FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

Desarrollo de un método de identificación automática de ataques spoofing de envenenamiento Arp en la suplantación de identidad en redes lan

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS

Autor (es)
Bach. Urrutia Vasquez Miguel
https://orcid.org/0000-0001-9533-1977

Bach. Julca Rojas Alex Rogelio

https://orcid.org/0000-0001-8942-6566

Asesor(a)
Dr. Ing. Mario Fernando Ramos Moscol
https://orcid.org/0000-0003-3812-7384

Línea de Investigación
Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú
2024

DESARROLLO DE UN MÉTODO DE IDENTIFICACIÓN AUTOMÁTICA DE ATAQUES SPOOFING DE ENVENENAMIENTO ARP EN LA SUPLANTACIÓN DE IDENTIDAD EN REDES LAN

Aprobación del jurado

_____

DR. TUESTA MONTEZA VICTOR ALEXCI

Presidente del Jurado de Tesis

_____

MG. ALVA ZAPATA JULIANA DEL PILAR

Secretario del Jurado de Tesis

_____

MG. ASENJO CARRANZA ENRIQUE DAVID

Vocal de Jurado

Reporte de similitud

NOMBRE DEL TRABAJO

TESIS _MIGUEL URRUTIA- FINAL TURNIT IN - MIGUEL URRUTIA VASQUEZ.docx

AUTOR

MIGUEL URRUTIA VASQUEZ

RECUENTO DE PALABRAS

14817 Words

RECUENTO DE CARACTERES

80794 Characters

RECUENTO DE PÁGINAS

85 Pages

TAMAÑO DEL ARCHIVO

2.0MB

FECHA DE ENTREGA

Jul 11, 2024 8:10 PM GMT-5

FECHA DEL INFORME

Jul 11, 2024 8:11 PM GMT-5

7% de similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 6% Base de datos de Internet
- 1% Base de datos de publicaciones
- Base de datos de Crossref
- Base de datos de contenido publicado de Crossref
- 3% Base de datos de trabajos entregados

Excluir del Reporte de Similitud

- Material bibliográfico
- Material citado
- Coincidencia baja (menos de 8 palabras)

DECLARACIÓN JURADA DE ORIGINALIDAD

Quien(es) suscribe(imos) la **DECLARACIÓN JURADA**, soy(somos) Miguel Urrutia Vasquez del Programa de Estudios de Ingeniería de Sistemas .de la Universidad Señor de Sipán S.A.C, declaro (amos) bajo juramento que soy (somos) autor(es) del trabajo titulado:

DESARROLLO DE UN MÉTODO DE IDENTIFICACIÓN AUTOMÁTICA DE ATAQUES SPOOFING DE ENVENENAMIENTO ARP EN LA SUPLANTACIÓN DE IDENTIDAD EN REDES LAN

El texto de mi trabajo de investigación responde y respeta lo indicado en el Código de Ética del Comité Institucional de Ética en Investigación de la Universidad Señor de Sipán (CIEI USS) conforme a los principios y lineamientos detallados en dicho documento, en relación a las citas y referencias bibliográficas, respetando al derecho de propiedad intelectual, por lo cual informo que la investigación cumple con ser inédito, original y autentico.

En virtud de lo antes mencionado, firman:

Urrutia Vasquez Miguel	75482511	
Julca Rojas Alex Rogelio	76563141	

Pimentel, 24 de setiembre de 2024.

DEDICATORIAS

A DIOS,

Por ser mi guía, brindarme salud y permitirme lograr mis objetivos, la cual me llena de mucha alegría haber cumplido mis sueños, para ser mejor cada día.

A MI PADRE,

Jhony, por haberme brindado sus apoyo y consejos en todo momento, motivándome constantemente para cumplir mis metas.

A MI HIJO,

Fabiano, ya que es el motor de mi vida, quien me da fuerzas para seguir adelante.

A MIS ABUELOS,

Bercelia y Segundo por haberme brindado su apoyo incondicional para formarme profesionalmente.

A LA MEMORIA DE

Mi Madre Marlene y mis hermanos Frank y Eduardo.

AGRADECIMIENTOS

Agradecer a mi familia, por haberme brindando la oportunidad de poder formarme en esta acreditada Universidad y ser mi apoyo durante todo este tiempo de vida universitaria.

De manera especial agradecemos a nuestro tutor de tesis Ing. Heber Mejía Cabrera, a nuestro asesor de tesis Dr. Ing. Mario Fernando Ramos Moscol por su apoyo y por habernos guiado durante el periodo de investigación.

A la Universidad Señor de Sipán por habernos brindado tantas oportunidades y enriquecernos en conocimiento.

Resumen

En la actualidad los ataques de intermediario son una amenaza constante, la cual viene ser preocupante para los usuarios que conforman una red de área local, siendo relevante para a seguridad informática. Este tipo de ataques de spoofing, tiene como principal objetivo perjudicar la privacidad de usuarios que se encuentre dentro de una red, ya sea doméstica o empresarial, puesto que es un problema que estaría provocando pérdidas financieras, por lo que estaría afectando la integridad de las personas, debido a que la información es leída o modificada, cuando el atacante logra interceptar el flujo de datos, afectando la confidencialidad de las posibles víctimas. Aprovechando la vulnerabilidad del protocolo de resolución de direcciones (ARP), ya que es un protocolo utilizado por la capa 2 de enlace a datos, que sirve para añadir direcciones IP a dirección de acceso a medios (MAC), por lo que acepta respuestas ARP sin verificar si se ha enviado una petición ARP. Este trabajo de investigación utiliza el aprendizaje automático y procesamiento de señales ARP extraídos directamente de un capturador de paquetes, cuyo contexto nace con el propósito de crear un mecanismo detección de ataques de intermediario, la cual se compone de 4 fases, siendo la primera fase de selección donde se evaluó los algoritmos que obtuvieron mejor precisión al clasificar ataques de spoofing ARP, en la segunda fase se realizó la creación del conjunto de datos la cual se obtuvo un total de 2285 registros, con la recopilación de la captura de datos, de donde se obtuvo información de paquetes atacados y paquetes limpios, en la tercera fase se implementó 6 algoritmos siendo entre ellos, Vecinos más cercanos, bosque aleatorio, SVC, regresión logística, árbol de decisión y aumento de gradiente, la cual se evaluó el desempeño de los 6 algoritmos a través de las métricas de rendimiento. Para el desarrollo del método propuesto, se tomaron el 70% de los registros en entrenamiento y el 30% restante en pruebas, los resultados demostraron que el clasificador Random Forest alcanzó el 99,32 % de precisión, por lo tanto, esta técnica es la más adecuada para clasificar los ataques de spoofing ARP.

Palabras Clave: Aprendizaje de máquina, Conjunto de datos, Suplantamiento ARP, Red de área local, Seguridad informática.

Abstract

Nowadays, man-in-the-middle attacks are a constant threat, which is a concern for users that make up a local area network, being relevant for computer security. This type of spoofing attack has the main objective of damaging the privacy of users within a network, whether domestic or business, since it is a problem that was causing financial losses, which would be affecting the integrity of the networks. people, because the information is read or modified, when the attacker manages to intercept the flow of data, affecting the confidentiality of the possible victims. Taking advantage of the vulnerability of the Address Resolution Protocol (ARP), since it is a protocol used by the data link layer 2, which is used to add IP addresses to the media access address (MAC), therefore it accepts ARP responses without checking if an ARP request has been sent. This research uses machine learning and processing of ARP signals extracted from a packet capturer, to later create a mechanism for detecting intermediary attacks, it is composed of 4 phases, the first phase of selection being where the algorithms obtained were evaluated. better precision when classifying ARP spoofing attacks, in the second phase the creation of the data set was carried out, from which information was obtained on attacked packets and clean packets, in the third phase 6 algorithms were implemented, among them, Nearest neighbors, random forest, SVC, logistic regression, decision tree and gradient increase, for this the performance of the 6 algorithms was evaluated through performance metrics, for this the results showed that the Random Forest classifier reached 99.32% precision, therefore this technique is the most suitable for classifying ARP spoofing attacks.

Keywords: Machine learning, Data set, Computer security, ARP spoofing.

ÍNDICE

I. INTRODUCCIÓN

1.1. Realidad Problemática.

En los últimos años los ataques a las redes de área local (LAN) de usuarios perjudica gradualmente la privacidad, aprovechando la vulnerabilidad del protocolo de resolución de direcciones (ARP), obtienen el tráfico de paquetes mediante suplantación de identidad permitiendo alterar la información de los usuarios para fines maliciosos, como ejemplo los ataques directos a causa de configuraciones inseguras, permitiendo el robo de información e interrupciones de sus operaciones que a corto o largo plazo, perjudican no solo la integridad de una persona o empresas, sino que también afecta en el ámbito financiero. (Scott, et al, 2017)

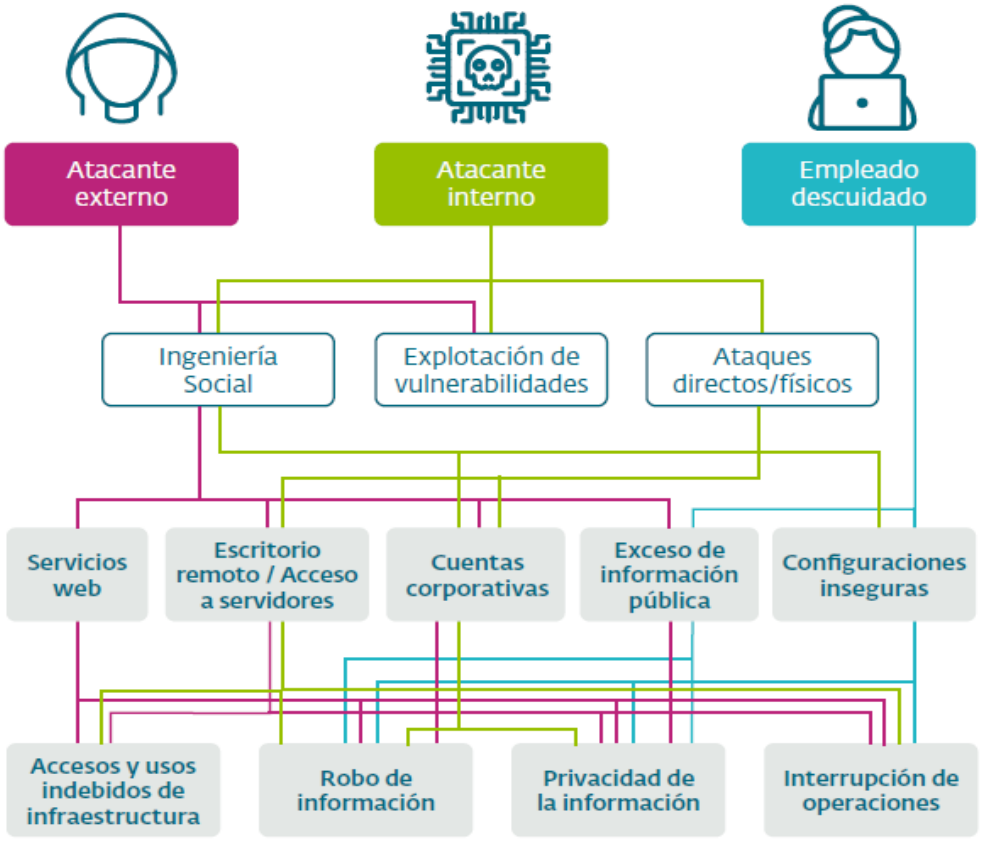


Figura 1. Tipo de atacantes, entre ellos el atacante interno que vulnera información de usuarios dentro de una red local.
Fuente: (Eset, 2020)

A nivel mundial el internet permite el intercambio de datos entre usuarios, de manera que genera diferentes riesgos cibernéticos perjudicando el correcto uso de información en distintos sectores. (Rohatgi & Shimpy, 2020)

Los ataques spoofing tienen como enfoque vulnerar los recursos de un sistema, afectando a muchos usuarios de una red, comprometiendo su información y generando pérdidas financieras tanto a clientes como empresas de comercio electrónico. (Ríos, et al, 2021)

Por otro lado la suplantación o robo de identidad hace referencia a phishing, que constantemente son utilizadas para extraer información importante de usuarios que pertenecen a una red, es por ello que los atacantes buscan vulnerar protocolos y hacerse pasar por su víctima para lograr sus objetivos, a causa de ello a corto o largo plazo genera diferentes tipos de problemas, como pérdidas de dinero, generando desestabilidad financiera a causa de movimientos inusuales en cuentas bancarias, según (Eset). Por otro lado, en Latinoamérica durante el 2018 a diario las cifras por ataques demostraron unas cifras relativamente elevadas con un total de 746.000 ataques, en vista de que los ataques phishing a través de correos electrónicos aparentemente confiables, pero camuflados usurparon la información de las víctimas con fines maliciosos. (kaspersky, 2018)

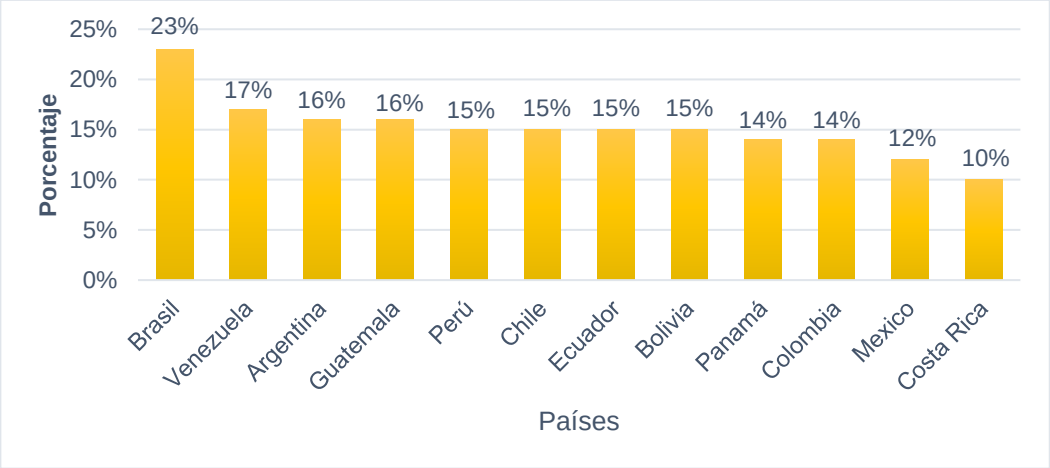


Figura 2. Clasificación de países Latinoamericanos con mayor porcentaje ante ataques Phishing durante enero a julio del 2018.
Fuente: (kaspersky, 2018)

Se puede mencionar además, en Latinoamérica se ha incrementado los ciberataques durante el 2020, a consecuencia de la pandemia Covid-19 las cifras de ataques son de mayor rango afectando gran mayoría de instituciones empresariales, es por ello que recomienda que los diferentes sectores tales como privados o públicos tengan mayor relación con la tecnología, en base a ello reducirán los riesgos de ataques que son peligro constante del día a día, dicho lo anterior la seguridad cibernética es de gran ayuda, sin minimizar la importancia que conlleva a reducir los delitos informáticos. (BID; OEA, 2020)

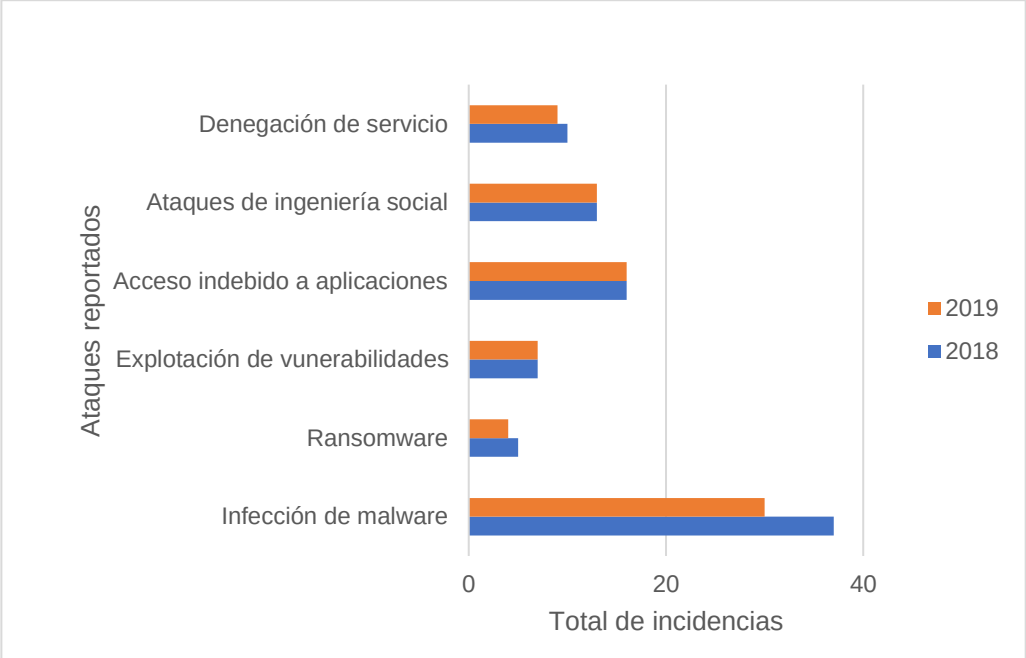


Figura 3. Tipos de ataques reportados por organizaciones.
Fuente: (Eset, 2020)

Todavía cabe señalar que las empresas tienen problemas con el presupuesto que debería ser destinado para invertir en un área de seguridad, para ser más específicos se quejan con mayor frecuencia. Sin embargo estas observaciones no son ejecutadas por las organizaciones con la finalidad de reducir los riesgo de pérdidas financieras, es así que en 2019 se obtuvo un porcentaje de 75% de las empresas que indicaron que su principal malestar es el poco presupuesto, cabe mencionar que los porcentajes que

se obtuvieron son el resultado que refleja peligro para las organizaciones, por consiguiente deberían obtener alternativas para implementar proyectos en seguridad, de manera que se esfuercen para obtener resultados óptimos en base a su tiempo y recursos invertidos. (Eset, 2020)

En el Perú, según un estudio realizado por las distintas empresas nacionales demostraron que el 27% consideran importante la ciberseguridad en sus planificaciones de desarrollo empresarial, aunque un 51% indicaron que no era necesario incluir la ciberseguridad en sus empresas, no obstante se demostró que el 70% sufrieron de incidencias de ataques cibernéticos, a pesar de que la gran mayoría es afectado por la ciberdelincuencia, el 8% confían de medidas para contrarrestar los ataques, por consiguiente es revelado que solo el 59% de empresas no cuenta con un encargado de ciberseguridad para que realice reportes a la gerencia (EY Perú, 2020), para concluir según reportes realizados, Perú no tiene creado aun estrategias de seguridad a nivel nacional, por consiguiente ya tienen en movimiento una política de seguridad cibernética, no obstante el país si cumple con componentes precisos para lidiar con las amenazas en el mundo no físico. (BID; OEA, 2020)

Un método que actualmente se usa en la ingeniería son el empleo de herramientas que permiten identificar tráfico inusual de datos. Una solución sería la creación de Caché ARP estáticas para almacenar la dirección de control acceso a medios de usuarios (MAC) conectado a la red. Sin embargo, este método no sería ideal para una red que trabaja con el protocolo de direcciones dinámicas. (Brandon, et al, 2017)

En contraste con lo anterior a causa del crecimiento de ataques informáticos buscan alternativas para minimizar los riesgos, ya que los antivirus y firewall no tienen un ritmo de acuerdo a los nuevos ataques, debido a que la detección de intrusos no es muy efectiva contra ataques de red muy grandes en cuanto a los tiempos de ejecución, los datos que se utilizan no tienen una precisión muy segura y los datos a evaluar no tienen una aproximación a los

datos que se realizan. Por lo que proponen usar el aprendizaje automático en diferentes campos de seguridad para detectar los atacantes, previamente entrenado un conjunto de datos con bastante información recopilada para identificar cual es la que corresponde a un ataque y cual no, indicando que no es peligro, mediante el algoritmo de K vecinos y subdivisión de zonas la cual usan criterios de cercanía y datos verificados en el espacio en el que se están trabajando. (Enciso, Salcedo, & Upegui, 2019)

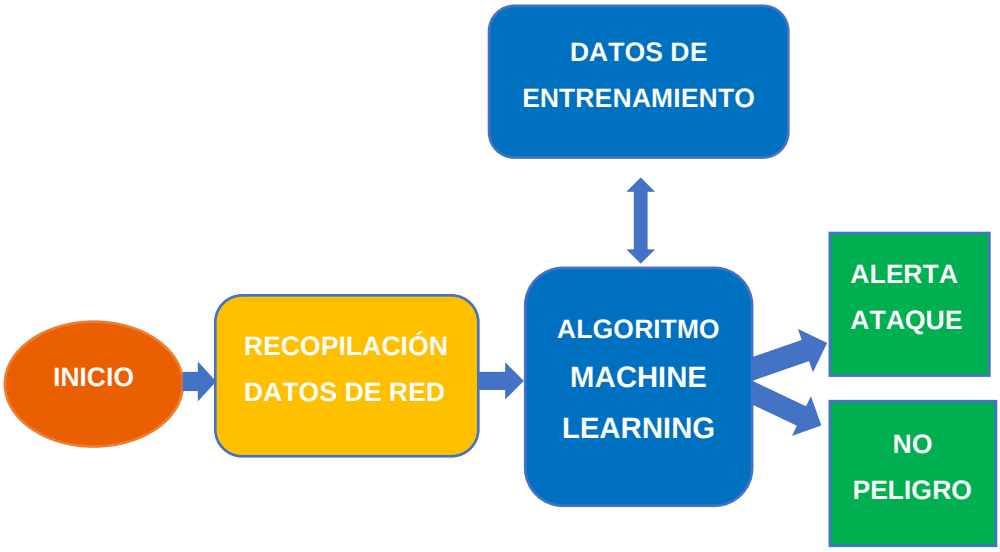


Figura 4. Representación gráfica del funcionamiento de software ante la detección de ataques arp.
Fuente: (Enciso, Salcedo, & Upegui, 2019)

1.2. Trabajos previos.

Mehak, Komal, Ghufran, & Shahbaz (2022)**,** realizó investigación *Predicting ARP spoofing with Machine Learning* en Karachi, Pakista. Los ciberataques en la red son una amenaza, y un desafío para los dispositivos que se encuentran conectados a la red ante el ataque Man in the Middle, por esta razón, presentaron un modelo de predicción que consta que consta de dos variantes, entre ellas se encuentran la técnica de aprendizaje profundo (LSTM) y la técnica de aprendizaje automático la cual es Árbol de decisiones para la predicción de la suplantación de identidad de ARP, evaluándose a través de un conjunto de datos el rendimiento de los modelos. . Los resultados obtenidos del modelo propuesto alcanzaron el 99.9 % por LTSM

y el 100% por la técnica de arbol de decisión. La técnica de árbol de decisiones demostró ser eficiente en cuanto a la evaluación de ejecución.

