológica.

Capítulo 3

Análisis de ciberataques

Una vez comprendido algunos de los algoritmos de aprendizaje automático mas conocidos y utilizados para la predicción e identificación es importante abordar la etapa de selección, análisis y preparación de datos. Este paso es necesario para garantizar que los algoritmos seleccionados en los pasos posteriores puedan operar de manera óptima, aprovechando al máximo las características relevantes y reduciendo el impacto de los datos inconsistentes o irrelevantes.

3.1. Fuentes de datos

Para la recopilación de los conjuntos de datos que se analizaran pueden ser utilizados diversos repositorios que se encuentran en internet como:

- Repositorios públicos: Son conjuntos accesibles a través de plataformas como Kaggle, Microsoft Research o conjuntos de datos de Google.
- Repositorios privados: Son conjuntos accesibles a través de plataformas de instituciones privadas como lo son las gubernamentales o académicas a las cuales se tienen acceso pagando o solicitando de manera personal.
- Bases de datos propias: Trata de información recolectada específicamente para un estudio mediante encuestas, experimentos o herramientas de captura de datos.
- Simulaciones: Son datos generados artificialmente para replicar escenarios específicos que permitan probar los algoritmos recabando los resultados en tiempo real.

Las fuentes elegidas para la recopilación de datos fueron repositorios públicos, destacando a Kaggle como la principal fuente, está siendo una plataforma en línea especialmente utilizada por científicos de datos y personas interesadas sobre el aprendizaje automático. A continuación, se muestra una captura de su página principal.

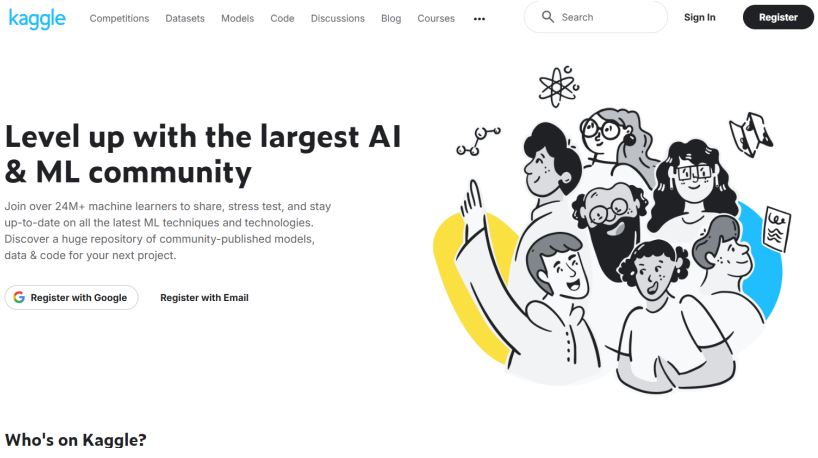


Figura 3.1: Kaggle [7].

La segunda fuente pública fue la Universidad de Nuevo Brunswick (UNB), específicamente a través de su Canadian Institute for Cybersecurity (CIC). Este instituto es reconocido internacionalmente por proveer conjuntos de datos de ciberseguridad diseñados para evaluar y desarrollar sistemas de detección de intrusos y estrategias de mitigación ante ciberataques. A continuación, se muestra una captura de su página donde se encuentra el banco de datos.

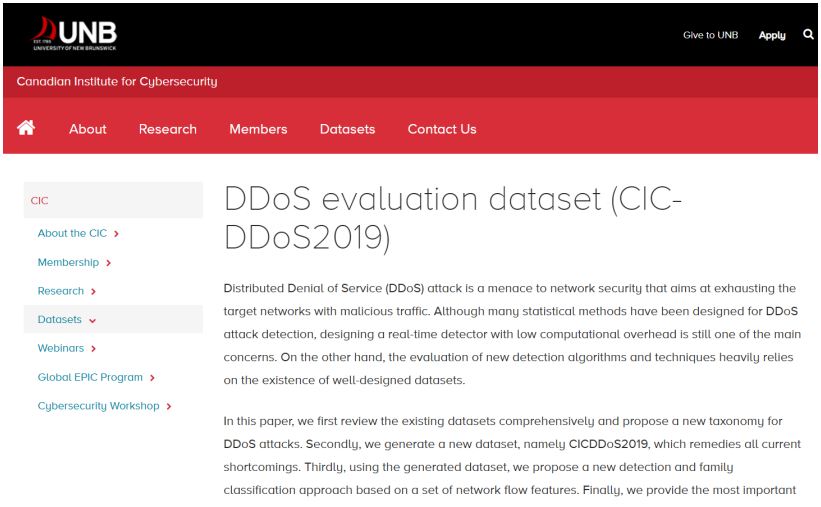


Figura 3.2: Universidad de Nuevo Brunswick [8].

Se limito la selección a tres conjuntos de datos para analizar tres diferentes tipos de ataques cibernéticos: phishing, ransomware y ataques de denegación de servicio (DoS y DDoS). Los conjuntos de datos relacionados con phishing y ransomware fueron obtenidos de Kaggle, mientras que el dataset correspondiente a DoS y DDoS proviene de la Universidad de Nuevo Brunswick (UNB). La combinación de estas fuentes permite un análisis amplio y representativo de amenazas cibernéticas contemporáneas, considerando tanto plataformas públicas de ciencia de datos como repositorios académicos especializados.

3.1.1. Conjunto de datos para Phishing

El primer conjunto de datos, titulado 'Phishing Dataset for Machine Learning' (Conjunto de datos de phishing para aprendizaje automático), fue publicado en el año 2021 por el usuario Shashwat Tiwari. Contiene 48 características y un total de 10,000 registros, divididos equitativamente entre 5,000 sitios de phishing y 5,000 sitios legítimos [45]. Los datos fueron recolectados entre los años 2015 y 2020, proporcionando una base sólida y representativa para el análisis y detección de ataques de phishing. Hasta la fecha, se han registrado más de 130,000 visualizaciones y 16,300 descargas, lo que refleja su popularidad y utilidad en la comunidad de ciberseguridad y ciencia de datos. Además, este conjunto de datos se encuentra disponible bajo la licencia Attribution 4.0 International (CC BY 4.0), permitiendo su uso y adaptación. A continuación,

se muestra una captura de la descripción del banco de datos en la pagina de Kaggle.

Figura 3.3: Conjunto de datos Phishing Dataset for Machine Learning.

3.1.2. Conjunto de datos para Ransomware

El segundo conjunto de datos, esta titulado 'Android Ransomware Detection' (Detección de Ransomware en el sitema operativo Android), fue publicado en el año 2023 por el usuario Cyber Cop. Este dataset contiene 85 caracteristicas y 203,556 registros, incluye diez tipos de ransomware de Android y de tráfico benigno. Los tipos de ransomware incluidos son: SVpeng, PornDroid, Koler, RansomBO, Charger, Simplocker, WannaLocker, Jisut, Lockerpin y Pletor [46]. El dataset se encuentra bajo la licencia GNU Affero General Public License 3.0, permitiendo su uso y modificación. Además, la fuente principal de los datos es el Canadian Institute for Cybersecurity (CIC), reconocido por su trabajo en ciberseguridad y recopilación de datos para la investigación académica. Hasta la fecha, el dataset ha registrado más de 7,087 visualizaciones y 1,144 descargas. Este conjunto de datos resulta valioso para entrenar y evaluar modelos de aprendizaje automático orientados a la identificación de ransomware, así como para estudiar el comportamiento y las características del tráfico malicioso en dispositivos Android. A continuación, se muestra una captura de la descripción del banco de datos en la página de kaggle.

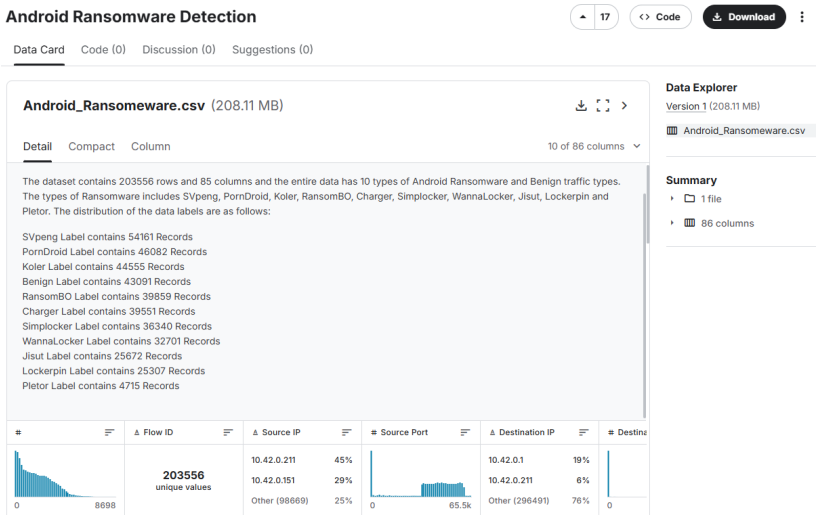


Figura 3.4: Conjunto de datos Android Ransomware Detection.

3.1.3. Conjunto de datos para DoS y DDoS

El ultimo conjunto de datos, esta titulado 'DDoS evaluation dataset (CIC-DDoS2019)' (Conjunto de datos de evaluación de DDoS (CIC-DDoS2019)), fue publicado en el año 2021 por un equipo de investigadores, incluyendo al Dr. Iman Sharafaldin, el Dr. Saqib Hakak y el Dr. Arash Habibi Lashkari. Este dataset contiene 80 caracteristicas y aproximadamente 50,063,112 registros en su versión original, incluye once tipos de ataques y de tráfico benigno. Los tipos de ataques de denegación de servicios incluidos son: DrDoS-UDP, UDP-lag, DrDoS-MSSQL, DrDoS-NetBIOS, Syn y WenDDoS [47]. El conjunto de datos se encuentra bajo la licencia CC BY-NC-SA 4.0, permitiendo su uso para investigación académica no comercial. Hasta la fecha, el dataset ha registrado más de 33,700 visualizaciones y 7,105 descargas en el repositorio de Kaggle, demostrando su relevancia en la investigación y desarrollo de modelos de detección de ataques cibernéticos complejos. A continuación, se muestra una captura de la descripción del banco de datos en la página de kaggle.

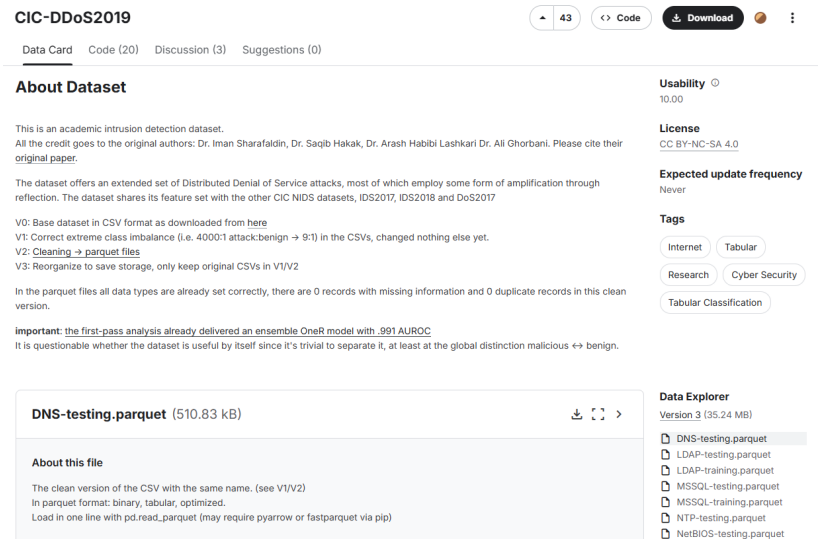


Figura 3.5: Conjunto de datos DDoS evaluation dataset (CIC-DDoS2019).

3.2. Preparación de datos

La preparación de datos es un paso fundamental en el análisis y modelado, ya que garantiza la calidad, consistencia y relevancia de la información utilizada. Antes de aplicar técnicas de aprendizaje automático y análisis exploratorio, es necesario limpiar, transformar y adaptar los datos para reducir errores y asegurar resultados precisos. En esta sección, se detallan los procesos específicos realizados para cada conjunto de datos (Phishing, Ransomware y DoS/DDoS). Se describen las técnicas de limpieza, transformación, manejo de datos faltantes y selección de características clave para obtener conjuntos de datos adecuados y equilibrados, listos para el análisis y la implementación de modelos predictivos.

3.2.1. Phishing

El conjunto de datos de phishing para aprendizaje automático contiene un total de 50 características y 10,000 registros extraidos de páginas web repartidas equitativamente en trafico de phishing y trafico benigno, sin embargo solo contamos con 48 características ya que una columna es para enumerar los registros y la otra es nuestra columna clave como se visualiza en la siguiente

figura.

	id	NumDots	SubdomainLevel	PathLevel	UrlLength	NumDash	NumDashInHostname	AtSymbol	TildeSymbol	NumUnderscor
0	1	3	1	5	72	0	0	0	0	
1	2	3	1	3	144	0	0	0	0	
2	3	3	1	2	58	0	0	0	0	
3	4	3	1	6	79	1	0	0	0	
4	5	3	0	4	46	0	0	0	0	

5 rows × 50 columns

Figura 3.6: Conjunto de datos phishing.

Como se observa en la siguiente figura, todos los datos son de tipo numérico, excepto la columna 'CLASS-LABEL', la cual es categórica y representa el valor 1 para casos de phishing y 0 para tráfico benigno. Se verificó que no existen valores nulos ni registros duplicados, lo cual asegura la integridad y calidad de la información para el análisis posterior.

```
rowsbefore = len(phiframe)

# Eliminar nulos
phiframe = phiframe.dropna()
rowsaftern = len(phiframe)

# Eliminar duplicados
phiframe = phiframe.drop_duplicates()
rowsafterd = len(phiframe)

# Cantidad de eliminados
ndelete = rowsbefore - rowsaftern
ddelete = rowsaftern - rowsafterd

print(f'Nulos eliminados: {ndelete} | Duplicados eliminados: {ddelete}')
```

```
Nulos eliminados: 0 | Duplicados eliminados: 0
```

Figura 3.7: Eliminar datos nulos y duplicados.

En relación con el balanceo de la columna clave 'CLASS-LABEL', se comprobó en la siguiente figura que los datos están completamente equilibrados. Esto es importante para asegurar una representación equitativa de ambas clases (phishing y tráfico benigno), permitiendo así un entrenamiento adecuado y evitando sesgos en los modelos de aprendizaje automático.

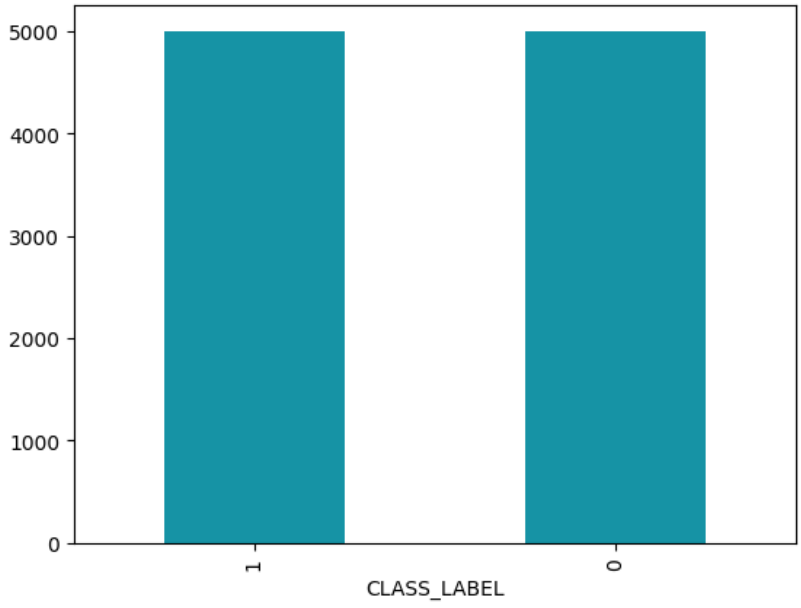


Figura 3.8: Balanceo de la columna 'CLASS-LABEL'.

El proceso de exploración del conjunto de datos no presentó mayores complicaciones. Por lo tanto, procedemos con la preparación de los datos para su aplicación en modelos de aprendizaje automático. Como se visualiza en la Figura 3.9, se creó un DataFrame llamado X, que contiene todas las características excepto las columnas 'CLASS-LABEL' e 'id'. También se generó otro DataFrame llamado y, que únicamente incluye la columna clave 'CLASS-LABEL'. Esta separación permite diferenciar las variables independientes de la variable dependiente, facilitando el entrenamiento y evaluación de los modelos.

```python
X = phiframe.drop(columns=['id', 'CLASS_LABEL']) #Asignar datos a 'X' sin las columnas 'id' (columna que
numera los registros) y 'CLASS_LABEL' (Columna clave)
y = phiframe['CLASS_LABEL'] #Asignar 'CLASS_LABEL' (columna clave) a 'y'
```

Python

Figura 3.9: Preparación de variables independientes y dependientes.

3.2.2. Ransomware

El conjunto de datos de detección de ransomware en Android contiene un total de 86 características y 203,556 registros totales, sin embargo solo contamos con 84 características ya que la columna 'Unnamed: 0' que se visualiza en la Figura 3.10 se utiliza para enumerar los registros y la otra es nuestra columna clave que contiene diez tipos de ransomware diferente de Android, los cuales son: SVpeng, PornDroid, Koler, RansomBO, Charger, Simplocker, WannaLocker, Jisut, Lockerpin y Pletor.

