Universidad de Buenos Aires
Facultad de Ciencias Económicas
Escuela de Estudios de Posgrado

MAESTRÍA EN CIBERDEFENSA Y CIBERSEGURIDAD

Trabajo final de tesis

Aportes en ciberseguridad a través de la Inteligencia Artificial en la detección de patrones compatibles con ciberataques en redes de Internet de las Cosas

Autor: Ingeniero Santiago Adrián Plohn

Director: Dra. Antonieta Kuz

Fecha: 10 de diciembre 2021

I. RESUMEN

La tendencia global a estar conectados constantemente conlleva un cambio radical en la manera como vivimos, trabajamos y nos desenvolvemos a diario. A los dispositivos que utilizamos usualmente y que se encuentran de alguna manera conectados a Internet se lo conoce como Internet de las Cosas, así mismo, recientemente potenciada con la incorporación de las redes 5G, ofrece una plataforma potencial poderosa desde donde lanzar ciberataques, ya que provee un vector de ataque a gran escala para diferentes amenazas tales como phishing, denegación de servicios distribuida, troyanos, etc. A estas amenazas no se escapan las redes de infraestructuras críticas, donde la incorporación de dispositivos de última generación implícitamente incluye la conectividad a Internet. Las investigaciones relacionadas con la detección de estas amenazas han sido un área de gran desarrollo en los últimos años en el ámbito de la ciberseguridad, proveyendo técnicas para su detección y mitigación. Sabiendo la problemática que hay, entonces el objetivo de este trabajo es tratar de dar visibilidad y estudiar, a través de un enfoque teórico metodológico la importancia de la utilización de técnicas basadas en Inteligencia Artificial para la detección de ciberataques en redes de IoT junto con un análisis de un caso de estudio mediante el uso de un set de pruebas con Kitsune un NIDS una herramienta en línea que puede aprender a detectar ataques en la red local, sin supervisión y de manera eficiente. Finalmente se presentan las conclusiones y trabajos futuros.

Palabras clave: internet de las cosas (IoT), sistema de detección de intrusiones de red (NIDS), malware, inteligencia artificial, redes.

II. ABSTRACT

The global trend to be constantly connected leads to a radical change in the way we live, work and function on a daily basis. The devices that we usually use and that are somehow connected to the Internet are known as the Internet of Things, likewise, recently enhanced with the incorporation of 5G networks, it offers a powerful potential platform from which to launch cyberattacks, since provides a large-scale attack vector for different threats such as phishing, distributed denial of services, Trojans, etc. Critical infrastructure networks do not escape these threats, where the incorporation of next-generation devices implicitly includes Internet connectivity. Research related to the detection of these threats has been an area of great development in recent years in the field of cybersecurity, providing techniques for their detection and mitigation. Knowing the problems that exist, then the objective of this work is to try to give visibility and study, through a theoretical methodological approach, the importance of the use of techniques based on Artificial Intelligence for the detection of cyberattacks in IoT networks with an analysis of a case study using a test set with Kitsune an NIDS, an online tool that can learn to detect attacks on the local network, without supervision and efficiently. Finally, the conclusions and future works are presented.

Keywords: internet of things (IoT), network intrusions detection systems, (NIDS), malware, artificial intelligence, network.

III. DEDICATORIAS

A Inés, mi amada madre, mi guía y soporte incondicional, que Dios la tenga en su gloria.

IV. AGRADECIMIENTOS

A mi directora de Tesis, la Dra. Antonieta Kuz, por su guía y apoyo en el armado de esta tesis, por el tiempo dedicado, los consejos, gran profesionalismo y por haberme motivado en los momentos de flaqueza.

A los profesores y directivos que me acompañaron en la maestría, especialmente al Dr. Roberto Uzal y al Ing. Carlos Amaya por haberme dado la posibilidad de formar parte de esta cohorte.

A todos mis compañeros de la cohorte 2018 de la Maestría en Ciberdefensa y Ciberseguridad de los cuales he aprendido mucho durante los dos años que cursamos juntos.