Ahuja, Singal, Mukhopadhyay , & Nehra , (2022), realizó investigación Ascertain the efficient machine learning approach to detect different ARP attacks en India. Las redes definidas por software son una arquitectura de red programable que controla dispositivos de red, pero son vulnerables a ataques tradicionales como la suplantación de identidad del protocolo de resolución de direcciones (ARP). Por esta razón, desarrollaron un sistema que recopila y registra las funciones necesarias para detectar un ataque a través de un archivo, la cual viene a ser el conjunto de datos del trafico capturado. Los resultados obtenidos del modelo propuesto alcanzaron una puntuación de precisión de 99.73 % durante el ataque, usando el 97% de la CPU y un alto uso de la memoria. El modelo propuesto demostró ser eficiente en cuanto al tiempo al momento de la detección de ataques.

Prasad & Chandra, (2022), realizó la investigación Defending ARP Spoofing-based MitM Attack using Machine Learning and Device Profiling en Babasaheb Bhimrao Ambedkar University. A través de la suplantación de identidad ARP (Protocolo de resolución de direcciones), un intruso lanza un ataque Man-in-the-middle (MitM), obteniendo el acceso para la manipulación de todo el tráfico de red entrante y saliente desde el dispositivo víctima. Por esta razón presentaron una técnica dual basándose en aprendizaje automático y creación de perfiles de dispositivos para detectar los ataques MitM en la suplantación de identidad ARP, la cual este enfoque basado en aprendizaje automático examina en la red la presencia de ataques suplantación de identidad de arp. Lo resultados obtenidos de la técnica propuesta alcanzaron el 99.4 de precisión usando métodos de clasificación automática.

Jerry John, Justice Owusu, & Griffith, (2020), realizó la investigación, *Detecting End-Point (EP) Man-In-The-Middle (MITM) Attack based on ARP, en la University of Science and Technology*, Ghana. Un atacante puede

vulnerar la red fácilmente y hacerse pasar por un Gateway de un conmutador para escuchar paquetes con fines maliciosos afectando la integridad de datos de la víctima. Por esta razón, presentó un método basado en la detección de los ataques que realiza un hombre del medio (MITM), para estimar la precisión, integrando dos técnicas de análisis que son procesamiento de señales y aprendizaje automático. Los resultados obtenidos alcanzaron un 99,72% de precisión usando métodos de clasificación basado en líneas. Machine Learning ha logrado ganar mucha popularidad en el análisis masiva de datos, en cuanto a la precisión en tiempo reducido, por lo que es importante incluir en la detección de ataques MitM en cualquier red.

(Al-Hababi & Sezer C, 2020), realizó la investigación, Man-in-the-Middle Attacks to Detect and Identify Services in Encrypted Network Flows using Machine Learning, en Universidad de Texas. Los redes inalambricas se han incrementado a medida que las personas requieren un servicio de acceso a internet de calidad por el uso de las redes sociales, no obstante se cataloga como trafico de red cifrado, por lo que un atacante puede interceptar y vulnerar los paquetes. Por esta razón, presentaron un experimento de laboratorio del ataque del hombre de en medio,que rastrea el trafico de su victima, por lo que se aplica los algoritmos de aprendizaje automatico para clasificar a gran escala los datos confusos de aplicaciones de redes sociales. El metodo propuesto no evidencia definiciones numéricas de lo desarrollado, sin embargo, se basa en clasificar las aplicaciones de redes sociales de los usuarios conectados a una red.

Nicolas Ricardo, Octavio, & Upegui, (2019), realizó la investigación, *Arp Attack Detection Software Poisoning and Sniffers in WLAN Networks Implementing Supervised Machine Learning*, en Universidad Nacional de Colombia. En la actualidad los ataques en la red se han ido incrementando, aprovechando las deficiencias del protocolo Arp. Por esta razón, desarrollaron una herramienta que permite detectar los ataques de un hacker, basándose en la identificación automática a través de un algoritmo

previamente entrenado con datos que ya han sido vulnerados, permitiendo así clasificar para luego notificar como amenaza. Los resultados obtenidos muestran que la efectividad es del 60%, después de realizar 20 pruebas de ataque, solo 14 funcionaron correctamente. El aprendizaje automático demuestra ser una herramienta potencial para predecir los ataques de intermediario.

Vijayakumar,et al, (2020), realizó la investigación, *A two-way approach for detection and prevention of IP spoofing attacks*, en la India. Las direcciones de protocolo de internet (IP), son vulnerables a ataques de suplantación, debido a los sistemas que provienen de los servicios de internet, eso trae consecuencias de pérdida de datos, falsificación, interferencia de red y mutilación. Por esta razón, presentó el método que propone el uso de agrupaciones K-Medias (K-MEANS clustering), que asocia archivos conforme a la función de distancia euclidiana mínima hacia el grupo que está más cercano que está basado en centroides y máquinas de vectores de soporte (SVM), que se encarga de clasificar fragmentos en distintas clases obtenidos por una API. Los resultados obtenidos muestran la tasa positiva genuina en un 9% y la exactitud de una estructura de interrupciones en un 9%. El modelo propuesto utiliza algoritmos de machine learning, para la cual se desarrolló un sistema de rastreo a indicios de suplantación IP, que se usará para la localización del atacante, para eludir los ataques de suplantación IP.

Sweta, Dayashankar, & Aanjey, (2019), realizó la investigación, *Two-Phase Validation Scheme for Detection and Prevention of ARP Cache Poisoning*, en India. Un atacante busca encontrar un bug en un sistema consiguiendo explotar el protocolo arp intercediendo en la comunicación de datos entre usuarios de una red, envenenando la caché de la víctima y obteniendo el tráfico para que se desvíe al atacante. Por esta razón, se estableció un esquema, en la que una víctima puede establecer una tabla secundaria donde se registran las direcciones IP y MAC, cuyo objetivo es tener fijado y poder sondear paquetes con la tabla principal validando las direcciones para

contrarrestar las anomalías presentadas en una red. Los resultados obtenidos disminuyeron de una manera significativa los ataques de suplantación. El mecanismo propuesto no evidencia definiciones numéricas de lo desarrollado, sin embargo, se basa en autenticar todos los usuarios de una red a través de una tabla de registro.

Ren, Tian, Kong, Zhou, & Li, (2020), realizó la investigación, *An detection algorithm for ARP man-in-the-middle attack based on data packet forwarding behavior characteristics*, en Chongqing, China. Un atacante puede vulnerar la red fácilmente y hacerse pasar por un Gateway de un conmutador y escuchar paquetes de datos de la víctima con fines maliciosos. Por esta razón, presentó un algoritmo que se encuentra instalado en un conmutador, que permite identificar al hombre de en medio (atacante), y rastrear en tiempo real la ubicación. El método desarrollado no trata sobre el aprendizaje de máquina, por lo que no evidencia definiciones numéricas de lo desarrollado, es decir, su método beneficia a los usuarios generando buen desempeño en la prevención de ataques de suplantación.

S, Soman, & Dr Pritam, (2018) , realizó la investigación, *Security with IP Address Assignment and Spoofing for Smart IOT Devices, en Bangalore*, India. Un enrutador normalmente verifica la dirección del protocolo de internet (IP) de destino y la dirección IP principal de inicio no es verificado, por lo que usuarios de cualquier red son atacados suplantando su IP sin que sospechen, que un hombre del medio puede alterar su red. Por esta razón, describió la defensa contra la suplantación de IP a través de un filtro de conteo de saltos, desarrollado en dos fases, la primera que es la fase de aprendizaje que explora el tráfico y filtrado de direcciones cuando se haya verificado la autenticidad de las direcciones IP de un paquete. El método propuesto no trata evidencias en definiciones numéricas, con relación al aprendizaje de máquina, sin embargo, puede ser considerado como preventivo de ataques a través de un cuadro de validación del control de acceso a medio (MAC).

P P, GV, TS, & AS, (2020), realizó la investigación, *The problem of security address resolution protocol*, en Rusia. Los envenenamientos ARP son muy peligrosos con respecto a seguridad para el acceso a los datos ARP. Por esta razón, presentó una metodología que detiene los envenenamientos ARP utilizando herramientas como Python para poder denegar el acceso a los atacantes. Los resultados obtenidos son la mejorabilidad de seguridad de redes para poder tener una mejor defensa con las suplantaciones de identidad falsas. El modelo propuesto ayuda a tener mayor seguridad mediante implementación de lenguajes y llegar cubrir la suplantación ARP.

1.3. Teorías relacionadas al tema.

1.3.1. Seguridad Informática

1.3.1.1 ¿Qué es la seguridad informática?

Según Romero, et al. (2018), la seguridad informática es la encargada de salvaguardar todo el medio informático para ser seguro, es decir debe evitar actos que afecten la integridad de información perteneciente a una persona y por otro lado tambien disminuir los riesgos ante eventuales amenazas.

1.3.2. Redes

Según Pérez Torres, (2018), una red es la comunicación entre usuarios mediante dispositivos cuya finalidad es la de intercambiar información, gracias a los elementos como el software que son fundamentales para que la comunicación se pueda dar y al hardware, como tarjetas de red inalámbricas o alámbricas, cableados (Pág.19). De la misma manera, (Ariganello, 2020, pág.23), sostuvo que las redes LAN que significa redes de área local y las redes WAN que quiere decir red área amplia, permite que los usuarios puedan establecer una conexión que puede ser interna o externa de una organización para compartir su información de distintas maneras. Sin embargo, menciona que los dispositivos no son tan fundamental para las redes, sino los protocolos ya que sin ellos los datos que se comparte no se enviarán, ni se interpretaría para que llegue a su destino final.

1.3.2.1. Estándares de Comunicación

1.3.2.1.1. Modelo TCP/IP

Es un estándar que sirve para establecer la comunicación, la cual se responsabiliza de que la información sea enviada a su destino sin haber perdido su legibilidad desde su origen. Para ello se logra gracias a cada uno de los protocolos que tienen su funcionalidad.

El modelo TCP/IP trabaja por capas y cada una tiene su funcionalidad, al igual que los protocolos para que se realicen distintas maneras de comunicación.

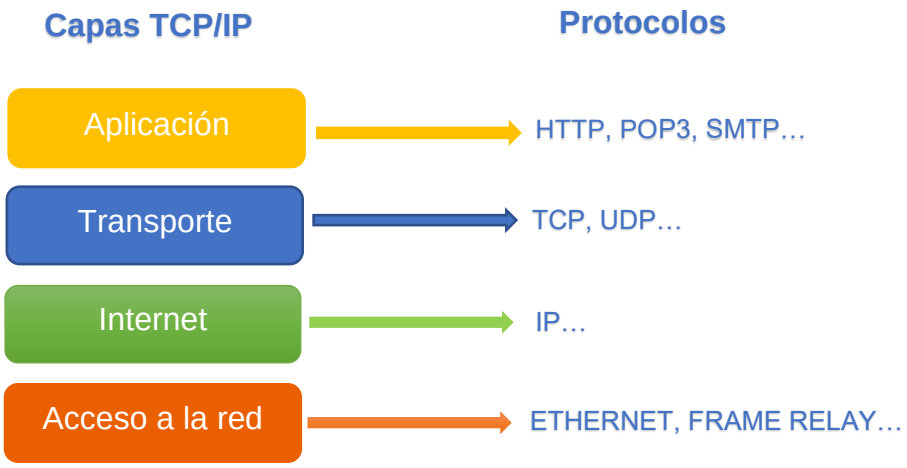


Figura 5. Agrupación Capas TCP/IP y respectivos protocolos. Fuente: Pérez Torres (2018).

1.3.2.1.1.1. Capa de aplicación

Proporciona los protocolos más requeridos a las aplicaciones para lograr ejecutar el proceso de la comunicación con la red, por ejemplo, se puede realizar intercambios de información entre cliente - servidor web, cuyo objetivo es revisar cómo se dirige la información en cada una de las capas. (Ariganello, 2020)

1.3.2.1.1.2. Capa de transporte

Según Pérez Torres, (2018), se encarga de que los paquetes sean sucesivos y sin fallas, permitiendo que los datos lleguen

sin importar lo que contenga. De donde resulta que el protocolo de transmisión de datos (TCP) es muy seguro antes de enviar los datos y el protocolo de diagrama de usuarios (UDP) que es menos seguro, ya que dicho protocolo se encarga del envío de los datos sin antes analizar que contenga errores para llegar a su destino.

1.3.2.1.1.3. Capa de Internet

Su función es añadir información y asegurar que los datos sean dirigidos a su destino correcto, dicho proceso se da a ejecutar por las direcciones IP, que se encargan de validar a los usuarios conectados en la red. (Ariganello, 2020)

1.3.2.1.1.4. Capa de acceso a la red

Según (Ariganello, 2020, pág. 71), es la capa en la que los usuarios pueden establecer conexión a la red, la cual es definida como área de trabajo, y es la que permite interconectar múltiples grupos de trabajo, además está integrada de varios protocolos, entre los más usados son: Protocolo de Internet (IP), Protocolos de resolución de direcciones (ARP), Protocolo de resolución inversa de direcciones (RARP), y el Protocolo de mensajes de control en Internet (ICMP).

1.3.2.1.1.4.1. Protocolos de red más usado

1.3.2.1.1.4.1.1. IP

Se interconecta con las redes más utilizadas. El protocolo se especifica en dos partes que son servicios IP y los protocolos IP.

Servicios IP:

Por consiguiente, son servicios que se conceden a las capas de protocolos inmediatos, se realizan en primitivos que son las tareas que se va a proporcionar y los indicadores que son para transitar información. Para su

Tabla 1.

Opciones de calidad del servicio IP

Precedencia	**Medida que proporciona preferencia en los datagramas.**
Seguridad	Pueden ser dos niveles de seguridad alto o bajo. Una seguridad de nivel alto, hace que el datagrama obtenga mínima posibilidad de resultar dañado o se pueda perder.
Retardo	Tiene dos niveles: alto o bajo. Si es un nivel bajo indica que tiene mínimo retardo para el datagrama.
Rendimiento	Son dos niveles, alto o bajo. Un nivel alto significa que tiene un rendimiento máximo para el datagrama.

Nota: Tomada de Stallings (2011)

calidad de los servicios IP se tienen varias opciones, las cuales son:

1.3.2.1.1.4.1.2. ARP

El protocolo de resolución de direcciones, es quien se encarga de realizar la resolución automática del proceso de mapeo de las direcciones MAC, cuando se inicia la transmisión de algún paquete entre la comunicación de dos computadores dentro de una red local, se indica a la aplicación la dirección IP que le corresponde. Pongamos, por ejemplo, Si el host 192.168.6.10 desea establecer conexión con 192.168.6.13, realizarían lo siguiente:

$ telnet 147.83.153.100

Donde aplican la máscara de red al protocolo de internet (IP) que ha solicitado, el host supone que está ubicado en su misma subred. Por consiguiente, ya no es necesario permitir que se le delegue en el direccionador, para que así las tramas puedan enviar los paquetes IP que la aplicación telnet ha generado, sin antes saber cuál es la dirección MAC.

Por lo cual, se encarga de enviar una solicitud, que es el encapsulamiento de un paquete que es dirigido sobre la trama de Ethernet (que viene a ser de tipo = 0x0806), que es remitido de su destino que tiene la ubicación del broadcast (EE; EE; EE; EE; EE; EE) para que el paquete se rea recibidos por todos los computadores que están dentro de una red LAN, conteniendo la dirección IP que requiere averiguar su dirección MAC, donde el host identifica su IP en la solicitud con su dirección MAC.

Para interpretar cómo funciona un host al realizar peticiones ARP, que generalmente son habituales, en efecto optan por una tabla emparejada de direcciones IP y sus respectivas direcciones MAC, de manera que, si una dirección IP ya está agregada en la tabla, no envían la petición de ARP. A la tabla se le conoce como caché ARP, que se mostrará a continuación:

Tabla 2.
Apariencia de Caché ARP

Tabla caché ARP		
IP	MAC	Interfaz
148.84.163.104	09:00:00:10:96:00	ether0
148.84.163.105	00:0c:ee:00:0f:e0	ether0

Nota: Tomado de Barceló, et al. (2004)

De las filas que se mostraron en la tabla 1, compete a un mapeado de la dirección IP y MAC con su interfaz de red que se ha pedido la petición, el cual se llena de manera automática mientras se van realizando actuales peticiones, cabe mencionar que puede ser manipulado haciendo una consulta a través del comando ARP.

En conclusión, la tabla ARP, generalmente se le denomina caché, ya que se comporta como memoria de ayuda para evitar hacer consultas de información a la red LAN, cuando actualmente cuente con un backup local propio. (Barceló, et al, 2004)

1.3.2.1.1.4.1.3. RARP

Es el protocolo de resolución de dirección inversa, que dispone direcciones IP de hardware en direcciones de red en el adaptador de red local. (Ariganello, 2020)

Es un protocolo que accede desde una máquina física a su dirección IP mediante el protocolo ARP de un servidor de puerta caché. Cuando se agrega una máquina configurada, el RARP del cliente solicita la dirección IP del enrutador RARP. El protocolo RARP está apto para Ethernet, Interfaz de datos y LAN Token Ring. (Jacobson, 2017)

Tabla 3.
Encabezado del protocolo RARP:

16		**32 bits**
Prototipo de hardware		Prototipo de protocolo
Distancia de la dirección de hardware	Distancia de la dirección del protocolo	Operación
Ubicación de hardware del remitente		
Dirección de protocolo del remitente		

Ubicación de hardware de destino
Ubicación de protocolo de destino

Nota: Tomada de Jacobson (2017)

1.3.2.1.1.4.1.4. ICMP

Es un protocolo de control de mensajes en internet, las herramientas ping y trace emplean el protocolo ICMP, enviando direccionamiento de paquetes específicos y así poder obtener respuesta. (Ariganello, 2020)

El ICMP tiene un campo de código de cabecera con los siguientes valores:

Tabla 4.
Campo de código ICMP.

Campo	Descripción
0	Contestación de eco.
3	Destino abrupto.
4	Pérdida del tráfico desde el lugar de inicio.
5	Reasignar ruta.
8	Petición de eco.
11	Duración extralimitada.
12	Incidencia de indicadores.
13	Solicitud de duración.
14	Contestación de traza de duración.
15	Petición de información.
16	Contestación de información.
17	Solicitud de máscara.
18	Contestación de la máscara.

Nota: Tomado de Ariganello (2020)

Según (Carvalho, 2016) es un protocolo adicional a IP, se utiliza para reportar mensajes de respuesta a paquetes de información indicando que un host puede ser localizado. A continuación, se muestra su funcionamiento del protocolo ICMP:

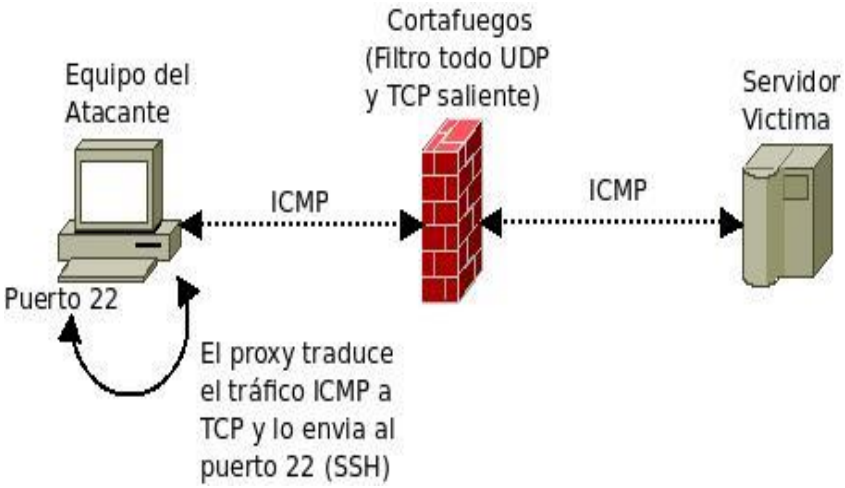


Figura 6. Funcionamiento del protocolo ICMP.
Fuente: (Carvalho, 2016)

1.3.2.1.2. Modelo OSI

Es un modelo conceptualizado para crear soluciones en una red y también diseñar redes, además permite establecer una comunicación en una red, cuya finalidad es que la información llegue a su destino.

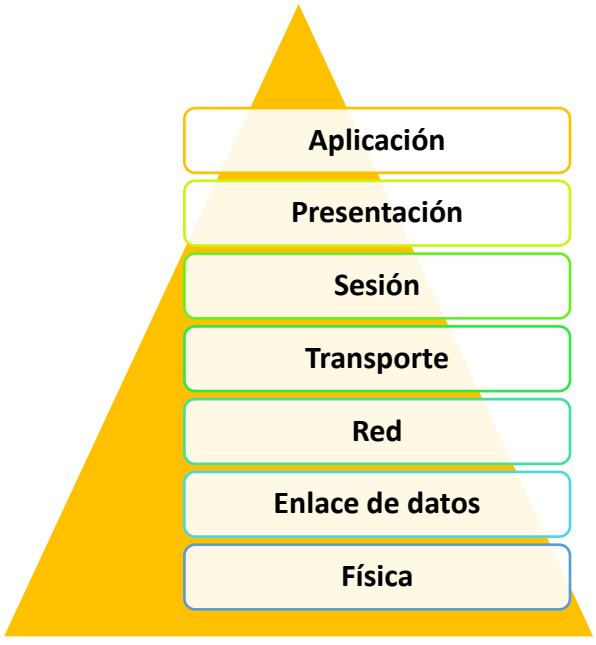


Figura 7. Las 7 capas del modelo OSI.
Fuente: Hallberg (2007)

1.3.2.1.2.1. Capa Aplicación

Es la capa con la que se relaciona el usuario, la cual brinda los protocolos a un software para que pueda iniciar una comunicación haciendo una conexión entre cliente y servidor.