	Unnamed: 0	Flow ID	Source IP	Source Port	Destination IP	Destination Port	Protocol	Timestamp	Flow Duration	Total Fwd Packets	...	min_se
0	0	172.217.2.174-10.42.0.211-443-51023-6	10.42.0.211	51023	172.217.2.174	443	6	16/06/2017 03:55:47	151054	6	...	
1	1	172.217.2.174-10.42.0.211-443-51023-6	10.42.0.211	51023	172.217.2.174	443	6	16/06/2017 03:55:47	349	2	...	
2	2	172.217.12.174-10.42.0.211-443-34259-6	10.42.0.211	34259	172.217.12.174	443	6	16/06/2017 03:55:52	119	2	...	
3	3	172.217.10.74-10.42.0.211-443-55509-6	10.42.0.211	55509	172.217.10.74	443	6	16/06/2017 03:55:53	37055	1	...	
4	4	172.217.2.174-10.42.0.211-443-44852-6	10.42.0.211	44852	172.217.2.174	443	6	16/06/2017 03:55:58	178727	6	...	

5 rows × 86 columns

Figura 3.10: Conjunto de datos ransomware.

La mayoría de los datos son numéricos, aunque también se observa una columna de tipo fecha y datos categóricos. La columna clave 'Label' identifica el nombre de cada tipo de ransomware y también incluye el tráfico benigno. Además, se confirmó que no existen valores nulos ni registros duplicados, lo que asegura la integridad y calidad de la información para el análisis posterior.

La columna clave 'Label' se distribuye en diez tipos de tráfico de ransomware y uno de tráfico benigno. La distribución esta relativamente balanceada de la siguiente manera:

- Ransomware SVpeng: 54,161 registros.
- Ransomware PornDroid: 46,082 registros.
- Ransomware Koler: 44,555 registros.
- Ransomware RansomBO: 39,859 registros
- Ransomware Charger: 39,551 registros.

- Ransomware Simplocker: 36,340 registros.
- Ransomware WannaLocker: 32,701 registros.
- Ransomware Jisut: 25,672 registros.
- Ransomware Lockerpin: 25,307 registros.
- Ransomware Pletor: 4,715 registros.
- Trafico Benigno: 43,091 registros.

En la siguiente figura también se puede observar de manera grafica la distribución de la columna 'Label':

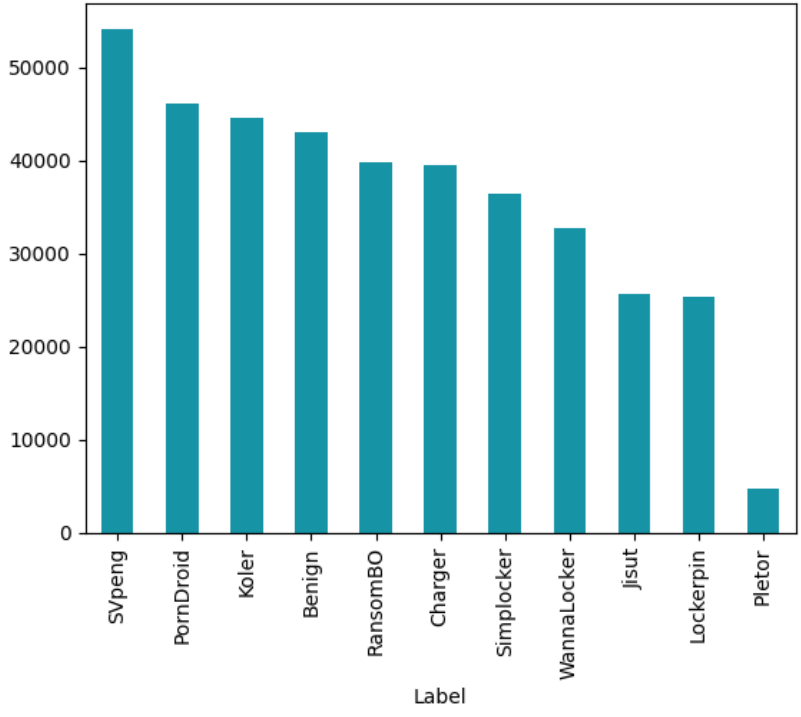


Figura 3.11: Balanceo de la columna 'Label'.

Dado que no todos los tipos de ransomware presentan un riesgo significativo para a los sectores mencionados debido a sus características de ataque, se descartaron los siguientes:

- PornDroid: Se disfraza como una aplicación de contenido para adultos, limitando su impacto a un segmento muy específico del público.
- Koier: Es una variante genérica de ransomware sin un patrón claro de propagación ni una orientación hacia sectores específicos, por lo que su relevancia en este estudio es limitada.
- PansomBO: Aunque tiene capacidad de cifrado, no está diseñado para maximizar daño en contextos como entornos laborales o de usuarios con baja preparación técnica.
- Pletor: Entra en el sector de dispositivos android y aplicaciones que se descargan de tiendas no oficiales, incluso mediante enlaces de sms o correo electrónico, pero debido a su bajo nivel de datos esta descartado.

Se decidió mantener seis tipos de ransomware en el análisis debido a sus características específicas y su posible impacto en entornos como sectores con bajo conocimiento tecnológico:

- SVpeng: Ataca apps bancarias móviles, afectando a usuarios sin protección que realizan transacciones desde su celular.
- Charger: Roba datos personales como contactos y mensajes, comunes en dispositivos sin medidas básicas de seguridad.
- Simplocker: Cifra archivos en Android, afectando documentos importantes en sectores que no hacen respaldos frecuentes.
- WannaLocker: Variante móvil de WannaCry, se propaga en redes mal protegidas como las de PyMEs o usuarios domésticos.
- Jisut: Bloquea el acceso total al dispositivo, afectando a quienes dependen del móvil para trabajo o comunicación.
- Lockerpin: Cambia el PIN del equipo, dejando inutilizable el dispositivo para usuarios sin conocimientos técnicos.

De nuevo observamos como finalmente quedara la distribución de manera grafica de la columna 'Label':

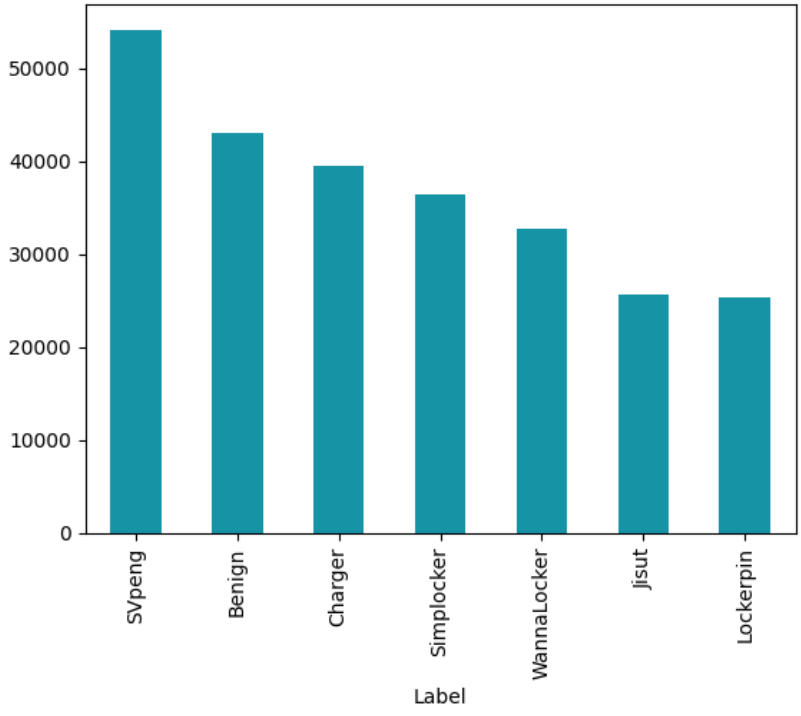


Figura 3.12: Descarte de tipos de ransomware.

Al calcular la media y la desviación estándar de todos los datos y tipos de ransomware, se observa una variabilidad considerable en la distribución. Esto sugiere que las clases no están completamente balanceadas, pero tampoco presentan un desbalance extremo. La desviación estándar representa cerca del 28 % de la media, lo cual se considera aceptable. Un desbalance preocupante ocurriría si la desviación estándar superara el 30 % de la media, ya que indicaría una distribución muy desigual de las clases. En este caso, la media se mantuvo estable como se muestra en la siguiente figura, lo que respalda la validez de los datos para el análisis.

```python
label_counts = ranframe["Label"].value_counts()

# Mostrar las cantidades de cada categoría
print("Cantidad de datos por categoría:")
print(label_counts)

# Calcular la media de las cantidades
mean_count = label_counts.mean()

print(f"\nLa media de los datos por categoría es: {mean_count:.2f}")
```

```
Cantidad de datos por categoría:
Label
SVpeng          54161
Benign          43091
Charger         39551
Simplocker      36340
WannaLocker     32701
Jisut           25672
Lockerpin       25307
Name: count, dtype: int64

La media de los datos por categoría es: 36689.00
```

Figura 3.13: Media y desviación del balance de los datos.

Para la preparación de nuestras variables en la aplicación de modelos de aprendizaje automático, se creó un DataFrame llamado X, que contiene todas las características excepto las siguientes columnas:

- Unnamed: 0: Representa el número de registro y no aporta valor al entrenamiento del modelo.
- Label: Es nuestra columna clave, por lo que se manejará de manera independiente.
- Timestamp: Indica la fecha en que ocurrió el tráfico del registro; sin embargo, no proporciona información útil para el modelo.
- Flow ID: Es un identificador único generado a partir de los datos del registro, por lo que no contribuye al aprendizaje del modelo.

Además, se creó otro DataFrame llamado y, que contiene únicamente la columna 'Label', la cual representa nuestra variable objetivo o dependiente mostrada en la siguiente figura.

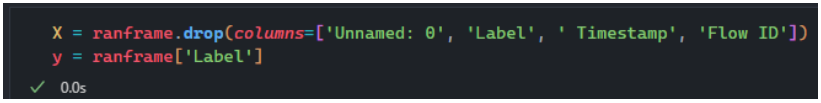


Figura 3.14: Preparación de variables independientes y dependientes.

3.2.3. DoS y DDoS

El conjunto de datos de evaluación de DDoS (CIC-DDoS2019) esta dividido en un total de 17 archivos, cada uno contiene trafico de diferentes tipos de ataques DDoS, específicamente de:

- Syn: Ataque basado en el protocolo TCP que inunda el servidor con solicitudes SYN sin completar la conexión, agotando sus recursos.
- UDP: Envía una gran cantidad de paquetes UDP a la víctima, saturando su ancho de banda y afectando la disponibilidad del servicio.
- MSSQL: Explotación del protocolo Microsoft SQL Server para generar tráfico malicioso y consumir recursos del sistema.
- LDAP: Ataque de amplificación que abusa del protocolo LDAP para enviar grandes volúmenes de tráfico a un objetivo.
- NetBIOS: Uso malintencionado del protocolo NetBIOS para generar tráfico excesivo y afectar la red del sistema objetivo.
- UDPLag: Variante del ataque UDP que introduce retrasos en la comunicación, afectando el rendimiento y disponibilidad del servicio.

Se unieron todos los archivos en dos listas separadas, una para el entrenamiento y otra para la prueba de los modelos de aprendizaje automático. Luego, se compararon ambas listas para conservar únicamente los prefijos comunes, asegurando que los tipos de tráfico en el entrenamiento también estuvieran en la prueba y evitando discrepancias en la distribución de datos.

Al final, como se visualiza en la siguiente figura se obtuvieron seis tipos de tráfico de ataques de denegación de servicio. Los datos de los archivos en cada lista se combinaron para formar dos DataFrames, uno para entrenamiento, con 120,065 registros y 78 características, y otro para prueba, con 38,973 registros y 78 características.

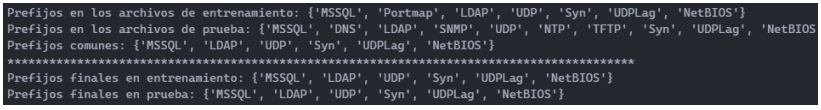


Figura 3.15: Prefijos comunes en listas de entrenamientos y prueba.

La información de nuestro DataFrame en la figura nos muestra que contamos con datos numéricos y con solo una columna categórica, la columna clave 'Label' que representa los diferentes tipos de tráfico.

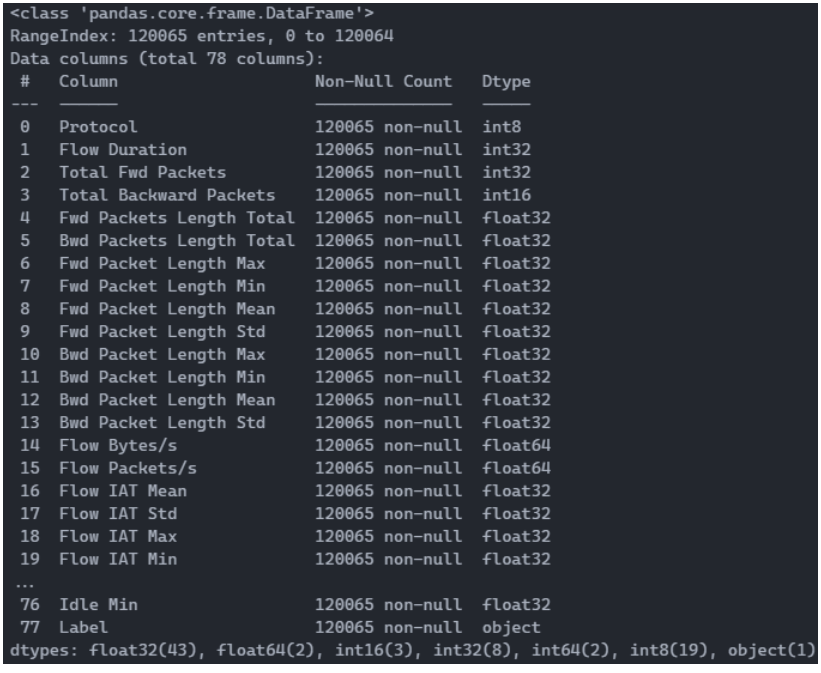


Figura 3.16: Tipos de datos de nuestro DataFrame.

Al analizar la distribución de los tipos de ataques de denegación de servicio en nuestros dos DataFrames, se identificó que el DataFrame de prueba contenía un tipo de ataque que no estaba presente en el DataFrame de entrenamiento. Para mantener el equilibrio en el proceso de modelado, este tipo de ataque fue eliminado del DataFrame de prueba, asegurando que ambos conjuntos de datos contengan los mismos tipos de tráfico y evitando posibles sesgos en el entrenamiento y evaluación del modelo. Se muestra en la figura la distribución

de datos de cada clase en el dataset.

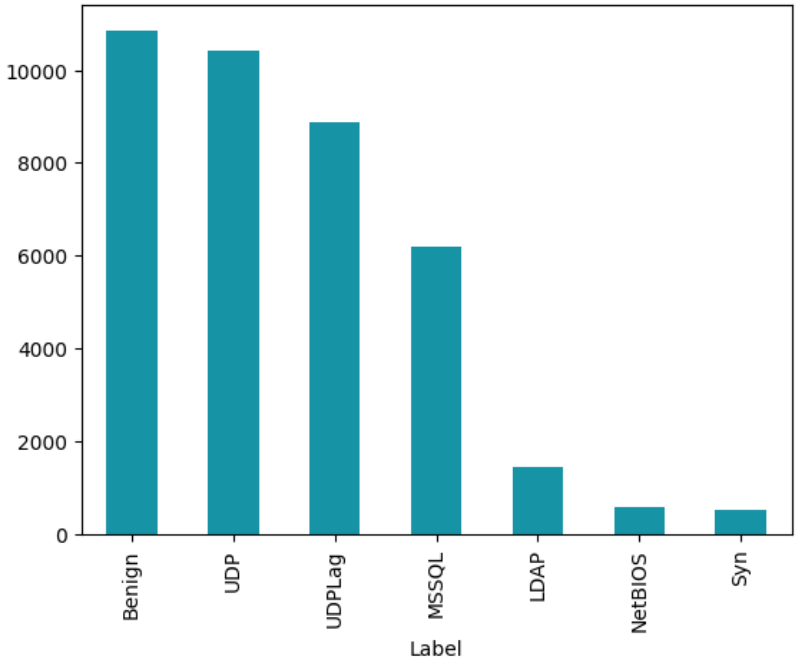


Figura 3.17: Clases finales.

El balanceo de nuestros datos con respecto a la columna clave 'Label' muestra una distribución visualmente dispareja. Al calcular la media y la desviación estándar, se observa que la desviación representa el 77 % de la media. Esta alta variabilidad indica un desbalance en la distribución de las clases, lo que sugiere la necesidad de aplicar técnicas de balanceo para mejorar el rendimiento de los modelos de aprendizaje automático. En este caso, debido a que la cantidad de registros de algunos tipos de ataques era considerablemente baja en comparación con otros, se decidió eliminar estos tres tipos de ataques: LDAP, NetBIOS y Syn. Al volver a calcular la media y la desviación estándar de los datos, se observó que la desviación estándar representa un 19 % de la media, lo que sugiere que la variabilidad no es alta y los datos se encuentran relativamente equilibrados, también se observa en la siguiente figura que la diferencia de las clases ya es menor.