Y especialmente a mi mujer Andrea y mis hijas Dolores y Guadalupe, sin su paciencia y apoyo nunca hubiera podido atravesar esta hermosa experiencia.

V. ÍNDICE

VI. ÍNDICE DE TABLAS Y FIGURAS

VII. GLOSARIO

AMQP	Advanced Message Queuing Protocol.
ANN	Artificial Neural Network.
APT	Advanced Persistent Tread.
Autoencoder	Tipo de red neuronal artificial que se utiliza para aprender codificaciones eficientes de datos no etiquetados. La codificación se valida y refina al intentar regenerar la entrada a partir de la codificación.
BLE	Bluetooth Low Energy.
Botnet	Red de equipos informáticos infectados con un software malicioso que permiten el control remoto a los mismos sin la aprobación del propietario.
CNN	Convolutional Neural Network.
CoAP	Constrained Application Protocol.
Cryptojacking	Es un software que se esconde en una computadora o dispositivo móvil y utiliza los recursos de la máquina para minar criptomonedas.
DDOS	Distributed Denial of Service.
DL	Deep Learning.
DLL	Dynamic Link Library.
DNN	Deep Neural Network.
DNS	Domain Name System.
DPI	Deep Packet Inspection.
DVR	Digital Video Recoder.
EPC	Electronic Product Code.

Exploit	Es un fragmento de software, de datos o secuencia de comandos y/o acciones, utilizada con el fin de aprovechar una vulnerabilidad de seguridad de un sistema de información para conseguir un comportamiento no deseado del mismo.
GAN	Generative Adversarial Network.
Gateway	Dispositivo utilizado para conectar dos redes diferentes.
GSMA	Global System for Mobile Communications Association.
Honeypot	Mecanismo de seguridad informática configurado para detectar, desviar o, de alguna manera, contrarrestar los intentos de uso no autorizados de los sistemas de información.
HPHC	High Processing High Connectivity.
HUB	Concentrador. Dispositivo utilizado para concentrar el cableado de una red y ampliarla.
IA	Inteligencia Artificial.
IEEE	Institute of Electrical and Electronics Engineers.
IOT	Internet of Things.
IP	Internet Protocol.
ITU-T	International Telecommunications Union – Telecommunications Standarization Sector.
Kernel	Programa de computadora que es el núcleo del sistema operativo, con un control completo sobre todo en el sistema. En la mayoría de los sistemas, es uno de los primeros programas cargados al inicio.
Keylogger	Es un programa malicioso para grabar las pulsaciones de teclas del usuario en la computadora con la finalidad de robar contraseñas y otra información confidencial.
Kitnet	Motor de detección de anomalías de Kitsune.
Kitsune	Sistema de detección de intrusiones de red plug-and-play que puede aprender a detectar ataques en la red local, sin supervisión y de manera eficiente en línea.
Lo-Ra	Long Range.
LoWPAN	Low-Power Wireless Personal Area Networks.

LPHC	Low Processing High Connectivity.
LPLC	Low Processing Low Connectivity.
LPWAN	Low-Power Wide Area Network.
LPWLAN	Low-Power Wireless Local Area Network.
Malware	Código malicioso. Software de computadora que es diseñado para dañar la manera en que una computadora funciona.
MDNS	Multicast Domain Name System.
MiTM	Man-in-the-Middle.
ML	Machine Learning.
MQTT	Message Queuing Telemetry Transport.
NB-IoT	Narrow Band Internet of Things.
NFC	Near Field Communications.
NIDS	Network Intrusion Detection System.
OSI	Open System Interconnection.
OT	Operation Technology.
Phishing	Un intento de engañar a alguien para que brinde información a través de Internet o por correo electrónico que le permita a otra persona sacar dinero de ellos, por ejemplo, sacando dinero de su cuenta bancaria.
PLC	Programmable Logic Controller.
Ransomware	Software que secuestra datos informáticos. Software diseñado por delincuentes para evitar que los usuarios de computadoras tengan acceso a su propio sistema informático o archivos a menos que paguen dinero.
RFID	Radio Frequency Identification.
RL	Reinforcement Learning.
RMSE	Root Mean Square Error.