Dentro de la capa de aplicación los protocolos más conocidos son:

Tabla 5.
Protocolos más comunes en la capa de aplicación.

PROTOCOLO	FUNCIÓN
DNS	Resolución de nombres de host a direcciones IP.
HTTP	Transferencias de páginas web.
SMTP	Envía emails.
POP	Recepciona emails.
FTP	Transferir ficheros.
DHCP	Proporciona a los hosts configuración automática de red.
TELNET	Conexiones virtuales para acceder remotamente.

Nota: Tomado de Pérez Torres (2018)

Según (Ariganello, 2020, pág. 27), es la única capa que no da soporte a otras, ya que es la única que se encuentra en el nivel superior y se asocia con un usuario.

1.3.2.1.2.2. Capa de presentación

Según, (Pérez , 2018, pág. 12), es la que se encarga de la codificación y conversión de los datos más importantes para luego ser enviado al destino. Estas funcionalidades se encargan de verificar que los datos emitidos a partir de la capa de aplicación origen sean recibidos en la de aplicación del destinatario, indica (Ariganello, 2020, pág. 27.)

1.3.2.1.2.3. Capa de sesión

Según (Ariganello, 2020, pág. 28), la comunicación en la capa de sesión debe establecer una solicitud de servicios y también en recibir las respuestas de aplicaciones que se encuentren conectadas entre dispositivos, también es la encargada de aplicar, supervisar y dar por finalizado las sesiones que se encuentre en la capa de presentación.

1.3.2.1.2.4. Capa de Transporte

Establece y especifica varias funciones a nivel de la red, es la principal encargada del control de flujo que hace supervisión a una comunicación de origen y destino, y es la única capa en los sistemas operativos de red, menciona (Pérez, 2018.)

1.3.2.1.2.5. Capa de Red

Según (Hallberg, 2007, pág. 30), la capa de red tiene las mismas funcionalidades que aquella capa de red ubicada en el modelo TCP/IP, siendo la encargada de iniciar el direccionamiento lógico, además elige una ruta para llegar a su destino mediante una tabla de enrutamiento a través de los protocolos.

1.3.2.1.2.6. Enlace de datos

Según (Hallberg, 2007, pág. 30), se encarga de que los estándares verifiquen los bits que viajan a la capa física, además analiza los errores y los repara para que los flujos de datos sean seguros. La función de la capa es convertir un medio de comunicación a una cola sin errores, y evitar que un receptor lento sea sobrecargado de datos, menciona (Tanenbaum & Wetherall, 2012),

1.3.2.1.2.7. Física

Según (Hallberg, 2007), esta capa se encarga que los paquetes sean transmitidos a través de un medio físico para lograr una conexión de red, es decir un cable de red, un cable coaxial, una fibra óptica, que es un medio por donde los bits viajan en nodo a través de una red física, también se puede decir que establece una adaptación de los bits generados por un pc a través de la codificación.

1.3.2.2. Machine Learning

Proviene de la inteligencia artificial, que es insertada de manera sistemática en los algoritmos para mejorar la forma en cómo se trabaja los datos y la información, tomando como ejemplo que actualmente se puede entrenar a los algoritmos de aprendizaje automático por voz, es este caso Siri de IPhone, que puede transformar datos acústicos a una secuencia de palabras. Así mismo se puede decir que ML (machine learning), no está bien establecido para poder desarrollarse de manera óptima, pero los errores se pueden entrenar para que de esa manera se pueda experimentar errores del aprendizaje, obteniendo como finalidad predicciones de distintos escenarios, que son incógnita para un ordenador. (Awad & Khanna, 2015, pág. 1)

1.3.2.2.1. Tipos de Machine Learning

1.3.2.2.1.1. Machine Learning supervisado.

Según (Ernesto, et al, 2018), el algoritmo visualiza variedad de ejemplos de entrada, es decir se encarga de ver un modelo que pueda predominar los datos y construir un patrón para pronosticar su salida, como si hubiera un encargado de enseñar. La aplicación de obtener variedad de muestras como ejemplo para que a futuro tenga excelentes predicciones con los nuevos datos obtenidos.

1.3.2.2.1.2. Machine Learning no supervisado.

Los algoritmos de identificación automática están basados en la creación de estructuras escondidas en agrupación de datos, sin saber el resultado esperado, su enfoque a nivel general es entrenar la aplicación a través de un modelo de datos. Generalmente estos algoritmos se encajan en estimaciones de máxima verosimilitud (MLE), además puede estar de la mano con pronósticos para otorgar recursos, indica (Awad & Khanna, 2015, pág. 7.).

Por ejemplo, si una base de datos de mucha clientela, que son clientes de un supermercado, y se pretende obtener una variedad de perfiles en base a compras que han realizado, pero no se determina el número de perfiles que tiene, a ello se percibe como las dificultades de clustering.

Se presenta un ejemplo sobre una agrupación de datos.

Se busca identificar un grupo de datos.

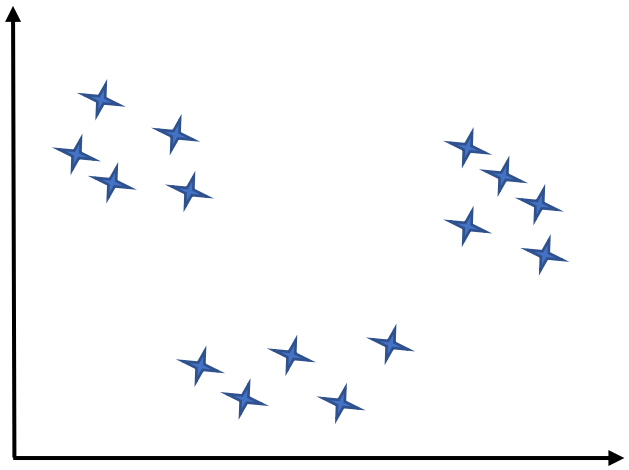


Figura 8: Determinación de datos en clases a identificar.
Fuente: (Mathivet, 2015)

De dicha forma se puede identificar la similitud de los datos de una clase, agrupándolos y separándolos de acuerdo a lo que identificó basado en el algoritmo de aprendizaje no supervisado.

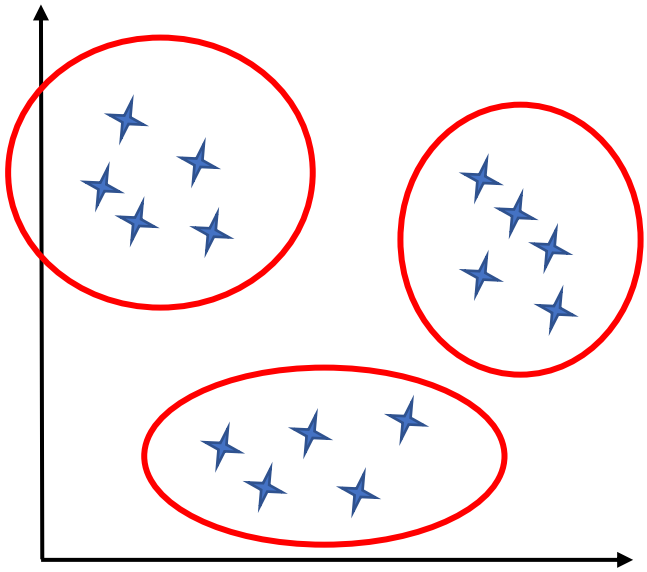


Figura 9. Determinación de datos en clases en base a similitud. Fuente: (Mathivet, 2015)

1.3.2.2.1.3. Machine Learning por refuerzo.

Según (Awad & Khanna, 2015, pág. 8), un agente inteligente se encarga de descubrir comportamientos en un marco de motivación para que pueda incrementar y adaptarse en un marco de motivación, la técnica de identificación debe sintetizar para que el modelo pueda ser entrenado en una agrupación de acciones experimentales. El algoritmo se encarga de determinar si las decisiones que obtiene son correctas luego de haber sido usado, sin embargo, aquel algoritmo no tiene una buena funcionalidad para determinar el tiempo de decidir, en efecto, no es muy usado en algunos dominios, menciona (Mathivet, 2015, pág. 328).

1.3.2.2.2. Tipos de algoritmos de machine learning

1.3.2.2.2.1. k-Means

(Awad & Khanna, 2015), es un algoritmo de agrupamiento repetitivo que está distribuido en un grupo N de puntos de datos

en K subgrupos para sintetizar la posición de la suma de cuadrados, a causa de ello la euclidiana al cuadrado es de capacidad media o más cercana.

$$J = \sum_{j=1}^{K} \left[\sum_{n \in S_j} | x_n - u_j |^2 \right]$$

Donde

$x_{n=}$ Es el vector que representa el enésimo punto de datos.

$u_{j=}$ Es el centroide geométrico de los puntos de datos en S_j.

El algoritmo de K means se basa la evaluación de un proceso sencillo para la estimación, y se considera dos puntos importantes dentro de ella, la primera que es la asignación, cuya finalidad es de encargarse que los datos sean agregados en un grupo, con el objetivo que el centroide esté cada vez más aproximado de dicho punto y la actualización, que se basa en realizar un cálculo de todos los puntos agregados al medio. Ambos de los puntos mencionados son muy importantes porque se turnan hasta iniciar el criterio de parada, es decir no se realizará más asignaciones de los datos.

1.3.2.2.2.2. Support Vector Machines

Son procedimientos de identificación que se encarga de ejecutar un análisis de datos e identificar patrones, cuyo objetivo es utilizar la clasificación, el análisis y detección de variantes. En efecto, un grupo de datos son preparados en una tarea de dos clases. Así mismo el algoritmo se encarga de crear una modalidad de clasificación, que agrega indicaciones a cualquiera de dichas clases y mapear, obteniendo una clasificación de los datos, para poder distribuirse de acuerdo a su ubicación, además la SVM se encuentran a una distancia muy cercana del hiperplano, que separa las clases.

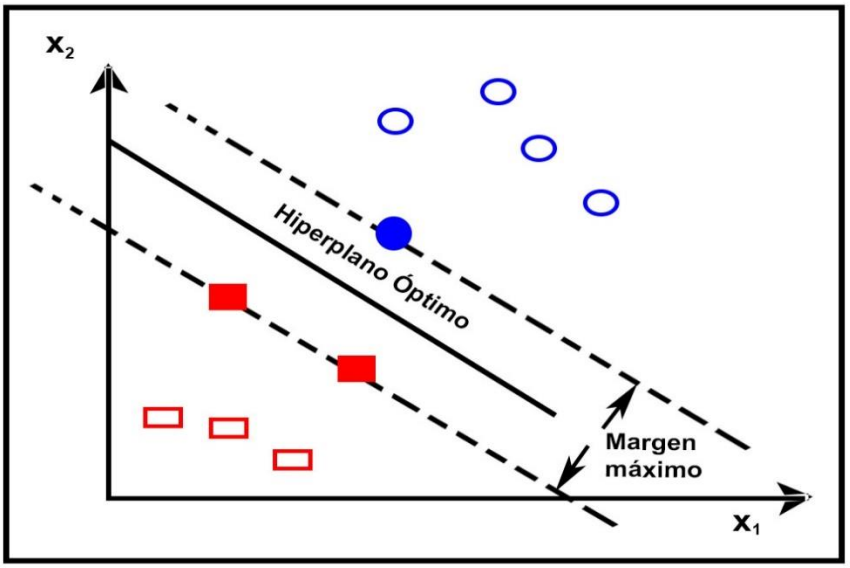


Figura 10. El SVM ubicado en el hiperplano con distancia entre vectores de soporte.
Fuente: (Awad & Khanna, 2015)

1.3.2.2.2.3. AdaBoost

Es un método que sirve para crear clasificadores reforzados en secuencia a partir de una base común que es entrenada, en tal sentido el algoritmo sostiene a varios usuarios para solucionar problemas con mejor predicción. Así mismo el clasificador puede hacer verificaciones, al igual que una combinación lineal de clasificadores fuertes, en efecto que:

$$H(x) = \sum_{t=1}^{T} B_t . h_t\,(x),$$

Donde

$H(x) =$ Clasificador fuerte.

$h_t(x) =$ Clasificador débil.

Adaboost se puede sintetizar de la siguiente manera:

Entrada

Conjunto de datos I $=\{(x_1, y_1)(x_{2,}, y_2)(x_{3,}, y_3), -, (x_{m,}, y_m)\}$.

Algoritmo de aprendizaje base L.
Número de rondas de aprendiza T.

Procesos

$D_1^i = \frac{1}{m}$ // Inicia la repartición de peso.

For(t=1 to T) DO // Ejecuta el ciclo para t = T iteraciones.

h_t = L(I,D_t) // Entrena un aprendiz débil h_t desde I usando D_t.

$\epsilon_t = \sum_i \boldsymbol{D_t^i} |h_t(x_1) - y_i|$ // Calcula el error de h_t.

$\beta_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$ // Calcula el peso de h_t.

$D_{t+1}^i = \frac{D_t^i}{Z_t}.e^{(-\beta_{t*}\, y_{i*} h_t(x_i))}$ // Actualiza la distribución.
// Z_t es el factor normalizado.

Salida

$H(x) = sign(\sum_{\tau=1}^{T} \beta_t h_t(x))$ // Clasificador fuerte

El algoritmo es acoplable a muchas iteraciones para lograr un aprendiz fuerte, que esté asociado con el clasificador real, es por ello que puede iterar añadiendo aprendices débiles formando parte del proceso de adaptación. Adaboost, es un algoritmo veloz y fácil para iniciar la implementación y adaptativo para unir con otros tipos de clasificadores, menciona (Awad & Khanna, 2015, pág. 13.).

Adaboost se puede esquematizar de la siguiente manera:

Se muestran datos en la parte izquierda, las barras indican los pesos agregados a cada una de las instancias, en efecto las predicciones son clasificadas para luego ser ponderado por los triángulos y finalmente se realiza la sumatoria en el círculo.

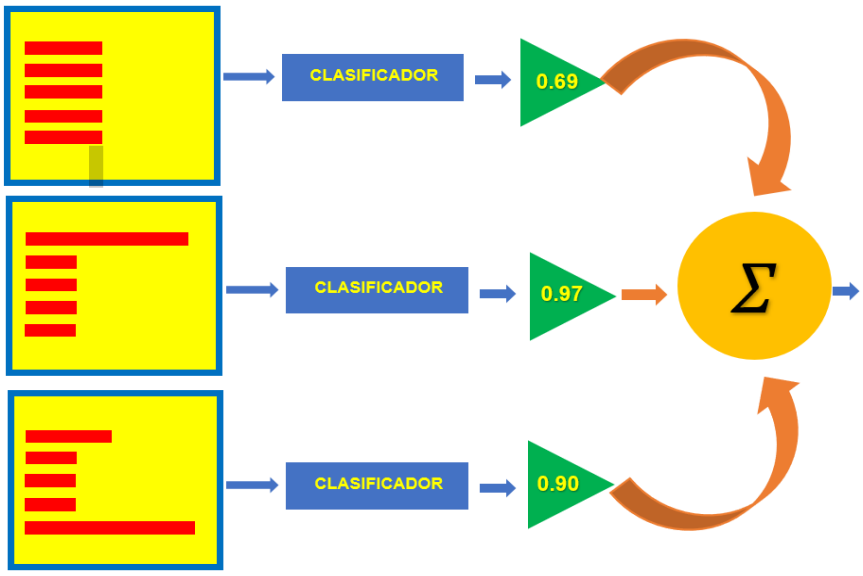


Figura 11. Representación gráfica de los procesos de Adaboost.
Fuente: (Harrington, 2012)

1.3.2.2.2.4. k-Nearest Neighbors

Es utilizado para solucionar problemas de clasificación, siendo un modelo de los más fáciles y con características intuitivas. De manera que su modo de aprendizaje es registrar ejemplos a la base de entrenamiento a través de su función de distancia y cifras de los vecinos más aproximados, como resultado, pronóstica de que categoría es el individuo. Desde otro punto de vista la eficiencia de aprendizaje es acelerado, aunque el tiempo que tarda en realizar la predicción de la distancia entre la instancia demora, siendo un método elevado con respecto al tiempo para predecir bases de datos como mayor volumen, a

menos que puedan equilibrarlas para obtener resultados eficaces, menciona (Polo, et al, 2018).

Según (Awad & Khanna, 2015, pág. 14), el algoritmo se encarga de detectar objetos dentro de una clase de aprendizaje de objetos K que están más aproximados a un objeto demostrativo y añadir la etiqueta a una clase más fuerte de un escenario. Ahora bien, se nombran tres componentes importantes, que son un grupo de objetos de etiquetas, la estimación de la distancia de objetos y cifras de los vecinos más cercanos (K).

Para detectar k vecinos más cercanos, se realiza un cálculo de la distancia del objeto y de los demás que son etiquetados, de manera que se pueda clasificar un objeto desconocido, como resultado sirve como referencias usarlo como referencia para un futuro objeto sin etiqueta.

Llegados a este punto el algoritmo K-NN se encarga de calcular la similitud entre las distancia de los objetos que están dentro de un conjunto de aprendizaje, (x,y) $\in$ I , y el objeto de prueba $z = (\hat{x}, \hat{y})$ para obtener con precisión la lista de vecinos que están cerca de I, 'x' simboliza el objeto de aprendizaje , 'y' simboliza la clase de entrenamiento. $(\hat{x}, \hat{y})$, representan el objeto de prueba y su clase. Para comprender mejor el algoritmo se puede sintetizar de esta forma:

Entrada:

Objeto de entrenamiento (x, y) $\in$ I y objeto de prueba z =$(\hat{x}, \hat{y})$ $\in$ I .

Proceso:

Calcular distancia d =$(\hat{x}, x)$ entre z y cada objeto (x, y) $\in$ I.

Salida (Clase mayor)

$$\hat{y} = arg_v\, max \sum_{(x_i, y_i \in I_z)} F(v = y_i)$$

F(.) = 1 si la prueba (.) es verdadera y el 0 en caso puesto, vendría a ser la etiqueta de la clase.

El valor de (K) se debe seleccionar con cuidado, en efecto un valor inferior puede originar un comportamiento escandaloso, donde resulta que mientras sean un valor superior puede integrar muchos puntos de otras clases.

1.3.2.2.2.5. Naive Bayes

EL algoritmo de Bayes ingenuos es un clasificador de probabilidad, que fue creado por Thomas Bayes (1701-1761) dado que decidió experimentar la existencia de Dios, a causa de ello intentó probar su método de Bayes donde ha basado su funcionamiento a través de un algoritmo:

$$p(a|b) = \frac{p(a,b)}{p(a)}$$

que por ley de la multiplicación probabilística se puede expresar de la siguiente manera:

$$p(b|a) = \frac{p(a|b).p(b)}{p(a)}$$

Naive bayes es un algoritmo que es usado en el aprendizaje de mañana para categorizar textos que están basados en la frecuencia de palabras, así mismo se pueda usar para detectar, sirva de ejemplo que se busca determinar si correo electrónico es spam o es un texto en particular. En efecto, debido a que es un algoritmo simple y rápido, tiene buen rendimiento a comparación de otros clasificadores, menciona (Faleiros, et al, 2018,pág. 30).

Asume que las características de entrada $x_1, x_2 \dots x_n$ por condición son independientes entre sí, dado la etiqueta de clase Y, tal como:

$$P(x_1, x_2 \dots x_n | \text{Y}) = \prod_{i=1}^{n} P(x_i|Y)$$

Para una clasificación de dos clases donde (i=0.1), se define como P (i | x) como la probabilidad del vector $x\{x_1, x_2 \dots x_n\}$ pertenece a la clase i. Donde definen el puntaje de la clasificación de la siguiente manera:

$$\frac{\boldsymbol{P(1|x)}}{\boldsymbol{P(0|x)}} = \frac{\prod_{j=1}^{n} f(\boldsymbol{x_j} \mid \boldsymbol{1}) \boldsymbol{P(1)}}{\prod_{j=1}^{n} f(\boldsymbol{x_j} \mid \boldsymbol{0}) \boldsymbol{P(0)}} = \frac{\boldsymbol{P(1)}}{\boldsymbol{P(0)}} = \prod_{\boldsymbol{j=1}}^{\boldsymbol{n}} \frac{f(\boldsymbol{x_j} \mid \boldsymbol{1})}{f(\boldsymbol{x_j} \mid \boldsymbol{0}}$$

$$\ln \frac{\boldsymbol{P(1|x)}}{\boldsymbol{P(0|x)}} = \boldsymbol{ln}\ \frac{\boldsymbol{P(1)}}{\boldsymbol{P(0)}} + \sum_{\boldsymbol{j=1}}^{\boldsymbol{n}} \boldsymbol{ln}\ \frac{f(\boldsymbol{x_j} \mid \boldsymbol{1})}{f(\boldsymbol{x_j} \mid \boldsymbol{0}},$$

Donde P (i | x) es proporcional a f (x | i) P(i) y f (x | i) es la estructura condicional de x para objetos de la clase i.

El algoritmo de Bayes es atractivo, a pesar que es simple y robusto a pesar de que no necesite de un esquema de predicción de parámetros iterativos y complejos, menciona (Awad & Khanna, 2015, pág.15).

1.3.2.2.2.6. Random Forest

Los bosques aleatorios son un algoritmo de fácil aprendizaje, ya que crea múltiples árboles de decisión, y los mezcla para lograr que las predicciones sean precisas y permanentes.