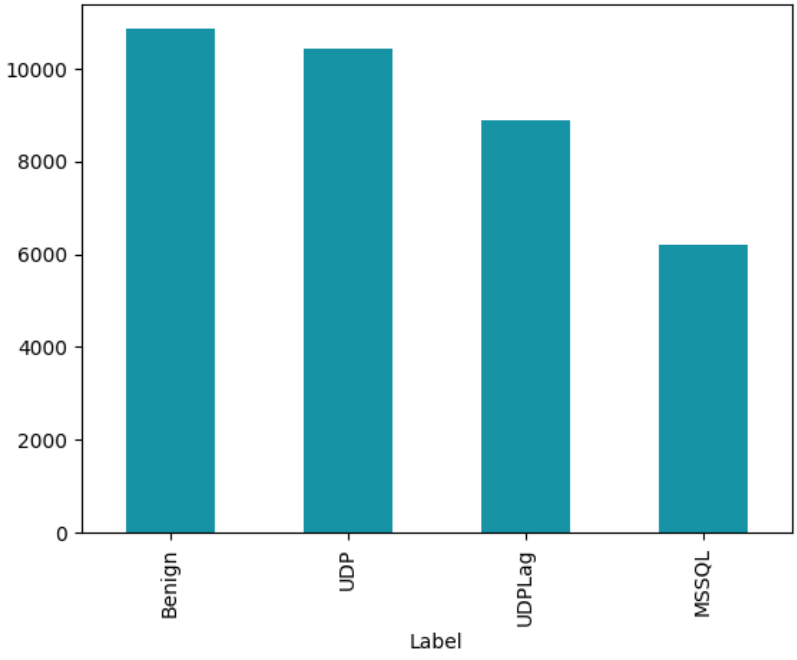


Figura 3.18: Clases finales equilibradas.

Con nuestras clases ahora equilibradas, procederemos a identificar aquellas columnas numéricas que deberían ser categóricas. Este tipo de columnas suelen tener pocos valores únicos, y su conversión a tipo categórico puede mejorar la eficiencia y el rendimiento de los modelos, además de reducir la complejidad computacional. En total, se identificaron 19 características para su conversión a formato categórico. Además, se revisaron las columnas categóricas de alta cardinalidad, que son aquellas que tienen demasiados valores únicos. En este caso, no se encontró ninguna columna con alta cardinalidad y finalmente, se combinarán las columnas categóricas y las numéricas que deben ser tratadas como categóricas y se identificaron las columnas numéricas que no fueron consideradas categóricas, eliminando aquellas que ya fueron clasificadas como numéricas. De esta forma, se logró separar correctamente las columnas numéricas para el análisis.

En cuanto a valores nulos, no se encontraron pero se detectaron valores duplicados, los cuales fueron eliminados. Como se observa en la siguiente figura se eliminaron 1,411 registros duplicados, quedando un total de 67,264 registros

y 78 características, los cuales se utilizarán para el posterior análisis con los modelos de aprendizaje automático.

```
print(f"Registros: {traindfc.shape[0]}")
print(f"Características: {traindfc.shape[1]}")
```

```
Registros: 67264
Características: 78
```

Figura 3.19: Datos para entrenamiento.

3.3. Análisis de los datos

El análisis datos es un paso esencial para comprender la estructura, distribución y relaciones dentro del conjunto de datos. A través de técnicas estadísticas y visualizaciones, se identifican patrones, comportamientos, anomalías y tendencias que pueden influir en el desempeño de los modelos predictivos. En esta sección, se presentan los métodos aplicados para examinar cada conjunto de datos (Phishing, Ransomware y DoS/DDoS), incluyendo análisis descriptivo, correlaciones entre variables y distribución de datos. Estos procedimientos permiten obtener información clave para la selección de características, optimización del modelado y toma de decisiones fundamentadas en el proceso de análisis y anterior al modelado.

3.3.1. Phishing

El conjunto de datos para phishing arrojo datos y patrones con correlaciones interesantes, comenzando por la distribución de la longitud de la URL por clase (0 legitimo y 1 phishing), en el grafico se muestra un patrón similar entre sitios legítimos y de phishing, con ambas categorías concentrándose principalmente entre 40-100 caracteres. Esta notable superposición sugiere que la longitud de URL por sí sola no constituye un factor determinante para identificar sitios maliciosos, ya que demuestra que los atacantes que crean sitios de phishing están construyendo URLs con longitudes que imitan el patrón de sitios legítimos.

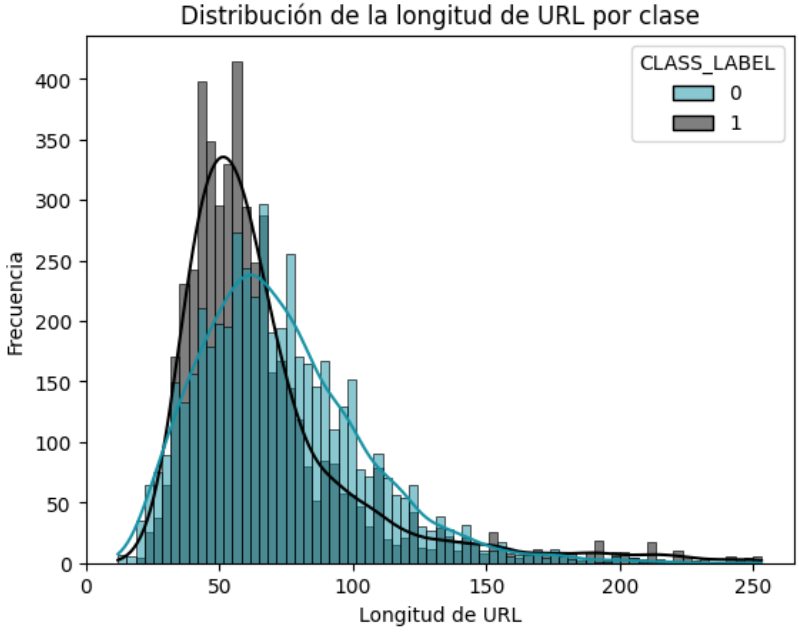


Figura 3.20: Distribución de la longitud de URL por clase.

Por otro lado, se observó que en la figura, la relación entre las características UrlLength y HostnameLength, mostrada por clase, revela un patrón distintivo: mientras que las URLs legítimas tienden a mantener nombres de host relativamente cortos, independientemente de la longitud total de la URL, las URLs de phishing muestran una tendencia a tener nombres de host significativamente más largos. En algunos casos, estos superan considerablemente el rango común. Esta observación sugiere que la longitud del nombre de host podría ser un indicador valioso para los sistemas de detección de phishing. Los atacantes parecen utilizar nombres de host extensos, posiblemente con el fin de ocultar elementos maliciosos o simular legitimidad mediante subdominios complejos.

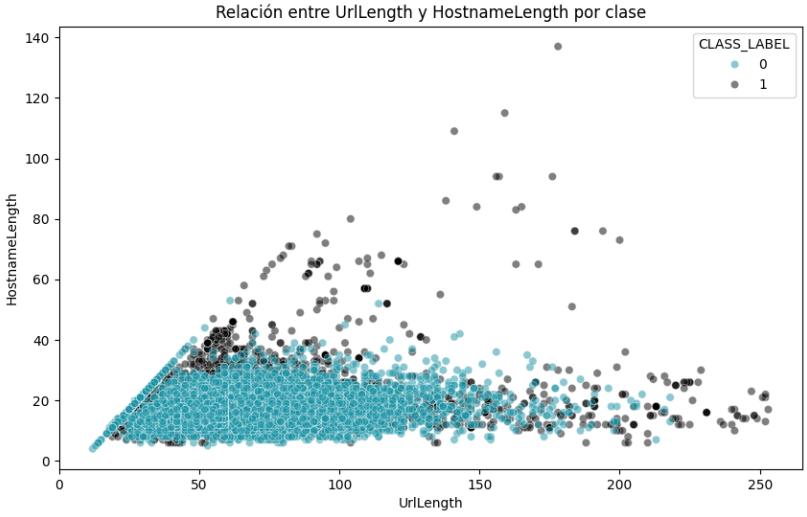


Figura 3.21: Relación entre la longitud de la URL y la longitud del nombre de host por clase.

También se observó una diferencia significativa en la cantidad de URLs que no utilizan el Protocolo de Transferencia de Hipertexto Seguro (https) entre las clases. Se puede visualizar en la figura un patrón claro, donde las URLs de phishing presentan una frecuencia de no uso de https que alcanza hasta 10,000, lo cual es considerablemente alto. En cambio, las URLs legítimas no superan una frecuencia de 1,000.

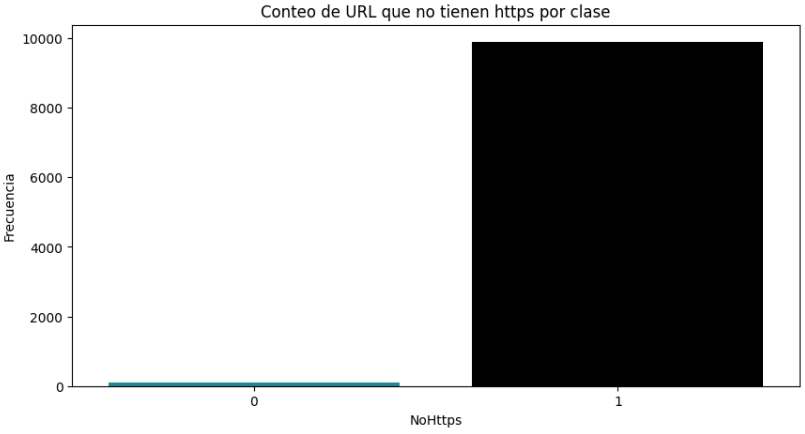


Figura 3.22: Conteo de URL que no tienen https por clase.

Al comparar otras características como NumDots y NumDash, que representan el número de puntos y guiones en las URLs, se observa en la grafica que la relación entre estas dos variables, en función de la tendencia según la clase, revela un patrón distintivo que podría ser crucial para la detección de phishing. Se identifica una tendencia inversa entre ambas variables, donde un aumento en NumDots se asocia con una disminución en NumDash. Específicamente, los datos etiquetados como phishing (1) tienden a concentrarse en la región donde NumDots es bajo y NumDash es relativamente alto, mientras que los datos legítimos (0) presentan una distribución más dispersa. Este patrón sugiere que la combinación de un bajo número de puntos (NumDots) y un alto número de guiones (NumDash) podría ser un indicador clave de sitios web de phishing.

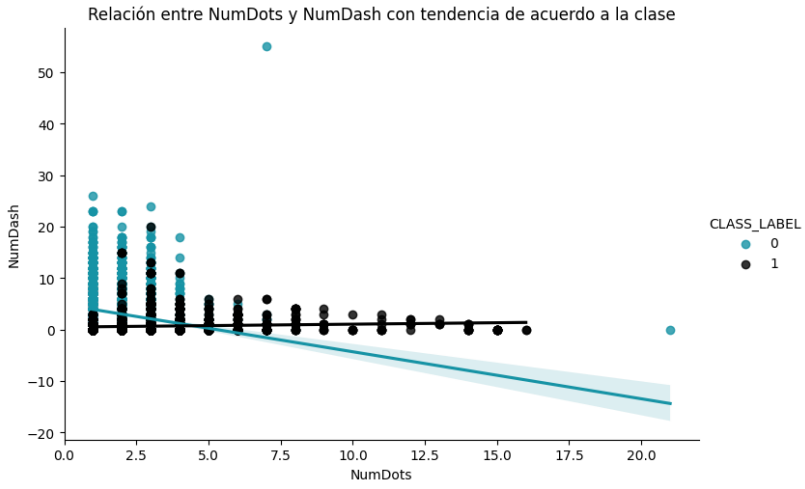


Figura 3.23: Relación entre NumDots y NumDash con tendencia de acuerdo a la clase.

Por ultimo respecto a la correlación de todas las características se destaco que hay una fuerte asociación entre longitud y complejidad de la URL y uso excesivo de parámetros, estos pueden ser indicadores útiles para clasificar una URL como phishing.

3.3.2. Ransomware

Este conjunto de datos, más extenso y distribuido entre diferentes tipos de ataques, evidenció relaciones claras entre el tráfico malicioso de ransomware y el tráfico benigno. A partir del análisis de la distribución de bytes transmitidos por cada tipo de tráfico, se observan diferencias significativas.

Como se muestra en la Figura 3.24, el tráfico relacionado con ransomware presenta irregularidades, alcanzando volúmenes cerca de 400,000 bytes, mientras que el tráfico legítimo se mantiene generalmente alrededor de los 100,000 bytes. Se puede deducir que si una red mantiene un patrón de tráfico promedio estable, cualquier desviación notable, sea por un volumen excesivamente alto o inusualmente bajo, podría facilitar la detección rápida de un ataque, tomando estos parámetros como referencia.

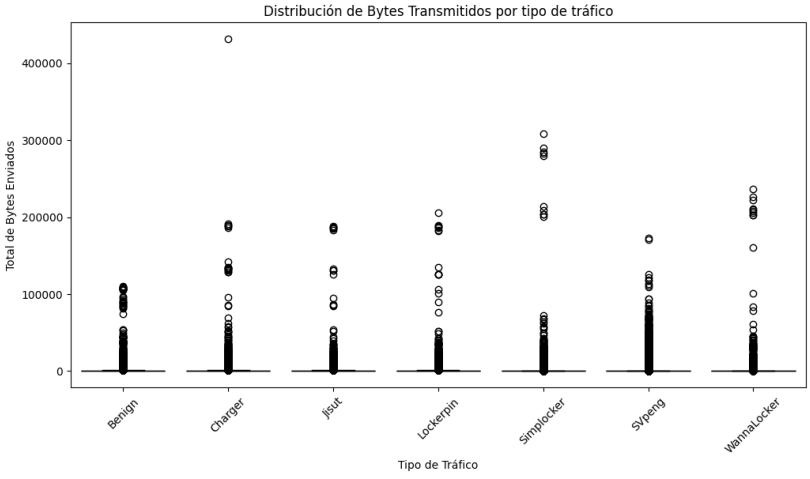


Figura 3.24: Distribución de Bytes transmitidos por tipo de tráfico.

Observando desde una perspectiva más general, en este caso la distribución en el número de paquetes por tipo de tráfico, se aprecia una diferencia mínima. Sin embargo, a diferencia del análisis anterior con los bytes, aquí la situación se invierte: el tráfico benigno presenta un mayor número de paquetes, mientras que el tráfico de ransomware muestra una cantidad similar pero ligeramente menor, con menos variabilidad. Aunque la diferencia no es demasiada, se puede aplicar un razonamiento similar al del análisis de bytes: si la red mantiene un comportamiento de tráfico estable, cualquier desviación significativa en la cantidad de paquetes puede ser una señal de alerta y posible indicio de actividad maliciosa.

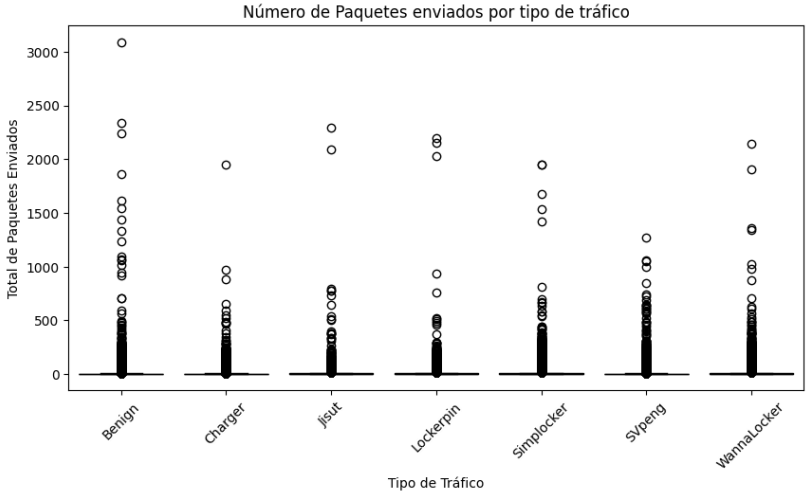
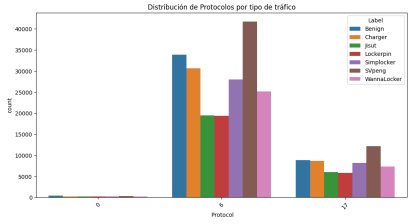


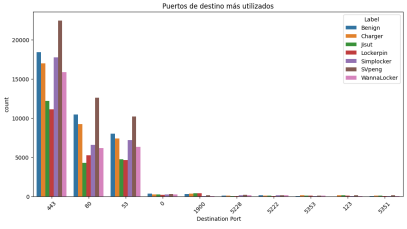
Figura 3.25: Distribución de Paquetes transmitidos por tipo de tráfico.

En las siguientes figuras se analizó el tráfico en los protocolos y puertos, comenzando por los protocolos mostrado en la Figura 3.26 (a), identificando la utilización de tres de ellos: 0 (IP sin definir o reservado), 6 (TCP - Transmission Control Protocol) y 17 (UDP - User Datagram Protocol). Principalmente se observaron los protocolos 6 y 17, mostrando un patrón de tráfico similar entre los distintos tipos de tráfico, aunque con menor volumen en el protocolo 17. Dado que los patrones no eran completamente claros en los protocolos, se procedió a analizar los puertos de destino más utilizados mostrados en la Figura 3.26 (b), siendo: 443 (HTTPS - navegación segura), 80 (HTTP - navegación web estándar) y 53 (DNS - resolución de nombres de dominio). Al igual que con los protocolos, los patrones de tráfico en estos puertos eran bastante similares entre las distintas categorías, tanto benignas como maliciosas.

