Inteligencia Artificial y Aprendizaje Automático para prevención de ataques DDoS en dispositivos IoT

Alumno: Juan Francisco Echazú

Supervisor: Rodrigo Ramele

Proyecto Final

Especialización en Ciencia de Datos

Lugar y Fecha: Buenos Aires, Argentina, Octubre de 2022

Abstract

IoT devices are becoming more and more widespread in our daily lives, and the implementation of 5G networks will make this technology more pervasive. As more devices are connected to the network, more we will depend on them and consequently devices' reliability becomes more important than ever. For example, devices with sensors that transmit the production status of a factory and according to that data a certain action is carried out cannot stop working because the entire production chain will be ruined or, even more importantly, medical devices that track vital signs and alert health personnel of any important changes must be available twenty-four hours a day because a patient's life depends on it.

Hence, this work explores artificial intelligence techniques used to prevent DDoS (Distributed Denial-of-Service) computer attacks on IoT devices that threaten data availability, which is one aspect that can disrupt these services and that affects is availability.

Resumen

Los dispositivos IoT son cada vez más comunes en nuestro día a día, y la implementación de redes 5G hará que esta tecnología crezca aún más. Cuantos más dispositivos haya conectados a la red más dependeremos de estos y más se nos complicará la vida cuando estos sistemas no estén disponibles. Por ejemplo, dispositivos con sensores que transmiten el estado de producción de una fábrica y según esos datos se realiza una acción determinada no puede dejar de funcionar porque se arruinará toda la cadena de producción o, aún más importante, dispositivos médicos que rastrean signos vitales y alertan al personal de salud sobre cualquier cambio importante deben estar disponibles las veinticuatro horas del día porque la vida de un paciente depende de ello.

Es por esto, que este trabajo analiza técnicas de inteligencia artificial utilizadas para la prevención de ataques informáticos DDoS (Distributed Denial-of-Service) en dispositivos IoT que atentan contra la disponibilidad de los datos.

Entre los trabajos evaluados, se destacan los resultados obtenidos con algoritmos de Aprendizaje Automático como Árboles de Decisión y Redes Neuronales Recurrentes (RNN).

1. Tabla de Figuras

2. Prefacio

2.1 Introducción

En un mundo cada vez más interconectado, las tecnologías inalámbricas están cobrando cada vez más y más protagonismo en todos los ámbitos. Gracias a la continua difusión de miles de millones de objetos conectados y dispositivos inteligentes en el contexto de Internet de las Cosas (IoT) se requiere de un medio de propagación y una red capaz de ofrecer servicios de calidad. Es por esto que surge la red 5G que se caracteriza por tasas de bits más altas, cuantificadas en más de 10 gigabits por segundo, así como una mayor capacidad y una latencia muy baja. De hecho, en la era emergente de IoT, 5G ciertamente permite superar los problemas actuales en términos de tiempos de respuesta de red y administración de recursos de red, considerando que IoT abarca tecnologías heterogéneas, que van desde redes de sensores inalámbricos (WSN) hasta sistemas de almacenamiento y recuperación de datos remotos por identificación de radiofrecuencia o RFID, NFC (Near Field Communication), actuadores, etc.

Sin embargo, junto con las capacidades funcionales, se debe cumplir con los requisitos de confidencialidad, integridad y disponibilidad con mayor foco en la seguridad en las comunicaciones. Los ataques de denegación de servicio distribuídos (DDoS) atentan contra la disponibilidad de los datos siendo su objetivo principal hacer colapsar los sistemas y romper su funcionamiento. Dicho ataque, se logra generando un gran flujo de información desde varios puntos de conexión hacia un mismo punto de destino. La forma más común de realizar un DDoS es a través de una red de bots o botnet, que es un grupo de equipos infectados por malware que se encuentran bajo el control de un agente malicioso, siendo esta técnica el ciberataque más usual y eficaz por su sencillez tecnológica. Esto puede lograrse tomando, de manera no autorizada, los distintos dispositivos IoT conectados a una red y usarlos de manera maliciosa. Es por esto, que el surgimiento de la red 5G, que permite una mayor cantidad de dispositivos conectados entre sí, crea la necesidad de proteger estos sistemas contra estos ataques de DDoS. Este es el motivo principal que guía los lineamientos de esta investigación.

Por otro lado, sabiendo que a medida que crece la creatividad y la tecnología utilizada para vulnerar los sistemas y generar ataques maliciosos, crece la necesidad de actualizar las medidas de seguridad de la información. En este trabajo, principalmente se hace hincapié en la utilización de sistemas basados en Inteligencia Artificial (AI), con algoritmos de Aprendizaje Automático (ML) y Redes Neuronales como posible solución para la protección de los datos que transmiten los dispositivos IoT a través de las redes 5G.

2.2 Alcance

Existen muchos posibles ataques informáticos sobre dispositivos IoT, pero por el auge de las redes 5G que permite el incremento exponencial de dispositivos conectados a estas redes, este estudio abarcará únicamente los ataques DDoS. Se analizarán los métodos de

inteligencia artificial más comunes utilizados para prevenir este tipo de ataques y así proteger los sistemas IoT.

Principalmente se mencionarán los algoritmos de Machine Learning clásicos y los tipos de redes neuronales utilizados, junto con su comparación.

Para facilitar el entendimiento del trabajo y la conexión con otros similares, diversas palabras del idioma inglés se conservarán en ese idioma. También los acrónimos se conservarán con la ortografía de los originales en inglés.

3. Antecedentes del Trabajo Realizado

Entre los trabajos más interesantes relacionados con este tema podemos nombrar como más antiguos a [21], del 2020, que detalla cómo se puede utilizar Machine Learning para brindar seguridad a dispositivos IoT explicando qué algoritmos se aplican. Por otra parte, también publicado en el 2020, [62] propone una arquitectura basada en Machine Learning para la detección de ataques entre los que se menciona al DoS y menciona casos donde falla un IDS basado en Machine Learning por ser un escenario complejo.

En [25] se encuentra un buen detalle y explicación de los diferentes tipos de DDoS que se nos puede presentar en la capa de red de la arquitectura IoT y realiza un estudio empírico a gran escala para evaluar la efectividad de los algoritmos de Machine Learning de última generación para la seguridad de las aplicaciones en red.

La publicación [40] del 2021, brinda una revisión de artículos y publicaciones orientadas a la seguridad en dispositivos IoT y al desarrollo de modelos que puedan integrar técnicas y tecnologías de vanguardia de Big Data y Machine Learning para la detección de ataques de IoT en tiempo real.

Finalmente, [46] analiza problemas y proporciona un modelo de comunicación basado en EASH (Energy Aware Smart Home). Se analiza la problemática en fallas de comunicación y tipos de ataques a la red EASH. Se evalúan algoritmos de clasificación de Machine Learning para generar un sistema de detección de intrusos en la red.

4. Conceptos Preliminares

4.1 Historia de Internet of Things

Se pueden mencionar a lo largo de la historia infinidad de avances tecnológicos aunque en este apartado nos enfocaremos en los relacionados con el Internet de las Cosas, término que tiene su origen en 1999, cuando Kevin Ashton, directivo de Procter & Gamble, tuvo la iniciativa de crear una agrupación de investigadores llamada Auto-ID Center en el Instituto Tecnológico de Massachussets (MIT), que se dedicaban a averiguar información sobre la identificación por radiofrecuencia en red (RFID) y tecnologías de sensores.[13]

A comienzos del siglo XXI, gracias a la popularización de la conectividad inalámbrica (celular o WiFi), se produjo la primera explosión en el crecimiento de los objetos conectados. Este crecimiento se consolidó especialmente en lo últimos años, según han ido surgiendo nuevos conceptos como el WSN (Wireless Sensor Networks) o las nuevas tecnologías de acceso radio como LPWA (NB-IoT, LTE-M,…), para finalmente dar paso al IoT que todos conocemos. [14]

- **Año 2000** - Internet Digital DIOS: LG lanza el primer refrigerador conectado a Internet. No fue un hecho tan relevante ya que su precio era muy elevado.[14]
- **Año 2005** - Nabaztag, primera mascota-asistente virtual conectado: la empresa francesa Violet lanzó al mercado Nabaztag (liebre en armenio) . Se trata de un dispositivo con forma de conejo que se conecta a Internet por ondas wifi. Se comunica con el usuario mediante mensajes de voz, y cambios de color o movimiento de sus orejas. Es capaz de reproducir, hablar, escuchar y responder a la voz de los usuarios. También les puede despertar a la mañana con las noticias de actualidad de diarios digitales, la música o la información del tiempo.[14]

Figura 1. Primera mascota virtual conectada

- **Año 2008** - Más dispositivos conectados que personas: fue el primer año en el que los dispositivos conectados a Internet superaron al número de personas conectadas.[14]
- **Año 2009** - Aunque el término era de uso corriente en círculos especializados desde 1999, no fue hasta el 2009, cuando Kevin Ashton lo introdujo para el gran público en su artículo *That «Internet of Things»Thing*, del que merece la pena traducir un pequeño extracto:

«*Somos entes físicos, como también lo es nuestro entorno. Nuestra economía, sociedad y supervivencia no se basan en ideas o información, se basan en cosas* (…)

Necesitamos dotar a los ordenadores de sus propios medios para recopilar información, para que puedan ver, oír y oler el mundo por sí mismos, en toda su gloria aleatoria. La RFID y la tecnología de sensores permiten a las computadoras observar, identificar y comprender el mundo, sin las limitaciones de los datos introducidos por el hombre»[14]

- **Año 2009** - Google comienza a trabajar en el proyecto de auto autónomo: *Google self-driving car project*, que posteriormente pasaría a ser conocido como Waymo**.** La tecnología desarrollada por Waymo permite a un automóvil conducirse de forma autónoma por la ciudad y ruta, detectando otros vehículos, señales de tráfico, peatones, etc.[14]
- **Año 2009** - Primer implante cardíaco monitorizado por IoT: la empresa Saint Jude Medical fabrica los primeros *implantes cardíacos conectados*. Un adaptador USB inalámbrico recibía los datos del implante y los transmitía posteriormente a los móviles del personal médico.[14]
- **Año 2010** - Electrodomésticos inteligentes: la compañía NEST empieza a fabricar electrodomésticos inteligentes. El primero fue un termostato que optimizaba el horario de la calefacción a partir de los patrones de uso de los usuarios.[14]
- **Año 2011** - Lanzamiento de IPv6: Los primeros pasos en IoT se dieron con la versión v4 (IPv4). Esto suponía una importante limitación, ya que el número de direcciones que se podían generar era muy reducido. A partir del año 2011 se diseña el protocolo de direccionamiento de Internet IPv6 posibilitando la identificación de una infinidad de direcciones. Supuso un gran impulso para el desarrollo del IoT ya que se proyectaba contar con miles de millones de dispositivos, sensores y actuadores conectados. Poco después Samsung, Google, Nokia y otros fabricantes anuncian sus proyectos NFC.[14]
- **Año 2013** - Google lanza las Google Glass: se puso a la venta (para desarrolladores calificados) Glass Explorer Edition, un dispositivo de visualización de tipo gafas de realidad aumentada presentado en el congreso I/O de junio de 2012. [14]
- **Año 2014** - Desarrollo de estándares industriales: Intel, Cisco, IBM, GE y AT&T se unen para mejorar la integración de IoT con la industria. Se crea la iniciativa IoT-GSI Global Standards para promover la adopción de estándares para IoT a escala global. Los participantes comparten informes de investigación, documentos técnicos y buenas prácticas de gran valor para el desarrollo del IoT industrial a escala empresarial y global.[14]
- **Año 2016** - Primer Malware de IoT (MIRAI): este año surgió Mirai, un botnet cuyo objetivo son dispositivos IoT, principalmente routers, grabadoras digitales de vídeo y cámaras IP de vigilancia. Este malware recopila las contraseñas por defecto que establecen los fabricantes de los dispositivos y que los usuarios muchas veces se

olvidan de cambiar. Luego, utiliza los dispositivos para realizar ataques de denegación de servicio (DoS) a terceros, normalmente, páginas web muy populares.[14]

- **Año 2017** - Servicios de IoT: los grandes fabricantes de servicios en la nube ofrecen soluciones IoT, por ejemplo *Azure IoT Edge*, *AWS IoT* y *Google Cloud IoT core*.[14]

De aquí en adelante, los hitos más importantes que vamos a mencionar tienen que ver con el crecimiento y la actualización de las redes móviles que brindan la infraestructura para que los dispositivos IoT puedan estar conectados.

4.2 Historia de las redes de comunicaciones móviles

Ninguna tecnología de comunicaciones, ya sea móvil o fija, pública o privada se despliega eliminando sus versiones previas. Hay muchos componentes que se reaprovechan buscando lograr una cierta convivencia entre las distintas versiones, y poco a poco, llevar la nueva tecnología al mercado. En la actualidad, en la mayoría de los países del mundo siguen existiendo las redes 2G, 3G y 4G aún cuando ya están surgiendo las 5G. La implementación de una nueva tecnología requiere de un proceso y es por eso que es importante conocer la historia de la migración y avance de la tecnología hasta llegar a las redes 5G actuales.[74]

La primera red de comunicación móvil comercial fue lanzada en Japón por NTT (Nippon Telegraph and Telephone) en 1979. Esta primera generación o 1G utilizaba canales de comunicación analógicos y servía exclusivamente para transmitir voz, con escasa seguridad en las comunicaciones. La arquitectura CS (Circuit Switched) del núcleo de red se basaba en la conmutación de circuitos.

Luego aparecen las redes 2G en 1991. Esta nueva generación de red móvil mejoró la seguridad de las comunicaciones al utilizar protocolos digitales cifrados, siendo el GSM (Sistema Global de comunicaciones móviles) el más extendido. Está compuesta por una red de acceso GSM basada en antenas/celdas llamadas BTS (Basic Transmission Station) y un core GSM compuesto por dispositivos llamados HLR (Home Location Register), donde se almacenan todas las tarjetas SIM (perfiles de los usuarios) y MSC (Mobile Switching Center) que proporciona la conmutación de llamadas, administración de movilidad y servicios de GSM, brindando transmisión de voz. Las antenas se conectan con el BSC (Basic Switching Controller) que son dispositivos que controlan a varias celdas, como puede observarse en la figura 2. Finalmente, esta infraestructura se conecta a las redes de voz PSTN (Public Switching Telephone Network). La red 2G sólo fue pensada para comunicación únicamente por voz.[74]

En este momento se da el cambio de comunicación por conmutación de circuitos a conmutación por paquetes a medida que se va aumentando con la digitalización. Con esta red nacen los SMSs y comienza el roaming.