RNN	Recurrent Neural Network.
SCADA	Supervisory Control and Data Acquisition.
Spam	Uso de servicios de mensajes (Mails, SMSs, etc.) para el envío de mensajes no solicitados aun gran número de destinatarios, con fines comerciales o publicitarios.
Streaming	Distribución digital de contenido multimedia a través de una red de computadoras, de manera que el usuario utiliza el producto a la vez que se descarga.
Stuxnet	Gusano informático malicioso, descubierto por primera vez en 2010, que se cree que ha estado en desarrollo desde al menos 2005. Stuxnet apunta a los sistemas SCADA y se cree que es responsable de causar daños sustanciales al programa nuclear de Irán.
TCP	Transmission Control Protocol.
UTP	Unshielded Twisted Pair.
XMPP	Extensible Messaging and Presence Protocol.
Zero-day	Ataque de día cero o zero-day se refiere a la explotación de una vulnerabilidad aún desconocida por aquellos interesados en mitigarla. Hasta que esta vulnerabilidad es mitigada, los hackers pueden explotarla afectando programas, datos y otros equipos en la red.

1. Introducción

El ciberespacio, también denominado Quinto Dominio, es definido por la Real Academia Española (RAE, 2021) como "*el ámbito virtual creado por medios informáticos*". Este ámbito se ha convertido en un componente clave sobre el que se basan tanto negocios como infraestructuras operativas críticas privadas, estatales o mixtas.

Los servicios basados en la nube, la IoT, los servicios en línea como pagos, servicios financieros, desde el típico soporte remoto de productos hasta la consulta médica en línea conviven con todo tipo de actividades ilícitas, desde tráfico de drogas, venta de armas, extorsión, hasta lavado de dinero y ciberataques a gran escala.

Este incremento en la actividad en Internet conlleva un incremento en el volumen de información que circula por la red. Al mismo tiempo crecen los ataques en cantidad y variedad, apoyados por herramientas y ciberarmas que permiten automatizarlos y ejecutarlos a gran escala (DDoS, campañas de *phishing*, *spam*, etc.).

Cada vez más la seguridad física depende de la seguridad en el ciberespacio, tanto por la información que en este ámbito circula, como los dispositivos interconectados utilizados por servicios que consumimos a diario, entre los que podemos mencionar servicios básicos como la provisión de agua potable o energía, los dispositivos de seguridad electrónica, servicios financieros o de salud. Es debido a esto que la detección temprana de estos ataques, así como su atribución son requisitos clave de cualquier estado nación que quiera demostrar sus capacidades de defensa en este ámbito.

Desde el punto de vista de la ciberdefensa, son necesarias herramientas basadas en técnicas avanzadas que permitan discriminar el tráfico normal del potencialmente dañino en tiempo real y con un alto grado de certeza ya que la dinámica del problema requiere de acciones automatizadas, tanto en la identificación como en la mitigación de un ciberataque. En relación con esto último, en su libro El Quinto Dominio, Clark afirma que "*La automatización y la inteligencia artificial tienen el potencial de borrar*

gran parte de la ventaja del atacante. Sin embargo, al mismo tiempo, los atacantes están viendo cómo pueden usar estas herramientas también." (Clarke & Knake, 2019, p. 6).

1.1. Objetivo

El presente proyecto tiene como finalidad abordar y estudiar la problemática de los ciberataques masivos en redes IoT desde el punto de vista de la detección temprana, a través del análisis de flujos mediante la utilización de técnicas de IA.