Sin embargo, una diferencia destacable fue observada en el caso del ransomware SVpeng, el cual presentó un volumen de tráfico superior en ambos graficos y en comparación con los demás tipos de ransomware, indicando que intenta generar más comunicación o transferencias durante su actividad, posiblemente como parte de su proceso para exfiltrar datos o contactar con servidores de comando y control.



(a) Distribución de Protocolos por tipo de tráfico.



(b) Puertos de destino más utilizados.

Figura 3.26: Trafico de protocolos y puertos utilizados.

Respecto a las correlaciones, se identificaron tres datos relevantes:

- Cuando el tráfico hacia atrás (respuestas del servidor) muestra mucha variabilidad en los tiempos entre paquetes, el flujo total de la conexión tiende a mantenerse más estable, reflejando cómo, en ataques de ransomware, los servidores maliciosos responden de manera irregular o errática, mientras que la conexión general intenta mantener un ritmo constante.

- Cuando el flujo completo de datos tiene alta variabilidad en los tiempos entre paquetes, los intervalos más largos en el tráfico hacia atrás tienden a ser menores, sugiruendi que el malware instalado por el atacante podría estar enviando paquetes al servidor de comando y control de forma irregular, pero las respuestas desde ese servidor malicioso tienden a ser más regulares y predecibles.

- Cuando los intervalos hacia atrás son muy largos, el flujo general tiende a ser más estable (con menor variabilidad). Este patrón puede indicar que, en algunos ataques de ransomware, las respuestas del servidor llegan con más demora, pero la conexión global mantiene un comportamiento estructurado y sin grandes cambios.

En la siguiente figura se muestra el mapa de correlación de todas las características, donde las correlaciones altas se representan con colores cercanos al rojo, mientras que las correlaciones negativas se indican con colores cercanos al azul.

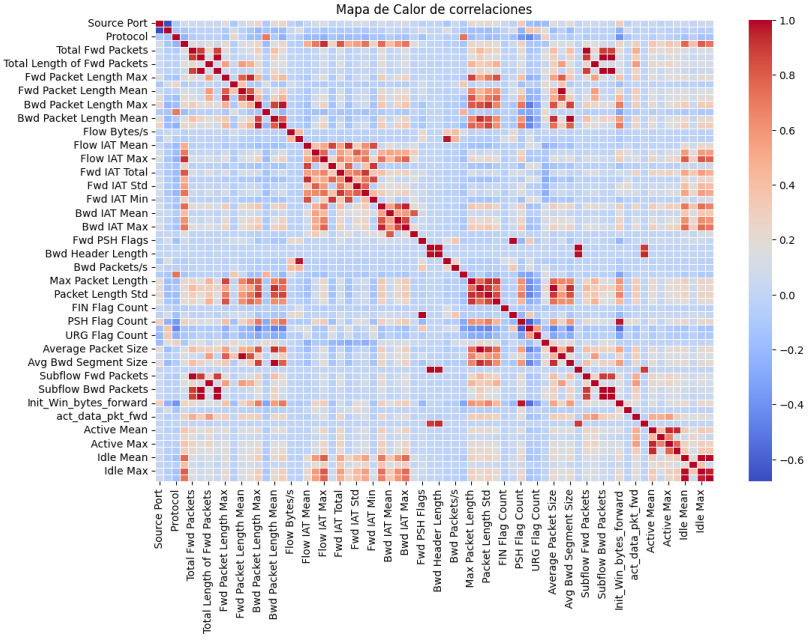


Figura 3.27: Mapa de calor que muestra las correlaciones entre características del tráfico de red.

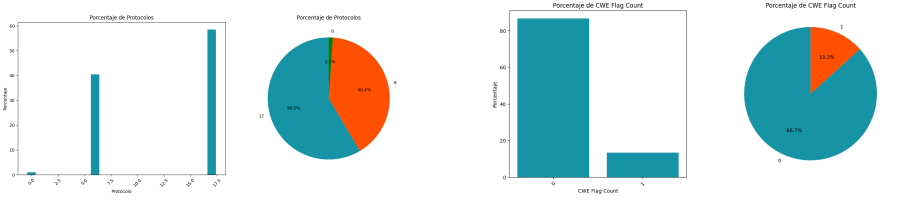
3.3.3. DoS y DDoS

En el conjunto que contiene ataques de denegación de servicio se identificaron distintos patrones importantes para reconocer este tipo de amenazas en el tráfico de red, destacándose el uso de dos protocolos específicos: el 17 (UDP - User Datagram Protocol) y el 6 (TCP - Transmission Control Protocol) como se muestra en la Figura 3.28 (a).

El puerto 17, asociado al servicio QOTD (Quote of the Day), es prácticamente obsoleto y su uso legítimo es casi inexistente, por lo que cualquier tráfico que lo involucre resulta sospechoso. Este servicio también ah sido explotado a través de UDP con ataques de amplificación mediante reflexión. Por otro lado, el protocolo 6, correspondiente a TCP, es el que aparece con mas frecuencia, esto podría tratarse de una manipulación de encabezados para evadir reglas de filtrado.

De acuerdo con los errores o fallos observados en el tráfico clasificado como legítimo (0) y malicioso (1) como se visualiza en el grafico 3.28 (b), la gran mayoría corresponde a tráfico legítimo, con un 92 % que no presenta indicios claros

de explotación de vulnerabilidades según el sistema de clasificación CWE. Esto podría indicar que, en caso de haber un ataque DDoS, este se encuentra mezclado con tráfico limpio, lo que complica su detección y mitigación. El pequeño porcentaje restante, un 7 % de tráfico clasificado como malicioso, podría representar intentos de explotación de vulnerabilidades específicas, concentrando así el comportamiento sospechoso vinculado a un posible ataque DDoS o a actividades de reconocimiento previas al mismo.



(a) Distribución de trafico por protocolos.

(b) Distribución de trafico de errores o fallos por tipo de trafico.

Figura 3.28: Distribución de trafico de protocolos y errores en el trafico.

Otro hallazgo importante está relacionado con el rango de duración. En este caso, la siguiente figura muestra que los tres tipos de ataques analizados presentan duraciones bajas cercanas a cero. Esto podría sugerir intentos de ataque rápidos y repetitivos, típicos de ataques DDoS de corta duración. En contraste, los valores altos de duración se observan principalmente en el tráfico benigno, lo que indica que este tipo de tráfico tiende a mantener conexiones más largas y estables.

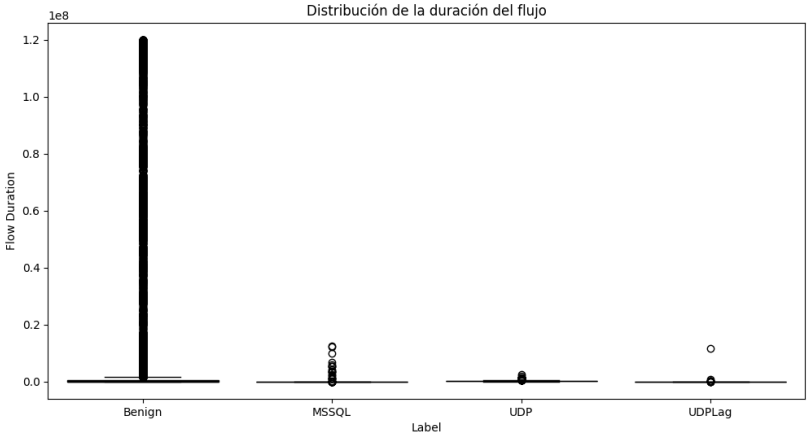


Figura 3.29: Distribución de la duración de flujo de los diferentes tipos de trafico.

También se identifican outliers (valores atípicos), especialmente en los ataques 'MSSQL' y 'UDPLag'. Estos valores elevados de duración podrían corresponder a intentos de explotación prolongados o sostenidos en el tiempo. Este patrón es evidente en una red donde el tráfico prolongado es común, la detección de conexiones inusualmente breves puede ser una señal de actividad anómala.

En cuanto a la longitud media del paquete en cada tipo de tráfico, en el siguiente grafico se observa un comportamiento distinto en dos de los tres protocolos analizados (0, 6 y 17). Para el protocolo 6 (TCP), el tráfico benigno muestra una duración de flujo generalmente larga y consistente, en contraste con el tráfico malicioso, que presenta duraciones más cortas.

Por otro lado, en el protocolo 17 (UDP) ocurre lo contrario, donde el tráfico benigno tiene flujos cortos pero también estables, mientras que el tráfico malicioso (MSSQL, UDP y UDPLag) muestra una mayor variabilidad en la duración, incluyendo valores atípicos muy elevados. Esto sugiere que los ataques DDoS sobre UDP pueden generar conexiones anómalamente largas e inestables.

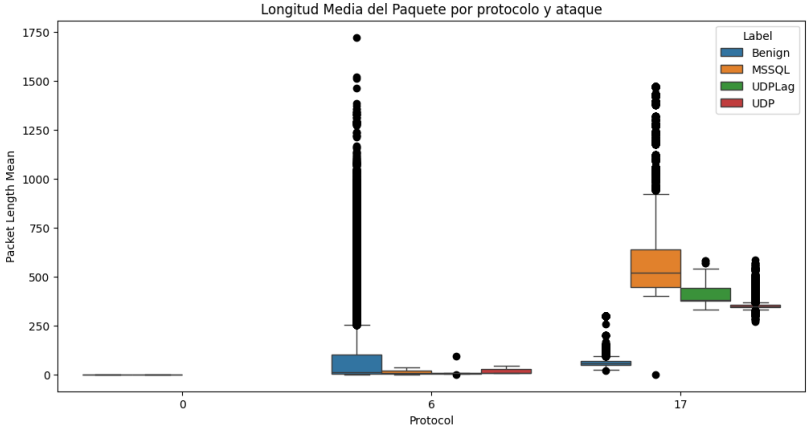


Figura 3.30: Distribución de longitud media del paquete por protocolo y trafico.

Respecto al mapa de calor y las correlaciones positivas, se mencionan las siguientes:

- La correlación entre Subflow Bwd Packets y Total Backward Packets muestra una relación perfecta, lo que sugiere un tráfico DDoS en el que se envía un volumen muy alto y consistente de paquetes hacia atrás. Este comportamiento es típico en ataques DDoS, donde se inunda la red con grandes cantidades de tráfico hacia atrás, lo que puede colapsar los dispositivos de red. Un patrón similar se observa en la correlación entre Bwd Packets Length Total y Subflow Bwd Bytes.
- La correlación entre RST Flag Count y Fwd PSH Flags indica que la activación conjunta de las banderas RST (Reset) y PSH (Push) puede ser un indicio claro de que los atacantes intentan interrumpir las conexiones de red de manera sincronizada. Este comportamiento es común en ataques como el TCP SYN Flood, donde los paquetes con la bandera RST intentan cerrar conexiones, mientras que los paquetes con la bandera PSH buscan enviar datos rápidamente, interrumpiendo así el flujo normal.

En cuanto a las correlaciones negativas, se mencionan las siguientes:

- La correlación negativa entre Protocol y ACK Flag Count indica que, a medida que se incrementa el uso de protocolos como UDP, disminuye significativamente la cantidad de respuestas ACK. Este patrón es característico de ataques DDoS basados en UDP, como los UDP Flood o DNS

Amplification, donde no se requiere establecer una conexión formal. En contraste, el tráfico benigno basado en TCP sí utiliza ACK, por lo que esta ausencia de confirmaciones es un fuerte indicio de tráfico malicioso.

- La correlación negativa entre Fwd IAT Mean y Protocol refleja que, cuando se utiliza un protocolo como UDP o se incrementa la actividad de tipo flood, el tiempo promedio entre la llegada de paquetes hacia adelante disminuye. Este comportamiento es típico en ataques DDoS, donde los paquetes se envían en ráfagas muy rápidas, reduciendo el intervalo entre ellos y generando un flujo constante y agresivo hacia el objetivo.
- La correlación negativa entre Protocol y Flow IAT Std sugiere que el tráfico generado bajo ciertos protocolos tiende a presentar intervalos de tiempo entre flujos más constantes y con poca variabilidad. Este patrón es común en ataques DDoS tipo Flood, donde el tráfico es uniforme, repetitivo y carece de la variabilidad natural del tráfico legítimo. Esta regularidad puede ser una señal clara de automatización y comportamiento malicioso.

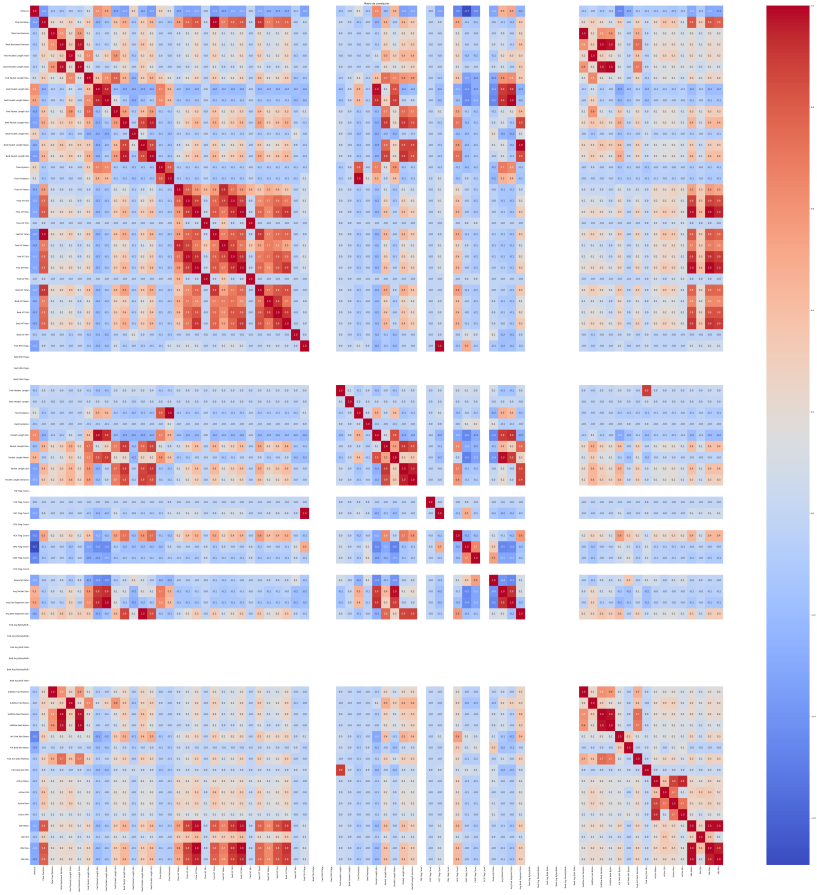


Figura 3.31: Mapa de calor que muestra las correlaciones entre características del tráfico de red.

Capítulo 4

Resultados

Tras la selección, análisis y preparación de los datos, continua la etapa final: la evaluación. En esta fase, cada uno de los conjuntos de datos será sometido varios tipos de algoritmos de aprendizaje automático, con el objetivo de comparar su rendimiento. Este procedimiento permitirá identificar la mejor opción para la detección de cada tipo de ataque cibernético.

Los algoritmos utilizados para los conjuntos de datos son:

- Random Forest (Supervisado)
- Support Vector Machine (SVM - Supervisado)
- CatBoost (Supervisado)
- XGBoost (Supervisado)
- Regresión Logística (Supervisado)
- K-Nearest Neighbors (KNN - Supervisado)

4.1. Phishing

Este conjunto de datos será evaluado utilizando los siguientes algoritmos: Random Forest, SVM (Support Vector Machine) y CatBoost. Random Forest suele ser más frecuente en análisis de clasificación debido a su robustez y efectividad. Por otro lado, CatBoost, aunque es un método relativamente más nuevo, ha ganado popularidad rápidamente gracias a su rendimiento en problemas con variables categóricas, pero aún no es tan común como Random Forest en este tipo de análisis.

4.1.1. Random Forest

El conjunto de datos fue dividido en un 80 % para entrenamiento y un 20 % para prueba. El modelo se entrenó utilizando 400 árboles de decisión, limitando el número máximo de nodos hoja a 400 y considerando hasta 10 variables aleatorias en cada división. El modelo alcanzó una precisión del 98.5 % en el conjunto de datos de prueba, lo que indica un rendimiento excelente en la clasificación de instancias de Phishing y No-Phishing. Este alto nivel de precisión sugiere que el modelo es altamente confiable para distinguir entre ambos tipos de instancias.

El rendimiento del modelo fue evaluado mediante una matriz de confusión 4.1, la cual proporciona una visión sobre los aciertos y errores en la clasificación. De un total de 2000 instancias, el modelo clasificó correctamente 973 legítimos y 997 de phishing. Los errores de clasificación fueron mínimos: apenas 15 legítimos fueron erróneamente clasificados como phishing (falsos positivos) y 15 de phishing fueron clasificados incorrectamente como legítimos (falsos negativos).