En 1998, nace la red 3G y se introduce el Core GPRS (General Packet Radio System) que es una central de paquetes con el SGSN (Serving GPRS Support Node) y el GGSN (Gateway

GPRS Support Node), que puede observarse en la figura 2. Estos nodos solo mueven paquetes y son los que asignan direcciones ip, DNS por defecto, arman la tabla de ruteo, etc. Aquí, la infraestructura 3G se acopla a la 2G existente y el BSC discrimina la comunicación por datos y por voz. De esta manera, en el 2006 se afianza Internet - HSDPA (High Speed Downlink Packet Access) donde contamos con una red de voz y de paquetes, donde por primera vez nace la posibilidad de ancho de banda a las telefonías móviles y empiezan a aparecer los smartphones con posibilidad de acceder a internet.[74]

En 2009 nace la red 4G donde hay un gran cambio tanto en la interfaz de acceso a la red como en la interfaz core. Esta nueva generación de red móvil implementa una Interfaz radio que ya cuenta con inteligencia y nace un core muy diferente al anterior, que podemos verlo en la figura 2, que era la combinación de 2G y 3G. Acá aparece el EPC (Enhanced Packet Core) que es el core pensado especialmente para paquetes. A través del IMS (Internet Multimedia Subsystem) sale la voz paquetizada y los datos también paquetizados salen a través del PDGw (Packet Network Data Gateway). Ofrece velocidades mucho mayores que 3G, con protocolos de Internet (IP) para soportar servicios multimedia (en especial aplicaciones de video y servicios: YouTube, video llamadas, etc.). El objetivo principal de la tecnología 4G es proporcionar alta velocidad, alta calidad, alta capacidad, seguridad y servicios de bajo coste para servicios de voz y datos, multimedia e internet a través de IP. Para usar la red de comunicación móvil 4G, los terminales de los usuarios deben ser capaces de seleccionar el sistema inalámbrico de destino. Para proporcionar servicios inalámbricos en cualquier momento y en cualquier lugar, la movilidad del terminal es un factor clave en 4G.[74]

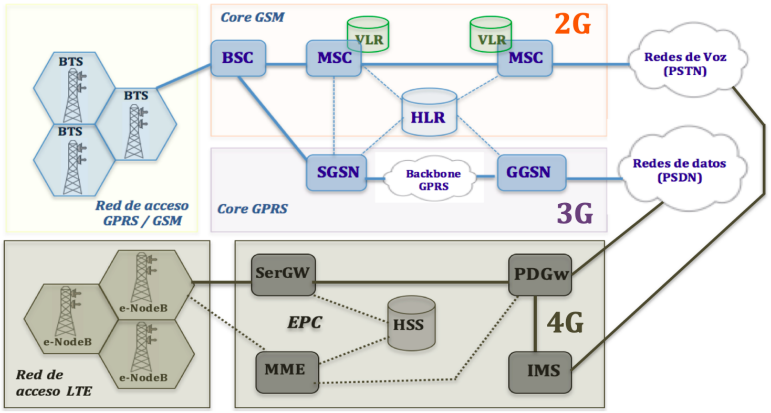

Figura 2. Evolución de las redes de comunicación móviles de 2G a 4G. [74]

Podemos ver un cuadro comparativo con algunas de las características más importantes.

	1G	2G GSM/GPRS	3G UMTS/HSPA	4G LTE
ESPECTRO:	Banda baja: 900 MHz	Banda baja: 900 MHz Banda media: 1,8 GHz	Banda baja: 900 MHz Banda media: 2,1 GHz	Banda baja: 800 MHz Banda media: 1,8 y 2,1 y 2,6 GHz
ESTACIÓN BASE:		BTS	Nodo B	eNodo B
NÚCLEO DE RED:	CS La arquitectura se basa en la conmutación de circuitos. **Transporta voz**	CS + PS La arquitectura se basa en la conmutación de circuitos y paquetes. **Transporta voz, SMS y datos**	CS + PS La arquitectura se basa en la conmutación de circuitos y paquetes. **Transporta voz, SMS y datos**	EPC La arquitectura se basa en la conmutación de paquetes. **Transporta voz IP, SMS y datos**

	1G	2G	2.5G	3G	4G
Inicio/ Evolución	1970/1984	1980/1991	1985/1999	1990/2002	2000/2006
Ancho de Banda	1.9 Kbps	14.4 Kbps	14.4 Kbps	2 Mbps	200 Mbps
Estándar	AMPS, FDMA	TDMA, CDMA, GSM	GPRS, EDGE, 1xRTT	WCDMA, CDMA-2000	Estándar Unificado
Tecnología	Analógica	Digital	Digital	Banda ancha CDMS y Tecnología IP	LAN/WAN/ PAN y WLAN
Servicio	Telefonía Móvil (Voz)	Voz digital y mensajes cortos	Mayor capacidad, paquetes de datos	Integración de alta calidad de audio, video y datos	Acceso a la información dinámica, dispositivos portátiles
Multiplexación	FDMA	TDMA, CDMA	TDMA, CDMA	CDMA	CDMA
Conmutación	Circuito	Circuito	Circuito para red de acceso, interfaz de aire	Paquetes excepto el circuito para interfaz de aire	Todos los paquetes
Núcleo	PSTN	PSTN	PSTN y Red de datos	Red de datos	Internet

Figura 3. Cuadro comparativo de redes móviles. [75]

Finalmente, en 2020, con la quinta generación o 5G aparece un nuevo gran cambio en la infraestructura de las comunicaciones móviles. La arquitectura 5G se orienta ahora a conectar personas, pero también objetos (Internet de las Cosas, IoT) con latencias (retardo en la transmisión) mínimas y velocidades de hasta 20 Gbps, que multiplicará exponencialmente el número de dispositivos conectados. Además, se complementa con las aplicaciones y servicios de red en la nube.

Esta red está compuesta también por una Interfaz radio distribuyendo y centralizando funciones, se conecta con el Backhaul al Core de red denominado SBA (Software Based

Architecture), mostrado en la figura 4, que está basado absolutamente en el software. En este momento las comunicaciones dejan a un lado todos los conectores de fibra óptica, RJ45 y otros, para pasar a convertirse en comunicaciones a través de APIs de un cliente a un servidor, utilizando peticiones JSON basadas en el protocolo HTTP2.

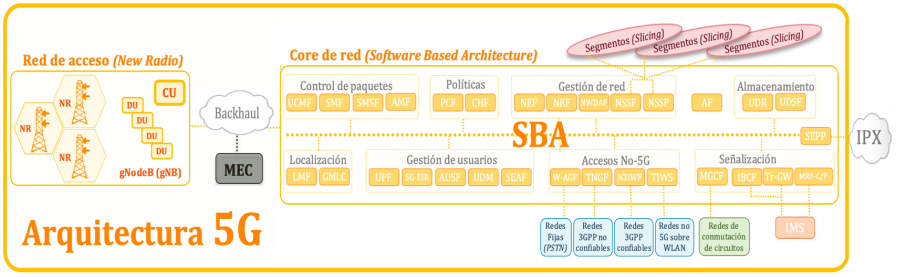

Figura 4. Arquitectura de la red 5G propuesta por [74]

El gran cambio con esta nueva versión de red, es una nueva interfaz radio que puede llegar a ser virtual, abierta (open RAN) o puede estar montada en la nube. Y por otro lado, se cuenta con un Core totalmente montado en forma de funciones, donde ya no hay más hardware.

4.3 Dispositivos IoT

El término IoT, o Internet de las cosas, se refiere a la red colectiva o ecosistema de dispositivos conectados y a la tecnología que facilita la comunicación entre los dispositivos y la nube, así como entre los propios dispositivos. Gracias a la llegada de los chips de ordenador de bajo coste y a las telecomunicaciones de gran ancho de banda, ahora tenemos miles de millones de dispositivos conectados a Internet. Esto significa que los dispositivos de uso diario, como los cepillos de dientes, las aspiradoras, los automóviles y las máquinas, pueden utilizar sensores para recopilar datos y responder de forma inteligente a los usuarios. [15]

El Internet de las cosas integra las “cosas” de uso diario con Internet. Los ingenieros en informática llevan agregando sensores y procesadores a los objetos cotidianos desde los años 90. Sin embargo, el progreso fue inicialmente lento porque los chips eran grandes y voluminosos. Los chips de ordenador de baja potencia llamados etiquetas RFID se utilizaron por primera vez para el seguimiento de equipos caros. A medida que los dispositivos de computación reducían su tamaño, estos chips también se hacían más pequeños, más rápidos y más inteligentes. [15]

4.3.1 Arquitectura IoT

Los dispositivos IoT tienen una arquitectura única que se puede definir en tres capas: la capa de percepción, de red y de aplicación [21]. Cada capa ofrece una funcionalidad diferente y a su vez deben estar conectadas entre sí para funcionar de manera correcta. Así mismo, podremos decir que cada capa tiene sus propias amenazas únicas. [4]

Capa de percepción: es la primera capa y consta de las capas física (PHY) y de control de acceso al medio (MAC).

La capa PHY se ocupa principalmente del hardware, es decir, sensores y dispositivos que se utilizan para transmitir y recibir información utilizando diferentes protocolos de comunicación, por ejemplo, RFID, Zigbee, Bluetooth, etc. [17]

La capa MAC establece un vínculo entre los dispositivos físicos y las redes para permitir una comunicación adecuada. MAC utiliza diferentes protocolos para vincularse con la capa de red, como LAN (IEEE 802.11ah), PAN (IEEE 802.15.4e, Z-Wave), red celular (LTE-M, EC-GSM). La mayoría de los dispositivos en las capas de IoT son del tipo plug and play desde donde se produce una gran parte de los grandes datos. [18]

Capa de red: es la capa más importante en los sistemas IoT porque actúa como un medio de transmisión de información y datos utilizando varios protocolos de conexión, como por ejemplo GSM, LTA, WIFI, 3-5G, IPv6 y IEEE 802.15.4 para conectar dispositivos con servicios inteligentes. En esta capa, existen servidores que almacenan y procesan la información funcionando como un middleware entre la red y la siguiente capa. [20]

Otro factor importante en la capa de red es el Big Data. La capa física produce una gran cantidad de información/datos continuamente que los sistemas IoT transmiten, procesan y almacenan. Dado que la información/datos son importantes para los servicios inteligentes en la capa de red, Machine Learning y Deep Learning se utilizan ampliamente hoy en día para analizar la información/datos almacenados y mejorar las técnicas de análisis. [19]

Capa de aplicación: es la tercera capa en los sistemas IoT que brinda servicio a los usuarios a través de software móvil y basado en la web. Según las tendencias y los usos recientes de cosas inteligentes, IoT tiene numerosas aplicaciones en este mundo tecnológicamente avanzado. Distintos ámbitos de nuestra vida como por ejemplo viviendas, transporte, salud, educación, agricultura, negocios/comercios, sistemas de distribución de energía y muchos más se han vuelto inteligentes gracias al sistema IoT y a los incontables servicios que brinda.

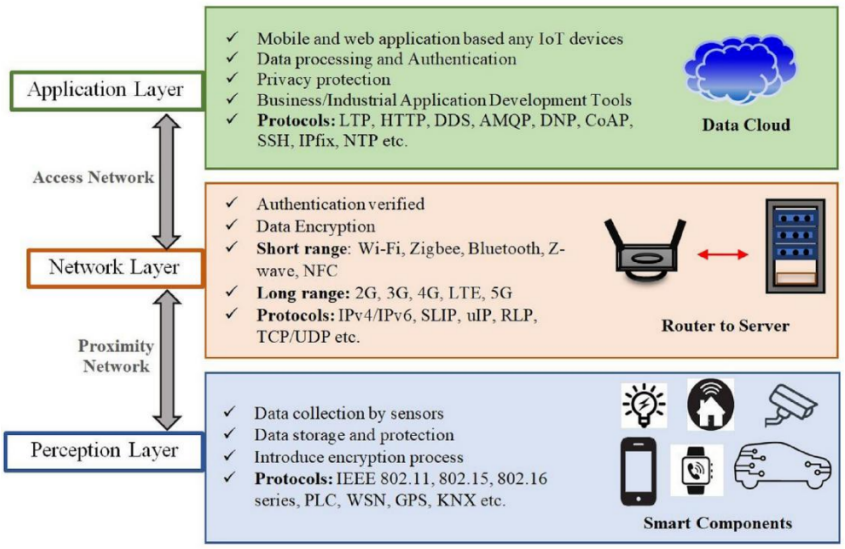

Figura 5. Arquitectura de capas IoT [21]

4.4 Seguridad de la Información

En primer lugar, es primordial mencionar que en cualquier entidad pública o privada e incluso en cualquier tipo de dispositivo electrónico personal se manejan datos privados y confidenciales. Es por esto, que se define a la seguridad de la información como aquel conjunto de medidas y técnicas empleadas para controlar y salvaguardar todos esos datos y asegurar que estos no salgan del sistema al que corresponden. Es, además, una pieza clave para que las organizaciones puedan actualmente llevar a cabo sus operaciones, ya que los datos manejados son esenciales para la actividad que desarrollan. [22]

4.4.1 Objetivos de la Seguridad de la Información

En la norma ISO 27001 se definen los objetivos de la seguridad de la información. Esta norma proporciona un modelo para la implantación de sistemas de gestión de seguridad de la información (SGSI), cuyo fin principal es la protección de los activos de información, es decir, equipos, usuarios e información.

Para establecer este sistema ISO de seguridad de la información hay que tener en cuenta tres aspectos fundamentales: integridad, confidencialidad y disponibilidad.

4.4.2 Integridad

Los sistemas que gestionan la información deben garantizar la integridad de la misma, es decir, que la información se muestra tal y como fue concebida, sin alteraciones o manipulaciones que no hayan sido autorizadas expresamente.

El objetivo principal es garantizar la transmisión de los datos en un entorno seguro, empleando protocolos seguros (cifrado) y técnicas para evitar riesgos.

4.4.3 Confidencialidad

La confidencialidad garantiza que solo las personas o entidades autorizadas tendrán acceso a la información y datos recopilados y que estos no se divulgarán sin el permiso correspondiente. Los sistemas de seguridad de la información deben garantizar que la confidencialidad de la misma no se vea comprometida en ningún momento.

4.4.4 Disponibilidad

El aspecto de disponibilidad garantiza que la información estará disponible en todo momento para aquellas personas o entidades autorizadas para su manejo y conocimiento. Para ello deben existir medidas de soporte y seguridad que permitan acceder a la información cuando sea necesario y que eviten que se produzcan interrupciones en los servicios.

Figura 6. Objetivos de la Seguridad de la Información. [22]

4.5 Inteligencia Artificial

La inteligencia artificial es la combinación de algoritmos planteados con el propósito de crear máquinas que presenten las mismas capacidades que el ser humano.

Los expertos en ciencias de la computación Stuart Russell y Peter Norvig diferencian en [76] varios tipos de inteligencia artificial:

- Sistemas que piensan como humanos: automatizan actividades como la toma de decisiones, la resolución de problemas y el aprendizaje. Un ejemplo son las redes neuronales artificiales.

- Sistemas que actúan como humanos: se trata de computadoras que realizan tareas de forma similar a como lo hacen las personas. Es el caso de los robots.

- Sistemas que piensan racionalmente: intentan emular el pensamiento lógico racional de los humanos, es decir, se investiga cómo lograr que las máquinas puedan percibir, razonar y actuar en consecuencia. Los sistemas expertos se engloban en este grupo.

- Sistemas que actúan racionalmente: idealmente, son aquellos que tratan de imitar de manera racional el comportamiento humano, como los agentes inteligentes.

4.5.1 Inicios de la IA

En 1936 se inició el proceso de la inteligencia artificial moderna, cuando Alan Turing, el experto matemático que descifró los códigos secretos nazis de la mítica máquina Enigma. Adelantó dos años el fin de la Segunda Guerra Mundial, ya que los aliados pudieron leer los mensajes secretos de los alemanes.

En 1936 Alan Turing publicó su concepto de máquina universal, que básicamente describía lo que era un algoritmo informático, y un ordenador. En 1950 formalizó el inicio de la Inteligencia Artificial con su Test de Turing, una prueba que define si una máquina es o no inteligente.

Si un humano y un agente, ente que tiene inteligencia artificial, IA se enfrentan a las preguntas de un interrogador y ese interrogador no puede distinguir si las respuestas provienen del humano o del agente, entonces el agente es inteligente.

En 1956 expertos como John McCarthy, Newell, Simon y Marvin Minsky, usaron por vez primera el término "*inteligencia artificial*" en una conferencia en Dartmouth (Estados Unidos) y en 2014, por primera vez una IA superó el Test de Turing.