Para determinar el sobreajuste es necesario la mezcla de todas las predicciones, con el fin de obtener una predicción única.

Para esto es necesario un conjunto de variables no relacionales y sus respectivas varianzas.

Promedio:

$$E\left[\frac{1}{n}\sum_{i=1}^{n} E[Y_i] = \frac{1}{n}.n\mu = \mu\right]$$

Donde:

n= número de predicciones.

μ=total de predicciones.

Varianza reducida:

$$Var\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right) = \left(\frac{1}{n}\right)^2 \sum_{i=1}^{n} Var(Y_i) = \frac{1}{n^2}.n\sigma^2 = \frac{\sigma^2}{n}$$

Donde:

n= Número de predicciones.

μ= Total de predicciones.

Var= Varianza.

Y_i= Predicciones.

1.3.2.2.2.7. Logistic Regression

Es una técnica que permite relacionar entre variable dependiente e independiente. (BSG, Institute, 2020)

La regresión logística modela la relación entre la dependiente "X" y un resultado característico "Y".

Su fórmula es:

$$P\left(\frac{y}{x}\right) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

Para generar los coeficientes de la regresión logística se tiene que volver a escribir y transformarlo a la inversa.

$$logit\left(P\left(\frac{y}{x}\right)\right) = In\left[\frac{P\left(\frac{y}{x}\right)}{1 - P\left(\frac{y}{x}\right)}\right] = \beta_0 + \beta_1 X$$

La función logística recibe valores de entrada ($\beta_0 + \beta_1 X$) entre infinito positivo y negativo, y la salida $P\left(\frac{y}{x}\right)$ esta limitado a los valores 0 y 1, como se aprecia en la imagen:

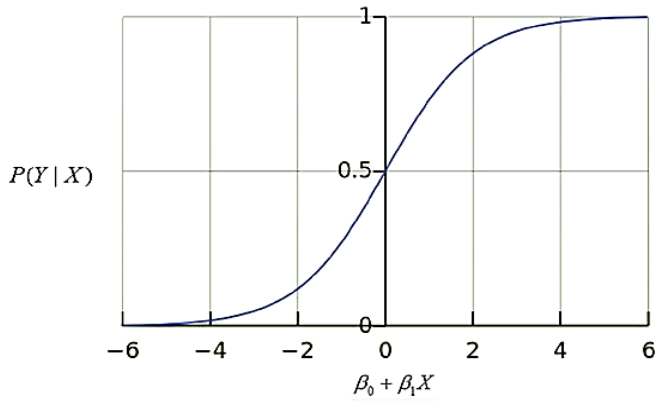


Figura 12. Función Logística y sus valores de entrada. Fuente: (Awad & Khanna, 2015)

La regresión logística usa una curva, utilizando los coeficientes de regresión β_0 y β_1, como se ve en la ecuación 1, en donde la salida es una variable binaria y X es numérico; en cambio en la ecuación 2 la probabilidad de éxito (Y) es $P\left(\frac{y}{x}\right)$ para un valor de hecho en X. Entonces la regresión logística tiene una ecuación formulada como:

$$logit\left(P\left(\frac{Y = 1}{X_1, X_2, X_3 \ldots X_n}\right)\right) = \beta_0 + \sum_{k=1}^{n} \beta_x X_k$$

Su resultado es en atribución a las variables que son netamente independientes o predictoras. (Awad & Khanna, 2015)

1.3.2.2.2.8. Binary Logistic Regression

Es un tipo de analizador lineal simple, su función es predecir los resultados de todas las variables z. La regresión logística binaria es un analizador que tiene como único objetivo comprobar la hipótesis. Para generar una distribución de probabilidad es necesario utilizar la sigmoidea, para esto se usa la fórmula:

$$s(z) = \frac{1}{1 + e^{-x}}$$

Donde:

S= Sigmoidea.

e= Constante exponencial (2,71828).

Figura:

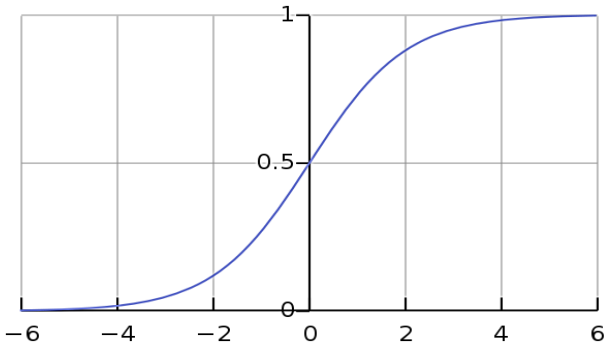


Figura 13. El eje horizontal es la salida de la función lineal y el eje vertical es la salida de la función logística, que se puede interpretar como una probabilidad entre 0 y 1.
Fuente: (Nasiriany, Garrett.Thomas, Wang, & Yang, 2019)

1.3.2.2.2.9. Multiclass Logistic Regression

Según (Nasiriany, Garrett.Thomas, Wang, & Yang, 2019), se utiliza para problemas de dos posibles resultados discretos. Para la pérdida de la regresión logísticas multiclase, se utiliza la teoría de la información perspectiva. Para la distribución de la regresión logística multiclase se utiliza la fórmula:

$$\mathrm{P}\,(\mathrm{Y}\hat{} = \mathrm{i}) = \frac{\mathrm{e}^{\mathrm{w_j}'\mathrm{x_i}}}{\sum_{\mathrm{k}=1}^{\mathrm{k}} \mathrm{e}^{\mathrm{w_j}'\mathrm{x_i}}}$$

1.3.2.2.2.10. Regresión linear

La regresión lineal es un modelo que es muy práctico de utilizar, se utiliza para predecir el valor de variables según el valor de otras variables. La variable que desea predecir se le conoce como variable dependiente, y la otra variable que se emplea para predecir se le llama variable independiente. (BSG, Institute, 2020)

La relación estadística se le conoce como predicción y respuesta. (Faleiros, Henrique, & Maia Polo, 2018)

La fórmula general de la regresión lineal es:

$$y_i = b + w_1 x_{1i} + w_2 x_{2i} + \ldots + + w_k x_{ki} + \varepsilon_i = b + x_i^T w$$
$$= \hat{y}_i + \varepsilon_i$$

Donde:

x_{ji}= conjunto de características observables.

b = intersección.

$\hat{y}$ = lo que se desea producir por el modelo.

ε_i= error –.

1.3.2.2.2.11. Decision Tree

Según el instituto BSG, los árboles de decisiones son un conjunto de reglas, que su forma de representación esquemática es de un árbol.

El árbol de decisión, es uno de los métodos de predicción que propone pruebas simples. Se representa mediante un árbol para las pruebas. Los árboles de decisiones se emplean para la regresión y la clasificación.

Un caso simple de saber que la característica del valor j es menor que el valor v:

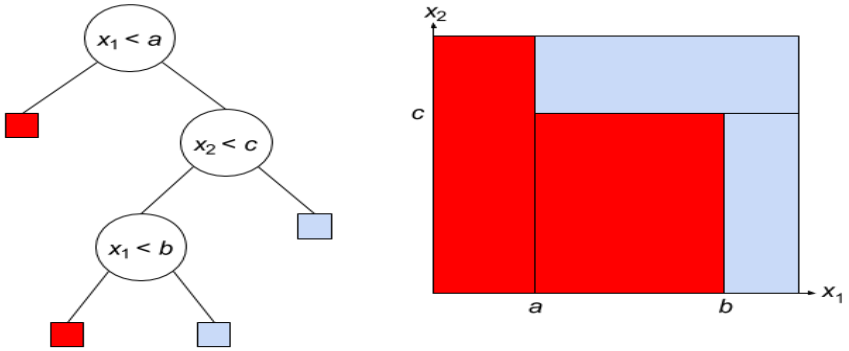


Figura 14. Representación de un árbol de decisiones arbitrariamente complejo.
Fuente: (Nasiriany, Garrett.Thomas, Wang, & Yang, 2019)

Capacitación:

La forma en que se entrenan los árboles de decisiones son codiciosas y recursivas, haciendo que se resuelva desde abajo, mediante las creaciones de nodos con divisiones mezcladas, y sus hijos son construidos con la misma técnica de árbol. Los datos almacenados son utilizados en subárboles izquierdos que satisfacen $x_j < v$, y los árboles derechos $x_j \geq v$.

Entropía e información:

La forma de medir la entropía y sorpresa es:

$$log\frac{1}{P(Y=k)} = -logP(Y=k)$$

Cuando $P(Y=k)$ = 0 la sorpresa se acerca a ∞, mientras que P$(Y=k)$ = 1, el valor se acerca a 0.

La sorpresa de la entropía es H(Y) que se denota en:

$$H(Y) = \mathbb{E}[-logP(Y)] = -\sum_k P(Y=k)logP(Y=k)$$

La gráfica de la entropía es:

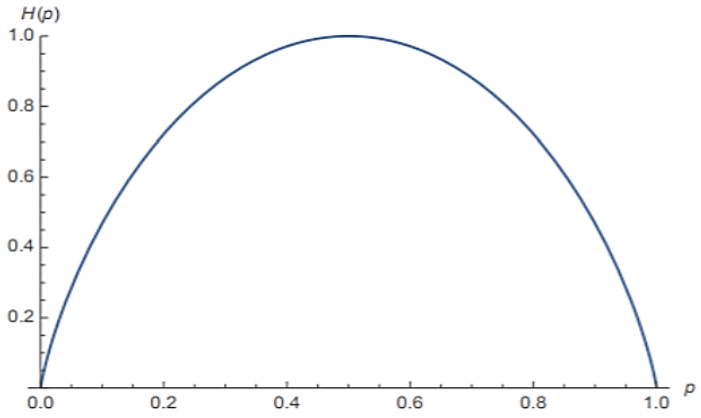


Figura 15. Representación de una entropía frente a una variable aleatoria.
Fuente: (Nasiriany, Garrett.Thomas, Wang, & Yang, 2019)

En la gráfica se observa que cuando la entropía su variable es 1, se denomina entropía cóncava, en general cuando la variable tiene más entropías los resultados son más idénticos en cambio cuando tiene menos entropías la distribución es muy quieta a salir. (Nasiriany, Garrett.Thomas, Wang, & Yang, 2019)

1.3.2.2.2.12. Perceptrón

Es una red neuronal unidireccional que tienen como función una entrada y otra de salida. Las entradas se relacionan con n y las salidas con p. Funciona mediante la salida si llega a 1 el perceptrón llega a ser positivo, en cambio si llega a -1 el perceptrón se vuelve negativo y si llega a 0 el perceptrón se vuelve nula. Se expresa de la siguiente manera:

$$sign(z) = \begin{cases} 1 & si\ z > 0 \\ 0 & si\ z = 0 \\ -1 & si\ z < 0 \end{cases}$$

Su modelo matemático se representa el perceptrón de la siguiente forma:

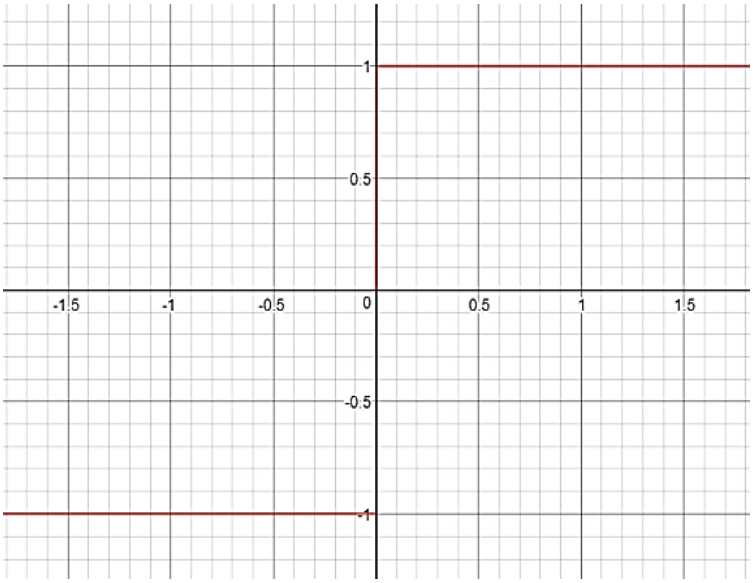


Figura 16. Representación del modelo matemático de Perceptrón.
Fuente: (Faleiros, Henrique, & Maia Polo, 2018)

En este algoritmo se tiene dos clasificaciones binarias (dos clases) de perceptrón, su fórmula es:

$$z = w_1 a_1 + w_2 a_2 + \cdots + w_n a_n$$

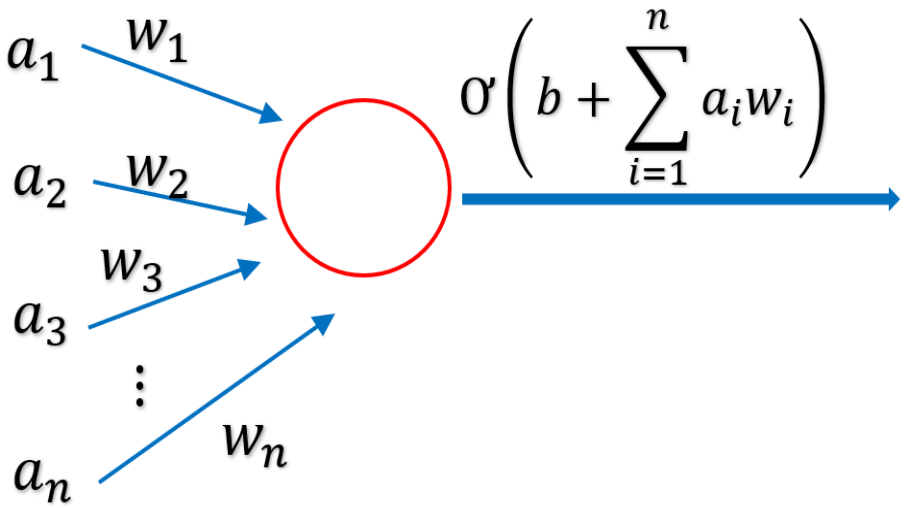


Figura 17. Representación gráfica de la fórmula de Perceptrón.
Fuente: (Faleiros, Henrique, & Maia Polo, 2018)

Se llega a la conclusión que el perceptrón es un clasificador lineal, ya que el plano bidimensional se nota que la línea se divide las clases. (Faleiros, Henrique, & Maia Polo, 2018)

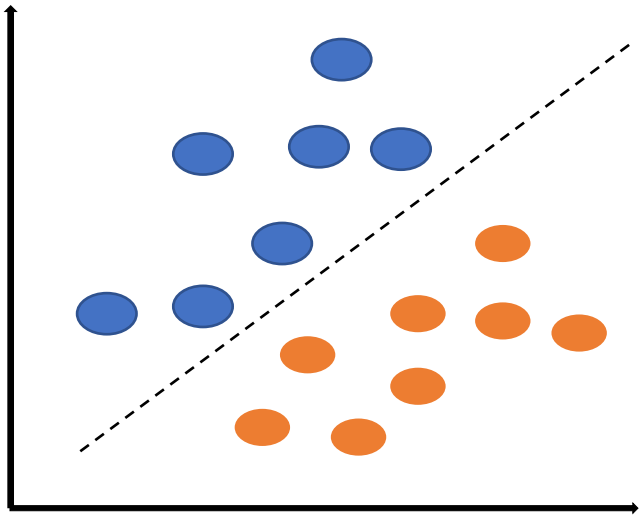


Figura 18. Representación de división de clases de perceptrón lineal.
Fuente: (Faleiros, Henrique, & Maia Polo, 2018)

1.3.2.2.2.13. Gradient Bosting

Es un algoritmo utilizado para el análisis de regresión supervisados de aprendizaje automático, el cual creará un modelo predictivo en la estructura del conjunto de modelos de predicción. Las predicciones suelen ser débiles y propensas al sobreajuste, pero si combinan en un conjunto generará un resultado mejorable. (Open Data Science, 2018)

Para el análisis se utilizan los clasificadores de árboles, se comienza con la ejecución y comprensión de la gradiente de los árboles de clasificación. (Gandhi, 2018)

Se comienza con el dominio de “x”, para que se obtenga la función limitante L, que se muestra en la fórmula:

$$F_0(x) = argmin_p \sum_{i-1}^{N} L(y_i, p)$$

Con esto se obtiene el valor de “y” y se podrá calcular la gradiente negativa, con la siguiente fórmula:

$$\overline{y_i} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F x_i}\right]$$

Posteriormente se utiliza la variable “α” que se usa para el ajuste de error obtenido, mediante la siguiente fórmula:

$$\alpha_m = \ argmin_{\alpha,\beta} \sum_{i=1}^{N} \lfloor \bar{y} - \beta h(x_i; \alpha_m \rfloor^2$$

Después se emplea la gradiente positiva mediante la siguiente fórmula:

$$\rho_m = \ argmin_{\rho} \sum_{i-1}^{N} L(\ y_i, F_m - 1(x_i) + \rho h(x_i; \alpha$$

Por último, se renueva la fórmula con los parámetros encontrados que se obtiene de los árboles de selección de datos.

$$F_x \rightarrow \ F_m(x) = F_{m-1}(x) + \rho_m h(x; \alpha_m)$$

La fórmula se utiliza para medir el gradiente desarrollado de un árbol de decisión, es importante el entendimiento de la ecuación para su funcionamiento.

1.3.2.2.2.14. Neuronal Network Es un modelo del funcionamiento cerebral del ser humano, que está conformado por un conjunto de nodos también llamados neuronas artificiales que se conectan y tramiten señales entre sí.

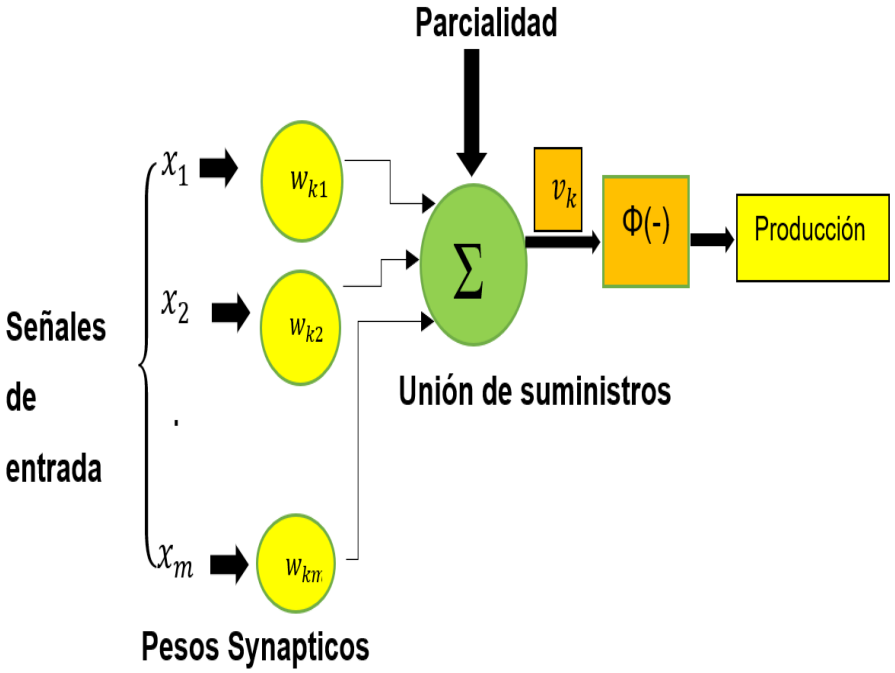


Figura 19. Diagrama de bloques de neurona artificial.
Fuente: *(Zayegh & Al Bassam, 2018)*

1.3.2.3. Ataques de red

1.3.2.3.1. Tipos de ataques

1.3.2.3.1.1. Ataque man in the middle (MitM)

Es la manera en la que el adversario puede escuchar sigilosamente paquetes de la víctima interviniendo en la comunicación, y a su vez alterar el contenido de manera que el usuario no se pueda percatar, por consiguiente los atacantes tienen como objetivo obtener información para realizar actos fraudulentos, como la falsificación de los datos y transferencias de dinero para su propio beneficio, para concluir el adversario en ocasiones al realizar la alteración de los datos, cuyo objetivo

es la de dificultar las comunicaciones y poner en riesgo la confianza de la víctima. (Henrichsen, Betz, & Lisosky, 2015)

1.3.2.3.1.2. Ataque de repetición

Es la manera en la que un atacante intercepta una copia del paquete que ya fue autenticado y lo envía después de un tiempo definido a su víctima, por consiguiente, al recibir los paquetes IP autenticados, pero duplicados puede afectar un servicio como consecuencia de la interrupción de la comunicación. (Stallings, Fundamentos de seguridad en redes: aplicaciones y estándares, 2004)

1.3.2.3.1.3. . Ataque de reconocimiento

Consiste en encontrar y realizar mapeo de aplicaciones, servicios ejecutados sin estar autorizados, dado que aprovechan la debilidad de una red vulnerable, pongamos por caso un ladrón espía una residencial, para ello buscará una casa con la ventana fácil de abrir, una puerta entreabierta o que la casa no esté ocupada con habitantes para poder robar. En conclusión, la finalidad del ataque de reconociendo es ingresar a una red sin ningún permiso interrumpiendo la operatividad. (Ariganello, 2020)

1.3.2.3.1.4. Ataques de acceso

Los ataques de acceso aprovechan la debilidad de los servicios de autenticación, protocolos de transferencia de archivos (FTP) y web, cuyo objetivo del atacante es de ingresar a cuentas de los sitios web, base de datos e información confidencial, haciendo uso de los ataques de diccionario que generalmente es aplicado para descifrar las claves de acceso al sistema. (Ariganello, 2014)

1.3.2.3.1.5. Ataque de denegación de servicios (DoS)

Los ataques de DoS generan una gran cantidad de solicitudes y son enviadas a una red, la cual limita la calidad de servicio de un dispositivo perjudicando el funcionamiento, como resultado dejando inactivo para que puedan acceder y realizar su correcto uso. En conclusión, los ataques DoS tienen como finalidad afectar la comunicación colapsando las aplicaciones y procesos que intervienen. (Ariganello, 2014)