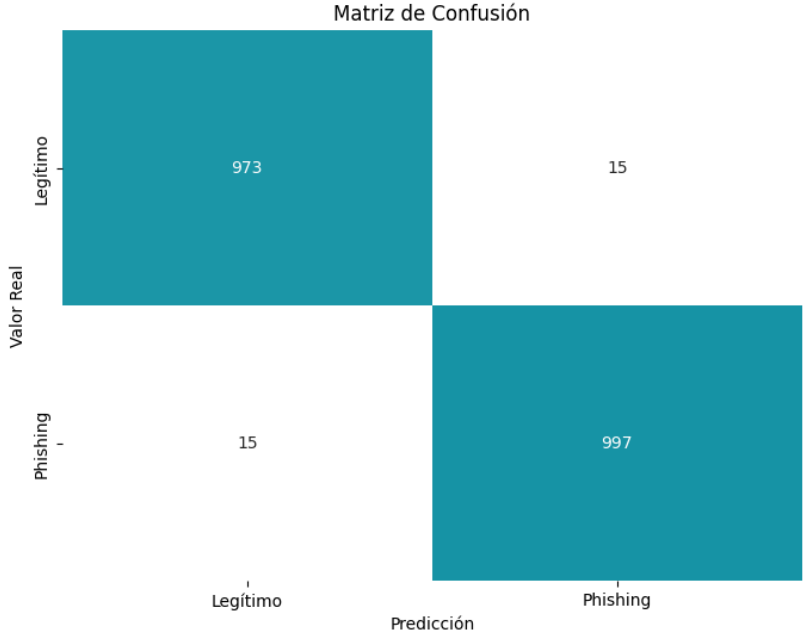


Figura 4.1: Matriz de confusión para la detección (Random Forest).

Se evaluó la importancia de las variables utilizadas por el modelo para

realizar las predicciones, mediante la medida de importancia de características propia del algoritmo Random Forest. Esta evaluación permite identificar qué variables tienen un mayor peso en la clasificación.

El análisis reveló que las características más relevantes fueron:

- PctExtHyperlinks (22.81 % de importancia)
- PctExtNullSelfRedirectHyperlinksRT (19.65 %)
- FrequentDomainNameMismatch (8.88 %)

En particular, la alta relevancia de PctExtHyperlinks y PctExtNullSelfRedirectHyperlinksRT sugiere que la proporción de hipervínculos externos y el comportamiento anómalo en redireccionamientos son fuertes indicadores para identificar páginas web asociadas a actividades de phishing. Como se muestra en la siguiente figura como resaltan las dos características mencionadas.

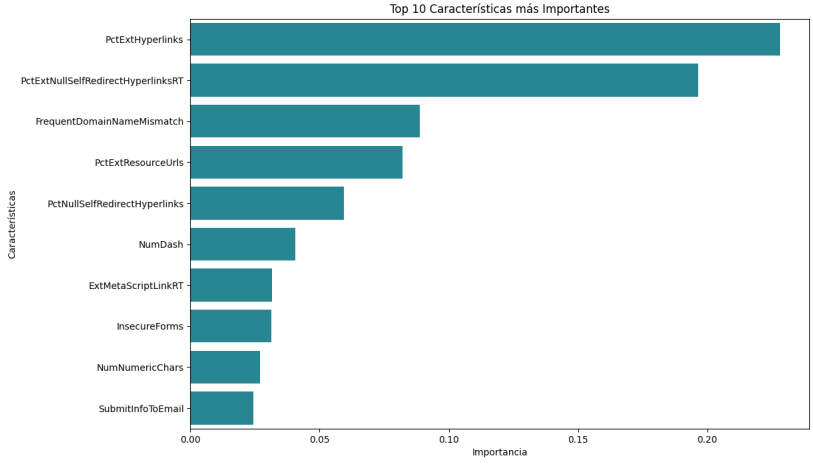


Figura 4.2: Características mas importantes para predicción (Random Forest).

La curva de aprendizaje muestra que el modelo alcanza una precisión cercana al 100 % en el conjunto de entrenamiento, mientras que en el conjunto de validación varía entre 95 % y 98 %. Esta pequeña diferencia sugiere que el modelo generaliza bien, sin signos de sobreajuste, lo que indica un buen rendimiento en la detección de sitios web de phishing.

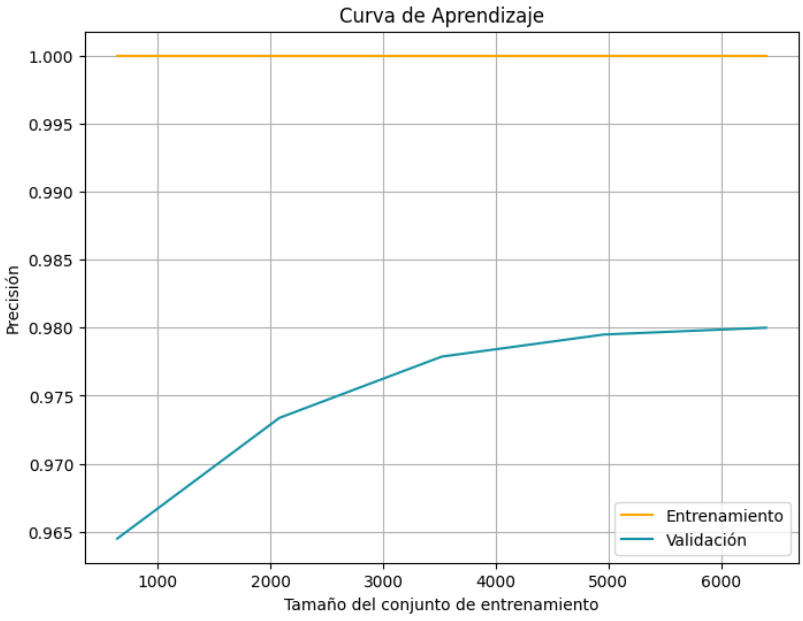


Figura 4.3: Curva de aprendizaje del modelo de detección de sitios web (Random Forest).

4.1.2. Support Vector Machine - SVM

El modelo SVM (Máquinas de Vectores de Soporte) los datos se dividieron en un 80 % para entrenamiento y un 20 % para prueba. El modelo fue entrenado con el conjunto de entrenamiento y evaluado en el conjunto de prueba. Los resultados mostraron una precisión del 99.9 %, evidenciando un rendimiento sobresaliente en la clasificación correcta de URLs tanto legítimas como de phishing. La precisión perfecta del modelo en las predicciones resalta su efectividad en la detección de sitios web fraudulentos.

La matriz de confusión 4.4 muestra que el modelo SVM clasificó correctamente 988 URLs legítimas y 1010 URLs de phishing. Sin embargo, se presentaron 2 falsos negativos, es decir, URLs de phishing que fueron clasificadas erróneamente como legítimas. A pesar de estos pocos errores, la precisión del modelo sigue siendo extremadamente alta, reflejando su capacidad para diferenciar eficazmente entre sitios web legítimos y fraudulentos.

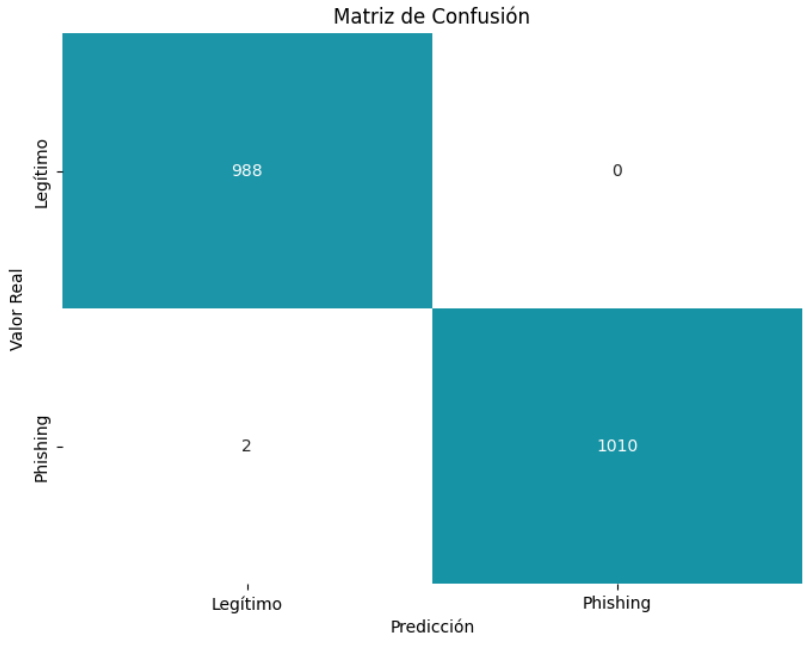


Figura 4.4: Matriz de confusión para la detección (SVM).

La curva de aprendizaje muestra un rendimiento sobresaliente del modelo SVM tanto en el conjunto de entrenamiento como en el de validación. Durante el entrenamiento, la puntuación AUC osciló entre 0.995 y 0.999, mientras que en la validación se mantuvo entre 0.992 y 0.999. Estos valores reflejan una capacidad excelente del modelo para discriminar entre URLs legítimas y de phishing, indicando una clasificación robusta y precisa en ambos conjuntos.

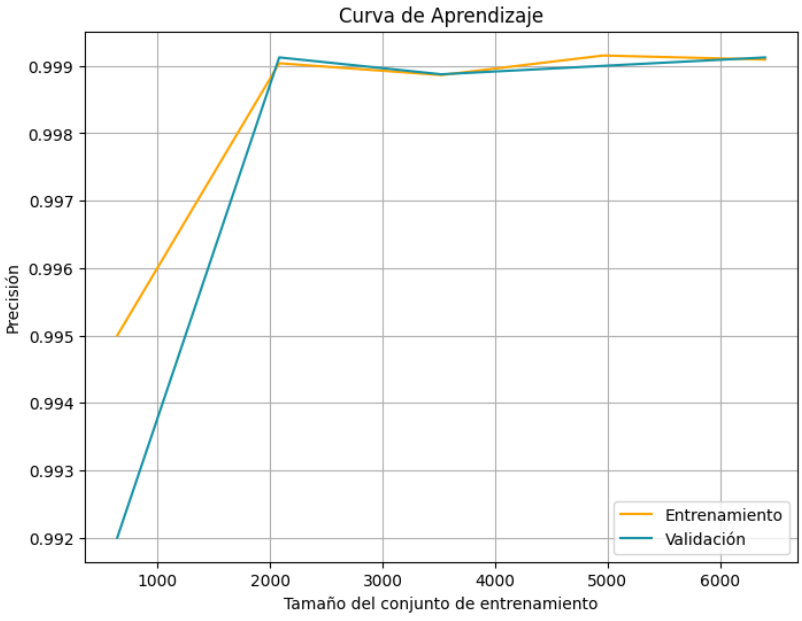


Figura 4.5: Curva de aprendizaje del modelo de detección de sitios web (SVM).

4.1.3. CatBoost

Por ultimo se utilizó el modelo CatBoost y tras dividir los datos en un 80 % para entrenamiento y un 20 % para prueba, el modelo fue entrenado y evaluado con los datos de prueba. El modelo alcanzó una precisión solida del 99 % para las URLs legitimas y con phishing. Estos resultados sugieren que el modelo es bastante bueno para detectar phishing.

La matriz de confusión 4.6 muestra que el modelo CatBoost clasificó correctamente 3969 URLs legítimas y 3976 URLs de phishing. También cometió 31 falsos positivos (URLs legítimas clasificadas como phishing) y 24 falsos negativos (URLs de phishing clasificadas como legítimas). Este patrón indica que el modelo tiene solidez al clasificar las URLs de phishing.

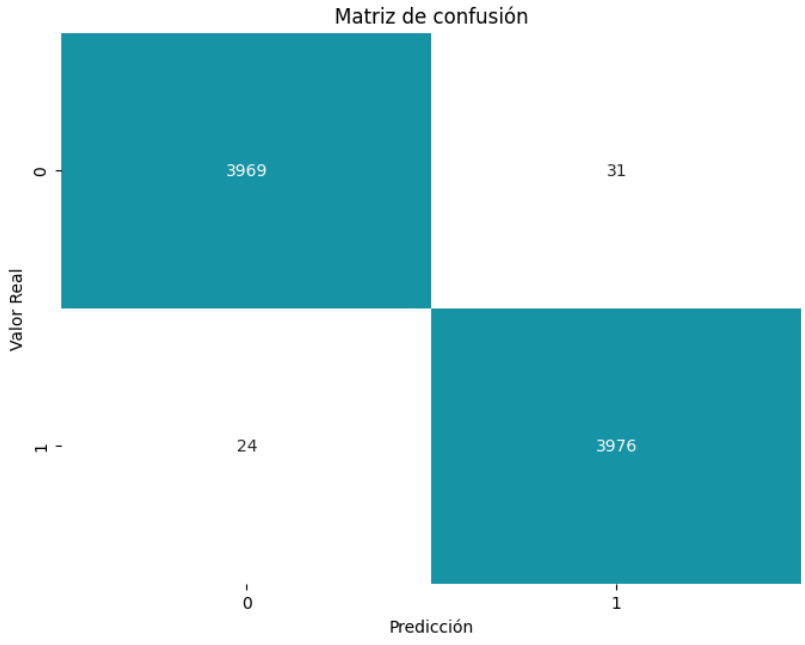


Figura 4.6: Matriz de confusión para la detección (CatBoost).

La curva de aprendizaje del modelo CatBoost mostró un valor solido en en entrenamiento mientras que en la validación la precisión osciló entre 0.978 y 0.984. Estos resultados indican que el modelo generaliza de manera consistente.

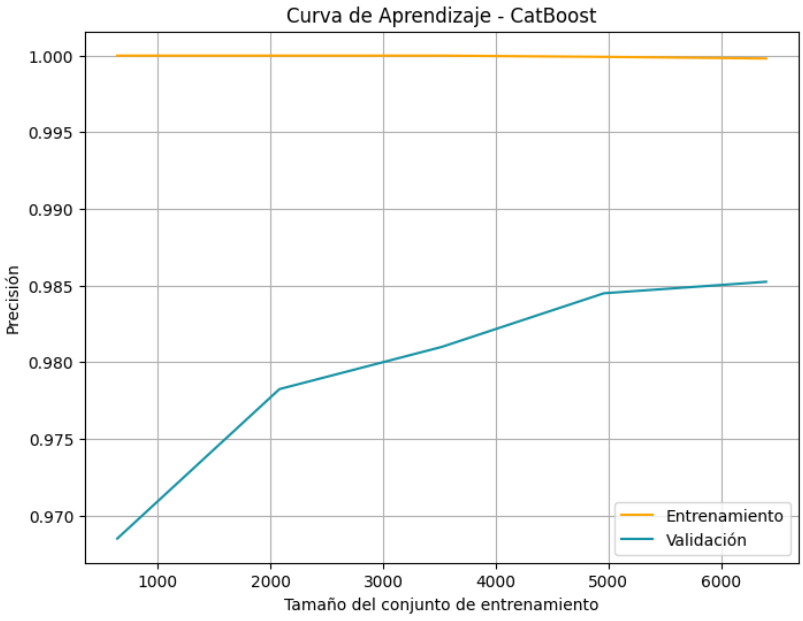


Figura 4.7: Curva de aprendizaje del modelo de detección de sitios web (CatBoost).

4.1.4. Comparación de Modelos

El modelo CatBoost obtuvo el mejor desempeño, alcanzando un rendimiento casi perfecto, con errores mínimos en la clasificación. RandomForest presento un rendimiento muy poco por debajo, mientras que SVM presento un desempeño inferior a los demás, evidenciando mayores dificultades en la correcta clasificación de las URLs, especialmente en la detección de sitios legítimos. A continuación en la tabla se muestran resumidos todos los resultados de los algoritmos entrenados.

Algoritmo	Entrenamiento		
	Precisión	**Recall**	**F1-Score**
Random Forest	0.98	0.98	0.98
No-Phishing	0.98	0.98	0.98
Phishing	0.98	0.98	0.98
SVM	0.86	0.86	0.86
No-Phishing	0.90	0.82	0.86
Phishing	0.84	0.91	0.87
CatBoost	0.99	0.99	0.99
No-Phishing	0.99	0.99	0.99
Phishing	0.99	0.99	0.99

Tabla 4.1: Comparación de modelos Random Forest, SVM y CatBoost en ataques Phishing.

4.2. Ransomware

En este conjunto de datos se emplearon tres algoritmos de aprendizaje automático: Random Forest y XGBoost siendo una técnica de ensamble que utiliza gradiente boosting para optimizar la predicción, destacándose por su eficiencia y rendimiento en problemas de clasificación complejos.

4.2.1. Random Forest

El conjunto de datos fue dividido en un 80 % para entrenamiento y un 20 % para prueba. El modelo se entrenó utilizando 100 árboles de decisión, limitando el máximo número de nodos hoja a 400 y la profundidad máxima de los árboles a 10. Se exigió un mínimo de 5 muestras por hoja y se utilizó la raíz cuadrada del número total de características para seleccionar los atributos en cada división. Tras el entrenamiento, el modelo alcanzó una precisión de 97 % y en el conjunto de validación de 71 % en la clasificación de las diferentes variantes de ransomware y tráfico benigno. En consecuencia, es posible que el modelo necesite ajustes adicionales o técnicas de regularización para mejorar su rendimiento en validación.

En la matriz de confusión obtenida por el desempeño del modelo Random Forest en la clasificación de las diferentes categorías de tráfico, se observa que el modelo logra una correcta clasificación en la mayoría de los casos, parti-

cularmente en la clase Simplocker, que presenta un número muy elevado de verdaderos positivos, con mínimas confusiones hacia otras clases. Sin embargo, existe cierta confusión entre las clases Lockerpin, Charger, y Jisut, donde se detectan varios errores de clasificación mutua, indicando similitudes en sus patrones de tráfico. Además, para las clases Benign y WannaLocker, aunque la mayoría de las instancias se clasificaron correctamente, también se observa una cantidad significativa de confusiones entre ambas, lo que sugiere que sus características pueden ser parcialmente similares para el modelo. En general, la matriz evidencia un desempeño robusto, aunque con áreas específicas donde podría mejorarse la discriminación entre clases relacionadas.