Pero el momento en el que la IA se convirtió en algo real y tangible para la mayoría de la gente fue en 1997 cuando el ordenador Deep Blue de IBM venció en una partida de ajedrez al que por aquel entonces era el mejor jugador de ajedrez de la historia, el ruso Gary Kaspárov. Se inició así una tradición en la que sucesivos ordenadores dotados de inteligencia artificial han vencido a los mejores jugadores en todo tipo de juegos.

Por otro lado, el verdadero auge de la inteligencia artificial, a un nivel práctico, llegó cuando comenzaron a aparecer ordenadores potentes y baratos, capaces de experimentar con la IA a un nivel global y cotidiano. Primero aparecieron los agentes inteligentes, entidades capaces de dar una respuesta analizando los datos según una reglas, o los populares chatbots que eran capaces de mantener una conversación como un humano.[47]

4.5.2 Concepto de Inteligencia Artificial

La Inteligencia Artificial, en Ciencias de la Computación, es un conjunto de técnicas, inspiradas en la percepción que tenemos los seres humanos de cómo opera la inteligencia animal y humana, donde se establecen estructuras de procesamiento genéricas, derivadas de casos concretos y acotados, que se intentan extrapolar a situaciones más generales. [77]

Una familia de técnicas que impulsa esta revolución de la Inteligencia Artificial está orientada a que un sistema aprenda a través de ejemplos, también llamados datos de entrenamiento, previo a su puesta en funcionamiento y/o incorporando la experiencia adquirida mientras está en uso. La forma en que se diseña este proceso de aprendizaje y los datos con que se alimenta son claves en el futuro del comportamiento del sistema. Claramente, quienes son responsables de hacer el diseño son, somos, seres humanos. [77]

Una segunda alternativa, la cual impulsa el éxito de estos días, está basada en la idea de utilizar algoritmos que permiten mapear un conjunto de entradas en salidas deseadas mediante el ajuste iterativo en un proceso de optimización matemática de parámetros libres. Esto es Machine Learning. A ese ajuste de parámetros, se le llama “aprendizaje”, y ese ajuste se realiza mediante datos, usando un dataset. [77]

Las redes neuronales y deep learning, como un nombre nuevo para describirlas pero que esencialmente son equivalentes, son un conjunto de técnicas (i.e. algoritmos, programas) que hacen esto mismo, un ajuste iterativo de parámetros libres para encontrar el mejor mapeo de una función de entrada en una de salida mediante un proceso de optimización matemática, pero con estructuras jerárquicas recurrentes aglomeradas inspiradas en modelos muy simplificados de cómo funcionan los sistemas nerviosos biológicos animales y humanos, y en cómo funcionan, de manera muy simplificada, las neuronas. [77]

A continuación se detallan los algoritmos de Machine Learning y Redes Neuronales más utilizados con una breve explicación de cada uno para comprender sus diferencias.

4.5.3 Aprendizaje Automático o Machine Learning (ML)

Entre los algoritmos de ML más importantes podemos mencionar a: Naive Bayes, K-Nearest Neighbors (KNN), K-Means Clustering, Support Vector Machine (SVM), Random Forest (RF) y Árbol de decisión (DT).

4.5.3.1 Naive Bayes

Es una técnica que ayuda a construir clasificadores, que son los modelos que clasifican el problema y les dan etiquetas de clase que se representan como predictores o valores de características. Este algoritmo funciona bien para grandes conjuntos de datos, por lo que es más adecuado para predicciones en tiempo real. Se considera uno de los algoritmos más simples de implementar y posiblemente uno de los más antiguos, ya que se estudió por primera vez en la década de 1960. El algoritmo se basa en el teorema de Bayes, que proporciona una forma de calcular la probabilidad de que un dato pertenezca a una clase,

dado nuestro conocimiento previo. El Clasificador Naive Bayes asume que la presencia de una característica particular en una clase no está relacionada con la presencia de otras características [26].

4.5.3.2 K-Nearest Neighbors (KNN)

KNN es otro algoritmo de aprendizaje automático popular debido a su simplicidad y facilidad de implementación. Se ha utilizado comúnmente para redes, específicamente para la detección de intrusos en la red. KNN tiene en cuenta la similitud cuando trabaja con datos. En otras palabras, clasifica los datos en función de cómo se clasifican sus puntos de datos más cercanos. La idea básica detrás de este algoritmo es que los elementos similares pueden estar más cerca. KNN usa la variable k, que representa el número de vecinos más cercanos. A medida que aumenta el número de vecinos más cercanos, el valor de k, la precisión puede aumentar. A diferencia de otros algoritmos de aprendizaje automático, KNN no parece tener una función de pérdida. Esto se debe a que no hay capacitación que realmente suceda con KNN. El único entrenamiento que se lleva a cabo es el algoritmo que memoriza los datos. Dado que técnicamente no se ajusta ninguna función a los datos, no se realiza ninguna optimización [27].

4.5.3.3 K-Means Clustering

El agrupamiento de K-means trata de categorizar los datos en diferentes categorías [28]. En otras palabras, cada punto de datos solo pertenece a un grupo. Para el agrupamiento de K-means, el enfoque radica en poder detectar patrones dentro de los datos dados. Para lograr esta tarea, debemos buscar un número fijo de grupos en el conjunto de datos. Cuanta menos variación exista dentro de estos grupos, más probable es que los puntos de datos sean homogéneos. El algoritmo de agrupación en clústeres K-means se ha utilizado para la detección de anomalías y la gestión del ancho de banda en las redes. Su simplicidad lo hace popular para su implementación. Sin embargo, el agrupamiento de K-means sufre en el rendimiento cuando hay una gran cantidad de datos superpuestos.

4.5.3.4 Support Vector Machine (SVM)

SVM es uno de los algoritmos clásicos de aprendizaje automático utilizados para redes. Su aplicabilidad en redes va desde el manejo de datos de red, clasificación de red y optimización. Puede manejar problemas de clasificación y regresión, lo que lo hace adecuado para diversas áreas de redes y aprendizaje automático. SVM también funciona bien con una cantidad limitada de datos. La idea de este algoritmo particular es crear una línea (también llamada hiperplano) para separar los datos en dos clases. Se supone que el hiperplano maximiza el margen entre dos clases. En el contexto de las redes, esto significaría separar el tráfico de red normal del tráfico de red malicioso. Los puntos más cercanos a la línea se llaman vectores de soporte y el margen es la distancia. SVM tiene numerosas ventajas, como poder manejar datos desequilibrados. También funciona bien con datos no estructurados y tiene menos probabilidades de que se produzca un ajuste excesivo. Desafortunadamente, la

principal desventaja de SVM es que es difícil de entender e interpretar lo que sucede en el algoritmo. Además, si bien este algoritmo es poderoso, tiene un largo tiempo de entrenamiento para grandes conjuntos de datos [29].

En [40] se menciona que SVM proporciona una mejor velocidad y rendimiento si el tamaño del conjunto de datos no es muy grande, en comparación con las redes neuronales. El clasificador ofrece buenos resultados a la hora de detectar tráfico malicioso IoT de tráfico benigno. Además, se explica que SVM proporciona diferentes funciones del kernel que se utilizan para mejorar el modelo de aprendizaje y optimizar el rendimiento.

[41] usó el kernel Radial Basis Function (RBF) con un optimizador de C-support (c-SVM) para distinguir entre IoT benigno y tráfico malicioso. Se usó RBF porque arrojó mejores resultados de clasificación que otras funciones lineales como la sigmoide y polynomial kernel. [42] explicó al optimizador de C-support que cuántos puntos de datos podrían clasificarse erróneamente. A un valor más alto de C significa que todos los puntos de entrenamiento deben clasificarse correctamente. SVM siempre busca mejores resultados que producir un margen mayor en el hiperplano.

4.5.3.5 Random Forest (RF)

RF es un algoritmo de aprendizaje supervisado. Se usa mucho debido a su simplicidad y su capacidad para manejar tareas de clasificación y regresión. Random forest crea múltiples árboles de decisión y los fusiona para obtener una predicción más precisa y estable. Random Forest se ha utilizado para redes gracias a su tiempo de entrenamiento más rápido en comparación con SVM, y el algoritmo es más fácil de implementar y, por lo tanto, más fácil de comprender [30]. Esto da como resultado una amplia diversidad que generalmente da como resultado un mejor modelo. Además, otra ventaja de este algoritmo es que es muy fácil medir la importancia relativa de cada característica en la predicción. Además, evita un problema de aprendizaje automático: el ajuste excesivo, que puede afectar negativamente a los modelos. Además, puede manejar valores faltantes.

Por otra parte, en [21] se menciona que el Random Forest se usa típicamente en la detección de ataques DDoS [35], Detección de anomalías e identificación de dispositivos IoT no autorizados en ataques a la superficie de la red. Random Forest da mejores resultados en la detección de ataques DDoS que SVM, ANN y KNN. A pesar de que RF no es útil en aplicaciones en tiempo real, necesita una mayor cantidad de conjuntos de datos de entrenamiento para construir un Árboles de Decisión que identifiquen intrusiones repentinas no autorizadas.

Ahmad & Alsmadi et al [40] explican que uno de los beneficios de usar RF sobre los tradicionales árboles de decisión es que evita la correlación de características dividiendo las características al azar y en muestras pequeñas. También menciona que en [43] se propuso una técnica de aprendizaje automático para detectar ataques DDoS en dispositivos IoT utilizando múltiples algoritmos. Entre ellos, se comparó los resultados de SVM, RF, Árbol de Decisión

y algoritmos de regresión logística. En ese estudio de investigación, el RF supera a otros métodos de aprendizaje automático con una precisión del 99,17 % para detectar ataques DDoS.

[44] propuso una técnica para capturar botnets Mirai mediante la evaluación de un tráfico de IoT en vivo utilizando Naïve Bayes, Random Forest y Técnicas de KNN. Los resultados muestran que el clasificador Random Forest funciona un poco lento pero proporciona el mejor rendimiento de precisión del 99% en comparación con otros algoritmos de aprendizaje automático.

[45] analizó los resultados del aprendizaje automático (SVM, KNN, Random Forest, Árbol de Decisión) y Red Neuronal Artificial (ANN) para detectar ataques DDoS en redes IoT. En su análisis, Random Forest y ANN brindan los mejores puntajes de precisión del 99% al analizar el tráfico malicioso IoT.

4.5.3.6 Árbol de Decisión (DT)

Similar a Random Forest y SVM, el algoritmo del árbol de decisiones también es capaz de manejar tareas de clasificación y regresión. También se considera uno de los algoritmos de aprendizaje automático más útiles y sencillos [29]. En un sentido técnico, un árbol de decisiones es una estructura similar a un diagrama de flujo en el que cada nodo representa una prueba de alguna característica (por ejemplo, si al lanzar una moneda sale cara o cruz). Cada nodo hoja representa una etiqueta de clase (decisión tomada después de calcular todas las características) y las ramas representan conjunciones de características que conducen a esas etiquetas de clase [27]. Los caminos desde la raíz hasta la hoja representan reglas de clasificación. Para los árboles de decisión, el enfoque principal está en cómo dividir mejor el conjunto de datos en función de ciertas condiciones.

Los Árboles de Decisión se utilizan ampliamente como clasificadores en aplicaciones de seguridad como DDoS y detección de intrusos. [21]

4.5.3.7 Regresión Lineal

Una regresión lineal se utiliza para generar predicciones sobre una variable que llamamos la variable dependiente, generalmente denominada con una “y”, dadas una o varias variables independientes generalmente denominadas las “x”.

Cuando usamos regresión lineal, la variable dependiente y siempre es numérica. Por ejemplo, precios de las casas, estaturas, la distancia de los planetas al sol, etc.

Una regresión lineal en su forma más simple (cuando contamos con solo una variable independiente x) es una línea recta en dos dimensiones que mejor se ajusta a los valores de los datos. Esto se puede ver en la Figura 7.

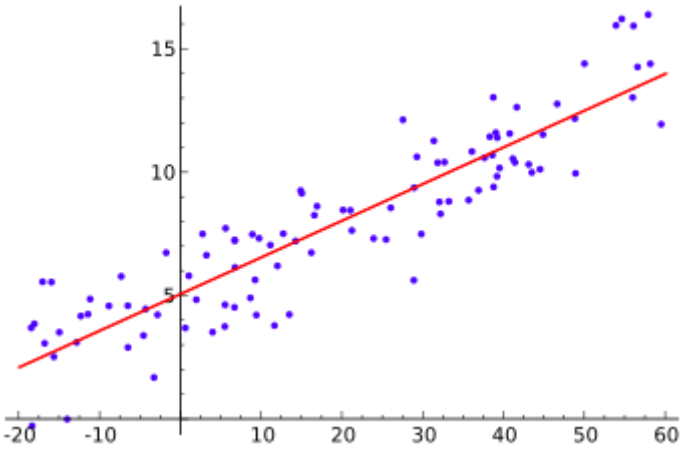

Figura 7. Ejemplo de regresión lineal de una variable dependiente y una independiente. [84]

En esta figura vemos puntos, los cuales representan a los datos verdaderos. También se observa una línea recta que representa ese modelo de regresión lineal. Esto lo podemos llevar a un ejemplo donde podríamos predecir la estatura de una persona dada su peso. Para ese caso los puntos representarían los valores verdaderos de peso y estatura de personas. La línea recta es la línea que se usaría para predecir la estatura de una persona dada su peso. [85]

4.5.3.8 Support Vector Regression (SVR)

La Regresión de Vectores de Soporte (SVR) es un algoritmo de regresión basado en los mismos algoritmos que usan las SVM para la creación de modelos de clasificación. Aunque existen algunas diferencias debido a que la salida de una regresión es un valor real y no una etiqueta, lo que hace que sea muy difícil predecir los valores objetivos. En el caso de una regresión existen infinitas posibilidades, frente al número limitado existentes en los problemas de clasificación

Mientras que en la regresión lineal se busca minimizar una función de error, generalmente el error cuadrático, en todos los puntos del conjunto de entrenamiento, en la SVR se define una zona en torno al hiperplano donde se ignora los errores. Lo que significa que en SVR se busca aproximar el mejor valor dentro de un margen dado por ε, tal como se muestra en la Figura 8.

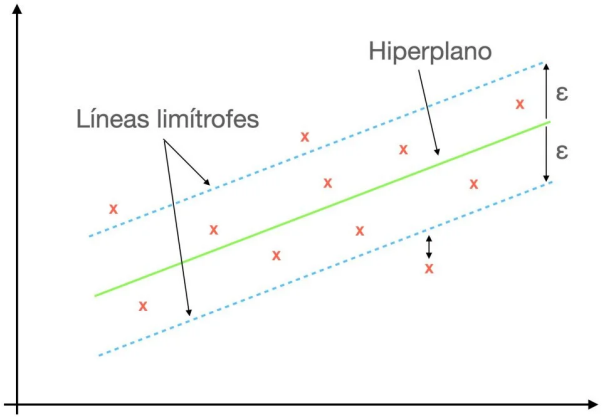

Figura 8. Funcionamiento básico de la regresión de vectores de soporte (SVR)

4.5.4 Redes Neuronales

El área de las redes neuronales va de la mano con la inteligencia artificial, el control inteligente, la Biomímesis, entre otros temas y ello ofrece un campo de trabajo novedoso, donde las aplicaciones y mejoras a sistemas avanzan a pasos cada vez más rápidos. [78]

Existen muchos procesos biológicos que pueden ser representados como un proceso de optimización con un propósito básico, control automático, automatización y toma de decisiones. Por tanto, es preciso observar que, a través del tiempo, la vida ha evolucionado para funcionar de una manera eficiente y robusta, logrando prolongar su existencia en una variedad de formas. Algunos ejemplos de ello son: las moscas, que poseen un diseño único de alas con una distribución de masa que optimiza la eficiencia del músculo; las águilas, que pueden detallar objetos a gran distancia debido a la gran densidad de conos y bastones, en comparación con el ojo humano; los camaleones, los cuales imitan el color del entorno para camuflarse de los depredadores [79], entre muchos otros. Lo anterior permite entender que en la naturaleza hay una gran fuente de inspiración para la aplicación en problemas de ingeniería. Las técnicas evolutivas a emplear son variadas, dependiendo de factores como el propósito al cual está dirigida (control de posición, encuentro de puntos óptimos, etc.) y el enfoque o inspiración natural tomada en cuenta (neuronas, bacterias, abejas, etc.).