1.3.2.3.2. Técnicas de ataque

1.3.2.3.2.1. Phishing

El phishing conformado por la mezcla de palabras inglesas flashing que traducido al español significa (pescar) y password significa (contraseñas), se basa en la suplantación de identidad organizaciones corporativas, a través de varias técnicas, que usualmente es usado para el envio abundante de correos electrónicos, por consiguiente, suele ser engañoso para los clientes porque a su vista sería aparentemente auténtico a condición de que se le pide al usuario información personal. (Madrid P, 2010)

1.3.2.3.2.2. Spam

Son correos electrónicos no solicitados de dudosa procedencia, enviados de manera automatizada y masivamente a una bandeja de entrada para fines comerciales mediante sistemas de mensajería electrónica, no obstante, son correos recibidos por el usuario sin haber solicitado. (Fundación Telefónica, 2008)

1.3.2.3.2.3. Spoofing

Los ataques spoofing se basa en interponerse en medio de dos sistemas, donde el adversario se encarga de interceptar los paquetes, visualiza los mensajes que se intercambia en la comunicación, además alterando la información que se

comparte una comunicación, de manera que ambos extremos no saben en realidad con quien se están comunicando. (Andreu, Pellejero, & Lesta, 2006)

1.3.2.3.2.4. Spoofing Web

Los ataques de envenenamiento web faculta al adversario acciones como visibilizar y hacer modificaciones en un sitio web que la víctima haya solicitado en el navegador, abarcando las conexiones totalmente seguras de la vía SSL, no obstante a través de código malicioso el adversario realiza una ventana con un aspecto confiable e inofensivo para la víctima, con la finalidad de lograr conseguir información como el historial de páginas visitadas, contraseñas o números de cuentas bancarias. (Villalón H, 2002)

1.3.2.3.2.5. Spoofing IP

Consiste en reemplazar las direcciones IP de inicio de un paquete TCP/IP por una dirección diferente, como consecuencia se suplanta la identidad logrando a través de herramientas que tengan implementado la técnica y en ocasiones alterando los paquetes sin hacer uso de un programa. (Jara & Pacheco, 2012)

1.3.2.3.2.6. Spoofing DHCP

Es un tipo de ataque de intermediario, donde el adversario sondea las peticiones DHCP para lograr responder de manera apresurada antes que un servidor de una red local, en efecto confiere al atacante facilitar configuraciones fraudulentas de red al host receptor, como por ejemplo la puerta de enlace imponiendo a todo el tráfico pasar a la máquina que es manipulado para su beneficio, por consiguiente, el intermediario logra el aprisionamiento y transformación de los datos. (Rocha & Launchbury, 2011)

1.3.2.3.2.7. Spoofing DNS

El ataque de suplantación DNS spoofing hace relación con la falsificación de direcciones IP cuando se realiza una petición de decisión de nombre, cuya finalidad es la de solucionar con una dirección fraudulenta con una denominación DNS o, al contrario, por consiguiente, deducen que puede conseguir varias maneras para alterar las entradas de un servidor que es el encargado de resolver peticiones para lograr que se falsifiquen la comunicación entre dirección y nombre. (Villalón H, 2002)

1.3.2.3.2.8. Spoofing EMAIL

La técnica de email spoofing es mayormente utilizada en la ingeniería social, la cual a través de correos electrónicos engañosos la utilizan para realizar spam y phishing, no obstante, el atacante realiza modificación de la dirección de la persona que envía y el asunto con la finalidad de que tenga una apariencia confiable. (Jara & Pacheco, 2012)

1.3.2.3.2.9. Spoofing ARP

El objetivo del spoofing de ARP es lograr aprisionamiento del tráfico de la víctima sin obligación de colocar en manera promiscua las interfaces de red, como efecto envenenan la tabla del protocolo de resolución de direcciones de los hosts comprometidos en la correspondencia de la cual requieren capturar y alterar el flujo de datos remitidos desde la máquina de la víctima que viaja por la puerta de enlace y se dirija a la máquina del atacante.

Como ejemplo en la presente figura se representa el proceso donde el atacante es la máquina E y se interpone entre dos máquinas que son (D y F) y remite paquetes de tipo arp- reply.

De tal manera que la comunicación entre D y F recorre por la máquina E, de donde resulta que los paquetes son enviados a la dirección MAC 0E:0E: 0E:0E:0E:0E , por consiguiente el flujo de respuesta es persistente para que la lista ARP de ambas máquinas que se están comunicando puedan eludir y que actualice los datos de forma correcta, en conclusión suceso de suplantación es efectivo siempre y cuando los paquetes de D y F sean dirigidos a E que es el atacante, para ilustrar mejor se muestra una representación de lo mencionado anteriormente

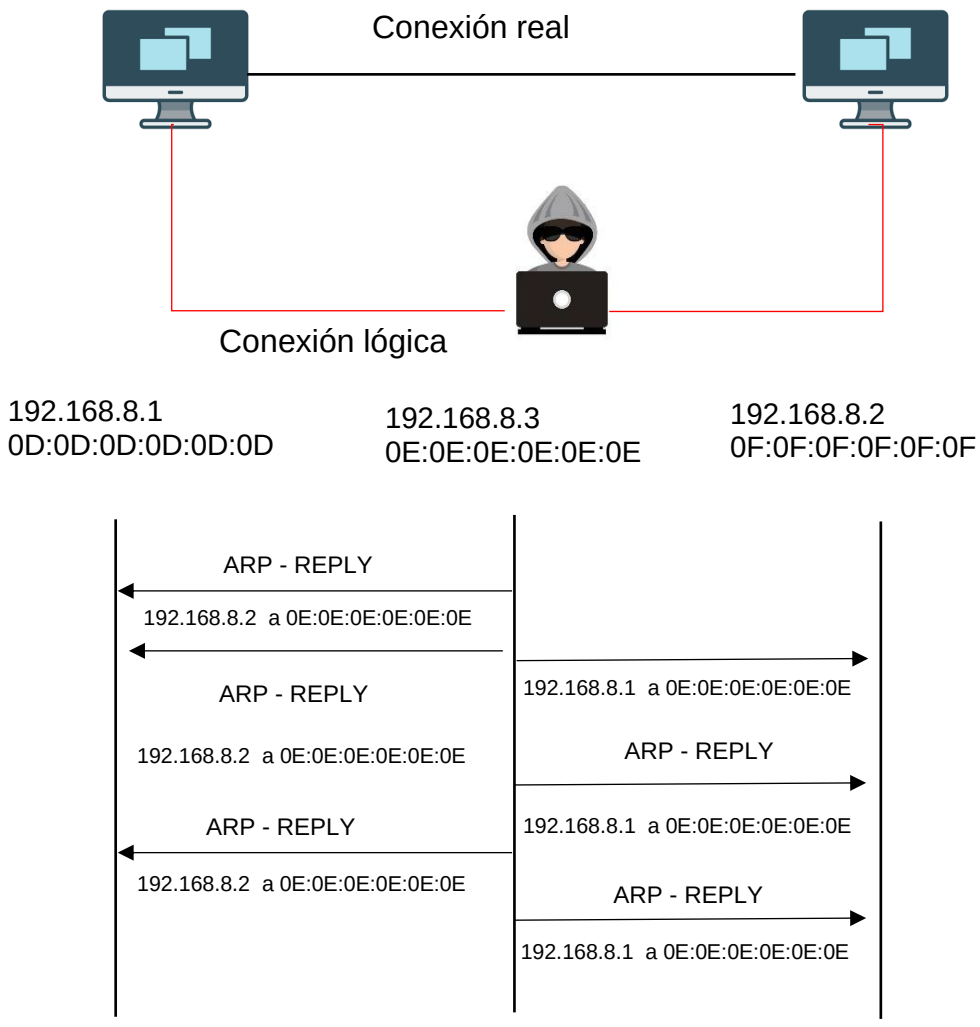


Figura 20. Proceso de envenenamiento ARP, de donde resulta que el atacante se interpone en la comunicación entre dos máquinas. Fuente: (Herrera, Garcia, & Perramón, 2004)

1.4. Formulación del Problema.

¿Cómo identificar en forma eficiente ataques spoofing de envenenamiento ARP en la suplantación de identidad en redes LAN?

1.5. Justificación e importancia del estudio.

Este trabajo investigación se realizó con el propósito de mejorar la privacidad de usuarios ante los ataques de suplantación de identidad, con ello se logrará proteger los sistemas informáticos con total seguridad ante cualquier amenaza cibernética.

Hoy en día las empresas que cuentan con tecnología, son propensas amenazas informáticas, ya que tienen poca información con respecto a seguridad informática lo que genera que las organizaciones empresariales invierten lo mínimo en protección de seguridad.

Hipótesis.

Desarrollando un método automático se podrá identificar en forma eficiente los ataques Spoofing de envenenamiento ARP en la suplantación de identidad de redes LAN.

1.6. Objetivos.

1.6.1. Objetivo general.

Desarrollar un método de identificación automática de ataques Spoofing de envenenamiento ARP en la suplantación de identidad de redes LAN.

1.6.2. Objetivos específicos.

a) Seleccionar los algoritmos de aprendizaje.

b) Elaborar un conjunto de datos para entrenamiento y pruebas.

c) Implementar los algoritmos de aprendizaje automático

d) Evaluar los resultados de obtenidos.

II. MÉTODO

2.1. Tipo y Diseño de Investigación.

2.1.1. Tipo de investigación

El presente trabajo es de tipo cuantitativa, tecnológica y aplicada porque se aplicará conocimientos científicos para brindar apoyo y realizar los objetivos propuestos a resolver, de tal manera que se busca comprobar los resultados más eficientes de cada uno de los algoritmos que se seleccionaron con respecto a la detección de ataques spoofing arp en las redes LAN.

Diseño de investigación

El tipo de diseño empleado en el presente trabajo de investigación es Cuasiexperimental, ya que la muestra será utilizada para obtener buenos resultados, de tal manera se elegirá un grupo de estudio experimental que incluye la detección de ataques de suplantación arp, por consiguiente, la evaluación de los algoritmos de identificación que se integran en el muestreo, se seleccionara los que tienen mejor precisión para detectar los ataques spoofing ARP en las redes LAN.

2.2. Variables, Operacionalización.

2.2.1. Variable Independiente

Algoritmos de identificación automática.

2.2.2. Variable dependiente

Detección de ataques spoofing.

Variables	Dimensión	Indicador	Ítem	Técnica e instrumentos de recolección de datos
Algoritmos de Clasificación	Consumo de Recursos	Grado de consumo de memoria	$C_m = \sum_i^n \frac{cm_i}{n}$	Instrumentos mecánicos o electrónicos / Registro Electrónico
		Grado de consumo de CPU	$C_c = \sum_f^n \frac{cc_j}{n}$	
		Promedio de tiempo de respuesta	$T_r = \sum_f^n \frac{tf_j + tf_i}{n}$	

Detección de ataques de spoofing arp	Rendimiento	Exactitud	$E = \frac{TP + TN}{TP + TN + FP + FN}$
		Precisión	$P = \frac{TP}{TP + FP}$
		Recall	$R = \frac{TP}{TP+FN}$

2.3. Población y muestra.

2.3.1. Población

La población de la presente investigación está compuesta por un top de 14 algoritmos de identificación automática (Ver Anexo N° 01)

2.3.2. Muestra

Se consideró seleccionar un muestreo no probabilístico por conveniencia, para ello se elaboró un top de 14 algoritmos de identificación automática el cual indica la precisión de cada uno de ellos, el cual se puede visualizar en el anexo 01, por consiguiente luego de haber analizado el desempeño de acuerdo a la precisión, se ha elegido 6 de los siguientes algoritmos con mayor porcentaje de acuerdo a la precisión para el muestreo (K-Nearest Keighbors, ,Random Forest, SVC, Logistic Regression, Decisión Tree y Gradient Boosting).

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.

Técnicas e Instrumentos

Instrumentos mecánicos o electrónicos:

Son recursos que permiten realizar una investigación abordando diferentes problemas, será utilizado con el objetivo recopilar información para luego detectar y analizar los ataques spoofing ARP que contenga indicios de presentar ataques de suplantación en una red LAN, de acuerdo a los resultados se evaluará la precisión de cada algoritmo que permitirá conocer cuál tiene mejor rendimiento para la detección de ataques.

Ficha de registro electrónico:

Es un instrumento que facilitara la recolección de los datos que se obtienen al desarrollar los indicadores, lo cual cada uno de ellos proporcionará un resultado, dado que la presente investigación estará evaluada por métricas de rendimiento, por lo que se hará uso de un formato de registro de matriz de confusión, que se visualiza en el Anexo N° 3, la cual registrará los resultados de identificación en un estado normal y un estado de ataque por intermediario (MITM), por consiguiente, se mostrará el registro de métricas de precisión, exactitud y Recall, por otro lado también se hará uso de métricas de consumo de CPU, Memoria y promedio de tiempo de respuesta (Ver Anexo 04, 05, 06) que son proporcionados cuando se carga el dataset a Python, que como respuesta brinda los resultados esperado que corresponde a la investigación.

2.5. Procedimiento de análisis de datos

2.5.1. Consumo de recursos

Con respecto a los algoritmos de identificación automática, a través de la variable del registro electrónico permite medir el consumo de recursos, en efecto evaluando los indicadores, las cuales son basadas en fórmulas matemáticas. Por consiguiente, se describe cada una de ellas a continuación:

Grado de consumo de memoria:

Es el tiempo utilizado para ejecutar los procesos y se representa en la siguiente fórmula:

$$C_m = \sum_i^n \frac{cm_i}{n}$$

A continuación, se muestran las descripciones de las variables de los indicadores de consumo de memoria.

Donde:

C_m: Es el grado de consumo en memoria.

cm_i: Es el grado de consumo de memoria en la prueba i.

n: Es el total de pruebas.

Grado de consumo de CPU:

Es el tiempo utilizado para ejecutar los procesos y se representa en la siguiente fórmula:

$$C_c = \sum_f^n \frac{cc_j}{n}$$

A continuación, se muestran las descripciones de las variables de los indicadores de consumo de CPU.

Donde:

C_c: Es el grado de consumo en memoria.

cc_i: Es el grado de consumo de CPU en la prueba i.

n: Es el total de pruebas.

Promedio de tiempo de respuesta:

Es el tiempo utilizado que tarda para ejecutar los proceso y se representa en la siguiente fórmula:

$$T_r = \sum_f^n \frac{tf_j + tf_i}{n}$$

A continuación, se muestra las descripciones de las variables de los indicadores del tiempo de respuesta:

Donde:

T_r: Es el tiempo de respuesta.

tf_j: Es el tiempo final de respuesta.

tf_i: Es el tiempo inicial de respuesta.

n: Es el total de pruebas.

2.5.2. Rendimiento

Con respecto a los algoritmos de identificación automática para la detección de ataques spoofing ARP, permite medir el rendimiento, en efecto evaluando los indicadores, lo cual se plasma en una matriz de confusión. Por consiguiente, se describe cada una de ellas a continuación:

Exactitud:

Es definida como la media armónica de la precisión, muestra la aproximación de los resultados de cálculo de las predicciones de los ataques de spoofing ARP con relación a un valor verdadero, se representa en la siguiente fórmula:

$$Exactitud = \frac{TP + TN}{TP + TN + FP + FN}$$

A continuación, se muestra las descripciones de las variables de los indicadores de exactitud.

Donde:

TN: Es el total de Negativos Verdaderos.

TP: Es el total de Verdaderos Positivos.

FP: Es el total de Falsos Positivos.

FN: Es el total de Falsos Negativos.

Precisión:

Permite la medición del grado de confiabilidad y proporciona los mismos resultados en cálculos desiguales, pero con un grado de similitud dependiendo de la situación, con respecto a la detección de los ataques de spoofing ARP que se han pronosticado. Este indicador se representa en la siguiente fórmula:

$$Precisión = \frac{TP}{TP + FP}$$

A continuación, se muestran las descripciones de las variables de los indicadores de exactitud:

Donde:

FP: Es el total de Falsos Positivos.

TP: Es el total de Verdaderos Positivos.

Recall:

Permite saber cuán sensible es el algoritmo de acuerdo la cantidad de valores verdaderos positivos de los ataques spoofing ARP que se detectaron correctamente. Este indicador se representa en la siguiente fórmula:

$$Recall = \frac{VP}{FN + VP}$$

A continuación, se muestran las descripciones de las variables de los indicadores de Recall.

Donde:

FN: Es el total de Falsos Positivos.

VP: Es el total de Verdaderos Positivos.

2.6. Criterios éticos.

Confidencialidad: En la presente investigación protege la identidad de los autores que participan como investigadores.

Originalidad: El presente proyecto respeta los derechos de autor, con la finalidad de comprobar que exista plagio, por consiguiente, se respalda la autenticidad de la información utilizada en la investigación a través de citas y referencias.

2.7. Criterios de Rigor Científico.

Consistencia: La investigación consta con datos verdaderos los cuales han sido procesados con formalidad, aplicando conocimiento de metodología de investigación científica, dando consistencia a los artículos mencionados que se han encontrado.

Validez: Se utilizará los indicadores que se han detallado anteriormente en la tabla de Operacionalización, con el objetivo de realizar la medición de las

variables para recolectar datos, y en efecto se pueda validar por un especialista del área.

III. RESULTADOS.

3.1. Resultados en Tablas y Figuras.

Se realizó el análisis de los algoritmos de identificación automático para la exploración de ataques de Spoofing ARP, la cual estos ataques se producen a causa de un adversario, que escucha tráfico interponiéndose entre la comunicación de dos dispositivos en una red local, es por ello que se realizó el análisis de esta investigación, utilizando Anaconda Navigator, que es una suite de código abierto, que contiene aplicaciones y librerías utilizadas en la ciencia de datos. Posteriormente para escribir código se utilizó Spyder, que es un IDE utilizado para la programación científica. Por consiguiente, para el desarrollo se utilizó una portátil HP, con las posteriores propiedades: Microprocesador Intel Core I3: 1.70 giga Hertz , RAM instalada de 8 GB.

3.1.1. Extracto de los resultados

Al entrenar el conjunto de datos, se obtuvieron las medidas de rendimiento de algoritmos de aprendizaje de máquina, de los cuales los indicadores son: Precisión, Recall y Exactitud, logrando obtener los siguientes resultados que se mostrarán a continuación.

Tabla 6.

Medidas de rendimiento de seis clasificadores de aprendizaje de máquina.

Algoritmo Clasificador	Exactitud	Precisión	Recall
k-Nearest Neighbors	97.95%	99.30%	95.94%
Gradient Boosting	99.12%	99.31%	98.64%
DecisionTree	97.81%	98.28%	96.62%
SVC	97.37%	96.33%	97.63%
Random Forest	99.27%	99.32%	98.98%

Logistic Regression	95.62%	93.18%	96.95%

Nota: Las medidas de rendimiento fueron calculadas usando la misma base de datos de entrenamiento y prueba. Fuente: Elaboración propia

Los datos que se visualiza en la tabla x, muestran que los algoritmos k-Nearest Neighbors , Gradient Boosting y Random Forest obtuvieron buenos resultados en cuanto a la precisión mide el porcentaje de los registros que son de categoría suplantado que se predijo de manera correcta, de los cuales el algoritmo Random Forest es más exacto para la detección de ataques Spoofing ARP, alcanzando un 99.32% de precisión, para la cual se realizó una gráfico comparativo de los tres algoritmos, que se mostrará en la figura 21.

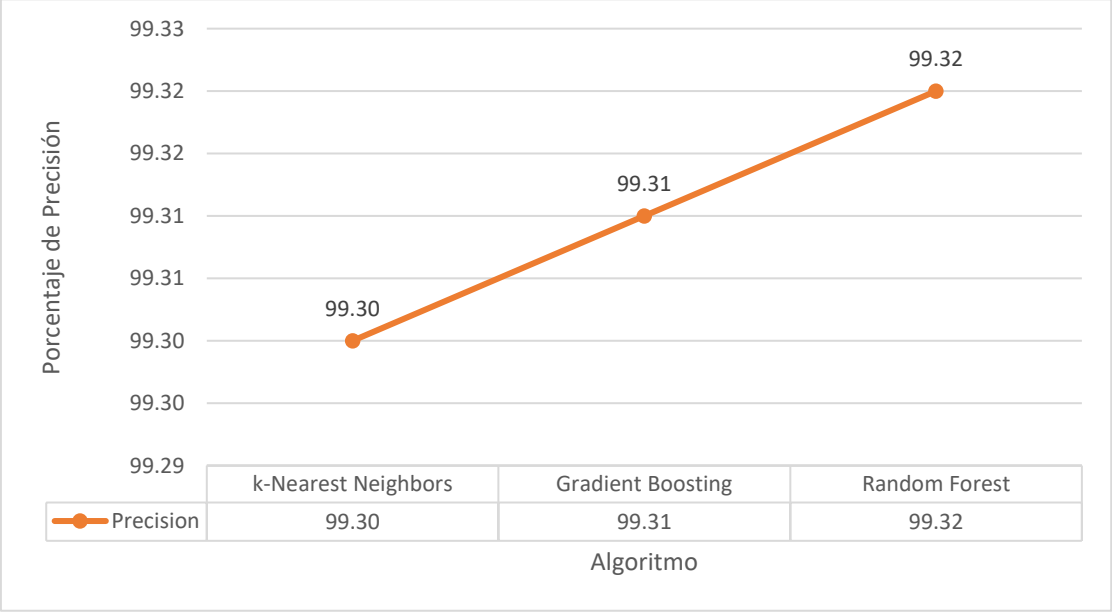


Figura 21. Porcentaje de precisión de los algoritmos de clasificación automática Fuente: Elaboración Propia

Al entrenar el conjunto de datos, se obtuvieron los resultados utilizando métricas de rendimiento en relación a los recursos de consumo de los algoritmos de clasificación automática, de las cuales se utilizó Promedio de tiempo de respuesta, grado de consumo de memoria y grado de consumo de CPU, logrando obtener los siguientes resultados que se mostraran a continuación.