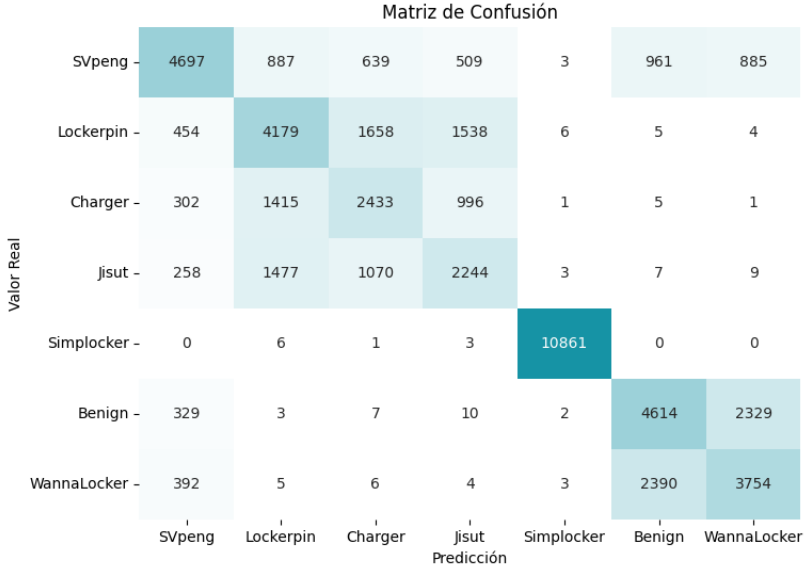


Figura 4.8: Matriz de confusión para la detección (Random Forest).

El análisis también reveló que las características más relevantes fueron:

- Source IP (31.76 % de importancia)
- Destination IP (5.75 %)
- Source Port (3.73 %)

Indicando que la dirección IP de origen, seguida de la dirección IP de destino y el puerto de origen, contienen patrones distintivos relevantes para la detección

de actividad maliciosa. En la siguiente figura se muestra la relevancia de las características.

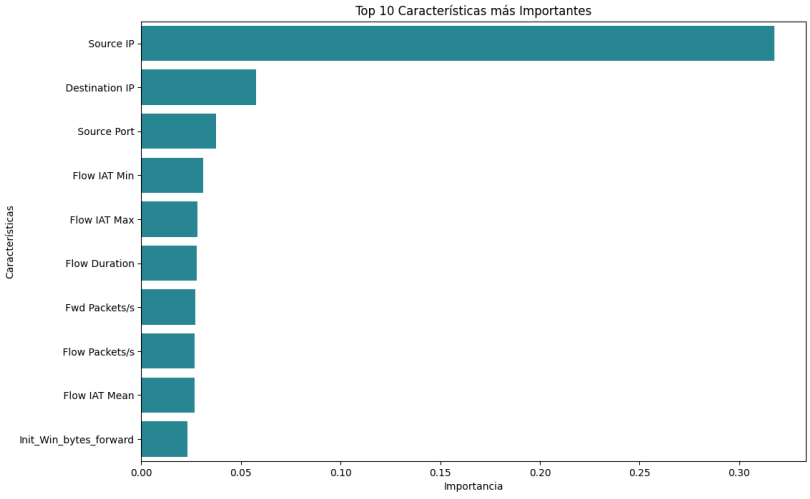


Figura 4.9: Características mas importantes para la detección (Random Forest).

4.2.2. XGBoost

En este conjunto de datos se eliminaron atributos irrelevantes como direcciones IP, identificadores de flujo y marcas de tiempo; posteriormente, las variables categóricas fueron codificadas numéricamente mediante codificación one-hot. Se utilizó una división de los datos del 80 % para entrenamiento y 20 % para prueba. El modelo XGBoost alcanzó una precisión de entrenamiento del 67 %. Respecto a la validación solo logro alcanzar un 61 % de precisión, demostrando que no fue tan eficaz al clasificar los diferentes tipos de ataques.

4.2.3. Comparación de Modelos

El modelo Random Forest mostró el mejor desempeño, alcanzando una precisión buena durante el entrenamiento, sin embargo al momento de la validación se queda corto el algunos tipos de ataques. Su capacidad para manejar grandes volúmenes de datos y su eficiente uso de características lo convierten en una opción robusta para la detección de amenazas pero si tenemos un conjunto de datos con poca calidad de datos es complicado encontrar un buen

rendimiento. Por otro lado, el modelo XGBoost se mostró con desempeño bajo en el entrenamiento y validación, una prueba mas sobre la baja calidad del conjunto de datos. Esto sugiere que, aunque los modelos tienen un buen rendimiento en datos robustos, si se encuentran con datos con mucho ruido y poca varianza tienen dificultades para entrenar y después identificar datos nuevos. A continuación en la tabla se muestran resumidos todos los resultados de los algoritmos entrenados.

Algoritmo	**Entrenamiento**		
	Precisión	**Recall**	**F1-Score**
Random Forest	0.98	0.97	0.97
SVpeng	1.00	1.00	1.00
Lockerpin	0.99	0.91	0.95
Charger	0.97	0.98	0.97
Jisut	0.97	0.94	0.96
Simplocker	0.99	0.98	0.98
Benign	0.97	0.99	0.98
WannaLocker	0.98	0.96	0.97
XGBoost	0.68	0.67	0.67
SVpeng	0.97	0.99	0.98
Lockerpin	0.67	0.35	0.46
Charger	0.54	0.48	0.41
Jisut	0.55	0.60	0.57
Simplocker	0.63	0.49	0.55
Benign	0.74	0.94	0.83
WannaLocker	0.65	0.60	0.63

Tabla 4.2: Comparación de modelos Random Forest y XGBoost en ataques Ransomware.

4.3. DoS y DDos

En este conjunto de datos se emplearon tres algoritmos de aprendizaje automático: Random Forest, Regresión Logística fue utilizada por su simplicidad y efectividad en escenarios lineales y K-Nearest Neighbors (KNN) se aplico

como un enfoque no supervisado para explorar la capacidad del modelo de agrupar patrones de tráfico malicioso sin necesidad de etiquetas predefinidas.

4.3.1. Random Forest

En el conjunto de entrenamiento, el modelo alcanzó una precisión del 96.41 %. Al ser evaluado sobre el conjunto de validación, los resultados se mantuvieron consistentes con una precisión del 96.69 %. Estos valores reflejan la capacidad del modelo para generalizar de manera eficaz sobre datos no vistos, lo que lo convierte en una herramienta sólida para la detección de tráfico malicioso.

La matriz de confusión revela un rendimiento robusto en la clasificación de las clases analizadas. La clase Benign fue clasificada correctamente en la mayoría de los casos. Para la clase MSSQL, el modelo identificó correctamente 4,940 instancias, con solo 3 casos erróneamente clasificados como Benign y 19 como UDP. En el caso de UDP, se alcanzaron 8,132 aciertos, aunque se observaron 10 errores de clasificación como Benign, 156 como MSSWL y 1 como UDPLag, por ultimo UDPLag clasifico correctamente 5,060 teniendo 281 errores como Benign, 47 como MSSQL y 1727 de UDP, siendo la clase que mostró mayor dificultad. Estos resultados evidencian un desempeño sobresaliente en clases mayoritarias, aunque con leves debilidades en la detección de clases menos representadas.

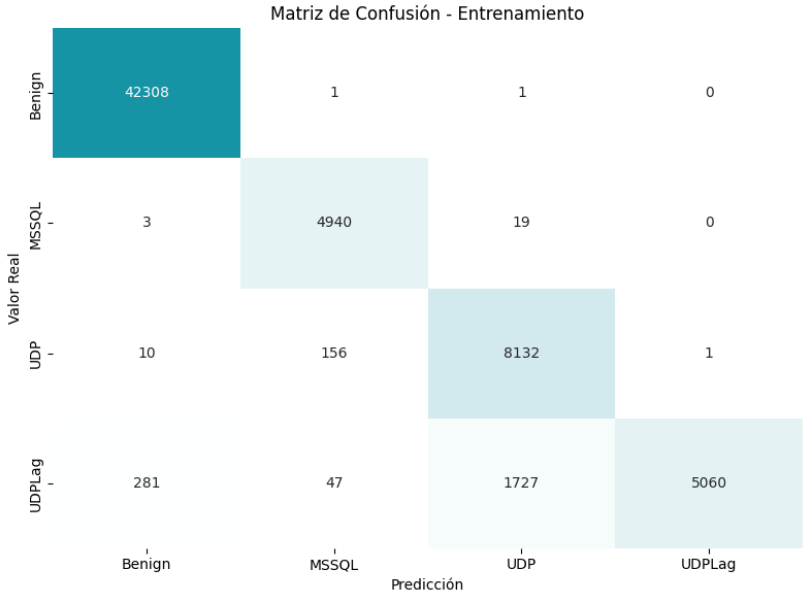


Figura 4.10: Matriz de confusión para la detección (Random Forest - Entrenameinto).

4.3.2. K-Nearest Neighbors

En el conjunto de entrenamiento, el modelo alcanzó una precisión del 99.67 %, con buena capacidad de ajuste, lo que indica que se adaptó bien a los patrones del tráfico malicioso. Al ser evaluado sobre el conjunto de validación, el modelo mostró resultados igualmente sólidos con una precisión del 95.99 %, lo que demuestra su capacidad para generalizar de manera efectiva nuevos datos. Estos resultados son indicativos de la eficiencia del modelo para detectar ataques de denegación de servicio distribuido (DDoS), ya que mantiene un rendimiento estable tanto en el entrenamiento como en la validación.

La matriz de confusión mostro un rendimiento sobresaliente en función de sus clases, observando que el modelo clasificó correctamente la mayoría de las instancias en la clase Benign con 8240 predicciones correctas, sin confundirlas con otras clases. En cuanto a la clase MSSQL, se identificaron 1638 casos correctamente, con 19 instancias clasificadas erróneamente como UDP. Para la clase UDP, el modelo mostró una alta precisión con 3461 predicciones correctas, aunque cometió algunos errores al clasificar 75 instancias como MSSQL y 2 como UDPLag. Finalmente, para la clase UDPLag, el modelo identificó

correctamente 5 instancias, pero cometió algunos errores de clasificación, con 7 instancias mal clasificadas como UDP y 3 como MSSQL, los resultados reflejan que, aunque el modelo es eficaz en la clasificación general, existen algunos casos de confusión entre clases, especialmente entre UDP, MSSQL y UDPLag.

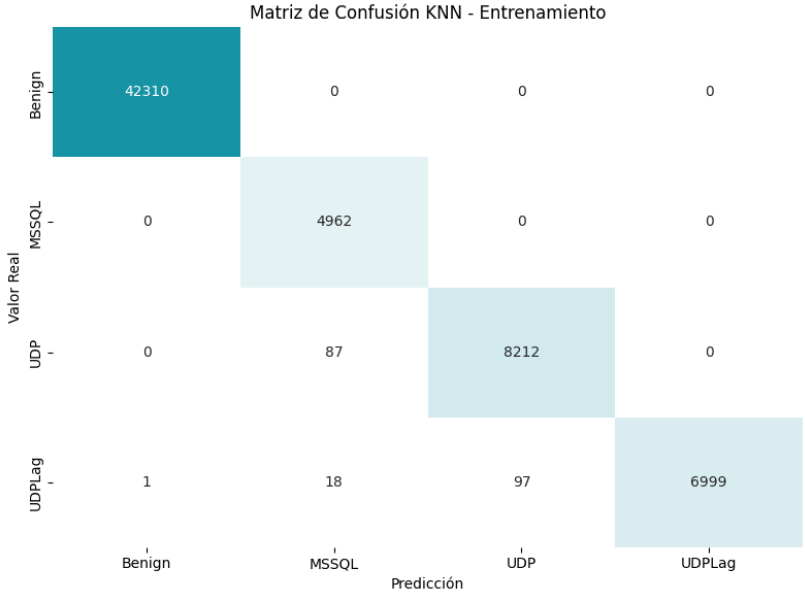


Figura 4.11: Matriz de confusión (K-Nearest Neighbors - Entrenamie).

De acuerdo con el gráfico de t-SNE (t-Distributed Stochastic Neighbor Embedding) 4.12, se muestra cómo las clases del conjunto de datos se agrupan en un espacio reducido a dos dimensiones. Las agrupaciones o clusters claros indican que el modelo KNN es capaz de distinguir entre ciertas clases de manera efectiva. Sin embargo, también se pueden observar algunas zonas donde los colores se mezclan levemente, lo que refleja áreas de solapamiento entre clases. Esto indica que el modelo KNN puede enfrentar dificultades en esas regiones, ya que las instancias de diferentes clases están cercanas unas de otras, lo que podría generar confusión al momento de hacer predicciones.

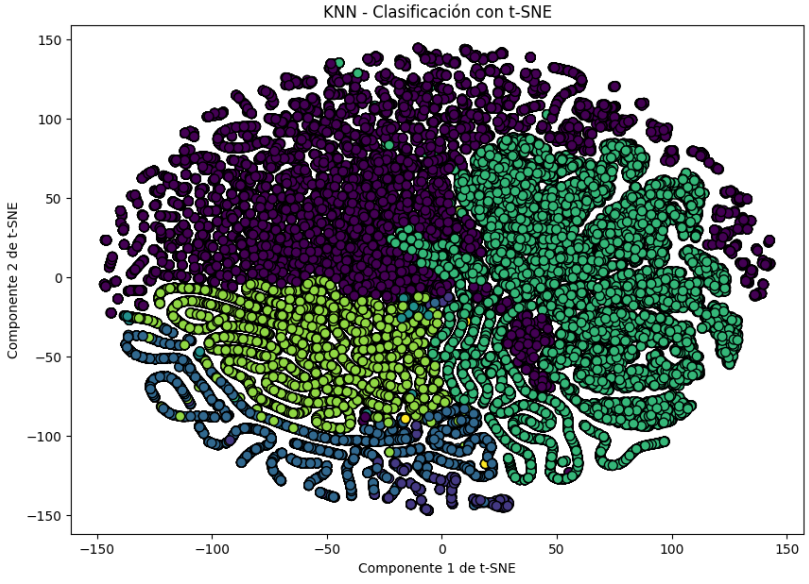


Figura 4.12: Gráfico de clasificación con t-SNE (K-Nearest Neighbors).

En el grafico PCA (Análisis de Componentes Principales) 4.13 se identifica a los vecinos más cercanos (en azul) de un punto seleccionado aleatoriamente del conjunto de validación (en rojo), proyectados en un espacio bidimensional. El punto rojo se encuentra claramente inmerso en una agrupación densa de datos de entrenamiento, lo que favorece una predicción confiable por parte del modelo. La proximidad del punto a múltiples vecinos cercanos, bien definidos y relativamente compactos, sugiere que el modelo tiene suficiente información contextual para tomar una decisión precisa. Aunque existen otras agrupaciones distantes en el espacio, el modelo opera de forma local, por lo que esas separaciones no afectan la predicción directa de este caso. Este tipo de visualización es útil para comprobar que KNN se desempeña de forma adecuada cuando los datos de entrada se sitúan en regiones bien pobladas y separadas.

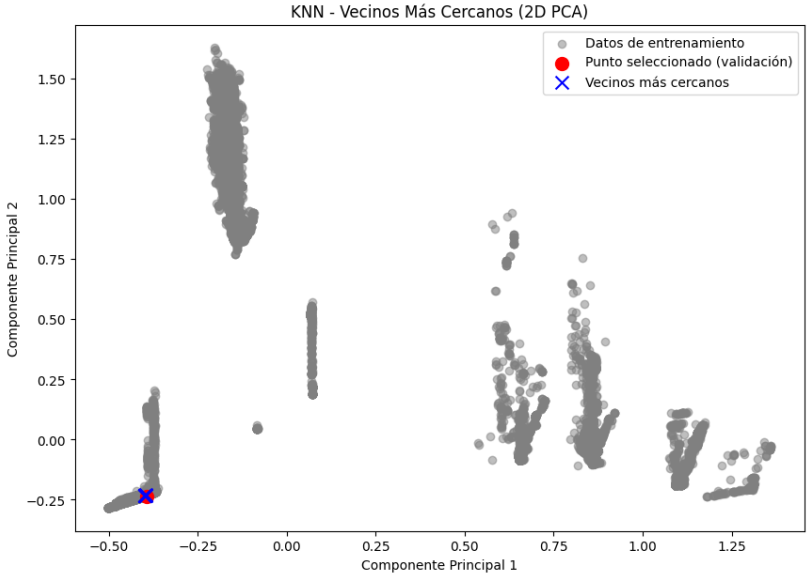


Figura 4.13: Gráfico de vecinos más cercanos con PCA (K-Nearest Neighbors).

4.3.3. Logistic Regression

En el conjunto de entrenamiento, el modelo alcanzó una precisión del 88.55 %, mostrando correcta clasificación en el trafico legitimo pero no en el malicioso. Al ser evaluado sobre el conjunto de validación, el modelo mostró consistencia con una precisión del 88.69 %, lo que resalta su capacidad para generalizar bien a datos no vistos previamente mas allá de la precisión con lo que lo hace.