Los métodos de la inspiración biológica pueden ser clasificados de acuerdo con las células, tejidos, órganos, organismos y jerarquía de poblaciones en la biología. Es posible extraer ideas de células u órganos (neuronas y funciones del ser humano), organismos (humanos que toman decisiones para solucionar un problema de control), o del comportamiento coordinado de un grupo de organismos (modelado de sistemas de control en vehículos autónomos con base en microorganismos que buscan nutrientes y evitan ambientes peligrosos). Además, existen adaptaciones evolutivas que suceden en sistemas biológicos y estas pueden ser útiles para mejorar el rendimiento de sistemas de control, a través de una interacción prolongada con su entorno [80]. Un ejemplo, de lo que podríamos llamar

bioinspiración, puede ser considerar entonces una neurona como se muestra en la Figura 9 para producir un modelo sencillo, como se muestra en la Figura 10.

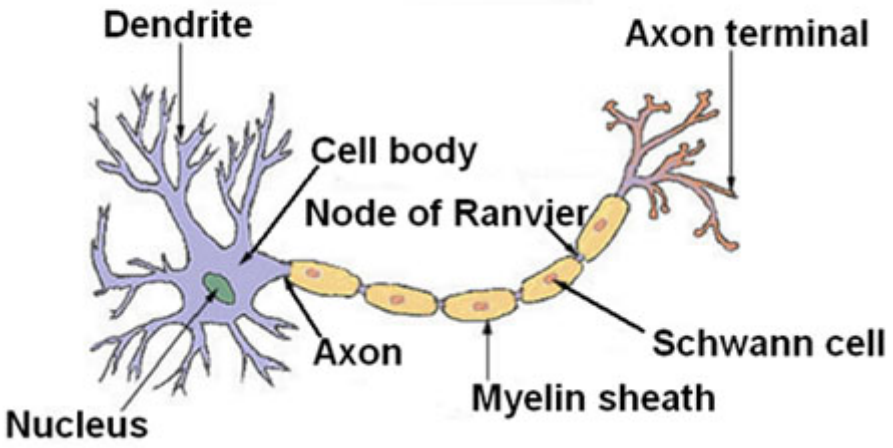

Figura 9. Estructura de una neurona típica tomada de [81]

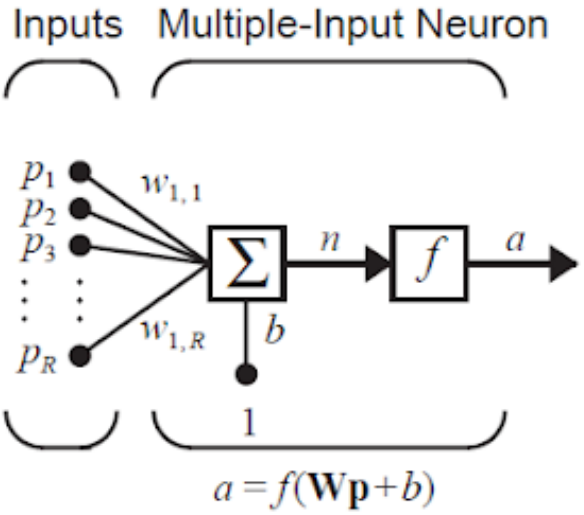

Figura 10. Modelo de una neurona [83]

La ecuación de la figura 8 nos muestra a **W** que corresponde a un vector fila con $w_1, w_2,..., w_R$ y **P** que corresponde a un vector columna con $p_1, p_2,..., p_R$ para **R** entradas. Físicamente **w** corresponde a la fuerza que posee la sinapsis entre neuronas, y es también el grado de importancia que posee una determinada señal de entrada a la neurona. El valor **p** corresponde a la señal que entra en las dendritas. El cuerpo de la neurona está representado por la sumatoria, la función de activación **f** y la salida **a**, que representa la señal sobre el axón, el cual haría parte de la señal de entrada **p**, para otra neurona. Una cadena de neuronas de este estilo puede formar una red, que puede poseer un nivel de aprendizaje, variando los parámetros **w** y **b** para reconocer patrones, agrupar datos, etc. Básicamente, aquí se ve un modelo sencillo de una neurona. Con agrupaciones de neuronas y pequeñas variaciones sobre el modelo de red de neuronas, es posible desarrollar soluciones a diversas problemáticas. [78]

En este trabajo se hará mención a las siguientes redes neuronales: ANN, CNN y RNN

4.5.4.1 Redes Neuronales Artificiales (ANN)

Son un subcampo del aprendizaje automático que se ocupa de la inteligencia artificial. El objetivo de ANN es desarrollar sistemas de aprendizaje automático basados en un modelo biológico del cerebro. Las ANN están compuestas por varias capas. Están la capa de entrada, la capa oculta y la capa de salida. En la capa de entrada, cada neurona representa cada característica de nuestro conjunto de datos. Toma las entradas y las pasa a la siguiente capa. La capa de nodos ocultos es un conjunto de neuronas donde cada neurona tiene un peso (parámetro) asignado. Toma la entrada de la capa anterior y hace el producto de entradas y pesos, aplica la función de activación y produce el resultado y pasa los datos a la siguiente capa. La capa de salida da el resultado final. Las ventajas de las ANN son que pueden trabajar con datos inadecuados y almacenan la información en toda la red. Las ANN también son competentes en el manejo de datos con alta volatilidad. También son capaces de modelar relaciones no lineales y complejas, lo que las hace altamente aplicables a varios escenarios de la vida real [29]. Sin embargo, pueden ser costosos desde el punto de vista computacional, ya que las redes neuronales a menudo exigen más potencia y pueden ser propensas al sobreajuste.

4.5.4.2 Redes Neuronales Convolucionales (CNN)

Las CNN se originaron en la década de 1950, inspiradas en experimentos biológicos que se estaban realizando durante ese período de tiempo. En 1959, David Hubel y Torsten Wiesel propusieron ambos tipos de células: simples y complejas, que se utilizaron en el reconocimiento de patrones [31].

Las CNN son otro tipo de ANN. Por lo general, las CNN se utilizan para la visión artificial y el procesamiento de imágenes, pero no necesariamente tienen que ser utilizadas para esas tareas solamente. Las CNN también se han utilizado para la creación de redes, en particular para desarrollar Network Intrusion Sistemas de Detección (NIDS). La razón por la que las CNN se han utilizado en Networking es que pueden extraer representaciones de características de alto nivel de conexiones de tráfico de red. Para las CNN, el objetivo es aprender representaciones de características adecuadas de los datos de entrada.

Una cosa a tener en cuenta acerca de las CNN es que requieren mucho tiempo de capacitación, pero pueden ser computacionalmente eficientes. También son buenos extractores de características. Sin embargo, pueden ser más difíciles de implementar debido a que CNN tiene más matemáticas involucradas en comparación con otras redes neuronales.

4.5.4.3 Redes Neuronales Recurrentes (RNN)

Las RNN son otro tipo de red neuronal, pero lo que las distingue de las redes neuronales tradicionales es que las RNN toman en cuenta los estados anteriores. Por lo general, las entradas y las salidas son independientes entre sí. Como resultado de esto, la mayoría de las

redes neuronales no recordará qué entradas recibieron hace unos momentos. Las RNN tienen bucles en su arquitectura de red, lo que permite que la información persista. La característica principal de la Arquitectura RNN es su Capa Oculta, la cual se encarga de recordar alguna información.

Las RNN tienen memoria, que considera toda la información sobre lo que se ha calculado y tienen la ventaja de recordar estados anteriores. También tienen la ventaja de poder procesar la entrada de cualquier longitud, e incluso si el tamaño de entrada es mayor, el tamaño del modelo no aumenta. Los pesos también se pueden compartir en diferentes pasos de tiempo. Las desventajas de RNN son que el cálculo es lento y entrenar modelos RNN puede ser difícil cuando el número de parámetros es extremadamente grande [26].

4.5.5 Optimización de hiperparámetros en algoritmos de Machine Learning

La optimización de hiperparámetros en ML tiene por objeto encontrar los hiperparámetros de un determinado algoritmo de ML que ofrezcan el mejor rendimiento medido en un conjunto de validación. Los hiperparámetros, a diferencia de los parámetros de los modelos, son establecidos por el feature engineering antes del entrenamiento. El número de árboles en un bosque aleatorio es un hiperparámetro, mientras que los pesos en una red neuronal son parámetros del modelo aprendidos durante el entrenamiento. Por lo tanto, los hiperparámetros son los ajustes del modelo para que el modelo pueda resolver de manera óptima el problema de aprendizaje automático.

4.5.5.1 Random Search

En Random Search, se utiliza una combinación aleatoria de hiperparámetros. El objetivo es encontrar la mejor solución para el modelo. Se ha demostrado que tiene mejores resultados porque no hay necesidad de tener en cuenta todas las posibilidades. La posibilidad de obtener parámetros óptimos es mayor gracias al patrón de búsqueda aleatorio que tiene. Como tal, existe la posibilidad de que el modelo se entrene sin requerir un esfuerzo adicional. Otro beneficio de Random Search es que puede proporcionar soluciones más rápidas y sencillas en comparación con algunos otros métodos. También tienen el potencial de resolver problemas de mayor escala, lo que puede ser útil al abordar algunos de los desafíos de redes más complejos. El Random Search se basa principalmente en examinar varias funciones, razón por la cual este método suele ser más rápido de implementar [32].

4.5.5.2 Optimización Bayesiana

El método de Optimización Bayesiana utiliza el Teorema de Bayes, que es una forma de averiguar la probabilidad de un evento. Está presentado como $P(A|B) = P(B|A) * P(A)/P(B)$ y en realidad podemos eliminar una parte de la ecuación, resultando en una forma más simplificada: $P(A|B) = P(B|A) * P(A)$. En la Optimización Bayesiana, se utiliza lo que se llama una función sustituta, que es una estimación de la

función objetivo. La función objetivo se puede escribir en la forma más simple como: $P(score/hyperparameters)$

La optimización bayesiana puede ser beneficiosa cuando se trata de funciones que son ruidosas, complejas o problemáticas para evaluar. Este método también realiza un seguimiento de los resultados de optimización anteriores, lo que lo hace más informado que otros métodos de optimización. Dado que es más eficiente, se requiere menos tiempo para la optimización. Por lo tanto, podemos encontrar hiperparámetros en menos tiempo. Como resultado, se garantiza que este método en particular tendrá un mejor rendimiento [33].

4.5.5.3 SGD (stochastic gradient descent)

SGD es uno de los muchos métodos de optimización populares y comunes para el aprendizaje automático. A diferencia del típico algoritmo de gradiente descendiente, SGD se puede utilizar para grandes conjuntos de datos. Asimismo, en lo que respecta a la actualización de coeficientes, la actualización de los mismos se realiza para cada escenario, en lugar de al final de un escenario.

Para que nuestros datos cooperen con SGD, necesitamos asegurarnos de que nuestro conjunto de datos de entrenamiento sea aleatorio. Esto es para modificar el orden de cómo se implementan las actualizaciones de los coeficientes. Debido a que los coeficientes se actualizan después de cada evento de entrenamiento, las actualizaciones pueden ser muy variadas. Al mezclar el orden de las actualizaciones de los coeficientes, evita que se distraiga o se atasque. Para actualizar los coeficientes, el costo de actualizar estos coeficientes es determinado para un evento de entrenamiento [27].

Dado que SGD puede trabajar con grandes conjuntos de datos, puede aprender más rápido y solo requiere una pequeña cantidad de iteraciones a través del conjunto de datos.

4.5.5.4 Adam

El método de optimización de Adam es una combinación de SGD y RMS-Prop (explicado en la Sección 6.3.5). El método fue presentado por Kingma y J. Ba. El método de optimización de Adam puede combinar ambos aspectos de SGD y RMS-Prop, lo que lleva a una multitud de ventajas. Adam tiene requisitos de memoria bajos y funciona bien incluso si los parámetros no se ajustan mucho. Es sencillo de implementar y es capaz de manejar gradientes ruidosos o de repuesto. También es adecuado para grandes problemas en términos de datos y/o parámetros.

4.5.5.5 RMS-Prop

El método RMS-Prop es un algoritmo de tasa de aprendizaje adaptativo propuesto por Geoff Hinton. Este método está basado en gradientes. RMS-Prop también elimina la necesidad de ajustar la tasa de aprendizaje y lo hace automáticamente. Además, RMS-Prop puede resolver eficazmente problemas de optimización que varían en dimensionalidad.

4.6 Seguridad por capa en IoT

Como se mencionó anteriormente, cada capa dentro de la arquitectura IoT posee distintas funcionalidades, por lo que se ven afectadas por diferentes amenazas. Esto hace que se requiera de medidas distintas para securizar cada capa.

4.6.1 Capa de percepción

Es la responsable de la recolección de los datos. Involucra distintos tipos de sensores que captan datos sobre una condición o evento. Es importante asegurar esta capa ya que ingresan grandes cantidades de datos al sistema y estos podrían ser dañinos o maliciosos. [11]

Dentro de esta categoría se detallan los ataques físicos que pueden afectar tanto a la red como a los dispositivos individualmente.

Interferencia de radio: se da cuando un atacante usa señales electromagnéticas para detener la conectividad entre el dispositivo y la red.

Manipulación del dispositivo: es posible cuando el dispositivo se encuentra en la fase previa a la implementación o en las fases de desarrollo/fabricación/empaquetado. En estos ataques, se manipula o modifica el desarrollo de dispositivos para que no funcionen de la manera que corresponde a la hora de la implementación.

4.6.2 Capa de aplicación

Es la capa más diversa y compleja dentro de esta arquitectura debido a la infinidad de productos, dispositivos y fabricantes diferentes que existen. Los principales motivos de preocupación son los permisos de accesos a los datos y la autenticación, donde se dificulta su administración por las transacciones masivas de datos e información por las diversidad de aplicaciones y usuarios que acceden a ellos. [4]

Los vectores de ataque dentro de esta categoría están basados en el software. Cualquier falla, desde una configuración incorrecta hasta la inyección de código SQL que permita acceder a datos privados provenientes de la base de datos puede ser un atentado contra la seguridad. Cuando se trata de software IoT, nos ocupamos principalmente de APIs y aplicaciones web específicamente.[11]

Interfaz web insegura: un atacante podría aprovechar una web insegura cuando se permiten contraseñas fáciles, el mecanismo de bloqueo de cuentas no es adecuado, no se utiliza el protocolo HTTPS para proteger la información transmitida ni firewalls en las aplicaciones o cuando la aplicación web es vulnerable a ataques como XSS, SQLi, CSRF, etc.

Autenticación insuficiente: se da cuando se utiliza un único factor de autenticación, los mecanismos de recuperación de contraseñas no están bien implementados, o los sistemas de autenticación se configuran de manera correcta.

Encriptación: un atacante puede acceder a datos confidenciales si no se encriptan los mismos tanto en la etapa de almacenamiento como en la comunicación.