1.2. Objetivos Específicos

- Analizar el estado del arte en cuanto a la utilización de técnicas de IA aplicadas a la detección de software malicioso (*malware* a partir de ahora)
- Estudiar los beneficios de la utilización de estas técnicas para la detección de patrones de comportamiento compatibles con ciberataques.
- Estudiar y presentar un caso práctico de aplicación sobre tráfico real a través de la implementación de un modelo analítico para tal efecto.

1.3. Metodología

La tesis es una investigación metodológica mixta[1], donde se presentará una revisión de metodologías y técnicas de IA utilizadas para la resolución de los problemas planteados, y a partir de dicho análisis se realizará una propuesta tecnológica de aplicación concreta.

El tipo de investigación a realizar en una primera etapa será exploratoria y descriptiva, con la aplicación de un enfoque del tipo de cualitativo y un proceso deductivo. En la segunda etapa el enfoque será cuantitativo donde se analizarán los últimos trabajos relacionados con el tema bajo tratamiento y se aplicarán algunas de las técnicas investigadas en un caso concreto.

Para sustentar la solución propuesta se realizará una introducción al marco teórico, donde se explicará la problemática de los ciberataques masivos, la complejidad de su mitigación en redes IoT, también se describirán las principales características de los protocolos utilizados y se introducirá al lector en los beneficios de la utilización de técnicas de IA como mecanismo de detección.

1.4. Problema de Investigación

La problemática de detección de ciberataques ha sido un tema de preocupación desde el uso masivo de las redes de comunicaciones y contempla diferentes métodos; en general la mayoría se basan en técnicas estadísticas o de firmas, aunque también son muy valorados los métodos que utilizan el análisis de bajo nivel utilizando Inspección Profunda de Paquetes (*Deep Packet Inspection* en inglés).

---

[1] "Los métodos mixtos o híbridos representan un conjunto de procesos sistemáticos, empíricos y críticos de la investigación e implican la recolección y el análisis de datos tanto cuantitativos como cualitativos, así como su integración y discusión conjunta, para realizar inferencias producto de toda la información recabada (denominada meta inferencias) y lograr una mayor entendimiento del fenómeno bajo estudio" (Hernández Sampieri & Mendoza Torres, Metodología de la Investigación: Las rutas cuantitativca, cualitativa y mixta, 2018)

En lo referente a las técnicas estadísticas o de firmas, su ventaja está dada por la rapidez de procesamiento, pero tienen la contra de no adaptarse bien a los cambios en las técnicas de ataque.

En cuanto a las técnicas de inspección profunda de paquetes, las mismas son mucho más detalladas al momento del análisis, pero consumen muchos más recursos y dependen de que el tráfico no esté cifrado para poder aplicarse, situación que tiende a desaparecer.

Como reacción a la constante evolución de las amenazas se ha incrementado la utilización de técnicas basadas en comportamiento para mejorar el rendimiento que se obtenía con las técnicas tradicionales basadas en estadísticas, firmas o en reglas.

El caso de estudio que presentaremos busca trazar un análisis e inspección del comportamiento del tráfico de paquetes a través de la detección de patrones mediante el uso de técnicas de IA. La IA es un activo clave en el kit de herramientas de seguridad de IoT para aprender de la inteligencia de amenazas recolectada en el campo y proveer mejores acciones para lidiar con estas al tiempo que las mismas se adaptan y evolucionan. De esta forma se podrá detectar y responder a las situaciones observadas y brindar asistencia para manejar la complejidad y la escala de dicho desafío.

2. Marco Teórico

Los ataques por *malware*[2] son el tipo de ataque más exitoso, efectuado tanto a compañías privadas como a entes gubernamentales e infraestructuras críticas. Entre los ataques más comunes podemos mencionar el *phishing*[3], el *ransomware*[4], el comprometer mails corporativos, los ataques de Protocolo de Escritorio Remoto (RDP en inglés, por *Remote Desktop Protocol*) y el *crytojacking*[5].