La matriz de confusión mostró que clasificó correctamente la mayoría de las instancias en la clase Benigno con 41,502 predicciones correctas, erroneamente solo con 33 hacia UDP y 775 hacia UDPLag, respecto a la clase MSSQL, se identificaron correctamente 4,133 casos, con 2 instancias clasificadas erróneamente como Benign, 820 para UDP y solo 6 hacia UDPLag. Para la clase UDP, el modelo mostró 5,560 predicciones correctas, cometiendo errores al clasificar 2,355 instancias como Benign, 383 como MSSQL y solo 1 como UDPLag. Finalmente, para la clase UDPLag, el modelo identificó correctamente 4,319 instancias con 1,543 como Benign, 146 como MSSQL y 1,107 como UDP. Los resultados reflejan que, aunque el modelo es eficaz en la consistencia de la clasificación general, existen algunos casos de confusión entre clases, especialmente entre UDP y UDPLag.

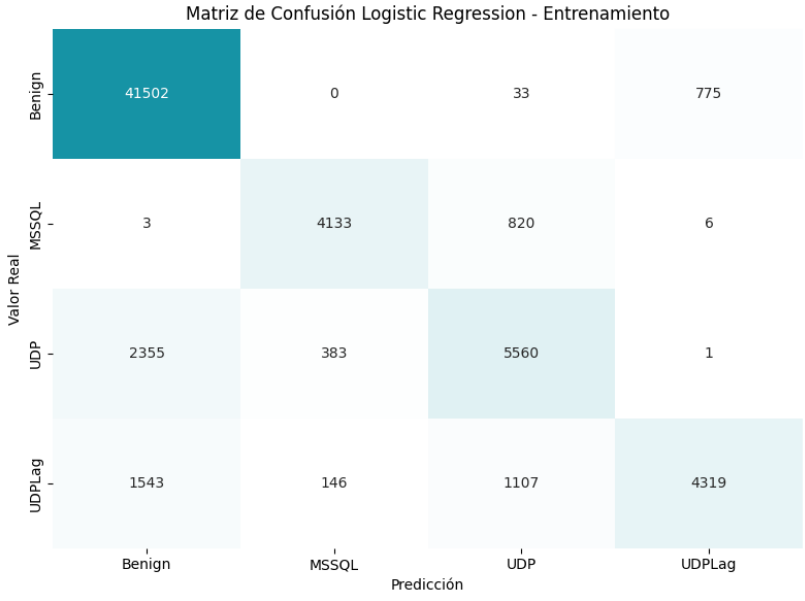


Figura 4.14: Matriz de confusión (Logistic Regression - Entrenamiento).

4.3.4. Comparación de modelos

En la evaluación comparativa de los modelos aplicados para la detección de tráfico malicioso asociado a ataques DDoS, se observó que tanto K-Nearest Neighbors (KNN) como Random Forest superaron significativamente en precisión al modelo de Regresión Logística. KNN logró un 98.99 % alcanzo un 99 % de precisión, mientras que Random Forest alcanzó un 96 % de precisión, destacando por sus capacidades de generalización y adaptación a los patrones del tráfico. Estos resultados refuerzan la robustez y confiabilidad para entornos donde la detección temprana y precisa es crítica. En contraste, aunque el modelo de Regresión Logística obtuvo un rendimiento bueno con una precisión del 88 %, su desempeño fue inferior al de los otros dos modelos, lo que sugiere una menor capacidad para capturar la complejidad de los datos en escenarios multiclase. A continuación en la tabla se muestran resumidos todos los resultados de los algoritmos entrenados.

Algoritmo	Entrenamiento		
	Precisión	**Recall**	**F1-Score**
Random Forest	0.94	0.92	0.93
Benigno	0.99	1.00	1.00
MSSQL	0.96	1.00	0.98
UDP	0.82	0.98	0.89
UDPLag	1.00	0.71	0.83
KNN	0.99	0.99	0.99
Benigno	1.00	1.00	1.00
MSSQL	0.98	1.00	0.99
UDP	0.99	0.99	0.99
UDPLag	1.00	0.98	0.99
Regresión Logística	0.85	0.77	0.80
Benigno	0.91	0.98	0.95
MSSQL	0.89	0.83	0.86
UDP	0.74	0.67	0.70
UDPLag	0.85	0.61	0.71

Tabla 4.3: Comparación de modelos Random Forest, KNN y Regresión Logística en ataques DDoS.

4.4. Hallazgos

En esta sección se presentan los hallazgos obtenidos tras el análisis de los tres tipos de ciberataques anteriores: phishing, ransomware y ataques de denegación de servicio distribuido (DDoS). Se busca revelar comportamientos comunes que ayuden a cualquier persona, incluso sin formación en informática, a reconocer indicios de posibles ciberataques en su entorno cotidiano.

4.4.1. Phishing

Uno de los hallazgos más relevantes es que los sitios web falsos diseñados para engañar a los usuarios, suelen usar enlaces web (URLs) más largos y confusos que los sitios legítimos. Aunque puedan parecer profesionales a simple vista, estas direcciones contienen nombres de dominio y subdominios excesiva-

mente extensos, lo cual es una táctica para ocultar la verdadera identidad del sitio o simular que pertenecen a una empresa confiable.

Por ejemplo un banco, suelen tener enlaces cortos y claros como:

- https://www.bancomx.mx/

En cambio una dirección falsa suele ser como:

- https://seguridad-cliente.banco.com.actualizacion.inf123.com/login

Esto no se presenta en todos los casos, siempre tenemos que tener en cuenta el como estamos accediendo a esa dirección de internet, realizando estos cuestionamientos:

- ¿Accedí desde una aplicación o sitio web oficial de la empresa u organización?
- ¿El enlace me llegó por correo, mensaje de texto o red social de alguien que no conozco o que me pareció sospechoso?
- ¿El contenido del mensaje o enlace me genera urgencia o miedo para que actúe rápido (como "tu cuenta será bloqueada", "última oportunidad", etc.)?

Tomando en cuenta estas preguntas tenemos el objetivo de generar duda, reflexionar la situación y tomar la mejor decisión. Si accedemos al enlace desde un sitio oficial de la empresa u organización si nos podremos encontrar con enlaces largos, esto no quiere decir que se trate de un ataqué, en cambio si accedemos desde un correo, mensaje o anuncio no antes visto, tenemos que tener en cuenta lo siguiente:

- ¿El correo es legitimo?
- ¿El correo se marca como 'spam'?
- ¿El contenido es de urgencia o miedo para que actúe rápido?
- ¿El anuncio es de una organización que conocemos?
- ¿El mensaje es de un numero anónimo o extraño?

Otro patrón preocupante es que muchas de estas páginas no utilizan el protocolo HTTPS, el cual indica una conexión segura (reconocible por el ícono de un candado en la barra del navegador). Pero los navegadores modernos han incorporado alertas visuales que advierten al usuario cuando una página

no es segura, siendo una herramienta útil para evitar este tipo de amenazas prestando la debida atención.

Esta figura muestra una página con el protocolo https incluido:

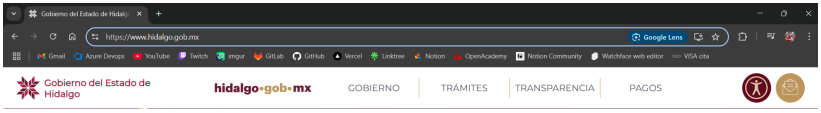


Figura 4.15: Página con protocolo Https.

Esta figura muestra una página que no contiene el protocolo https incluido e inmediatamente salta la pagina del sitio 'no seguro'.

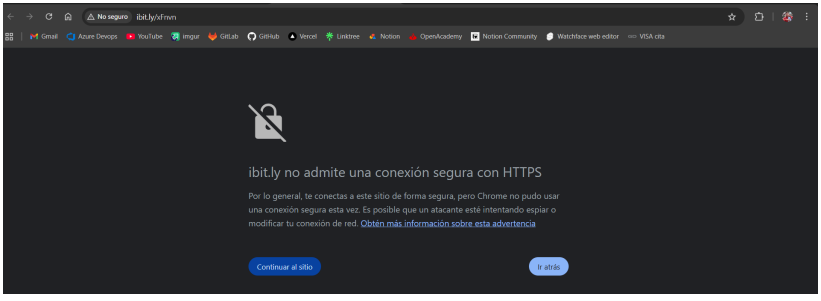


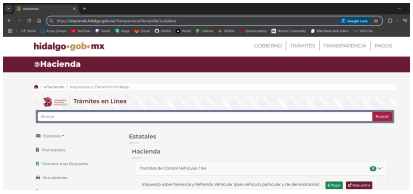
Figura 4.16: Página sin protocolo Https.

Esto tampoco quiere decir que en el protocolo HTTPS no existan paginas con engaños, según PhishLabs, para finales del primer trimestre de 2019, más del 50 % de los sitios de phishing ya utilizaban certificados SSL/TLS, alcanzando alrededor del 58 % [48]. En 2021-2022, según un reporte de Infosecurity, este porcentaje sigue creciendo (de 32 % en 2021 a más de 49 % en 2022)[49].

Se observaron comportamientos anormales en los enlaces dentro de estas páginas con phishing. Una gran cantidad conducen a páginas externas o realizan redirecciones vacías sin razón clara. Este tipo de comportamiento no es común en sitios auténticos, y representa una señal clara de posible engaño.

La siguiente figura muestra un enlace seguro en una misma página.

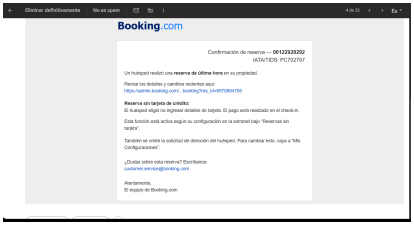
(a) Botón con enlace.



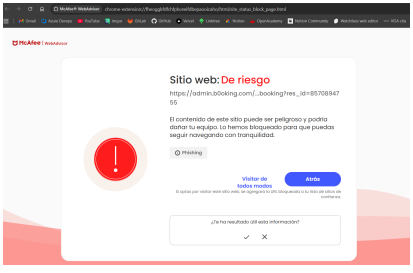
(b) Enlace seguro.

Figura 4.17: Enlace seguro en una página.

La siguiente figura muestra un correo sospechoso con un enlace no seguro a otra pagina con diferente dominio.



(a) Correo con enlace no seguro.



(b) Página con enlace no seguro.

Figura 4.18: Enlace no seguro en una página.

4.4.2. Ransomware

Una de las señales más útiles para detectar un ataque de este tipo es la alteración del tráfico normal de una red. Comúnmente, las redes académicas para usos normales mantienen un flujo estable de datos [50]. Debido a esto, los cambios repentinos o anormales siendo aumentos excesivos o disminución significativa del tráfico, puede ser una señal de advertencia temprana, pero también tenemos que tomar en cuenta:

- La institución se encuentra en hora pico.
- La institución tiene un evento.
- La institución se encuentra en periodo vacacional.

Cuando se observa que, las respuestas del servidor (tráfico de regreso) presentan tiempos irregulares o erráticos y la conexión general se mantiene estable, suele ser un patrón con comportamientos típicos de ransomware, donde el servidor malicioso responde de manera impredecible, mientras que el malware en el equipo infectado intenta mantener un flujo constante de comunicación. Esta sincronización puede estar diseñada para ocultar la actividad maliciosa dentro del tráfico normal.

También se identificaron conexiones en las que, a medida que las respuestas del servidor tardan más en llegar (mayor latencia), el flujo total del tráfico aparenta mayor estructura y menor variabilidad. Este comportamiento puede estar orientado a evadir mecanismos de detección que se basan en identificar cambios bruscos en el tráfico. Al mantener un ritmo controlado y sin picos evidentes, los atacantes buscan ocultar la comunicación con sus servidores de comando y control.

4.4.3. DoS y DDoS

El análisis del tráfico muestra que UDP y TCP son los protocolos más utilizados, lo que sugiere que posibles ataques DDoS tipo flood están aprovechando la naturaleza sin conexión de UDP. Por su parte, el uso elevado de TCP podría estar relacionado con manipulación de encabezados para evadir filtros, ocultando tráfico malicioso dentro de conexiones aparentemente legítimas 3.28.

De igual manera se observa que el 86.7 % del tráfico no presenta indicios de vulnerabilidad, indicando que el tráfico es considerado benigno, sin embargo, un 13.3 % del tráfico esta marcado con una vulnerabilidad conocida, lo que sugiere intentos específicos de explotación. Esta pequeña proporción considera escenarios de ataque encubiertos entre tráfico aparentemente legítimo 3.28.

Dentro del trafico, el 86 % no contiene la bandera para confirmar la recepción de los datos, lo cual sugiere un comportamiento anómalo. Este patrón es típico en ataques DDoS como los UDP Flood o TCP SYN Flood, donde los atacantes envían grandes cantidades de paquetes sin completar el proceso de conexión. En el caso del protocolo UDP, no se utilizan estas confirmaciones y en el caso de un SYN Flood, los paquetes se envían sin esperar respuesta, dejando conexiones incompletas.

Al analizar el rango de duración del tráfico, se observa un patrón claro: en redes donde el tráfico legítimo suele ser prolongado y constante, la presencia de conexiones inusualmente breves puede indicar actividad anómala. Los ataques muestran duraciones bajas, lo que sugiere intentos rápidos y repetitivos, típicos de ataques DDoS. Además, se detectan valores atípicos (outliers), especialmente en los ataques MSSQL y UDPLag, que podrían representar intentos de explotación prolongados o sostenidos 3.29.

Al analizar la longitud media de los paquetes por tipo de tráfico, se identifica un comportamiento diferente entre los protocolos 6 (TCP) y 17 (UDP). En el caso de TCP, el tráfico benigno presenta flujos largos y consistentes, mientras que el tráfico malicioso muestra duraciones más cortas. Por el contrario, en UDP, el tráfico legítimo tiende a ser corto pero estable, y el malicioso (como MSSQL, UDP y UDPLag) exhibe una alta variabilidad, con valores atípicos muy elevados. Esto sugiere que los ataques DDoS basados en UDP pueden generar conexiones anómalamente largas e inestables 3.30.

Se observó una correlación significativa entre las banderas RST (Reset) y PSH (Push), lo que sugiere intentos coordinados de interrumpir conexiones activas. Este patrón es típico de ataques como el TCP SYN Flood, donde los atacantes alternan entre cerrar conexiones abruptamente (RST) y enviar datos forzados (PSH), generando una carga anómala en los servidores y dificultando la gestión del tráfico legítimo.

Al aumentar el uso de protocolos como UDP, disminuye el tiempo promedio entre paquetes enviados hacia adelante. Este comportamiento refleja ataques, donde los paquetes se envían en ráfagas continuas con muy poco tiempo entre ellos, generando una carga constante sobre el sistema objetivo. Esta rapidez y regularidad en el envío de paquetes es poco común en actividades normales, lo que sugiere una alta probabilidad de automatización maliciosa.

4.5. Predicciones

Se presentan los resultados de los modelos predictores seleccionados por su alto desempeño durante su entrenamiento, con el objetivo de evaluar su capacidad de generalización ante nuevos datos. Para la detección de tráfico Phishing, el modelo CatBoost fue el elegido, al alcanzar una precisión del 99 %, en el caso del tráfico Ransomware, se optó por Random Forest, con una precisión del 98 % y por último, para el tráfico DDoS, el modelo seleccionado fue K-Nearest Neighbors (KNN), con una precisión del 99 %.

4.5.1. Phishing - CatBoost

Para la predicción de nuevos datos, en este caso URLs, se utilizaron 2,000 enlaces. Estos registros están equilibrados de forma equitativa: 1,000 URLs legítimas y 1,000 maliciosas. Tras realizar la predicción, las cuatro métricas generales (accuracy, precisión, recall y F1-score) arrojaron el mismo porcentaje, lo que evidencia la solidez y consistencia del modelo al clasificar correctamente los casos. A continuación en la tabla se muestran los resultados para las diferentes métricas.

Métrica	Valor
Accuracy	0.98
Precision (macro)	0.98
Recall (macro)	0.98
F1-score	0.98

Tabla 4.4: Métricas generales de predicción (Phishing - CatBoost).

En la siguiente tabla de visualizan los resultados desglosados por clase: URLs legítimas y URLs con phishing. En ambos casos, se observa un rendimiento consistente del modelo. Esta uniformidad en el desempeño sugiere que el modelo no favorece a una clase sobre la otra, lo cual es crucial en problemas de clasificación binaria como el análisis de URLs, donde los falsos negativos pueden representar un riesgo importante para la seguridad.

Clase	Precision	Recall	F1-score
Legítima	0.98	0.98	0.98
Phishing	0.98	0.98	0.98

Tabla 4.5: Métricas por clase de predicción (Phishing - CatBoost).

La matriz de confusión proporciona una visión más detallada sobre el desempeño del modelo, mostrando cuántos registros fueron clasificados de manera correcta e incorrecta. En el caso de las URLs legítimas, el modelo clasificó correctamente 984 instancias y cometió 16 errores. Por otro lado, para las URLs de phishing, se identificaron correctamente 978 registros, mientras que 22 fueron clasificados erróneamente.