Configuración incorrecta: las aplicaciones IoT dependen de la configuración de muchos sistemas y componentes para funcionar correctamente. Por lo tanto, cada uno de estos componentes requiere de una configuración de seguridad adecuada. Si no están configurados correctamente, pueden ser explotados fácilmente por un atacante. Los sistemas operativos, servidores, sistemas de gestión de bases de datos y cualquier otra aplicación deben configurarse correctamente para un entorno IoT seguro.

4.6.3 Capa de red

Como se mencionó anteriormente, esta capa se encarga de la transmisión de los datos, por lo tanto, se incluye cualquier ataque que utilicé la red para acceder a datos confidenciales o para afectar el funcionamiento del sistema. Estos ataques, entre otros, pueden ser DoS o DDos, suplantación de identidad, man-in-the-middle, etc.

Man-in-the-middle: El concepto de hombre en el medio es cuando un atacante intercepta una comunicación entre dos sistemas. Es un ataque peligroso porque es aquel en el que el atacante se hace pasar por el remitente original. Como el atacante tiene la comunicación original, puede engañar al destinatario haciéndole creer que todavía está recibiendo un mensaje legítimo.

Dentro de IoT, podemos imaginar un escenario en el que una parte malintencionada quiera falsificar datos de temperatura de un dispositivo de monitoreo para forzar el sobrecalentamiento de una máquina y, por lo tanto, detener la producción. Además de ser un inconveniente para el negocio, esto también podría causar daños físicos y financieros a la organización operativa. [11]

DoS y DDoS: los ataques de denegación de servicio, DoS (Denial of Service) por sus siglas en inglés, son ataques cuyo objetivo es inhabilitar el uso de un sistema informático. Este tipo de ataques puede tener como objetivos servidores que mantienen, por ejemplo, webs o aplicaciones. La base principal del funcionamiento de estos ataques consiste en enviar gran cantidad de peticiones de diferentes tipos a un mismo punto para que el servidor o la red a la que se envía no soporte la cantidad de paquetes recibidos y como consecuencia se produzca una interrupción del servicio proporcionado.

Los ataques de denegación de servicio distribuidos, DDoS (Distributed Denial of Service) por sus siglas en inglés, son un tipo de DoS en el cual el envío de peticiones es realizado por varios atacantes. Generalmente, estos ataques se realizan a través de botnets, que habitualmente están compuestas por ordenadores que han sido infectados y son controlados a distancia por los atacantes.

Estos ataques incrementan el daño que pueden causar a medida que existan más dispositivos conectados a la red y es por esto que la implementación de una nueva tecnología como la red 5G, debe ser evaluada. Esta última será la amenaza que se estudiará en este trabajo.

4.7 Ataques DDoS en IoT

Asegurar los dispositivos IoT es un desafío cada vez mayor para los fabricantes y los consumidores. Las configuraciones predeterminadas y los datos almacenados en línea sin contraseña son algunos de los principales desafíos de seguridad identificados por los investigadores. Los dispositivos IoT, con frecuencia, se envían con una contraseña predeterminada, fácil de recordar o sin contraseña, que hace que un atacante pueda explotar esta vulnerabilidad fácilmente y obtener acceso a estos dispositivos.

Dicha vulnerabilidad pone en riesgo la privacidad de los consumidores y permite a los atacantes utilizar dispositivos IoT para generar ataques a gran escala como DDoS. [23] informó que un registro médico no cifrado de más de 5 millones de pacientes en los EE. UU. e incluso tamaños más grandes en todo el mundo están disponibles en línea. Estos datos se almacenan en línea en más de 187 servidores y cualquiera puede acceder a ellos ejecutando una simple consulta en un navegador web. Según el informe Semantics [24], las diez principales contraseñas predeterminadas y fáciles de adivinar utilizadas en ataques a dispositivos IoT se muestran en la Fig. 11:

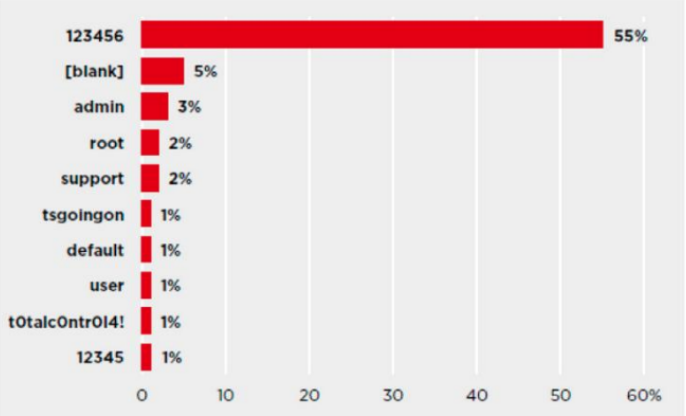

Figura 11. Top 10 de contraseñas en ataques IoT [24]

Los ataques a la red, según la escala y la gravedad, pueden costarle a una organización hasta millones de dólares. En [25] se analizan varios tipos de ataques de red que se muestran en la Figura 12.

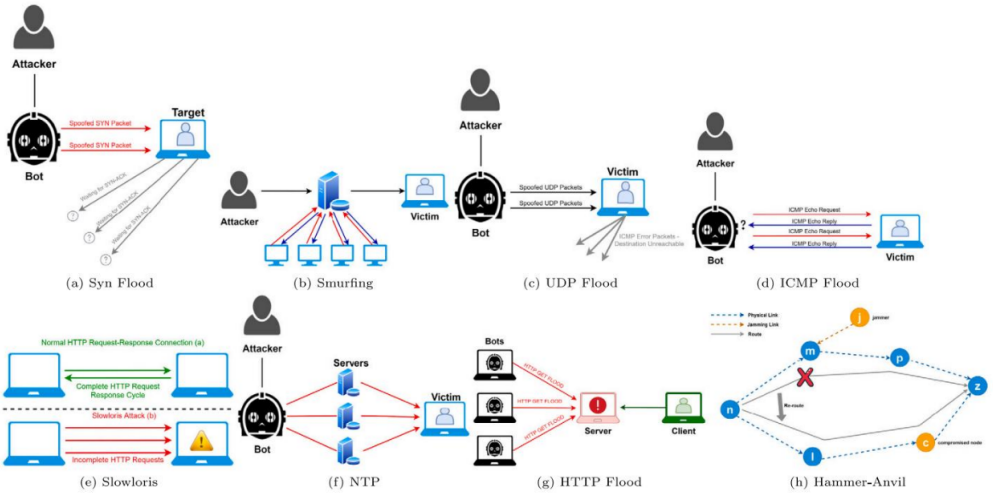

Figura 12. Resumen de varios ataques de Red. [25]

4.7.1 Syn flood attack

Este tipo de ataque se conoce desde los años 80, pero realmente atrajo la atención en 1996. Este ataque utiliza una falla en el protocolo TCP/IP para interferir con la operación. Normalmente, la interacción TCP/IP sigue los siguientes pasos (Fig. 12a):

1. El cliente quiere conectarse al servidor, por lo que envía un mensaje.
2. El servidor recibe el mensaje, por lo que le informa al cliente que lo reconoció con una notificación.
3. El cliente se comunica y se establece la conexión.

El objetivo de este ataque es lograr que los servidores no estén disponibles para un tráfico válido utilizando todos los recursos disponibles. Esto se logra mediante el envío constante de mensajes. Por lo tanto, el agresor puede interferir con todas las puertas de red disponibles en el dispositivo específico. Esto hace que dicho dispositivo no responda al tráfico de red válido en absoluto, y si responde al tráfico, lo hace con bastante lentitud.

4.7.2 Smurfing

Este ataque en particular explota el Protocolo de mensajes de control de Internet (ICMP), como se muestra en la Fig. 12b. ICMP permite a los administradores de red intercambiar información sobre el estado de la red. También se puede usar para hacer ping a otros nodos con respecto al estado de su red. Smurfing explota las características de las redes de transmisión. En un escenario típico, el host X envía un ping al host Y y esto desencadena una respuesta automática. La distancia se mide por el tiempo que tarda en llegar una respuesta. Con Broadcast Networks, envían solicitudes de ping a cada host. Entonces, los atacantes aprovechan esta falla al intentar amplificar su tráfico. Su funcionamiento es el siguiente:

1. Para Smurfing, se crea una solicitud falsa que contiene una IP de origen falsificada. Esta es en realidad la dirección del servidor de destino.
2. Luego, la solicitud se envía a una red de transmisión.
3. Esa solicitud luego se transmite a todos los otros hosts en la red.
4. Cada uno de los hosts envía una respuesta ICMP a la dirección de origen suplantada.
5. Después de tener suficientes respuestas ICMP, el servidor de destino se cae.

4.7.3 Ataque DDoS

Como se mencionó anteriormente, este tipo de ataque se da cuando se intenta forzar que un servicio en línea no esté disponible para los usuarios. Dichos ataques siguen siendo relevantes incluso hoy en día y pueden ser facilitados por los ataques Syn Flood. Los ataques DDoS pueden venir en varias categorías, tres de las cuales son importantes para tener en cuenta. Estas categorías son las siguientes:

1. Ataques a la capa de aplicación: por lo general, se dirigen a una capa de aplicación específica. Generan paquetes de protocolo de tal manera que los servidores tienen varias sesiones abiertas que conducen al agotamiento de los recursos. Estos paquetes se ven muy similares a un paquete normal y no son detectados por los algoritmos comunes de detección de anomalías.
2. Ataques basados en volumen: este tipo de ataques generan un gran volumen de tráfico de red hacia un dispositivo de destino y saturan los recursos del dispositivo que está bajo ataque. Algunos ataques DDoS típicos son UDP Flooding, ICMP Flooding, Synchronize Flooding y otros.
3. Ataques basados en protocolos: este tipo de ataques se aprovechan de las vulnerabilidades presentes en protocolos específicos como TCP. También consumen recursos del servidor, lo que hace que los servicios dejen de estar disponibles para los usuarios legítimos.

Uno de los puntos importantes que hay que tener en cuenta en los ataques DDoS es que los más lentos son más difíciles de identificar debido a su parecido con el flujo de red típico. En estos escenarios, el atacante abre una sesión con la red y la mantiene abierta durante un largo período de tiempo, enviando poco tráfico. Como resultado, la red no interrumpirá el tiempo de espera del atacante y el tráfico parece legítimo. Por lo tanto, pasará desapercibido para las herramientas de detección más comunes.

En [25] se explican algunos de los ataques DDos más comunes:

- UDP (User Datagram Protocol) Flooding: tiene como objetivo comprometer las capacidades del dispositivo de red para procesar y responder al tráfico.
- ICMP (Internet Control Message Protocol) Flooding: se envía una cantidad inusualmente alta de paquetes de red y esto hace que el servidor tenga dificultades para procesar cada paquete.

- Ping of Death: se transmiten paquetes más grandes que el tamaño permitido, lo que provoca que la máquina se bloquee o falle.
- Slowloris: el agresor intenta abrir y mantener abiertas tantas conexiones como sea posible entre él y el objetivo.
- NTP Amplifying Attack: se utiliza un servidor con el fin de generar tráfico basura. Lo hace a través de solicitudes breves, lo que puede llevar a obtener respuestas mucho más grandes. El agresor utiliza la dirección IP de la persona objetivo como fuente. Como resultado, el servidor de la persona objetivo se llena de tráfico basura.
- HTTP Flooding: en este escenario, el agresor se concentra principalmente en saturar un servidor con varios mensajes HTTP.
- Zero-Day DDoS Attack: se ocupa de cualquier nueva falla de seguridad que no se haya solucionado.

4.7.3.1 UDP Flooding

En este caso se sobrecarga las puertas de enlace de red aleatorias en un host específico con paquetes IP que contienen datos como se muestra en la Fig. 12c. El host a cargo de recibir estos paquetes necesita examinar varios elementos de la red. Si no encuentra nada, el host devuelve un paquete de destino inalcanzable. Con el tiempo, a medida que se transfieren más paquetes, el sistema se inunda y no puede comunicarse. El ataque UDP Flood recibe su nombre del Protocolo de datagramas de usuario (UDP), que es un protocolo de red menos seguro que TCP. A diferencia de TCP, UDP no tiene sesiones. Este protocolo en particular es adecuado para aplicaciones como video o chat de voz. Dado que UDP no tiene un protocolo de enlace de tres vías como TCP, la sobrecarga es menor. Sin embargo, esto hace que UDP sea vulnerable a actividades maliciosas.

4.7.3.2 ICMP Flooding

En este tipo de ataque, el atacante intenta “desarmar” la computadora de la víctima sobrecargándola con pings, como se muestra en la Fig. 12d. Aquí, el atacante inunda la red de la víctima con paquetes de solicitud y luego la red responde con la misma cantidad de paquetes de respuesta.

Esto puede afectar negativamente tanto a los canales entrantes como salientes de una red. Esto da como resultado que se consuma una cantidad significativa de ancho de banda. Por lo general, las solicitudes de ping se usan para probar las conexiones entre dos computadoras, pero en escenarios de ataque, se pueden usar para sobrecargar una red con paquetes de datos. Para ejecutar este ataque con éxito, los atacantes necesitan conocer la dirección IP de la víctima.

4.7.3.3 Slowloris

Este ataque fue desarrollado por Robert Rsnake Hanson. Este se considera uno de los ataques más simples porque requiere una cantidad mínima de ancho de banda. El ataque opera implementando múltiples conexiones al host de destino y manteniéndolas durante

mucho tiempo. Esto se logra a través de solicitudes incompletas, que nunca se realizan. Los servidores comprometidos generan aún más conexiones, por lo que pueden esperar a que se ejecute cada solicitud. En última instancia, la capacidad del host está llena, luego se rechazan los intentos válidos de conexión, como se ilustra en la Fig. 12e. Al enviar paquetes sin terminar, este ataque no es detectado por los sistemas tradicionales de detección de intrusos.

4.7.3.4 NTP amplification

Este es un tipo de ataque DDoS que explota uno de los protocolos más antiguos, el NTP (Network Time Protocol) mediante el envío de tráfico UDP, como se muestra en la Fig. 12f. NTP se define como un tipo de protocolo que se utiliza para garantizar que las horas de reloj de la computadora para una red estén sincronizadas correctamente. El protocolo NTP fue diseñado para ser tolerante a fallas y muy escalable. La versión anterior de NTP a menudo tendrá un servicio de monitoreo que permite a los administradores obtener un recuento del tráfico del servidor. El comando que utilizan envía información sobre los hosts conectados previamente. El ataque de amplificación NTP se considera un ataque de reflexión porque implica obtener una respuesta de un servidor a una dirección IP falsificada entre el mismo protocolo en ambas direcciones. Los ataques de reflexión pueden ser peligrosos, pero lo son aún más cuando se amplifican. En esta situación, el atacante envía repetidamente comandos al host, consiguiendo que la dirección IP del host coincida con la del servidor de la víctima. Posteriormente, la NTP entrega el expediente a la dirección obtenida. La cantidad de flujo de red transferido aumenta considerablemente, lo que genera un servicio deficiente para las interacciones válidas.

4.7.3.5 HTTP Flooding

Estos tipos de ataques son particularmente difíciles de resolver porque son casi indistinguibles del tráfico de red válido. Los ataques de inundación HTTP son difíciles de detectar porque no se basan en paquetes con formato incorrecto, suplantación de identidad o técnicas de reflexión. Como resultado, caen por debajo del umbral de detección de un sistema. Estos ataques suelen tener como objetivo la capa 7 del modelo OSI en el nivel de la aplicación. En este ataque, la idea es sobrecargar a la víctima con solicitudes HTTP para que no pueda responder al tráfico, como se muestra en la Fig. 12g. Con HTTP Flood, las redes de bots generalmente se usan para mejorar el impacto del ataque. Muchos de los dispositivos utilizados para este ataque están infectados con malware y el atacante puede lanzar un mayor volumen de tráfico.