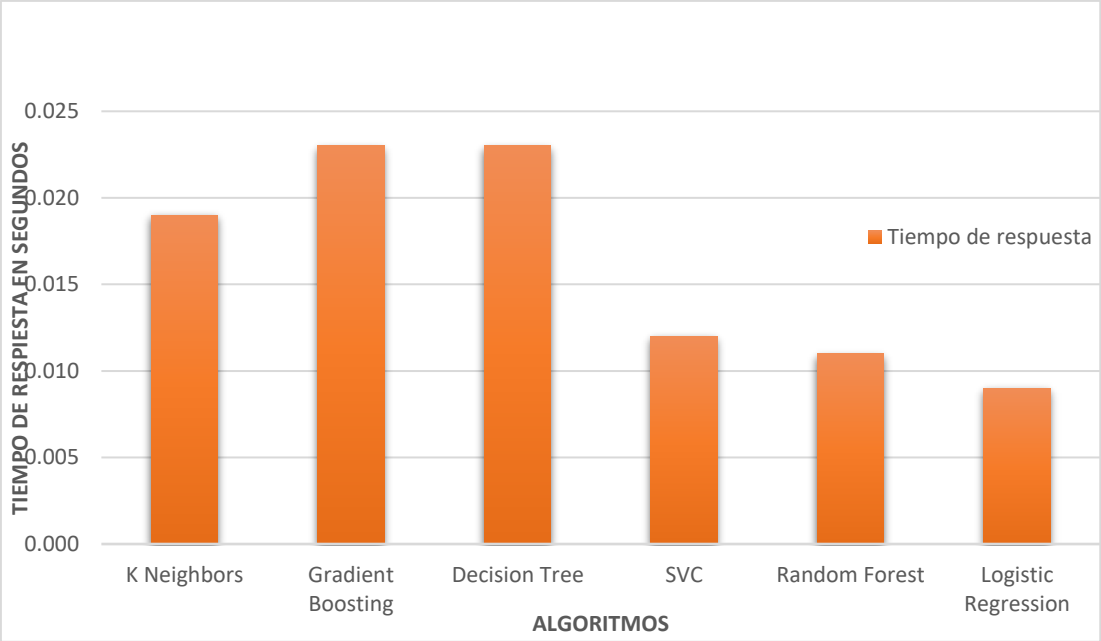
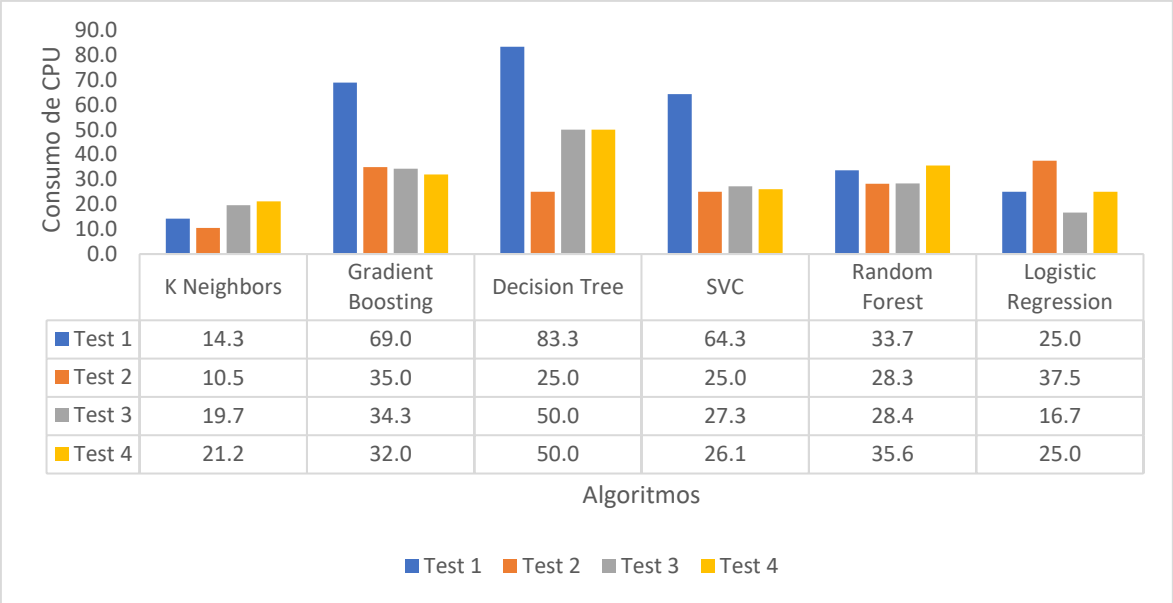


Figura 22. Promedio de tiempo de respuesta de los algoritmos
Fuente: Elaboración Propia

En la Figura 22, se muestra el tiempo promedio por clasificadores de acuerdo al entrenamiento que se realizó, de los cuales el algoritmo Random Forest obtuvo un tiempo de 0.011 segundos, siendo un algoritmo certero al momento de entrenar, dando estimaciones de las variables que son importantes al momento de clasificar. Por consiguiente, el algoritmo Neighbors, es el segundo mejor clasificador, obtuvo un 0.19 segundos, ya que es un algoritmo cuyo objetivo es clasificar de manera correcta las instancias nuevas, y por último el tercer mejor algoritmo Gradient Boosting obtuvo 0.023 segundos, siendo útil al momento de explorar, permitiendo detectar de manera rápida las variables predictoras que son muy importante.



	K Neighbors	Gradient Boosting	Decision Tree	SVC	Random Forest	Logistic Regression
Test 1	14.3	69.0	83.3	64.3	33.7	25.0
Test 2	10.5	35.0	25.0	25.0	28.3	37.5
Test 3	19.7	34.3	50.0	27.3	28.4	16.7
Test 4	21.2	32.0	50.0	26.1	35.6	25.0

Figura 23. Grado de consumo de CPU de los algoritmos
Fuente: Elaboración Propia

En la figura 23, se aprecia los resultados al realizar 4 test de entrenamiento, de los cuales, se tomará en cuenta los algoritmos que obtuvieron mujer precisión, tal como se observa en la figura 24, el algoritmo K Neighbors resultó ser uno de los algoritmos con menor consumo de CPU alcanzando un porcentaje de 10.5 % en el test 2, posteriormente el algoritmo Random Forest obtuvo el 28.3% y por consiguiente el algoritmo como mayor porcentaje en consumo de CPU, fue Gradient Boosting

3.1.2. Resultados del algoritmo Bosque Aleatorio

De un total de 696 de los registros el algoritmo Random Forest logró detectar 391 registros con estado normal y 295 registros que son ataques Spoofing ARP. Por consiguiente, de todos los registros con estado normal que se predijeron 388 fueron correctos y 3 fueron erróneos, debido a que el clasificador lo predijo como normal, siendo en realidad un registro con ataque de Spoofing ARP. Por otro lado, los ataques de spoofing Arp que se predijeron fueron detectados correctamente con un total de 293 registros, y 2 con error, ya que el algoritmo lo clasifico como ataque spoofing Arp, siendo en realidad un registro con estado normal.

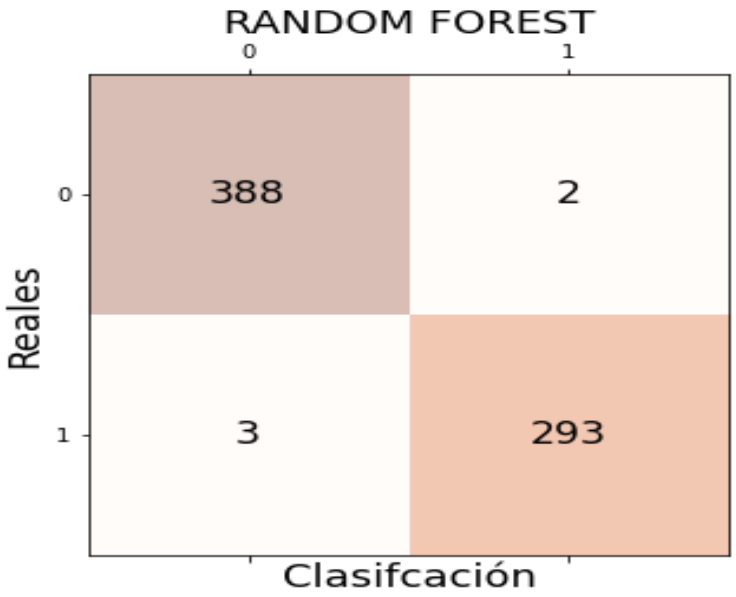


Figura 24. Matriz de confusión Random Forest
Fuente: Elaboración Propia

Para obtener las medidas de rendimiento, se utilizó las siguientes fórmulas para hallar la precisión, exactitud y Recall

$$\text{Precisión} = \frac{388}{(388+2)} = 0.9932$$

$$\text{Exactitud} = \frac{388+293}{(388+2\ +3\ +293)} = 0.9928$$

$$\text{Recall} = \frac{388}{(388+3)} = 0.9932$$

3.1.3. Resultados del algoritmo Vecinos más cercanos

De un total de 696 de los registros el algoritmo Random Forest logró detectar 400 registros con estado normal y 286 registros que son ataques Spoofing ARP. Por consiguiente, de todos los registros con estado normal que se predijeron 388 fueron correctos y 12 fueron erróneos, debido a que el clasificador lo predijo como normal, siendo en realidad un registro con ataque de Spoofing ARP. Por otro lado, los ataques de spoofing Arp que se predijeron fueron detectados correctamente con un total de 284 registros, y

2 con error, ya que el algoritmo lo clasifico como ataque spoofing Arp, siendo en realidad un registro con estado normal.

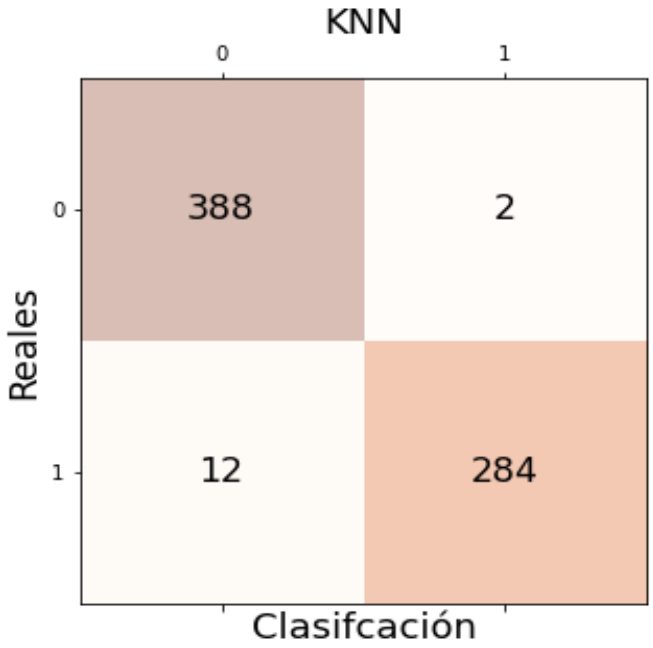


Figura 25. Matriz de confusión Vecinos más cercano

Fuente: Elaboración Propia

Para obtener las medidas de rendimiento, se utilizó las siguientes fórmulas para hallar la precisión, exactitud y Recall

$$\text{Precisión} = \frac{388}{(388+2)} = 0.9930$$

$$\text{Exactitud} = \frac{388+293}{(388+2\ +12+284)} = 0.9795$$

$$\text{Recall} = \frac{388}{(388+12)} = 0.9594$$

3.1.4. Resultados del algoritmo Gradient Boosting

De un total de 696 de los registros el algoritmo Random Forest logró detectar 402 registros con estado normal y 294 registros que son ataques Spoofing ARP. Por consiguiente, de todos los registros con estado normal que se predijeron 388 fueron correctos y 4 fueron erróneos, debido a que el clasificador lo predijo como normal, siendo en realidad un registro con ataque de Spoofing ARP. Por otro lado, los ataques de spoofing Arp que se predijeron fueron detectados correctamente con un total de 292 registros, y

2 con error, ya que el algoritmo lo clasifico como ataque spoofing Arp, siendo en realidad un registro con estado normal.

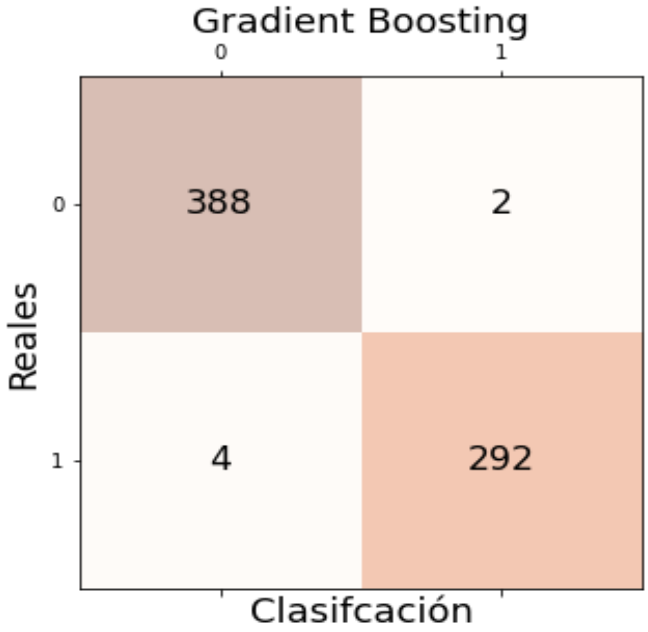


Figura 26. Matriz de confusión Gradient Boosting
Fuente: Elaboración Propia

Para obtener las medidas de rendimiento, se utilizó las siguientes fórmulas para hallar la precisión, exactitud y Recall

$$\text{Precisión} = \frac{388}{(388+2)} = 0.9931$$

$$\text{Exactitud} = \frac{388+292}{(388+2\ +4+292)} = \ 0.9912$$

$$\text{Recall} = \frac{388}{(388+4)} = 0.9864$$

3.2. Discusión de resultados.

Se evaluó los algoritmos de aprendizaje supervisado, utilizando las métricas de rendimiento, que se proporcionó en una representación visual del rendimiento de los algoritmos al haber clasificado los ataque de Spoofing ARP.

Los resultados obtenidos de los algoritmos con mayor rendimiento son KNN, Gradient Boosting y Random Forest, que se muestran en las Figuras N.º 24,25 y 26. Por consiguiente a través de la matriz de confusión, que permitió visualizar el desempeño de los algoritmos, se procedió a realizar el calculó de la exactitud, precisión, y Recall.

En la investigación de (Jerry John, Justice Owusu, & Griffith, 2020), se utilizó para entrenamiento el 80% del conjunto de datos y el 20% para evaluar el rendimiento de cada modelo , compuesto por un total de 5300 filas, a diferencia del conjunto de datos de la presente investigación que utilizó 2285 registros , utilizando para el entramiento el 70% y el 30 % para el rendimiento, la cual fue asignado el nombre de SpoofingArp2021, logrando obtener una precisión de 99,32 % y los investigadores del artículo científico, lograron obtener el 99.44%, debido a que solo utilizaron 2 características para el entrenamiento, logrando así obtener mejores resultados, a diferencia del conjunto de datos que se creó para esta investigación, la cual empleo más características para el análisis.

El promedio para el algoritmo de Random Forest , en precisión es del 99,32 %, exactitud 99.27% y Recall del 98.98%, con un tiempo de respuesta de 0.011 segundos, obteniendo así los mejores resultados, mientras que k-Nearest Neighbors obtuvo una precisión de 99.30%, una exactitud de 97.95% y Recall con un 95.94%, con un tiempo de respuesta de 0.19 segundos, por otro lado el algoritmo Gradient Boosting tan solo alcanzo el 99.31%, , exactitud de 99.12% y Recall de 98.64%, alcanzando un tiempo de respuesta de 0.023 segundos.

3.3. Aporte práctico

3.3.1. Método propuesto para el desarrollo de investigación

El método de desarrollo que se utilizó en esta investigación consiste de las presentes fases:

Fase de selección de algoritmos: Se realizó un análisis minucioso, para evaluar la eficiencia de algoritmos de identificación automática en la detección de ataques spoofing Arp, estimando los resultados de distintos algoritmos, usando métricas de desempeño, utilizando varios artículos, visitados en las bases de datos Ieee, ScienceDirect y Scopus, conformando un ranking de los 6 algoritmos con mejor precisión de clasificación para la estimación en el desarrollo propuesto.

Fase de elaboración del Conjunto de datos: Recopilando los registros en diferentes tipos de escenario, creando así el nuevo conjunto de datos, denominado con nombre (spoofingArp2021) de spoofing ARP, con 3 características y una etiqueta, la cual se utiliza para el análisis de los datos en lenguaje de programación Python, posteriormente se aplicó la normalización de los datos para realizar el estudio del desarrollo propuesto.

Fase de Diseño del método: Se seleccionó 6 algoritmos de aprendizaje automático con mayor precisión en la detección de ataques Spoofing ARP, la cual se implementará en lenguaje de programación Python para la evaluación de desempeño en esta investigación.

Fase de Evaluación: En esta fase se recopila los resultados obtenidos del analisis de 6 los clasificadores (K-Nearest Keighbors, Random Forest, SVC, Logistic Regression, Decisión Tree y Gradient Boosting), usando métricas de rendimiento como son: la exactitud, Recall y precisión y en las métricas de Consumo de recursos que son: Promedio de tiempo de respuesta, Grado de consumo de memoria, Grado de consumo de CPU.

A continuación, se muestra un esquema del desarrollo de un método de identificación automática para detectar los ataques de suplantación ARP

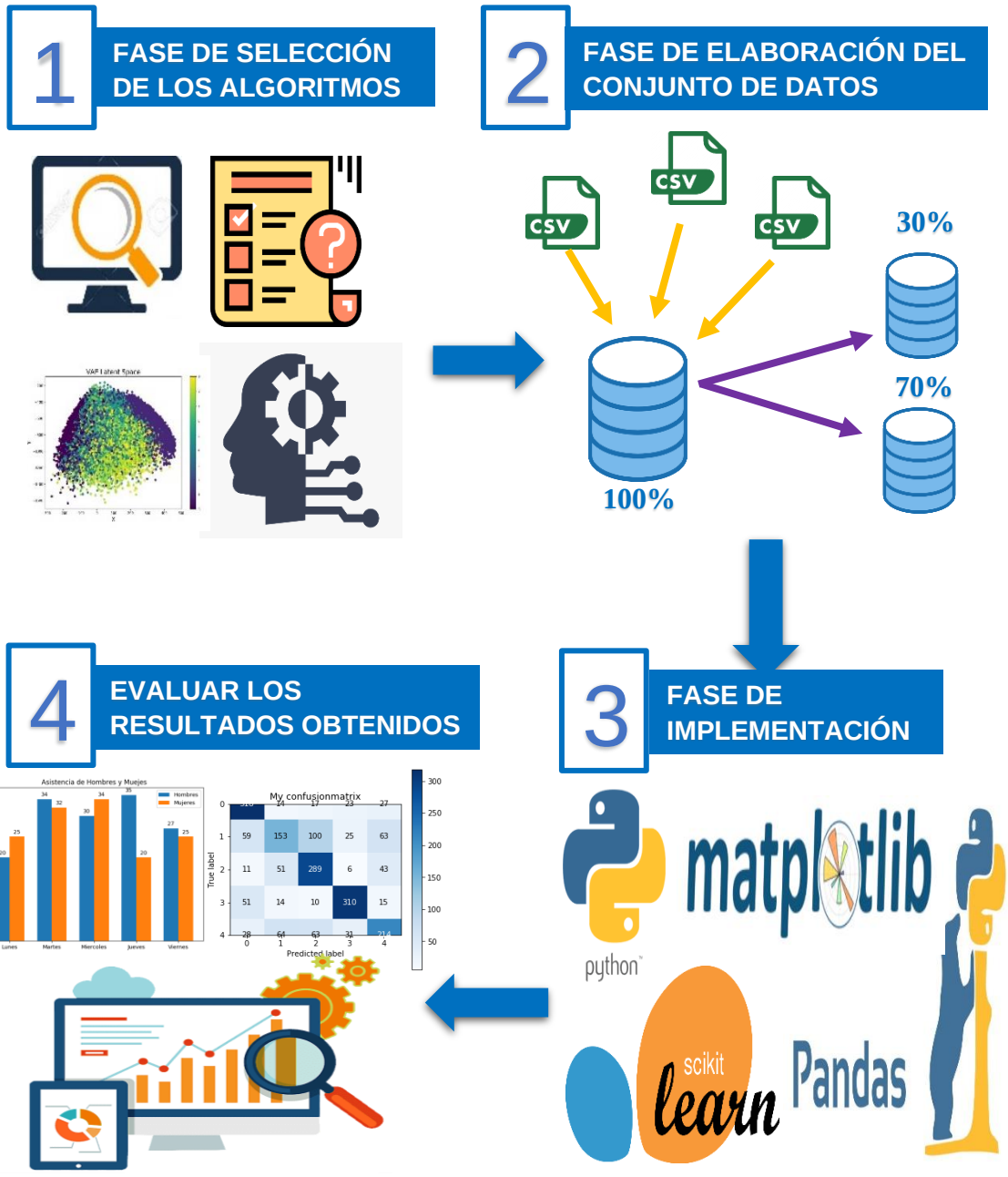


Figura 27. Diseño del método de desarrollo propuesto para identificación automática de ataques spoofing ARP.
Fuente: Elaboración Propia

3.1.1.1. Fase de selección de los algoritmos

En esta investigación se realizó una revisión de la literatura científica publicadas con una antigüedad de 5 años hasta el año 2021, la cual fueron extraídas de la base de datos de la IEEE Xplore, Sciencedirect y Scopus para ello se utilizó las palabras clave como “Arp” , “Machine Learning”, utilizando los operadores booleanos “AND”, “OR”, dicha cadena de búsqueda encuentra varios tipos de artículos en los temas necesarios para la investigación, de tal manera que en Scopus se encontraron 2 artículos, de los cuales uno se utilizó a fondo para el estudio, para la cual se utilizaron las siguientes bases de datos:

Tabla 7.