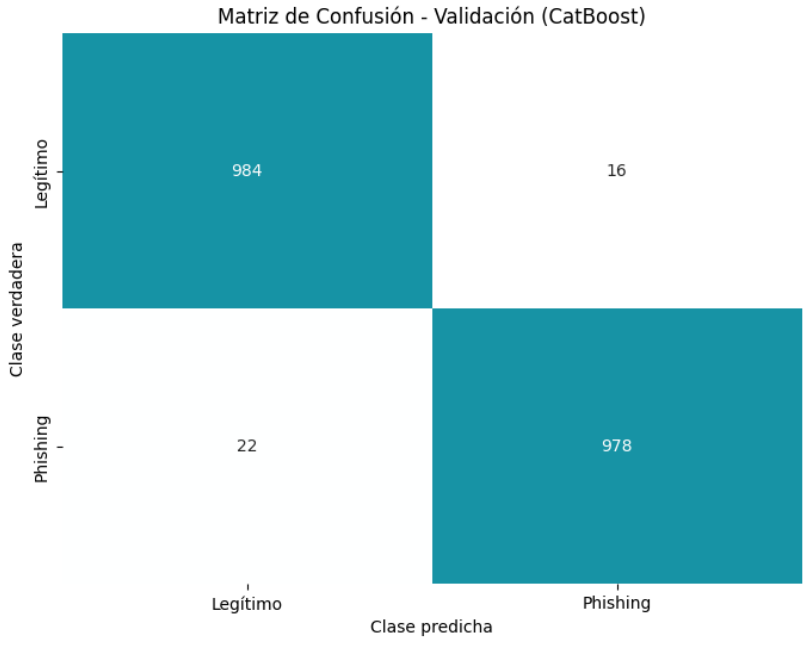


Figura 4.19: Matriz de confusión de predicciones (Phishing - CatBoost).

4.5.2. Ransomware - Random Forest

La predicción de ataques causados por ransomware fue evaluada utilizando el modelo de Random Forest. Para esta prueba se utilizaron 51,364 nuevos registros de tráfico, distribuidos de la siguiente manera: 10,832 de SVpeng, 8,618 de Benign, 7,910 de Charger, 7,268 de Simplocker, 6,540 de WannaLocker, 5,135 de Jisut y 5,061 de Lockerpin. Al realizar la predicción, las métricas generales (accuracy, precisión, recall y F1-score) arrojaron porcentajes similares y bajos. Este desempeño sugiere un posible problema en el proceso de entrenamiento del modelo. A pesar de haber obtenido una buena precisión durante la fase de entrenamiento, los resultados indican que el modelo no logra generalizar correctamente sobre nuevos datos. Esto podría deberse a la calidad o representación de los datos con los que fue entrenado. A continuación en la tabla se muestran los resultados para las diferentes métricas.

Métrica	Valor
Accuracy	0.71
Precision (macro)	0.71
Recall (macro)	0.70
F1-score	0.70

Tabla 4.6: Métricas generales de predicción (Ransomware - Random Forest).

Al analizar los resultados detallados por clase, se observa en la tabla que tres clases presentan un desempeño superior en comparación con las demás. En general, el modelo muestra un rendimiento limitado en la mayoría de las clases, con excepción de SVpeng, que además de ser la clase con mayor número de registros, fue clasificada correctamente en su mayoría. Este comportamiento sugiere que el modelo puede estar favoreciendo clases con mayor representación, o bien que logró aprender mejor los patrones asociados a ciertas clases, en detrimento de otras con menor frecuencia o características menos distintivas.

Clase	Precision	Recall	F1-score
Benign	0.83	0.93	0.88
SVpeng	1.00	1.00	1.00
Charger	0.59	0.57	0.58
Simplocker	0.62	0.55	0.59
WannaLocker	0.66	0.64	0.65
Jisut	0.61	0.62	0.61
Lockerpin	0.68	0.47	0.55

Tabla 4.7: Métricas por clase de predicción (Ransomware - Random Forest).

Según la matriz de confusión, se logró clasificar correctamente un total de 37,052 registros, mientras que 6,559 fueron clasificados de forma errónea. Se observa un buen desempeño del modelo al identificar correctamente clases como Benign, SVpeng, Charger, Simplocker y WannaLocker. Sin embargo, también se evidencian errores significativos en las clases Simplocker y WannaLocker, lo

cual indica cierta debilidad del modelo para distinguir adecuadamente entre algunas categorías, afectando negativamente su capacidad general de clasificación.

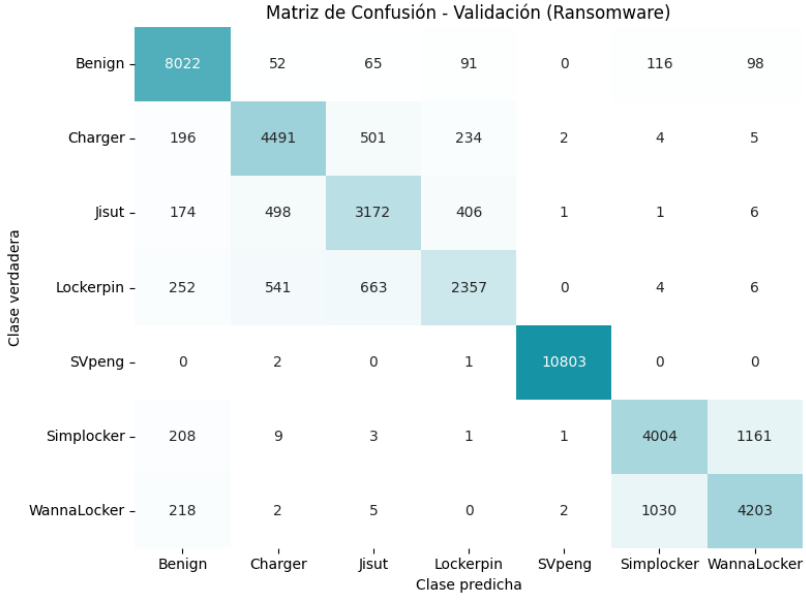


Figura 4.20: Matriz de confusión de predicciones (Ransomware - Random Forest).

4.5.3. DDoS - k-Nearest Neighbors (KNN)

El trafico nuevo de los ataques DDoS fue evaluado utilizando el modelo de K-Nearest Neighbors. Para esta prueba se emplearon 15,672 nuevos registros, distribuidos de la siguiente manera: 10,544 de Benign, 2,121 de UDP, 1,757 de UDPLag y 1250 de MSSQL.

Después de la predicción, las métricas generales (accuracy, precisión, recall y F1-score) arrojaron porcentajes altos y similares, manteniendo una correcta clasificación respecto al entrenamiento. El desempeño refleja claramente la importancia de la calidad de datos y diferencias respecto a cada tipo de ataques. A continuación en la tabla se muestran los resultados para las diferentes métricas.

Métrica	Valor
Accuracy	0.96
Precision (macro)	0.92
Recall (macro)	0.92
F1-score	0.92

Tabla 4.8: Métricas generales de predicción (DDoS - K-Nearest Neighbors).

Los resultados mostrados en la tabla se presentan consistentes, con métricas altas en la mayoría de los casos. La única excepción se encuentra en la clase UDPLag, que presenta una leve disminución en la métrica de recall. A pesar de ello, las demás clases mantienen valores superiores a 0.80 en las métricas evaluadas. En términos generales, el rendimiento del modelo es sólido, con especial eficacia en ciertas clases, lo que sugiere una capacidad robusta para identificar ataques de tipo DDoS.

Clase	Precision	Recall	F1-score
Benign	1.00	1.00	1.00
MSSQL	0.96	0.97	0.97
UDP	0.83	0.90	0.87
UDPLag	0.89	0.79	0.83

Tabla 4.9: Métricas por clase de predicción (DDoS - K-Nearest Neighbors).

La matriz de confusión revela un total de 15,054 registros correctamente clasificados y 618 clasificados de forma incorrecta. Se observa una clasificación precisa para las clases Benign y UDP. Sin embargo, los errores más significativos se presentan en la clase UDPLag, donde 360 registros fueron erróneamente clasificados como UDP, lo que indica una confusión frecuente del modelo entre estos dos tipos de tráfico.

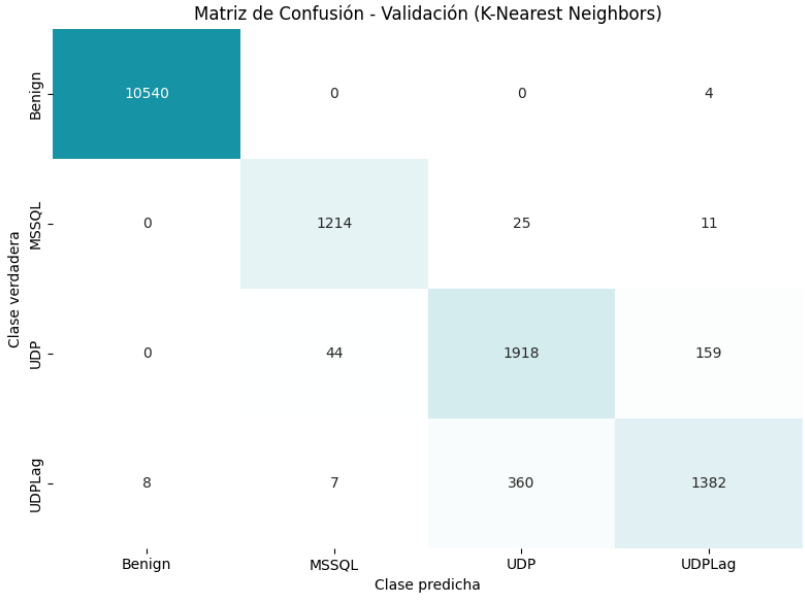


Figura 4.21: Matriz de confusión de predicciones (DDoS - K-Nearest Neighbors).

Capítulo 5

Conclusiones

El análisis y evaluación de ataques cibernéticos realizados en este estudio permitieron cumplir con el objetivo general de identificar patrones comunes de amenazas digitales que afectan a la comunidad estudiantil y al público en general. A través del uso de algoritmos de aprendizaje automático y del procesamiento de bases de datos públicas y académicas especializadas, se lograron detectar comportamientos característicos de ataques como el phishing, el ransomware y los ataques de denegación de servicio (DoS/DDoS).

Los resultados obtenidos demuestran que los modelos de machine learning aplicados pueden alcanzar altos niveles de precisión en la clasificación de amenazas cibernéticas, destacando el desempeño de algoritmos como SVM (99.9 % en phishing), XGBoost (99.83 % en ransomware) y Random Forest (99.34 % en DDoS). Estos resultados evidencian el potencial de la inteligencia artificial como una herramienta eficaz para mejorar la ciberseguridad y apoyar la detección de ataques.

Además, se identificaron áreas de mejora relacionadas con la confusión entre variantes similares de amenazas y la dificultad para clasificar ataques menos representados en los datos, lo que sugiere la necesidad de utilizar conjuntos de datos más balanceados y estrategias avanzadas de optimización.

Este trabajo ofrece una base técnica que puede ser aprovechada en desarrollos futuros orientados a la educación en ciberseguridad y la creación de herramientas preventivas. Los resultados muestran la importancia de fomentar una cultura digital más segura e informada, en la que el conocimiento técnico, combinado con soluciones inteligentes, puede desempeñar un papel clave en la mitigación de amenazas cibernéticas actuales.

Como trabajo a futuro de esta investigación, se plantea desarrollar una plataforma web educativa que facilite la difusión accesible y clara de información

sobre las amenazas cibernéticas identificadas en previo análisis. Esta investigación sentó las bases fundamentales para la creación de dicha plataforma, que no solo tendrá como objetivo informar y concienciar al público sobre los riesgos cibernéticos, sino que también incorporará un predictor de phishing. Esta herramienta permitirá a los usuarios ingresar URLs para verificar si contienen phishing, además de proporcionar recomendaciones prácticas para evitar este tipo de ataques. La implementación de esta funcionalidad fortalecerá el propósito de ofrecer recursos accesibles y efectivos para la prevención de ciberataques, consolidando el sitio web como una fuente confiable, dinámica y útil en materia de seguridad cibernética.

Bibliografía

[1] G. Florencia, "Los ciberataques aumentaron un 50 % en 2021 y alcanzaron peak histórico en diciembre - ITSitio." Section: Seguridad.

[2] gmcdouga, "Check point research warns every day is a school day for cyber criminals with the education sector as the top target in 2024."

[3] Y. K. Saheed and M. O. Arowolo, "Efficient cyber attack detection on the internet of medical things-smart environment based on deep recurrent neural network and machine learning algorithms," vol. 9, pp. 161546–161554.

[4] K. Rahman, M. A. Aziz, N. Usman, T. Kiren, T. A. Cheema, H. Shoukat, T. K. Bhatia, A. Abdollahi, and A. Sajid, "Cognitive lightweight logistic regression-based IDS for IoT-enabled FANET to detect cyberattacks," vol. 2023, pp. 1–11.

[5] A. Delplace, S. Hermoso, and K. Anandita, "Cyber attack detection thanks to machine learning algorithms."

[6] A. Almalaq, S. Albadran, and M. Mohamed, "Deep machine learning model-based cyber-attacks detection in smart power systems," vol. 10, no. 15, p. 2574.

[7] "Kaggle: Your Machine Learning and Data Science Community."

[8] "DDoS 2019 | datasets | research | canadian institute for cybersecurity | UNB."

[9] A. Sarmiento, "Digitalización en méxico: empresas avanzan, pero con brechas." Section: Notícias.

[10] Creze, "80 % de las PyMEs en méxico ya son parte de la digitalización -."

[11] J. Bravo, "Pymes en méxico: urge su digitalización."

[12] E. Demos and C. A. García, “La jornada - nueve de cada 10 ciberdelitos podrían prevenirse, señalan especialistas.” Section: Sociedad.

[13] UNICEF, “Mantener seguros a niñas, niños y adolescentes en internet | UNICEF.”

[14] E. ECONOMISTA, “En aumento, ciberataques a instituciones públicas.”

[15] kaspersky, “Más de la mitad de las PyMEs de AL reporta incremento en ciberataques, pero 20 % no está preparada para enfrentarlos.”

[16] “Algoritmos de aprendizaje automático | microsoft azure.”

[17] I. Belcic and C. Stryker, “¿qué es el aprendizaje supervisado? | IBM.”

[18] IBM, “What is a decision tree? | IBM.”

[19] MathWorks, “Introducción a support vector machines (SVM).”

[20] Daniel, “CatBoost: Una herramienta esencial para el machine learning.”

[21] IBM, “¿qué es XGBoost? | IBM.”

[22] AWS, “¿qué es la regresión logística? - explicación del modelo de regresión logística - AWS.”

[23] “¿qué es el aprendizaje no supervisado? | IBM.”

[24] E. Kavlakoglu and V. Winland, “¿qué es la agrupación en clústeres k-means? | IBM.”

[25] IBM, “¿qué es el algoritmo de k vecinos más cercanos? | IBM.”

[26] J. Patiño, “Más de 13 millones de víctimas por fraudes cibernéticos en méxico.” Section: México.

[27] E. Kaspersky, “Estos son los grandes retos de ciberseguridad para las pequeñas empresas.”

[28] G. Ana, “El sector educativo y de investigación es el más atacado con 2.256 ciberataques semanales por organización - Éxito educativo.” Section: Actualidad directiva.

[29] S. Alba, “Digitalización de la educación también impacta en el crecimiento de los ciberataques.”

[30] S. Infochannel, “México top 4 mundial en ciberataques vs sector educativo.”

[31] Alejandro Gonzales, "México, por encima del promedio mundial de ciberataques: Check point." Section: Ciberseguridad.

[32] V. Analitik, "El 57 % de las pymes en latinoamérica reporta un incremento de ciberataques: ¿cómo prepararse para enfrentarlos?." Section: Noticias empresariales.

[33] J. García, "Estadísticas de ciberseguridad: Pronóstico para el 2024."

[34] Redacción, "México registró 119 millones de intentos de phishing en 2024."

[35] Redacción, "México dentro de los 15 países más ciber atacados por medio de emails."

[36] H. Hernandez, "El ciberataque a bimbo y el ransomware medusa - capital software."

[37] G. CLULEY, "The phishing swindle that conned $100 million out of google and facebook."

[38] Articulo19, "Ataque DDOS a revista espejo, segundo medio atacado en sinaloa este mes – ARTICLE 19 MX-CA."

[39] F. Fernández, "Estudio experimental de ciberataques a través de códigos QR," p. 40.

[40] J. Alsamiri and K. Alsubhi, "Internet of things cyber attacks detection using machine learning," vol. 10, no. 12.

[41] M. Panda, A. A. A. Mousa, and A. E. Hassanien, "Developing an efficient feature engineering and machine learning model for detecting IoT-botnet cyber attacks," vol. 9, pp. 91038–91052.

[42] Muhammad Arslan Ajmal, Muhammad Imran, Muhammad Asif Raza, and Ali Raza, "Cyber threats prediction model using advanced data science approaches," vol. 3, no. 2.

[43] H. Karimipour, A. Dehghantanha, R. M. Parizi, K.-K. R. Choo, and H. Leung, "A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids," vol. 7, pp. 80778–80788.

[44] A. Nusret Özalp and Z. Albayrak, "Detecting cyber attacks with high-frequency features using machine learning algorithms," vol. 19, no. 7, pp. 213–233.

[45] S. Tiwari, “Phishing dataset for machine learning.”

[46] Cyber Cop, “Android ransomware detection.”

[47] I. Sharafaldin, S. Hakak, A. H. Lashkari, and G. Ali, “DDoS 2019 | datasets | research | canadian institute for cybersecurity | UNB.”

[48] J. Ryan, “More than half of phishing sites now use HTTPS | PhishLabs.”

[49] P. Muncaster, “Volume of HTTPS phishing sites surges 56 % annually.”

[50] J. A. L. Moreno, “Implementación de análisis de tráfico y de flujos de red con tecnologías netflow y sflow en equipos de red de la unam utilizando software libre,” tesis de maestría, Universidad Nacional Autónoma de México, Facultad de Ingeniería, Ciudad de México, México, 2019.