4.7.3.6 Zero day DDoS attack

Esto se conoce como ataques que explotan nuevas vulnerabilidades de seguridad. Desafortunadamente, actualmente no existen enfoques adecuados para protegerse de este tipo de ataques. El motivo es que puede pasar mucho tiempo desde el momento en que se detecta la vulnerabilidad hasta el lanzamiento y la instalación de un nuevo parche. Mientras se espera esa solución, la vulnerabilidad puede explotarse para bloquear recursos potenciales o robar

información valiosa. Con enfoques avanzados de aprendizaje profundo, existen varios estudios prometedores sobre cómo combatir tales ataques de seguridad.

5. Desarrollo

En un proceso de Machine Learning, la definición de la técnica utilizada está relacionada con el objetivo deseado y con los datos y tipos de datos disponibles. Es por este motivo, que se analizaron trabajos en los cuales se utilizaron distintos algoritmos de Machine Learning con el objetivo de securizar dispositivos IoT o para la prevención y detección de ataques DoS o DDoS. La evaluación de los algoritmos se basó en las métricas típicas que son: accuracy, precision, recall y métrica F1, que serán explicadas más adelante.

5.1 Uso de Machine Learning para securizar las capas IoT

5.1.1 Capa de percepción

Distintos algoritmos de ML se han propuesto para securizar la capa de percepción o capa física. En [36] se propuso un esquema centralizado en ML para la seguridad de los dispositivos IoT. Básicamente, permite que ciertos usuarios con autorización se comuniquen con el sistema y almacenen de forma segura la información de los usuarios autorizados. En el esquema de protocolo de seguridad peer-to-peer propuesto, los clientes deben registrarse primero en el servidor de la nube antes de iniciar la comunicación en el sistema IoT. Además, propusieron un modelo para evitar ataques y asegurar dispositivos IoT utilizando Neural Network (NN) y el algoritmo ElGamal. Aquí se utilizaron claves privadas y públicas para controlar su criptosistema. Los datos manipulados se segmentan en grupos y luego se comparan con los datos de entrenamiento.

5.1.2 Capa de aplicación

Los métodos K-NN, Random Forest, Q-learning, ML basados en Dyna-Q se utilizaron ampliamente para proteger los dispositivos IoT de ataques basados en aplicaciones/web, especialmente para la detección de malware [38]. En [39] se utilizaron técnicas de aprendizaje automático supervisado (tanto K-NN como Random Forest) para detectar ataques de malware e informaron que los métodos de Random Forest con conjuntos de datos de MalGenome ofrecen una mejor tasa de detección que K-NN. En otra investigación, Q-learning muestra un mejor rendimiento en términos de detección de latencia y precisión que el método de aprendizaje de detección basado en Dyna-Q [38].

5.1.3 Capa de red

A la hora de querer securizar el sistema IoT frente a ataques DDoS, la capa principal que queremos proteger es la capa de red. Para esto se utilizan diferentes algoritmos de ML supervisado como SVM, NN y KNN para detectar el ataque de intrusión.

En [37] se propuso un modelo para ataques DDoS usando un algoritmo ANN. En el esquema propuesto, solo los paquetes de información real tienen permiso para transmitir a

través de la red en lugar de los falsos. ANN se desempeñó mejor en la detección de ataques DDoS solo si fue entrenado con conjuntos de datos actualizados.

En 2018, Doshi con sus colegas en [35] presentó una forma de detectar ataques DDoS en dispositivos IoT locales utilizando algoritmos de aprendizaje automático de bajo costo y datos de tráfico independientes del protocolo y basados en flujo.

En este modelo propuesto, se han considerado algunos comportamientos limitados de la red IoT, como el cálculo de los puntos finales y el tiempo necesario para viajar de un paquete a otro (intervalos de tiempo entre paquetes). Compararon una variedad de clasificadores para la detección de ataques, incluidos KNN, el algoritmo KDTree, SVM con el kernel lineal (LSVM), Árbol de Decisión con puntajes de impurezas de Gini, Random Forest con puntajes de impurezas de Gini, NN. Se informó que las técnicas propuestas pueden identificar ataques DDoS en dispositivos IoT locales utilizando routers de puerta de enlace domésticos y otras cajas intermedias de red.

La precisión del conjunto de prueba para cinco algoritmos es superior a 0,99.

5.2 Machine Learning para la prevención de DDoS en IoT

En los últimos años, las técnicas de ML avanzaron notablemente brindando seguridad a los dispositivos IoT [34].

En [25] se utilizaron un total de nueve algoritmos divididos en dos categorías: algoritmos clásicos de ML (Naive Bayes, K-Nearest Neighbors, K-Means Clustering, SVM, Random Forest y Árbol de decisión) y redes neuronales (ANN, CNN y RNN), proporcionando una evaluación empírica a gran escala con optimización y sin técnicas de optimización.

En [25] se llegó a la conclusión de que la aplicación de optimización en los métodos de ML tiene un impacto positivo en el rendimiento de los modelos. En lo que respecta a los métodos clásicos de ML, el algoritmo del árbol de decisión fue el que mejor se desempeñó con y sin optimización. Por otro lado, K-means tuvo el peor rendimiento incluso cuando se le aplicaron optimizaciones.

En cuanto a las redes neuronales, en [25] se vió que RNN se desempeñó mejor en los tres conjuntos de datos.

Por último, el análisis de las proporciones de entrenamiento y prueba reveló que 80% de entrenamiento y 20% de prueba fue el más óptimo.

En [61] se propuso un método de detección de intrusos para ataques de inyección que también podría utilizarse para ataques DDoS. En este artículo se utilizaron tres algoritmos clásicos de ML de clasificación: SVM, Random Forest y Decision Tree. Por otra parte, se utilizaron cuatro parámetros de evaluación para comparar los algoritmos:

- Accuracy: relación entre el número de registros que se clasifican correctamente y el número total de registros. Cuanto mayor sea la precisión, mejor será el modelo aplicado.
- Precision: relación entre el número de registros positivos que se clasifican correctamente y el número total de registros positivos. Esto significa que cuanto mayor sea la tasa de falsos positivos, mayor será el resultado de precision. La medida de precision es una buena medida cuando el costo de un falso positivo es alto.
- Recall: relación entre el número de registros positivos que se clasifican correctamente y el número total de clasificaciones en la clase real. Esto significa que cuanto mayor sea la tasa de FN (falso negativos), menor será el valor de recall. La medida de recall es importante cuando queremos seleccionar el mejor modelo en caso de que la tasa de FN sea muy alta.
- Métrica F1: esta medida se conoce como la media armónica de las medidas de precision y recall. Se considera una buena medida de evaluación para datos desequilibrados.

Finalmente, para que la comparación sea lo más adecuada posible se realizaron tres experimentos tomando distintas cantidades de características, que es el proceso de reducir la dimensionalidad de los datos para mejorar el rendimiento.

Los resultados obtenidos fueron los siguientes:

Classification results using different set of features.

	Accuracy	Precision	Recall	F1 Score
	Results of using 76 features			
Decision tree	0.9681	0.6731	0.8500	0.7513
Random forest	0.9888	0.9511	0.8470	0.8960
SVM	0.9758	0.7008	0.9999	0.8240
	Results of using 13 features			
Decision tree	0.9908	0.9539	0.8807	0.9158
Random forest	0.9887	0.9507	0.8439	0.8941
SVM	0.9757	0.7001	0.9999	0.8235
	Results of using 8 features			
Decision tree	0.9891	0.9521	0.8500	0.8982
Random forest	0.9891	0.9537	0.8492	0.8985
SVM	0.9756	0.6989	0.9999	0.8227

Figura 13. Tabla de comparación de resultados obtenidos en [61]

En esta comparación de los tres algoritmos con el mismo dataset AWID (conjunto de registros públicos que se asemejan a rastros reales de tráfico WiFi), se pudo notar que los resultados obtenidos, en el primer experimento con 76 características, con RF fueron los mejores con accuracy, precision y F1 del 98.88%, 95.11% y 89.60% respectivamente. Aún así, SVM tuvo el mejor resultado de recall con un 99.99%. Por otra parte en el segundo experimento, con trece características se obtuvo el mejor resultado con DT logrando un accuracy del 99.08% y un F1 del 91.58%. Por último, en el tercer experimento, utilizando 8

características, se puede ver que el DT y RF dieron el mismo valor de accuracy, siendo este el más alto con 98.91%.

Analizando todo el contexto, en [61] se concluye que el árbol de decisión es el mejor clasificador que se puede usar para detectar intrusos en la red (para ataques de inyección) y esto puede servir también para casos en los que se quiere prevenir un ataque de DDoS.

En [62] se propone una arquitectura inteligente que integra la tecnología de procesamiento de eventos complejos (CEP) y el paradigma de aprendizaje automático (ML) para detectar diferentes tipos de ataques de seguridad de IoT en tiempo real. En particular, dicha arquitectura es capaz de gestionar fácilmente patrones de eventos cuyas condiciones dependen de los valores obtenidos por los algoritmos de ML. En este artículo las técnicas de ML utilizadas se basan en el aprendizaje supervisado. En particular, se utiliza un método de regresión lineal para modelar la relación entre una respuesta escalar y muchas variables explicativas. El modelo propuesto se puede utilizar para ajustar un modelo predictivo generado a partir de un conjunto de datos observados de valores extraídos de un escenario de red regular. También se utiliza el SVR y los resultados se muestran a continuación:

Results of Linear Regression as predictor.

	Predicted as negative	Predicted as positive
Is negative	7930 (True negative)	0 (False positive)
Is positive	0 (False negative)	104,223 (True positive)
Precision = 1	Recall = 1	F1-score= 1

Figura 14. Resultados de las predicciones con Regresión Lineal. [62]

La figura 14 muestra que se obtiene una puntuación perfecta y que este modelo de regresión lineal funciona muy bien en un escenario real contra ataques desconocidos.

Results of SVR as predictor.

	Predicted as negative	Predicted as positive
Is negative	7928 (True negative)	2 (False positive)
Is positive	0 (False negative)	104,223 (True positive)
Precission = 0.99998	Recall = 1	F1-score= 0.99998

Figura 15. Resultados de las predicciones con SVR. [62]

Por otra parte, la figura 15 demuestra que el modelo basado en SVR funciona peor que el de regresión lineal pero aun así sigue funcionando bastante bien.

De todas maneras, en [62] se menciona que estos resultados son los obtenidos en un escenario en particular, con un cierto dataset y un determinado proceso de feature engineering. No se descarta que ante otro escenario pueda funcionar mejor el SVR.

En [40] se muestra un resumen de publicaciones de investigación, sus modelos y métricas de rendimiento. Ya se vió anteriormente que hay muchos parámetros a tener en cuenta a la

hora de elegir el mejor algoritmo de clasificación, porque principalmente depende mucho del dataset con el que estemos trabajando y de qué tan equilibrados sean los datos, pero a grandes rasgos puede apreciarse que en todos los estudios que se usaron redes neuronales y aprendizaje automático para comparar, las redes neuronales revelan resultados mucho mejores que los algoritmos de aprendizaje automático.

Por otro lado, también se revela en [40] que las redes neuronales son la solución más popular para los ataques de IoT a gran escala debido a que son capaces de procesar conjuntos de datos de manera efectiva sin necesidad de manipular las características. Los métodos de ML clásicos requieren un tedioso proceso de feature engineering para extraer primero las features creando features nuevas o más pequeñas y luego seleccionando un subconjunto de features más relevantes. Este proceso requiere mucho tiempo y es propenso a errores. También depende en gran medida del conocimiento del dominio. Se manifiesta también cuál es la relación entre los distintos tipos de IA según el crecimiento de datos:

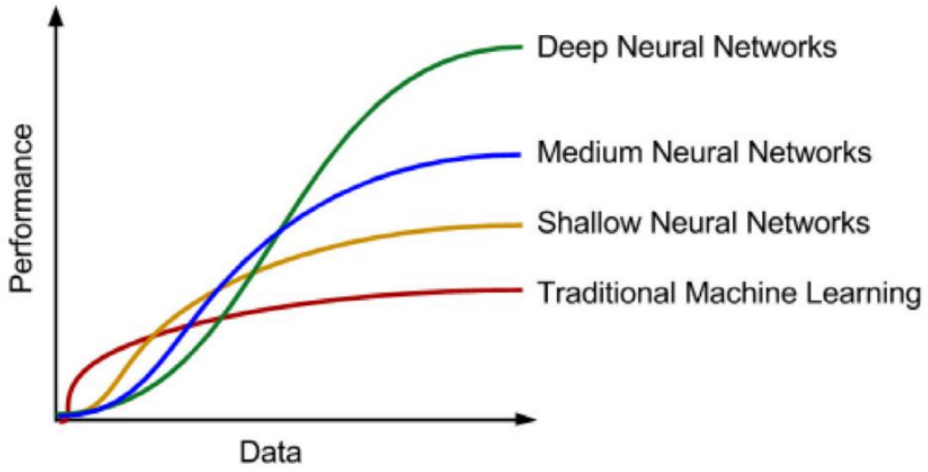

Figura 16. Rendimiento de modelos de IA con crecimiento de datos.[40]

La Figura 16 muestra que inicialmente, cuando el tamaño de los datos es pequeño, los modelos tradicionales de aprendizaje automático mejoran rápidamente y superan a los modelos de aprendizaje profundo. Sin embargo, a medida que el tamaño de los datos comienza a crecer, el rendimiento de los modelos de aprendizaje automático se estanca en algún momento y permanece estancado después de eso. Por el contrario, el rendimiento de la red neuronal profunda crece rápidamente con el crecimiento del conjunto de datos. Este fenómeno encaja muy bien con nuestra infraestructura IoT, donde el tamaño de los datos crece exponencialmente con miles de millones de dispositivos conectados y el enorme volumen de tráfico de red generado por estos dispositivos. De manera similar, cuando ocurre un ataque DDoS, el volumen de tráfico generado es masivo y diverso. Los algoritmos tradicionales de aprendizaje automático entrenados en un pequeño conjunto de datos sin capacidad de aprendizaje para datos no vistos no podrán funcionar bien y detectar con precisión el tráfico malicioso.

5.4 Optimización aplicada a los algoritmos de ML

Cinco métodos de optimización fueron utilizados en [25] para la optimización de los algoritmos. Para los algoritmos de Aprendizaje Automáticos Clásicos, se aplicaron dos métodos: Random Search y Optimización Bayesiana. Por otro lado, para las redes neuronales se utilizaron tres métodos de optimización: SGD, ADAM y RMS-Prop.

Utilizando tres conjuntos de datos distintos (KDD, NSL-KDD, ADFA-17) se demostró en [25] que no existe un método que sea absolutamente mejor que el resto sino que depende del conjunto de datos que se utilice. En este caso puntual, los métodos de optimización de hiperparámetros que mejor funcionaron fueron Optimización Bayesiana para los métodos clásicos de ML y Adam para redes neuronales. Los resultados obtenidos se muestran a continuación:

Optimized results for classic ML methods using Bayesian optimization.