Bases de datos utilizadas

N°	Base de datos	URL
1	IEEE Xplore	https://ieeexplore.ieee.org/
2	SCOPUS	https://www.scopus.com/
3	Science Direct	https://sciencedirect.com/

Nota: Elaboración propia

Después de realizar la revisión de los artículos científicos, en base a los algoritmos de aprendizaje automática con mejor desempeño en precisión se encontró lo siguiente:

Tabla 8.

Algoritmos de aprendizaje automático con mejor desempeño en precisión

N°	Algoritmo	Precisión	Aplicación	Autor	Año
1	k-Means	79,33%	Predicción de lealtad de clientes	Sardjoeni Moedjiono, Yosianus Robertus Isak, Aries Kusdaryono.	2016
2	Logistic Regression	99,62 %		Jerry John Kponyo, Justicia	
3	Gradient Bosting	99,34 %	Detección de ataques de	Owusu Agyemang, y Griffith Selorm	2020
4	Naive Bayes	99,72%	intermediario	Klogo	
5	Decision Tree	99,34%	basado en el análisis del		
6	Random Forest	99,44%	protocolo arp		
7	k-Nearest Neighbors	99,34%			
8	Perceptron	95%	Detección de Malware en Android	Anıl UTKU, İbrahim Alper DOĞRU, M. Ali AKCAYOL	2018
9	Regression linear	82 %	Predicción del modelo para el mercado	Hiral R. Patel , Satyen M. Parikth Dhara N. Darji	2018

N°	Algoritmo	Precisión	Aplicación	Autor	Año
10	AdaBoost	98,8%	Detección de sitios web falsificados	Ekta Gandotra y Deepak Gupta	2020
11	Neuronal Network	97.67%	Detección de sms spam	Amani Alzahrani and Danda B. Rawat	2019
12	Super Vector Machine (SVM)	94.26%			
13	Binary Logistic Regression	98%,	Datos médicos	Qais Abdulqader	2017
14	Multiclass Logistic Regression	98,71	Predicción de peligro de incendios forestales	Lei Wang, Qingjian Zhao, Zuomin Wen, Jiaming Qu	2018

Nota: Top de algoritmos de clasificación autmatica con mejor precisión extraído de las bases de datos IEEE, Scopus, ScienceDirect. Fuente: Elaboración propia.

De los 14 algoritmos de aprendizaje automático con mejor precisión en diferentes aplicaciones, se seleccionó 06 algoritmos con mejor desempeño en precisión para la detección de ataques spoofing ARP.

Tabla 9.

Algoritmos de aprendizaje automático seleccionados

Nº	Algoritmo	Aplicación
1	Logistic Regression	Ataques spoofing ARP
2	Gradient Bosting	Ataques spoofing ARP
3	SVC	Ataques spoofing ARP
4	Decision Tree	Ataques spoofing ARP
5	Random Forest	Ataques spoofing ARP
6	k-Nearest Neighbors	Ataques spoofing ARP

Nota: Los algoritmos fueron seleccionados en base a los resultados mostrados en un artículo de Detección de ataques de intermediario basado en el análisis del protocolo ARP Fuente: Elaboración propia.

3.1.1.2. Fase de elaboración del conjunto de datos

Para el desarrollo de esta investigación se llevó a cabo la creación de un conjunto de datos, obteniendo los registros desde la herramienta Wireshark, que es utilizado para la captura de trafico de red, bajo un criterio de filtro ARP, para capturar los paquetes necesarios para el estudio, se realizó 7 la extracción de los registros, en estado normal y de suplantación, la cual las tramas ARP que contienen una longitud de 42 bytes son de estado suplantado, y las tramas con longitud de 60 bytes son de tamaño real, luego se exporto la data en formato .csv, la cual se construyó el dataset con 2285 filas

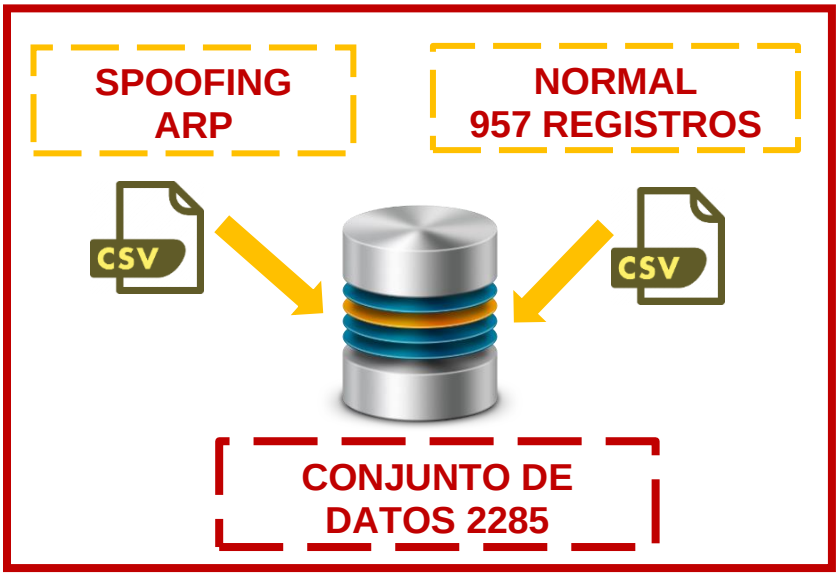


Figura 28. Esquema de elaboración del conjunto de datos
Fuente: Elaboración Propia

Con la recopilación de las capturas de tráfico, se exportó en formato .csv los archivos que conforman el conjunto de datos final, se obtuvo un total de 2285 registros, recopilados con ataque de suplantación arp, que conforma el 58,12 % y en estado normal que conforma el 41,88 % cómo se observa en la figura 22. Para el desarrollo del método propuesto, se tomaron el 70% de los registros para entrenamiento y el 30% restante para pruebas

3.1.1.2.1. Eliminación de las columnas

Para entrenar el conjunto de datos, se procedió a eliminar la columna protocolo, puesto que este indicaba el nombre ARP, esto quiere decir que es un dato innecesario, considerando que columnas son las más aptas para el clasificar ataques de spoofing ARP.

```
import pandas as pd
df = pd.read_csv
("spoofingArp2021.csv")
df.drop(['protocolo']
,axis='columns',inplace=True)
```

Figura 29. Script para eliminar columnas innecesarias. Fuente: Elaboración Propia

Dicho lo anterior, quedando las columnas tiempo, ip_origen, ip_destino, longitudCat, que se utilizaría para el desarrollo del proyecto.

Tabla 10.

Columnas a utilizar después de la eliminación.

Tiempo	ip_origen	ip_destino	longitudCat
1781254000	3232235797	16843009	42
311462000	28295825933	3232235797	60
11279000	3232235797	28295825933	42
6233000	3232235797	16843009	42
833000	3232235857	3232235797	60

Nota: Elaboración propia

3.1.1.2.2. Transformación de los datos

Para proceder con la transformación, se utilizó la librería preprocessing, que es un paquete que sirve para cambiar características sin procesar, para poder tener mejores estimaciones en los resultados, es por ello que después definimos un objeto con nombre encoder y llamando al codificador LabelEnconder que cumple

la función de transformar características categóricas a valores numéricos entre 0 y 1, para ello luego se programó un script, para la transformación de la columna de longitudCat, ya que contenía características categóricos, siendo necesaria realizar su transformación para el entrenamiento.

```
import pandas as pd
from sklearn import preprocessing
df = pd.read_csv
("spoofingArp2021.csv")
df.drop(['protocolo'],axis='columns'
,inplace=True)

code_atr=preprocessing.LabelEncoder()
df['longitudCat']= code_atr.fit_transform
(df['longitudCat'])
```

Figura 30. Script para la transformación de la columna longitudCat.
Fuente: Elaboración Propia

Posteriormente se ejecutó el script en Python, la cual es mostrada a continuación con un antes y después.

Tabla 11.
Antes y después de la transformación de la columna categoría.

longitudCat	longitudCat
42	0
60	1
42	0
42	0
60	1

Fuente: Elaboración Propia

3.1.1.2.3. Conversión de las columnas ip_origen, ip_destino.

Se realizó la conversión de la columna ip origen y ip destino, ya que las direcciones ip estándar están en "base 256", por cual normalmente es una cadena que está separado por 4 números, cumpliendo un rango entre 0 a 255, siendo estos separados a través de puntos, es por esto que se realizó la conversión a número decimal entero en base 10 para poder realizar el entrenamiento, dichas columnas contenían dirección ip de origen,ip_destino teniendo en cuenta que estas columnas contiene datos importantes como la dirección ip de atacante y por otro lado la dirección ip de la víctima y con la ayuda de una hoja de cálculo, se separó la columna del conjunto de datos para poder realizar la transformación de los valores actuales, sin antes realizar la conversión con ayuda de un script, utilizando la librería ipaddress, el cual permitió la conversión.

```
import ipaddress
print('---CONVERTIR DIRECCION IP A NUMERICO---'
print('La conversión es:',
      int(ipaddress.ip_address('192.168.1.12'))
print('La conversión es:',
      int(ipaddress.ip_address('192.168.1.29'))
print('La conversión es:',
      int(ipaddress.ip_address('192.168.1.18'))
print('La conversión es:',
      int(ipaddress.ip_address('192.168.1.8')))
```

Figura 31. Script para conversión manual de las direcciones ip de la columna ip_origen a numerico. Fuente: Elaboración Propia

Ahora para conocer cómo funciona la conversión, a continuación, se mostrará la siguiente fórmula:

$$(a \times 256^3) + (b \times 256^2) + (c \times 256^1) + (d \times 256^0)$$

En donde a, b, c, d representará al número decimal de cada octeto, pongamos por caso, la dirección ip es 192.168.1.8, si aplicamos la formula quedaría así:

$$(192 \times 256^3) + (168 \times 256^2) + (1 \times 256^1) + (8 \times 256^0)$$

$$3221225472 + 11010048 + 256 + 8 = 3232235784$$

De donde resulta que 192.168.1.8, es lo mismo que 3232235784, ahora bien, para comprobar se realizó un ping al resultado de la conversión.

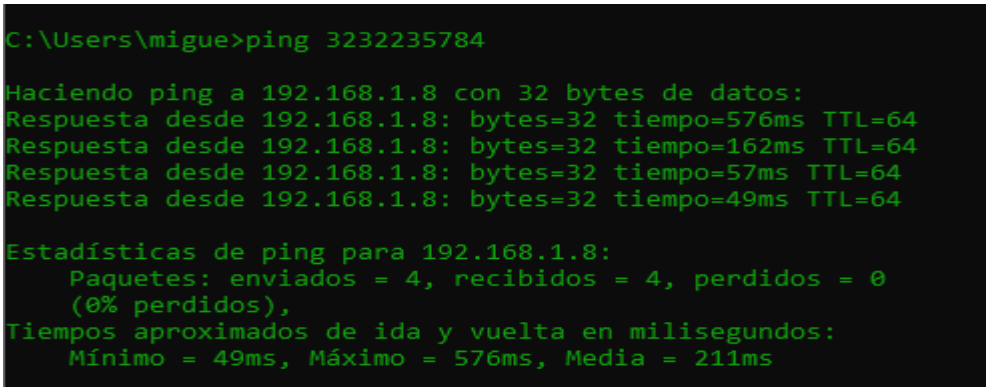


Figura 32. Comprobación del resultado de la conversión. Fuente: Elaboración Propia

Como se afirmó arriba, después de haber culminado con la conversión de los datos de la columna ip_origen, se actualizó la columna del conjunto de datos, quedando de esta manera.

Tabla 12.

Comparación antes y después de la conversión.

IP ORIGEN	IP ORIGEN
192.168.1.23	3232235799
192.168.1.29	3232235805
192.168.1.18	3232235794
192.168.1.1	3232235777

Nota: Elaboración propia

3.1.1.2.4. Normalización de los datos

Para proceder con la normalización, se utilizó la técnica de Min-Max, ya que tiene una característica especial, que es la de conservar la relación de los datos reales, la normalización se refiere a la escala de los datos de las variables numéricas en un rango de 0 a 1, es por ello que se procedió a normalizar la columna de tiempo, que contiene el tiempo de respuesta en la que los paquetes fueron obtenidos, por consiguiente, al construir el conjunto de datos el tiempo es exportado en formato numérico, no obstante también se normalizo las columnas de tiempo, ip_origen y ip_destino por consiguiente se procedió a su normalización y así no perder relación con los datos actuales**.**

Tabla 13.

Conjunto de datos antes de la normalización.

Tiempo	Ip_origen	Ip_destino
1781254000	3232235797	16843009
311462000	28295825933	3232235797
11279000	3232235797	28295825933
6233000	3232235797	16843009
833000	3232235857	3232235797

Fuente: Elaboración Propia

Ahora para conocer el proceso de cómo funciona la normalización, a continuación, se mostrará la siguiente fórmula:

$X_{nuevo} = (X - X_{min}) / (X_{max} - X_{min})$

Dónde

X: Es un conjunto de los valores observados presentes en X.

X min : Son los valores mínimos en X

X max : Son los valores máximos en X

Pongamos por caso, que el primer valor de tiempo sea = 833000, segundo valor sea = 11279000, y el tercer valor sea 1781254000.

Primer proceso: 833000

$$Min\ Max = \frac{(833000 - 833000)}{(1781254000 - 833000)}$$

$$Min\ Max = \frac{(0)}{(1780421{,}000)}$$

$$Min\ Max = 0$$

Segundo proceso: 11279000

$$Min\ Max = \frac{(11279000 - 833000)}{(1781254000 - 833000)}$$

$$Min\ Max = \frac{(10446000)}{(1780421000)}$$

$$Min\ Max = 0.005867$$

Luego de haber culminado la normalización de las columnas del conjunto de datos, se procedió a implementar en lenguaje de programación los algoritmos de aprendizaje de maquina

Tabla 14.
Conjunto de datos después de la normalización

Tiempo	Ip_origen	Ip_destino
0.059384	0.013653	0.000000
0.010383	0.863474	0.099531
0.000376	0.013653	0.875361
0.000207	0.013653	0.000000
0.000027	0.013653	0.099531

Fuente: Elaboración Propia

Concluyendo con la normalización de las columnas del dataset, se muestra a continuación el código escrito para realizar la n

```
import pandas as pd
from sklearn import preprocessing
df = pd.read_csv
("spoofingArp2021.csv")
df.drop(['protocolo'],axis='columns',inplace=True)

df['ip_destino']= (df['ip_destino']
-df['ip_destino'].min())/(df['ip_destino'].max()
-df['ip_destino'].min())

df['tiempo']= (df['tiempo']
-df['tiempo'].min())/(df['tiempo'].max()
-df['tiempo'].min())

df['ip_origen']= (df['ip_origen']
-df['ip_origen'].min())/(df['ip_origen'].max()
-df['ip_origen'].min())
```

Figura 33. Script para la normalización de las columnas del conjunto de datos. Fuente: Elaboración Propia

3.2. Fase de implementación de los algoritmos de identificación automática

En esta sección, se muestra el modelo general para la detección de ataques spooofing ARP, por la cual está compuesto en base a la recolección de los datos, por consiguiente se inició seleccionando la interfaz de red a la que se estuvo conectado , la cual fue interceptado las comunicaciones a través de la herramienta wireshark, a partir de la entrada de los datos de una trama ARP, posteriormente el análisis que está compuesto del algoritmo de aprendizaje automático que clasifican los datos proporcionados.

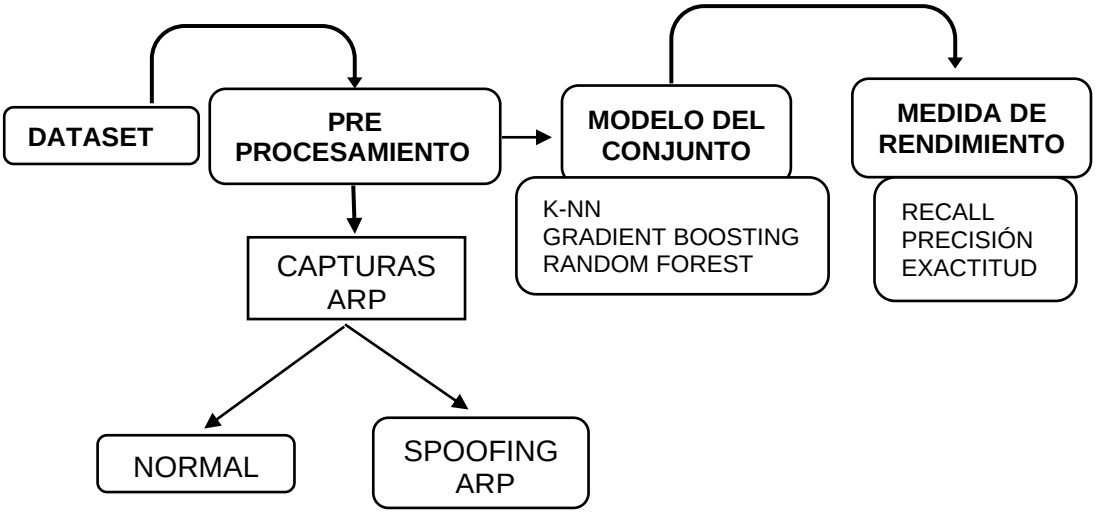


Figura 34. Esquema de funcionamiento del modelo propuesto para la detección de ataques Spoofing ARP.

Fuente: Elaboración Propia

3.2.1.2.1. Implementación del algoritmo Random Forest

El algoritmo bosques aleatorios, combina cientos de árboles de decisión, para luego realizar un entrenamiento de cada árbol de decisión en una muestra distinta de las observaciones, por lo tanto, sus predicciones finales, se realiza haciendo un promedio de las predicciones de cada árbol en particular después de haber logrado seleccionar características que son muy importantes del conjunto de datos.

Tabla 15.

Funcionamiento de algoritmo Random Forest.

Nª	Funcionamiento
1	Seleccionar muestras de manera aleatoria del dataset
2	Crea un árbol decisión en cada de las muestras que son seleccionados, por consiguiente, se muestra el resultado de la predicción
3	Realiza una votación para los resultados obtenidos
4	El algoritmo selecciona el resultado más votado como predicción final.

Fuente: Elaboración propia

A continuación, se mostrará una imagen del funcionamiento del algoritmo random forest.

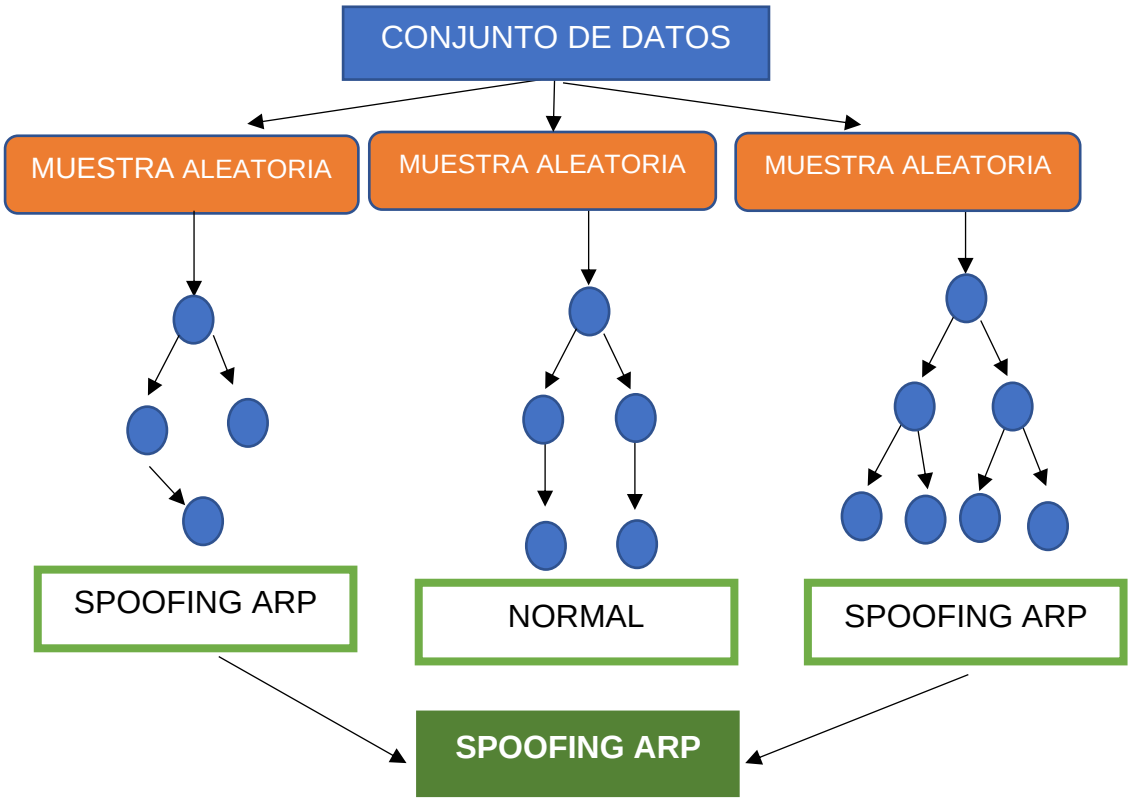


Figura 35. Funcionamiento del algoritmo Random Forest para la identificación automática de ataques spoofing ARP.
Fuente: Elaboración Propia

a) Formula del algoritmo

Los bosques aleatorios son un algoritmo de fácil aprendizaje, ya que crea múltiples árboles de decisión, y los mezcla para lograr que las predicciones sean precisas y permanentes.

Para determinar el sobreajuste es necesario la mezcla de todas las predicciones del modelo de entrenamiento, con el fin de obtener una predicción única de los ataques de spoofing ARP. Para esto es necesario un conjunto de variables no relacionales y sus respectivas varianzas.