Algorithms				
Decision tree				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.9937	0.9941	0.9942	0.9941
NSL KDD	0.9967	0.9967	0.9971	0.9969
ADFA-IDS-17	0.9082	0.9034	0.929	0.9087
Naive Bayes				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.9058	0.8805	0.9941	0.9185
NSL KDD	0.9068	0.8812	0.9948	0.9195
ADFA-IDS-17	0.754	0.8246	0.6461	0.7234
K-Means				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.463	0.3716	0.5009	0.4264
NSL KDD	0.1897	0.0034	0.002	0.0026
ADFA-IDS-17	0.5597	0.7768	0.4741	0.4121
Random forest				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.9216	0.9913	0.986	0.9304
NSL KDD	0.9456	0.9273	0.9873	0.9504
ADFA-IDS-17	0.8525	0.7951	0.9504	0.8657
SVM				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.9921	0.9941	0.9913	0.9926
NSL KDD	0.9956	0.9953	0.9963	0.9958
ADFA-IDS-17	0.6921	0.6241	0.98	0.7636
KNN				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.9915	0.9937	0.9926	0.992
NSL KDD	0.9955	0.9963	0.9966	0.9958
ADFA-IDS-17	0.8134	0.8842	0.7122	0.7939

Figura 17. Resultados de métodos de ML clásicos con optimización bayesiana. [25]

Optimized results for classic ML methods using random search.

Algorithms				
Decision tree				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.9937	0.9941	0.9941	0.9942
NSL KDD	0.9967	0.9967	0.9971	0.9969
ADFA-IDS-17	0.9082	0.9034	0.929	0.9087
Naive Bayes				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.9058	0.8805	0.9941	0.9185
NSL KDD	0.9068	0.8812	0.9948	0.9195
ADFA-IDS-17	0.754	0.8246	0.6461	0.7234
K-Means				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.463	0.3716	0.5009	0.4264
NSL KDD	0.1897	0.0034	0.002	0.0026
ADFA-IDS-17	0.5597	0.7768	0.4741	0.4121
Random forest				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.9209	0.9913	0.9869	0.9297
NSL KDD	0.9456	0.9273	0.9878	0.9504
ADFA-IDS-17	0.8499	0.7917	0.9504	0.8636
SVM				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.9921	0.9941	0.9913	0.9926
NSL KDD	0.9956	0.9953	0.9963	0.9958
ADFA-IDS-17	0.6921	0.6241	0.98	0.7636
KNN				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.9915	0.9937	0.9926	0.992
NSL KDD	0.9955	0.9963	0.9966	0.9958
ADFA-IDS-17	0.8134	0.8842	0.7122	0.7939

Figura 18. Resultados de métodos de ML clásicos optimizados con Random Search. [25]

El método del árbol de decisión tuvo los resultados de rendimiento más altos en los tres conjuntos de datos, especialmente para el conjunto de datos ADFA-IDS-2017. Ambos métodos, Random Search y Bayesiana, lograron resultados óptimos. Además, se encontró que la relación entrenamiento-prueba que logró los resultados más altos fueron la relación 90% entrenamiento, 10% prueba. El rendimiento de K-Means Clustering fue muy bueno para los conjuntos de datos KDD-Cup y NSL-KDD, pero muy malo en los conjuntos de datos ADFA-IDS-2017, incluso después de que se le aplicaron los métodos Bayesiano y Random Search Optimization. Es posible que el conjunto de datos ADFA-IDS-2017 era muy diferente a los otros dos conjuntos de datos en términos de estructura, lo que provocó que el agrupamiento de K-Means no funcionara tan bien.

Se debe tener en cuenta que, si bien el método del Árbol de Decisión logró los resultados más altos en general en los tres conjuntos de datos, el método clásico de ML más fuerte es el SVM. Sin embargo, el SVM no lo superó. Lo que esto podría indicar es que tal vez sea óptimo adoptar un enfoque de menos es más al elegir qué método de ML a aplicar. En

ocasiones, es posible que los métodos más potentes no siempre logren los mejores resultados, según los conjuntos de datos utilizados.

Por otro lado, tenemos los resultados de las redes neuronales:

Optimized results for neural network-based methods.

Neural Networks-Training: 70% Testing: 30%				
ANN				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.96	0.95	0.97	0.97
NSL-KDD	0.98	0.99	0.98	0.99
ADFA-17	0.88	0.82	0.81	0.60
RNN				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.98	0.98	0.98	0.98
NSL-KDD	0.95	0.98	0.98	0.99
ADFA-17	0.88	0.90	0.88	0.91
CNN				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.95	0.95	0.95	0.95
NSL-KDD	0.96	0.95	0.96	0.95
ADFA-17	0.86	0.86	0.90	0.88
Neural Networks-Training: 80% Testing: 20%				
ANN				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.97	0.98	0.97	0.97
NSL-KDD	0.98	0.98	0.98	0.98
ADFA-17	0.78	0.82	0.84	0.87
RNN				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.99	0.99	0.99	0.99
NSL-KDD	0.99	0.99	0.99	0.99
ADFA-17	0.91	0.89	0.90	0.93
CNN				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.95	0.95	0.95	0.95
NSL-KDD	0.95	0.95	0.96	0.95
ADFA-17	0.91	0.87	0.87	0.90
Neural Networks-Training: 90% Testing: 10%				
ANN				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.95	0.98	0.98	0.97
NSL-KDD	0.99	0.99	0.98	0.98
ADFA-17	0.88	0.90	0.90	0.75
RNN				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.99	0.98	0.99	0.99
NSL-KDD	0.99	0.99	0.99	0.99
ADFA-17	0.90	0.90	0.87	0.90
CNN				
Dataset	Accuracy	Precision	Recall	F1-Score
KDD	0.95	0.95	0.96	0.95
NSL-KDD	0.96	0.95	0.96	0.95
ADFA-17	0.87	0.88	0.87	0.87

Figura 19. Resultados optimizados de Redes Neuronales. [25]

Al estar optimizados, tanto ANN como CNN, sus resultados en el conjunto de datos ADFA-IDS-2017 fueron mucho mejores, aunque no superaron a RNN. Si bien los tres NN se pueden usar para redes, proponemos que sea mejor usar el RNN. A diferencia de las otras ANN y CNN, RNN puede recordar estados y entradas anteriores, lo que puede ser útil al examinar el comportamiento anterior de una red. Además, se notó que el método de optimización de Adam es el más óptimo para las NN.

6. Conclusiones

Este trabajo busca ser en un futuro la punta de flecha que logre mejorar la seguridad de la información en la arquitectura IoT implementada en redes 5G. Se hizo un análisis de seguridad empezando por la arquitectura de los dispositivos IoT ya que el primer punto para desarrollar un sistema seguro es reconocer los vectores de ataques. Luego, se analizaron varios artículos en los cuales se probaron y compararon distintos algoritmos y métodos de aprendizaje automático y redes neuronales para brindar solución a esta problemática planteada.

La nueva generación de la red 5G permite el desarrollo de nuevas aplicaciones y proporciona mejoras notables con respecto a las anteriores tecnologías de redes celulares. Sin embargo, al ser una tecnología aún en desarrollo que no ha sido implementada a nivel mundial, presenta vulnerabilidades a pesar de los grandes beneficios que trae. Este avance tecnológico, requiere de nuevas medidas de seguridad y estudio constante para hacer frente a los ataques que ya son conocidos y estar preparados para los que aún quedan por conocer. Los mecanismos para mitigar los ataques de seguridad en redes 5G deben ser diseñados e implementados en un entorno de desarrollo constante.

Este trabajo puntualmente se enfoca en la prevención de ataques DDoS contra los dispositivos IoT, con lo cual la capa de la arquitectura IoT que se quiere proteger es la capa de red. En esta capa, hay que tener en cuenta que el tráfico de red puede llegar a ser muy diverso y heterogéneo. Bajo estas condiciones, estoy convencido que la seguridad de los dispositivos IoT, cuando sea la hora de implementarse sobre la red 5G, debe estar basada principalmente en la IA para detectar anomalías monitoreando la red y de esta forma poder prevenir ataques maliciosos. Además, los vectores de ataque también mejorarán al aprender los modos de operar de la red.

Es por esto, que la mejor solución para hacer frente a estos problemas será la de implementar un modelo de Redes Neuronales a nivel de paquete para detectar, contener y prevenir ataques y amenazas. Esto es lo que se conoce como inteligencia de amenazas, y debido a la diversidad en el tráfico de las redes, un modelo de Redes Neuronales bien entrenado será capaz de detectar aquel tráfico malicioso y así poder actuar y prevenir este tipo de ataques.

7. Futuros trabajos

Los ataques DDoS serán cada vez más lentos, lo que dificultaría su detección. Por lo tanto, una vía de investigación futura sería la evaluación empírica de algoritmos de Machine Learning y Redes Neuronales en redes móviles 5G para estudiar el tráfico producido por dispositivos IoT y así poder identificar el tráfico malicioso con mayor precisión.

El primer gran desafío será encontrar un conjunto de datos que pueda representar de la mejor manera al tráfico generado en una red 5G en la cuál miles de millones de dispositivos IoT se conectan y transmiten datos constantemente. Una vez encontrados los datos, estos se utilizarán para entrenar distintos modelos y compararlos con el fin de determinar cuál es el que obtiene mejores resultados. También deberá realizarse una comparación antes y después de hacer feature engineering y optimización de hiperparámetros ya que se pudo demostrar que los modelos varían su efectividad según las características del conjunto de datos utilizados.

8. Referencias-Bibliografía

[1] Abomhara, M., & Koien, G. M. (2014). Security and privacy in the Internet of Things: Current status and open issues. *2014 International Conference on Privacy and Security in Mobile Systems (PRISMS)*. IEEE. Retrieved from http://dx.doi.org/10.1109/prisms.2014.6970594

[2] admin. (2020, February 6). Man-in-the-Middle Attacks in the IoT. Retrieved September 3, 2022, from GlobalSign website: https://www.globalsign.com/en/blog/man-in-the-middle-attacks-iot

[3] Afaq, A., Haider, N., Baig, M. Z., Khan, K. S., Imran, M., & Razzak, I. (2021). Machine learning for 5G security: Architecture, recent advances, and challenges. *Ad Hoc Networks*, *123*, 102667. https://doi.org/10.1016/j.adhoc.2021.102667

[4] Babar, S., Mahalle, P., Stango, A., Prasad, N., & Prasad, R. (2010). Proposed Security Model and Threat Taxonomy for the Internet of Things (IoT). In *Recent Trends in Network Security and Applications* (pp. 420–429). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://dx.doi.org/10.1007/978-3-642-14478-3_42

[5] Dorsemaine, B., Gaulier, J.-P., Wary, J.-P., Kheir, N., & Urien, P. (2016). A new approach to investigate IoT threats based on a four layer model. *2016 13th International Conference on New Technologies for Distributed Systems (NOTERE)*. IEEE. Retrieved from http://dx.doi.org/10.1109/notere.2016.7745830

[6] González, C. (2019). Desafíos de Seguridad en Redes 5G. Retrieved from Https://jadimike.unachi.ac.pa/ website: https://jadimike.unachi.ac.pa/bitstream/handle/123456789/132/paper%20technology%20inside.pdf?sequence=1&isAllowed=y

[7] Hossain, Md. M., Fotouhi, M., & Hasan, R. (2015). Towards an Analysis of Security Issues, Challenges, and Open Problems in the Internet of Things. *2015 IEEE World Congress on Services*. IEEE. Retrieved from http://dx.doi.org/10.1109/services.2015.12

[8] Jiménez, J. (2020, January 2). Ataques más comunes a los dispositivos IoT. *RedesZone*. Retrieved from https://www.redeszone.net/tutoriales/seguridad/ataques-comunes-dispositivos-iot-seguridad/

[9] Las 4 etapas de la arquitectura IoT. (n.d.). Retrieved September 2, 2022, from Digi International website: https://es.digi.com/blog/post/the-4-stages-of-iot-architecture (accedido 9/9/22)

[10] NIS Cooperation Group. (2019). *EU coordinated risk assessment of the cybersecurity of 5G networks*.

[11] Rizvi, S., Kurtz, A., Pfeffer, J., & Rizvi, M. (2018). Securing the Internet of Things (IoT): A Security Taxonomy for IoT. *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*. IEEE. Retrieved from http://dx.doi.org/10.1109/trustcom/bigdatase.2018.00034

[12] The global IoT market opportunity will reach USD4.3 trillion by 2024. (n.d.). Retrieved March 8, 2022, from https://machinaresearch.com/news/the-global-iot-market-opportunity-will-reach-usd43-trillion-by-2024/

[13] IoT: Origen, importancia en el presente y perspectiva de futuro. (n.d.). Retrieved September 17, 2022, from https://www.itop.es/blog/item/iot-origen-importancia-en-el-presente-y-perspectiva-de-futuro.html (accedido 10/9/22)

[14] Breve historia de Internet de las cosas (IoT). (2020, September 22). Retrieved September 17, 2022, from Think Big website: https://empresas.blogthinkbig.com/breve-historia-de-internet-de-las-cosas-iot/ (accedido 10/9/22)

[15] What is Cloud Computing? (n.d.). Retrieved September 18, 2022, from Amazon Web Services, Inc. website: https://aws.amazon.com/es/what-is/iot/ (accedido 11/9/22)

[16] School, T. (2022, April 19). Historia y evolución del Internet de las Cosas (IoT). Retrieved September 18, 2022, from Tokio School website: https://www.tokioschool.com/noticias/internet-de-las-cosas-evolucion/ (accedido 11/9/22)

[17] Asghari, P., Rahmani, A. M., & Javadi, H. H. S. (2018). Service composition approaches in IoT: A systematic review. *Journal of Network and Computer Applications*, *120*, 61–77. https://doi.org/10.1016/j.jnca.2018.07.013

[18] Tsai, C.-W., Lai, C.-F., Chiang, M.-C., & Yang, L. T. (2014). Data mining for internet of things: A survey. *IEEE Communications Surveys & Tutorials*, *16*(1), 77–97. https://doi.org/10.1109/surv.2013.103013.00206

[19] Ahmed, E., Yaqoob, I., Hashem, I. A. T., Khan, I., Ahmed, A. I. A., Imran, M., & Vasilakos, A. V. (2017). The role of big data analytics in Internet of Things. *Computer Networks*, *129*, 459–471. https://doi.org/10.1016/j.comnet.2017.06.013

[20] Singh, A., Payal, A., & Bharti, S. (2019). A walkthrough of the emerging IoT paradigm: Visualizing inside functionalities, key features, and open issues. *Journal of Network and Computer Applications*, *143*, 111–151. https://doi.org/10.1016/j.jnca.2019.06.013

[21] Tahsien, S. M., Karimipour, H., & Spachos, P. (2020). Machine learning based solutions

for security of Internet of Things (IoT): A survey. *Journal of Network and Computer Applications*, *161*, 102630. https://doi.org/10.1016/j.jnca.2020.102630

[22] Helena. (2020, July 14). Seguridad de la información: Aspectos a tener en cuenta. Retrieved September 23, 2022, from AyudaLeyProteccionDatos website: https://ayudaleyprotecciondatos.es/2020/07/14/seguridad-de-la-informacion/ (accedido 11/9/22)

[23] Gillum, J. (2019, September 17). Millions of Americans' medical images and data are available on the internet. Anyone can take a peek. Retrieved September 24, 2022, from ProPublica website: https://www.propublica.org/article/millions-of-americans-medical-images-and-data-are-available-on-the-internet (accedido 11/9/22)

[24] Threat landscape trends – Q1 2020. (n.d.). Retrieved September 24, 2022, from Broadcom Software Blogs website: https://symantec-enterprise-blogs.security.com/blogs/threat-intelligence/threat-landscape-q1-2020 (accedido 11/9/22)

[25] Aledhari, M., Razzak, R., & Parizi, R. M. (2021). Machine learning for network application security: Empirical evaluation and optimization. *Computers & Electrical Engineering*, *91*, 107052. https://doi.org/10.1016/j.compeleceng.2021.107052

[26] Jamshidi S. (2019). The applications of machine learning techniques in networking.

[27] Natalino, C., Schiano, M., Di Giglio, A., Wosinska, L., & Furdek, M. (2019). Experimental study of machine-learning-based detection and identification of physical-layer attacks in optical networks. *Journal of Lightwave Technology*, *37*(16), 4173–4182. https://doi.org/10.1109/jlt.2019.2923558

[28] Sharma, D., Thulasiraman, K., Wu, D., & Jiang, J. N. (2019). A network science-based k-means++ clustering method for power systems network equivalence. *Computational Social Networks*, *6*(1). https://doi.org/10.1186/s40649-019-0064-3

[29] Bhutani, G. (2014). Application of machine-learning based prediction techniques in wireless networks. *Int'l J. of Communications, Network and System Sciences*, *05*(07), 131–140. https://doi.org/10.4236/ijcns.2014.57015

[30] Fawagreh, K., Gaber, M. M., & Elyan, E. (2014). Random forests: From early developments to recent advancements. *Systems Science & Control Engineering*, *2*(1), 602–609. https://doi.org/10.1080/21642583.2014.956265

[31] Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017). Applying convolutional neural network for network intrusion detection. *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE. Retrieved from

http://dx.doi.org/10.1109/icacci.2017.8126009

[32] Bergstra J, Bengio Y. Random search for hyper-parameter optimization. J Mach Learn Res 2012;13:281–305.

[33] Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & de Freitas, N. (2016). Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proceedings of the IEEE*, *104*(1), 148–175. https://doi.org/10.1109/jproc.2015.2494218

[34] Alsheikh, M. A., Lin, S., Niyato, D., & Tan, H.-P. (2014). Machine learning in wireless sensor networks: Algorithms, strategies, and applications. *IEEE Communications Surveys & Tutorials*, *16*(4), 1996–2018. https://doi.org/10.1109/comst.2014.2320099

[35] Doshi, R., Apthorpe, N., & Feamster, N. (2018, April 11). Machine learning ddos detection for consumer internet of things devices. Retrieved from arXiv.org website: https://arxiv.org/abs/1804.04159

[36] Kiran, B. N., Radheshyam, S. G., Sagar, N., Sanath, A., & Balthar. (2018). SECURITY FOR IoT SYSTEMS USING MACHINE LEARNING. *Department of Information Science and Engineering,The National Institute Of Engineering, Karnataka, INDIA*, *4*(2).

[37] Saied, A., Overill, R. E., & Radzik, T. (2016). Detection of known and unknown DDoS attacks using Artificial Neural Networks. *Neurocomputing*, *172*, 385–393. https://doi.org/10.1016/j.neucom.2015.04.101

[38] Xiao, L., Li, Y., Huang, X., & Du, X. (2017). Cloud-Based malware detection game for mobile devices with offloading. *IEEE Transactions on Mobile Computing*, *16*(10), 2742–2750. https://doi.org/10.1109/tmc.2017.2687918

[39] Andrea, I., Chrysostomou, C., & Hadjichristofi, G. (2015). Internet of Things: Security vulnerabilities and challenges. *2015 IEEE Symposium on Computers and Communication (ISCC)*. IEEE. Retrieved from http://dx.doi.org/10.1109/iscc.2015.7405513

[40] Ahmad, R., & Alsmadi, I. (2021). Machine learning approaches to IoT security: A systematic literature review. *Internet of Things*, *14*, 100365. https://doi.org/10.1016/j.iot.2021.100365

[41] Ioannou, C., & Vassiliou, V. (2019). Classifying security attacks in iot networks using supervised learning. *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE. Retrieved from http://dx.doi.org/10.1109/dcoss.2019.00118

[42] C. Dawson, A Guide to SVM Parameter Tuning September 26, Medium, 2019 https://towardsdatascience.com/a-guide-to-svm-parameter-tuning-8bfe6b8a452c

[43] Chaudhary, P., & Gupta, B. B. (2019). DDoS detection framework in resource constrained internet of things domain. *2019 IEEE 8th Global Conference on Consumer*

Electronics (GCCE). IEEE. Retrieved from http://dx.doi.org/10.1109/gcce46687.2019.9015465

[44] Dwyer, O. P., Marnerides, A. K., Giotsas, V., & Mursch, T. (2019). Profiling iot-based botnet traffic using DNS. *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE. Retrieved from http://dx.doi.org/10.1109/globecom38437.2019.9014300

[45] Wehbi, K., Hong, L., Al-salah, T., & Bhutta, A. A. (2019). A survey on machine learning based detection on ddos attacks for iot systems. *2019 SoutheastCon*. IEEE. Retrieved from http://dx.doi.org/10.1109/southeastcon42311.2019.9020468

[46] Atul, D. J., Kamalraj, R., Ramesh, G., Sakthidasan Sankaran, K., Sharma, S., & Khasim, S. (2021). A machine learning based IoT for providing an intrusion detection system for security. *Microprocessors and Microsystems*, *82*, 103741. https://doi.org/10.1016/j.micpro.2020.103741

[47] Pascual, J. A. (2019, August 24). Inteligencia artificial: Qué es, cómo funciona y para qué se utiliza en la actualidad. *ComputerHoy*. Retrieved from https://computerhoy.com/reportajes/tecnologia/inteligencia-artificial-469917 (accedido 11/9/22)

[48] Niu, W., Zhang, X., Du, X., Zhao, L., Cao, R., & Guizani, M. (2020). A deep learning based static taint analysis approach for IoT software vulnerability location. *Measurement*, *152*, 107139. https://doi.org/10.1016/j.measurement.2019.107139

[49] Cremer, J. L., & Strbac, G. (2021). A machine-learning based probabilistic perspective on dynamic security assessment. *International Journal of Electrical Power & Energy Systems*, *128*, 106571. https://doi.org/10.1016/j.ijepes.2020.106571

[50] Meidan, Y., Sachidananda, V., Peng, H., Sagron, R., Elovici, Y., & Shabtai, A. (2020). A novel approach for detecting vulnerable IoT devices connected behind a home NAT. *Computers & Security*, *97*, 101968. https://doi.org/10.1016/j.cose.2020.101968

[51] Nassef, O., Sun, W., Purmehdi, H., Tatipamula, M., & Mahmoodi, T. (2022). A survey: Distributed Machine Learning for 5G and beyond. *Computer Networks*, *207*, 108820. https://doi.org/10.1016/j.comnet.2022.108820

[52] Ssengonzi, C., Kogeda, O. P., & Olwal, T. O. (2022). A survey of deep reinforcement learning application in 5G and beyond network slicing and virtualization. *Array*, *14*, 100142. https://doi.org/10.1016/j.array.2022.100142

[53] Olga, V., Ruslana, Z., Yuriy, F., & Joanna, N. (2021). Big data analysis methods based on machine learning to ensure information security. *Procedia Computer Science*, *192*, 2633–2640. https://doi.org/10.1016/j.procs.2021.09.033

[54] Miglani, A., & Kumar, N. (2021). Blockchain management and machine learning

adaptation for IoT environment in 5G and beyond networks: A systematic review. *Computer Communications*, *178*, 37–63. https://doi.org/10.1016/j.comcom.2021.07.009

[54] Sedjelmaci, H. (2021). Cooperative attacks detection based on artificial intelligence system for 5G networks. *Computers & Electrical Engineering*, *91*, 107045. https://doi.org/10.1016/j.compeleceng.2021.107045

[55] Sarker, I. H. (2021). CyberLearning: Effectiveness analysis of machine learning security modeling to detect cyber-anomalies and multi-attacks. *Internet of Things*, *14*, 100393. https://doi.org/10.1016/j.iot.2021.100393

[56] Lee, H. (2020). Home IoT resistance: Extended privacy and vulnerability perspective. *Telematics and Informatics*, *49*, 101377. https://doi.org/10.1016/j.tele.2020.101377

[57] Mandal, K., Rajkumar, M., Ezhumalai, P., Jayakumar, D., & Yuvarani, R. (2020). Improved security using machine learning for IoT intrusion detection system. *Materials Today: Proceedings*. https://doi.org/10.1016/j.matpr.2020.10.187

[58] Yi, H. (2021). Improving security of 5G networks with multiplicative masking method for LDPC codes. *Computers and Electrical Engineering*, *95*, 107384. https://doi.org/10.1016/j.compeleceng.2021.107384

[59] Sumathy, S., Revathy, M., & Manikandan, R. (2021). Improving the state of materials in cybersecurity attack detection in 5G wireless systems using machine learning. *Materials Today: Proceedings*. https://doi.org/10.1016/j.matpr.2021.04.171

[60] Hariyanti, E., Djunaidy, A., & Siahaan, D. (2021). Information security vulnerability prediction based on business process model using machine learning approach. *Computers & Security*, *110*, 102422. https://doi.org/10.1016/j.cose.2021.102422

[61] Gaber, T., El-Ghamry, A., & Hassanien, A. E. (2022). Injection attack detection using machine learning for smart IoT applications. *Physical Communication*, *52*, 101685. https://doi.org/10.1016/j.phycom.2022.101685

[62] Roldán, J., Boubeta-Puig, J., Luis Martínez, J., & Ortiz, G. (2020). Integrating complex event processing and machine learning: An intelligent architecture for detecting IoT security attacks. *Expert Systems with Applications*, *149*, 113251. https://doi.org/10.1016/j.eswa.2020.113251

[63] Al-Turjman, F. (2020). Intelligence and security in big 5G-oriented IoNT: An overview. *Future Generation Computer Systems*, *102*, 357–368. https://doi.org/10.1016/j.future.2019.08.009

[64] Kumar, A., Abhishek, K., Ghalib, M. R., Shankar, A., & Cheng, X. (2022). Intrusion detection and prevention system for an IoT environment. *Digital Communications and Networks*, *8*(4), 540–551. https://doi.org/10.1016/j.dcan.2022.05.027

[64] Schiller, E., Aidoo, A., Fuhrer, J., Stahl, J., Ziörjen, M., & Stiller, B. (2022). Landscape of IoT security. *Computer Science Review*, *44*, 100467. https://doi.org/10.1016/j.cosrev.2022.100467

[65] Mangla, C., Rani, S., Faseeh Qureshi, N. M., & Singh, A. (2022). Mitigating 5G security challenges for next-gen industry using quantum computing. *Journal of King Saud University - Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2022.07.009

[66] Madi, T., Alameddine, H. A., Pourzandi, M., & Boukhtouta, A. (2021). NFV security survey in 5G networks: A three-dimensional threat taxonomy. *Computer Networks*, *197*, 108288. https://doi.org/10.1016/j.comnet.2021.108288

[67] Abidi, M. H., Alkhalefah, H., Moiduddin, K., Alazab, M., Mohammed, M. K., Ameen, W., & Gadekallu, T. R. (2021). Optimal 5G network slicing using machine learning and deep learning concepts. *Computer Standards & Interfaces*, *76*, 103518. https://doi.org/10.1016/j.csi.2021.103518

[68] Mahbub, M. (2020). Progressive researches on IoT security: An exhaustive analysis from the perspective of protocols, vulnerabilities, and preemptive architectonics. *Journal of Network and Computer Applications*, *168*, 102761. https://doi.org/10.1016/j.jnca.2020.102761

[69] Rahman, Md. S., Safa, N. T., Sultana, S., Salam, S., Karamehic-Muratovic, A., & Overgaard, H. J. (2022). Role of artificial intelligence-internet of things (AI-IoT) based emerging technologies in the public health response to infectious diseases in Bangladesh. *Parasite Epidemiology and Control*, *18*, e00266. https://doi.org/10.1016/j.parepi.2022.e00266

[71] Catak, F. O., Kuzlu, M., Catak, E., Cali, U., & Unal, D. (2022). Security concerns on machine learning solutions for 6G networks in mmWave beam prediction. *Physical Communication*, *52*, 101626. https://doi.org/10.1016/j.phycom.2022.101626

[72] Mohanta, B. K., Jena, D., Satapathy, U., & Patnaik, S. (2020). Survey on IoT security: Challenges and solution using machine learning, artificial intelligence and blockchain technology. *Internet of Things*, *11*, 100227. https://doi.org/10.1016/j.iot.2020.100227

[73] Snehi, M., & Bhandari, A. (2021). Vulnerability retrospection of security solutions for software-defined Cyber–Physical System against DDoS and IoT-DDoS attacks. *Computer Science Review*, *40*, 100371. https://doi.org/10.1016/j.cosrev.2021.100371

[74] Youtube, D. L. C. C. (2021). Ciclo Webinar sobre Ciberseguridad en 5G - 1. Presentación e introducción a 5G [Video]. Retrieved from https://www.youtube.com/watch?v=P3M2FBZz3pI&t=1556s (accedido 02/9/2022)

[75] Fernandez, Y. (n.d.). Tecnología móvil. Retrieved September 29, 2022, from SlideShare iOS website: https://es.slideshare.net/YenniferFernandez/tecnologia-mvil (accedido 11/9/2022)

[76] Russell, S. J., & Norvig, P. (2004). *Inteligencia artificial: Un enfoque moderno*. PRENTICE HALL.

[77] Ación, L., Alemany, L. A., Ferrante, E., Lützow Holm, E., Martinez, V., Milone, D., … Uchitel, S. (2019). *Inteligencia artificial : una mirada multidisciplinaria* (1st ed., pp. 63–87). Manuel A. Solanet.

[78] Noguera, J., Portillo, N., & Hernandez, L. (2014). Redes Neuronales, Bioinspiración para el Desarrollo de la Ingeniería. *Ingeniare*, (17), 117–131. https://doi.org/10.18041/1909-2458/ingeniare.17.584

[79] Bruck, H. A., Gershon, A. L., Golden, I., Gupta, S. K., Gyger, L. S., Jr, Magrab, E. B., & Spranklin, B. W. (2007). Training mechanical engineering students to utilize biological inspiration during product development. *Bioinspiration & Biomimetics*, *2*(4), S198–S209. https://doi.org/10.1088/1748-3182/2/4/s08

[80] *Biomimicry for optimization, control, and automation*. (2005). London: Springer-Verlag. Retrieved from http://dx.doi.org/10.1007/b138169

[81] Neurons & glial cells. (n.d.). Retrieved September 30, 2022, from SEER Training website: https://training.seer.cancer.gov/brain/tumors/anatomy/neurons.html (Accedido 30-9-2022).

[82] Ogata, K. (2003). *Ingeniería de control moderna*. Pearson Educación.

[83] Redes neuronales. (n.d.). Retrieved September 30, 2022, from lo básico website: http://proyectogio.blogspot.com/2016/10/redes-neuronales-lo-basico.html

[84] Wikimedia, C. de los proyectos. (2022, September 21). Regresión lineal. Retrieved October 2, 2022, from Wikipedia website: https://es.wikipedia.org/wiki/Regresi%C3%B3n_lineal (accedido 30-9-2022)

[85] larispardo. (2022, February 10). ¿Qué es y para qué sirve una regresión lineal en machine learning? *Platzi*. Retrieved from https://platzi.com/blog/que-es-regresion-lineal/?gclid=CjwKCAjw7eSZBhB8EiwA60kCWxiloE6ndmahUPSyuawXUrDRI-_dSbnpXfc485i7vv43dmlwvSLYTxoCZK4QAvD_BwE&gclsrc=aw.ds (accedido 30-9-2022)

[86] Rodríguez, D. (2021, December 17). Regresión de vectores de soporte (SVR, support vector regression). Retrieved October 2, 2022, from Analytics Lane website: https://www.analyticslane.com/2021/12/17/regresion-de-vectores-de-soporte-svr-support-vector-regression/ (accedido 30-9-2022)