Promedio:

$$E\left[\frac{1}{n}\sum_{i=1}^{n} E[Y_i] = \frac{1}{n}.n\mu = \mu\right]$$

Donde:

n= número de predicciones.

μ=total de predicciones.

Varianza reducida:

$$Var\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right) = \left(\frac{1}{n}\right)^2 \sum_{i=1}^{n} Var(Y_i) = \frac{1}{n^2}.n\sigma^2 = \frac{\sigma^2}{n}$$

Donde:

n= Número de predicciones.

μ= Total de predicciones.

Var= Varianza.

Y_i= Predicciones.

b) Código del algoritmo

Tabla 16.

Código de implementación para el modelo Random Forest.

Random Forest

```
#Entrenamiento del modelo
A_entrenar, A_prueba, b_entrenar, b_prueba = train_test_split(A,b,test_size=0.30,ra
ndom_state= 70)
print(A_entrenar.shape, A_prueba.shape, b_entrenar.shape, b_prueba.shape))

clasificador = RandomForestClassifier(n_estimators=150,max_depth=7,random_stat
e=0).fit(A_entrenar, b_entrenar)
predecir = clasificador.predict(A_prueba)
```

Fuente: Elaboración Propia

En la tabla 16, se muestra el código escrito en lenguaje de programación, para el entrenamiento del conjunto de datos, importando la librería del algoritmo clasificador, y luego se carga los datos a entrenar, posteriormente realizamos un ajuste a los parámetros, para ello, le indicamos el número de árboles con un total de 100 parar tener buena calidad con respecto a la estimación para identificar ataques spoofing ARP, por ultimo también se utilizamos el parámetro Max_depth para establecer el número máximo de niveles en cada uno de los árboles de decisión. Por consiguiente, se realiza el entramiento del modelo, para luego imprimir los resultados de las métricas establecidas.

3.2.1.2.2. Implementación del algoritmo K-vecinos más próximos

Es utilizado para solucionar problemas de clasificación, siendo un modelo de los más fáciles y con características intuitivas. De manera que su modo de aprendizaje es registrar ejemplos a la base de entrenamiento a través de su función de distancia y cifras de los vecinos más aproximados, como resultado, pronóstica de que categoría es el individuo. Desde otro punto de vista la eficiencia de aprendizaje es acelerado, aunque el tiempo que tarda en realizar la predicción de la distancia entre la instancia demora, siendo un método elevado con respecto al tiempo para predecir bases de datos como mayor volumen, a menos que puedan equilibrarlas para obtener resultados eficaces.

a) Formula del algoritmo.

Llegados a este punto el algoritmo K-NN se encarga de calcular la similitud entre las distancia de los objetos que están dentro de un conjunto de aprendizaje, (x,y) $\in$ I , y el objeto de prueba $z = (\hat{x}, \hat{y})$ para obtener con precisión la lista de vecinos que están cerca de I, 'x' simboliza el objeto de aprendizaje , 'y' simboliza la clase de entrenamiento. $(\hat{x}, \hat{y})$, representan el objeto de prueba y su clase. Para comprender mejor el algoritmo se puede sintetizar de esta forma:

Entrada:

Objeto de entrenamiento (x, y) ∈ I y objeto de prueba z =$(\hat{x}, \hat{y})$ ∈ I .

Proceso:

Calcular distancia d =$(\hat{x}, x)$ entre z y cada objeto (x, y) ∈ I.

Salida (Clase mayor)

$$\hat{y} = arg_v \, max \sum_{(x_i, y_i \in I_z)} F(v = y_i)$$

F(.) = 1 si la prueba (.) es verdadera y el 0 en caso puesto, vendría a ser la etiqueta de la clase.

El valor de (K) se debe seleccionar con cuidado, en efecto un valor inferior puede originar un comportamiento escandaloso, donde resulta que mientras sean un valor superior puede integrar muchos puntos de otras clases.

b) Código fuente:

Para poder realizar la implementación del modelo, se realizó en lenguaje de programación Python, a continuación, se mostrará mediante código, importando la librería sklearn.neighbors.

Tabla 17.
Código de implementación para el modelo K-NN.

KNeighborsClassifier

```
#Entrenamiento del modelo
A_entrenar, A_prueba, b_entrenar, b_prueba = train_test_split(A,b,test_size= 0.30,random_state= 70)
print(A_entrenar.shape, A_prueba.shape, b_entrenar.shape, b_prueba.shape ))
clasificador = KNeighborsClassifier(n_neighbors=2).fit(A_entrenar, b_entrena r)
predecir = clasificador.predict(A_prueba)
```

Fuente: Elaboración Propia

El código que se visualiza en la Tabla 17, muestra la construcción del algoritmo de KNeighborsClassifier, para realizar el entrenamiento del

conjunto de datos, se inició importando el algoritmo llamando la librería de sklearn.neighbors, por consiguiente se realizó la creación del algoritmo, haciendo uso del parámetro n_neighbors, para así poder realizar un ajuste al clasificador y definir el número de vecinos más cercanos , para finalizar de entreno el clasificador con el conjunto de datos, obteniendo los resultas de la detección de ataques spoofing ARP, a través de una matriz de confusión.

3.2.1.2.3. Implementación del algoritmo Gradient Boosting

Es un potente clasificador de aprendizaje de máquina, que se utiliza en la creación de modelos de predicción, es usada para clasificar y procedimientos de regresión, la cual construye un modelo aditivo de una sola manera por diferentes etapas, permitiendo optimizar funciones de perdida de diferencias arbitrarias.

a) Formula del algoritmo

Se comienza con el dominio de "x", para que se obtenga la función limitante L, que se muestra en la fórmula:

$$F_0(x) = argmin_p \sum_{i-1}^{N} L(y_i, p)$$

Con esto se obtiene el valor de "y" y se podrá calcular la gradiente negativa, con la siguiente fórmula:

$$\overline{y_i} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F x_i}\right]$$

Posteriormente se utiliza la variable "α" que se usa para el ajuste de error obtenido, mediante la siguiente fórmula:

$$\alpha_m = argmin_{\alpha,\beta} \sum_{i=1}^{N} \lfloor \overline{y} - \beta h(x_i; \alpha_m \rfloor^2$$

Después se emplea la gradiente positiva mediante la siguiente fórmula:

$$\rho_m = argmin_\rho \sum_{i-1}^{N} L(\, y_i, F_m - 1(x_i) + \rho h(x_i; \alpha$$

Por último, se renueva la fórmula con los parámetros encontrados que se obtiene de los árboles de selección de datos.

$$F_x \rightarrow F_m(x) = F_{m-1}(x) + \rho_m h(x; \alpha_m)$$

La fórmula se utiliza para medir el gradiente desarrollado de un árbol de decisión, es importante el entendimiento de la ecuación para su funcionamiento.

b) Código fuente

El código que se muestra en la Tabla 18, muestra el código escrito para el entrenamiento del conjunto de datos, para ello primero se importó el algoritmo, desde su librería sklearn.ensemble, posteriormente creamos el conjunto de datos, declarando las variables de entra y salida, después se entrenó el modelo, se realizó la predicción del modelo para obtener los resultados de detección de ataques de spoofing arp a través de los indicadores.

Tabla 18.
Código de implementación para el modelo Gradient Boosting

Gradient Boosting

```
#Entrenamiento del modelo
A_entrenar, A_prueba, b_entrenar, b_prueba = train_test_split(A,b,test_size=0.30,ran
dom_state= 70)
print(A_entrenar.shape, A_prueba.shape, b_entrenar.shape, b_prueba.shape))

clasificador = GradientBoostingClassifier(max_depth=5,random_state=0).fit(A_entren
ar, b_entrenar)
predecir = clasificador.predict(A_prueba)
```

Fuente: Elaboración Propia

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones.

Los resultados que se obtuvieron en base del desarrollo de investigación, se lograron concluir gracias a los objetivos específicos planteados.

En esta investigación, se analizaron 10 artículos científicos de Spoofing ARP, obtenidos de las base de datos de IEE, scopus, y sciencedirect, de las cuales se seleccionó uno artículo.

Se realizó un método para la detección eficiente de ataques spoofing ARP en redes LAN, posteriormente se propuso un conjunto de clasificadores para obtener la mejor precisión, de los cuales se seleccionó seis algoritmos de aprendizaje de maquina como mejor precisión en la identificación de ataques de spoofing ARP, de los cuales Random Forest, Gradient Boosting y K-NN obtienen mejor rendimiento.

El uso del aprendizaje de maquina es de gran ayuda para la seguridad informática, la cual genera como resultado, el ser una poderosa herramienta que puede ayudar a combatir los ataques.

Se finalizó que el método propuesto alcanzó una precisión del 99,32 %, cuando se modela utilizando los algoritmos de aprendizaje de máquina, por lo tanto, el análisis del protocolo ARP, ha demostrado ser un buen método para detectar ataques de suplantamiento ARP.

4.2. Recomendaciones.

Se recomienda utilizar la herramienta ettercap en modo grafico de Kali Linux, para para realizar ataques de intermediario.

Se recomienda a futuros investigadores a realizar la recopilación de los datos, tanto como en las interfaces ethernet e inalámbrica.

Se recomienda que, para recopilar los datos, se debe analizar si son usuarios frecuentes en la red local, para poder capturar sin interrupciones los datos necesarios.

Finalmente se recomienda a los futuros investigadores, a utilizar los resultados obtenidos de esta investigación, para la aplicación en otros estudios similares basados en ataques.

REFERENCIAS

Ahuja, N., Singal, G., Mukhopadhyay , D., & Nehra , A. (2022). Ascertain the efficient machine learning approach to detect different ARP attacks. *Computers and Electrical Engineering*, 1-12.

Al-Hababi, A., & Sezer C, T. (2020). Man-in-the-Middle Attacks to Detect and Identify Services in Encrypted Network Flows using Machine Learning. *2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet)* (pág. 5). Marrakech, Morocco: IEEE.

Andreu, F., Pellejero, I., & Lesta, A. (2006). *Fundamentos y Aplicaciones de Seguridad en Redes WLAN: Fundamentos y Aplicaciones de Seguridad.* Barcelona, España: Marcombo Ediciones Técnicas.

Ariganello, E. (2014). *Redes CISCO. Guía de estudio para la certificación CCNA Security.* España: Ra-Ma.

Ariganello, E. (2020). *REDES CISCO.* España: RA-MA Editorial.

Ariganello, E. (2020). REDES CISCO Guía de estudio para la certificación CCNA 200-301. En E. Ariganello, *Introducción a las redes* (pág. 23). Paracuellos de Jarama, Madrid: Ra-Ma.

Awad, M., & Khanna, R. (2015). *Efficient Learning Machines.* Estados Unidos: Apress Open.

Baca Urbina, G. (2016). *Introduccion a la seguridad Informatica.* Mexico: Grupo Editorial Patria.

Barceló, J., Iñigo, J., Marti, R., Peig, E., & Perramo, X. (2004). *Redes de computadores.* Barcelona, España: Eureca Media.

BID; OEA. (2020). *https://observatoriociberseguridad.org.* Obtenido de https://observatoriociberseguridad.org: https://observatoriociberseguridad.org/#/final-report

BSG, Institute. (2020). *MACHINE LEARNING: APRENDIZAJE SUPERVISADO Y NO SUPERVISADO.* Colombia: AENOR.

Carvalho, R. T. (2016). *Fundamentos de redes de computadores.* Londrina: Editora e Distribuidora Educacional S.A.

E.V.A.Eijkelenboom, & B.F.H.Nieuwesteeg. (2021). An analysis of cybersecurity in Dutch annual reports of listed companies. *Computer Law & Security Review* (pág. 28). Holanda: ScienceDirect .

Enciso, N. R., Salcedo, O. J., & Upegui, E. (2019). Arp Attack Detection Software Poisoning and Sniffers in WLAN Networks Implementing Supervised Machine Learning. *International Conference on Mobile, Secure, and Programmable Networking*, 11.

Eset. (2020). *https://www.welivesecurity.com.* Obtenido de https://www.welivesecurity.com: https://www.welivesecurity.com/wp-content/uploads/2020/08/ESET-Security-Report-LATAM_2020.pdf

Eset. (s.f.). *https://www.eset.com/.* Obtenido de https://www.eset.com/: https://www.eset.com/es/robo-de-identidad/#

EY Perú. (2020). *https://www.ey.com/.* Obtenido de https://www.ey.com/: https://assets.ey.com/content/dam/ey-sites/ey-com/es_pe/topics/cybersecurity/ey-giss-como-pasar-seguridad-aislada-integrada.pdf

Faleiros, M. L., Henrique, S., & Maia Polo, F. (2018). *Guia de estudos Data Sciense Machine Learning.* Brasil: Grupo Neuron/USP.

Fundación Telefónica. (2008). *La Sociedad de la Información en España.* España: Editorial Ariel.

Gandhi, R. (5 de Mayo de 2018). *https://towardsdatascience.com/*. Recuperado el 2020 de 7 de 10, de https://towardsdatascience.com/: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

Hallberg, B. A. (2007). Fundamentos de redes. 4ª edición. En B. A. Hallberg, *EL modelo de interconexion OSI* (pág. 28). México, D.F.: McGRAW-Hill/Interamericana Editores, S.A. DE C.V.

Harrington, P. (2012). *Machine Learning in action.* United States of America: Manning Publications Co.

Henrichsen, J. R., Betz, M., & Lisosky, J. (2015). *Cómo desarrollar la seguridad digital para el periodismo.* Mexico: Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura.

Herrera J, J., & García Alfaro, X. P. (s.f.). *Aspectos Avanzados de Seguridad en Redes.*

Herrera, J., Garcia, J., & Perramón, X. (2004). *Aspectos avanzados de seguridad en redes.* España: Eureca Media.

Jacobson, V. (2017). TCP/IP Protocols. En I. Javvin Tecnologies, *Network Protocols Handbook* (págs. p85-86). United States: Technology Press.

Jara, H., & Pacheco, F. (2012). *Ethical Hacking 2.0.* Argentina: Fox Andina.

Jerry John, K., Justice Owusu, A., & Griffith, S. K. (2020). Detecting End-Point (EP) Man-In-The-Middle (MITM) Attack based on ARP. *Revista internacional de redes de comunicación y seguridad de la información (IJCNIS)*, 6.

kaspersky. (2018). *https://latam.kaspersky.com.* Obtenido de https://latam.kaspersky.com: https://latam.kaspersky.com/about/press-releases/2018_panorama-de-amenazas-phishing

Madrid P, A. (2010). *Derecho del sistema financiero y tecnología.* España: Marcial Pons Ediciones Jurídicas y Sociales, S.A.

Mathivet, V. (2015). *Inteligencia Artificial para desarrolladores.* España: Ediciones ENI.

Mehak, U., Komal, F., Ghufran, A., & Shahbaz, S. (2022). Predicting ARP spoofing with Machine Learning. *International Conference on Emerging Trends in Smart Technologies (ICETST)*, 1-6.

Ming, R., Yanhui, T., Siqi, K., Dali, z., & Danping, L. (2017). A detection algorithm for the ARP broker attack. *Conferencia Internacional sobre Sistemas Inteligentes y Sostenibles* (pág. 6). Chennai: IEEE.

Nasiriany, S., Garrett.Thomas, Wang, W., & Yang, A. (2019). *A Comprehensive Guide to Machine Learning.* Berkeley: Independently published.

Neminath, H., & Nikhil, T. (2017). An event based technique for detecting spoofed IP packets. *Journal of Information Security and Applications*, 32-43.

Open Data Science. (2 de Noviembre de 2018). *Medium*. Obtenido de https://medium.com/@ODSC/5-essential-neural-network-algorithms-9336093fdf56

P P, S., GV, N., TS, P., & AS, G. (2020). The problem of security address resolution protocol. *Serie de conferencias* (pág. 9). Rusia: IOP.

Pérez Torres, D. (2018). Redes Cisco, Guía de estudio para la certificación CCNA 200-301. En D. Pérez Torres, *Introducción a las redes informáticas* (pág. 19). Mexico: Alfaomega Grupo Editor, S.A. de C.V.

Polo, F. M., Faleiros, M., Leonardo, E., & Samuel, H. (2018). *Guia de estudos Data Science Machine Learning.* Brasil: Grupo de estudos em Data Science NEURON/USP.

Prasad, A., & Chandra, S. (2022). Defending ARP Spoofing-based MitM Attack using Machine Learning and Device Profiling. 12.

Ren, M., Tian, Y., Kong, S., Zhou, D., & Li, D. (2020). An detection algorithm for ARP man-in-the-middle attack based on data packet forwarding behavior characteristics. *5th Information Technology and Mechatronics Engineering Conference (ITOEC)* (pág. 6). Chongqing, China: IEEEE.

Ríos, V., RM Inácio, P., Magoni, D., & M Freire, M. (2021). Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms. *Computer Networks*, 20.

Rocha, R., & Launchbury, J. (2011). *Practical Aspects of Declarative Languages.* Estados Unidos: Springer.

Rohatgi, V., & Shimpy, G. (2020). Una encuesta detallada sobre técnicas de detección y mitigación contra la suplantación de ARP. *Cuarta Conferencia Internacional sobre I-SMAC (IoT en Social, Móvil, Analítica y Nube) (I-SMAC)* (pág. 5). Delhi, India: IEEE.

Romero, M. I., Figueroa, G. L., Álava, J. E., Parrales, R., Álava, C. J., Murillo, Á. L., & Castillo, M. A. (2018). *Introducción a la seguridad informática y el análisis de vulnerabilidades.* Manabí. Ecuador.: 3ciencias.

S, R., Soman, K., & Dr Pritam, G. S. (09 de 2018). Security with IP Address Assignment and Spoofing for Smart IOT Devices. pág. 5.

S, R., Soman, K., & Dr Pritam, G. S. (2018). Security with IP Address Assignment and Spoofing for Smart IOT Devices. *International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pág. 5). Bangalore, India: IEEE.

Santhosh, K. (2017). A dedicated setup to identify spoofing via IP-traceback. *Conferencia Internacional sobre Sistemas Inteligentes y Sostenibles*, 6.

Scott, B., Xu, J., Zhang, J., Brown, A., Clark, E., Yuan, X., . . . Williams, K. (2017). An Interactive Visualization Tool for Teaching Arp Spoofing Attack. *IEEE Frontiers in Education Conference (FIE)*, 5.

Sreenivas, S., Mahesh, M., & B R, C. (2020). Techniques To Secure Address Resolution Protocol. *11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pág. 7). Kharagpur, India: IEEE.

Stallings, W. (2004). *Fundamentos de seguridad en redes: aplicaciones y estándares.* España: Pearson.

Stallings, W. (2011). *Comunicaciones y Redes de Computadoras.* Madrid: PEARSON EDUCACION.

Sweta, S., Dayashankar, S., & Aanjey, M. T. (2019). Two-Phase Validation Scheme for Detection and Prevention of ARP Cache Poisoning. *Progress in Advanced Computing and Intelligent Engineering* (pág. 13). India: Springer Singapur.

Tanenbaum, A. S., & Wetherall, D. J. (2012). *Redes de computadoras.* Naucalpan de Juárez, Estado de México: Pearson Educación de México, S.A. de C.V.

Vijayakumar, K., Achyut, R., Senthil, K. G., TS Shiny, A., & Snehalatha, N. (2020). A two-way approach for detection and prevention of IP spoofing attacks. *AIP 2277*, 8.

Villalón H, A. (2002). *Seguridad en Unix y Redes.* España: Nau Llibres.

Zayegh, A., & Al Bassam, N. (2018). *Principios y aplicaciones de redes neuronales.* Muscat: IntechOpen. Obtenido de Principios y aplicaciones de redes neuronales.

Zheng, A., & Casari, A. (2018). *Feature Engineering for Machine.* Gravenstein Highway North: O'REILLY.

ANEXOS.

Anexo 01: Top de algoritmos de identificación automática

N°	Algoritmo	Precisión	Aplicación	Autor	Año
1	k-Means	79,33%	Predicción de lealtad de clientes	Sardjoeni Moedjiono, Yosianus Robertus Isak, Aries Kusdaryono.	2016
2	Logistic Regression	99,62 %	Detección de ataques de intermediario basado en el análisis del protocolo arp	Jerry John Kponyo, Justicia Owusu Agyemang, y Griffith Selorm Klogo	2020
3	Gradient Bosting	99,34 %			
4	Naive Bayes	99,72%			
5	Decision Tree	99,34%			
6	Random Forest	99,44%			
7	k-Nearest Neighbors	99,34%			
8	Perceptron	95%	Detección de Malware en Android	Anıl UTKU, İbrahim Alper DOĞRU, M. Ali AKCAYOL	2018
9	Regression linear	82 %	Predicción del modelo para el mercado	Hiral R. Patel , Satyen M. Parikth Dhara N. Darji	2018
10	AdaBoost	98,8%	Detección de sitios web falsificados	Ekta Gandotra y Deepak Gupta	2020
11	Neuronal Network	97.67%	Detección de sms spam	Amani Alzahrani and Danda B. Rawat	2019
12	Super Vector Machine (SVM)	94.26%			
13	Binary Logistic Regression	98%,	Datos médicos	Qais Abdulqader	2017
14	Multiclass Logistic Regression	98,71	Predicción de peligro de incendios forestales	Lei Wang, Qingjian Zhao, Zuomin Wen, Jiaming Qu	2018

Anexo 02: Formato de registro de matriz de confusión

		ACTUAL	
		Normal	MITM
PREDICCIÓN	Normal		
	MITM		

	Valor
Exactitud	
Precisión	
Recall	

Anexo 03: Formato de Consumo de CPU

CONSUMO DE CPU	
Ítem	**Valor**
Uso	
Velocidad	
Procesos	
Subprocesos	
Tiempo	

Anexo 04: Formato de Consumo de memoria

CONSUMO DE MEMORIA	
Ítem	**Valor**
Uso	
Disponibilidad	
Confirmada	
En caché	
Tiempo	

Anexo 05: Formato de Promedio de tiempo respuesta

PROMEDIO DE TIEMPO DE RESPUESTA	
Ítem	**Valor**
Velocidad	
Tiempo	
CPU	
Memoria	
Disco	