Estos ataques, en la fase previa a concretarse, tienen un comportamiento que, en caso de ser monitoreado como parte de la política de prevención, permitiría en muchas ocasiones detectarlos de manera temprana. Estos comportamientos incluyen acceso anómalo o abuso de comandos como en el caso de Powershell, que es un lenguaje de scripting muy utilizado por las empresas para ejecutar tareas programadas o de mantenimiento.

Los ciberataques no solo han crecido en volumen, sino también en inteligencia, desarrollando capacidades para movimiento lateral cada vez más avanzadas. Esto significa que una vez que ingresan a una red tienen más capacidad para expandirse y evitar su detección.

---

[2] Malware: Código malicioso. Software de computadora que es diseñado para dañar la manera en que una computadora funciona. https://dictionary.cambridge.org/dictionary/english-spanish/malware

[3] Phishing: Un intento de engañar a alguien para que brinde información a través de Internet o por correo electrónico que le permita a otra persona sacar dinero de ellos, por ejemplo, sacando dinero de su cuenta bancaria. https://dictionary.cambridge.org/dictionary/english-spanish/phishing

[4] Ransomware: Software que secuestra datos informáticos. Software diseñado por delincuentes para evitar que los usuarios de computadoras tengan acceso a su propio sistema informático o archivos a menos que paguen dinero. https://dictionary.cambridge.org/dictionary/english-spanish/ransomware

[5] Cryptojacking: Es un software que se esconde en una computadora o dispositivo móvil y utiliza los recursos de la máquina para minar criptomonedas.

Los puntos de mayor interés para los ciberdelincuentes son los relacionados con nuevas tecnologías; servicios en la nube, redes IoT, 5G, etc. Estas áreas al ser relativamente nuevas suelen traer vulnerabilidades inherentes que son explotadas por los atacantes.

Entre los ciberataques más complejos se encuentra el realizado a través de *botnets*. Una *botnet* es una red de equipos informáticos infectados con un software malicioso que permite el control remoto a los mismos sin la aprobación del propietario.

Estas redes son utilizadas con fines delictivos de diferentes maneras. Se utilizan para disparar desde la misma ataques de denegación de servicio (DoS), para ejecutar campañas de *spam*, para inutilizar las infraestructuras infectadas, realizar minería de criptomonedas, etc.

Las capacidades tecnológicas de estos *bots* (nombre derivado de la palabra robot) son muy amplias y sofisticadas. El software usualmente tiene capacidades para expandirse dentro de la red aprovechando vulnerabilidades conocidas; también poseen capacidades para no ser detectados como el polimorfismo o el auto cifrado. A través de la conexión con su Centro de Comando y Control pueden actualizarse y adquirir nuevas capacidades. Estas *botnets* están diseñadas para sacar provecho de estos dispositivos desprotegidos y utilizarlos para cometer delitos.

2.1. Conceptos acerca de Malware

2.1.1. Definición

La palabra *malware* proviene del acrónimo en inglés de ***Mal****icious Soft****ware***, y a continuación, tomamos algunas de sus definiciones:

- Diccionario en línea Merriam-Webster: *"Software designed to interfere with a computer's normal functioning"* (Meriam-Webster, 2021).

- Diccionario en línea Collins: *"Malware is a type of computer program that is designed to damage or disrupt a computer"* (Collins, 2021).
- Diccionario en línea de Cambridge: *"Computer software that is designed to damage the way a computer works"* (Cambridge, 2021).

Como surge de lo expuesto existen diversas definiciones, a los fines de este trabajo definiremos al *malware* como el software especialmente diseñado para modificar el uso normal o infringir un daño en los dispositivos informáticos donde se aloja y a los conectados al mismo. Como vemos la definición es muy amplia tal como sus tipos, objetivos y capacidades.

2.1.2. Clasificación

Existen diferentes tipos de *malware* y su clasificación puede variar dependiendo del autor. A continuación, vamos a presentar una clasificación basada en su funcionamiento en la Tabla 1 de Clasificación de *malware*:

Clasificación	Descripción
Virus	Los virus han estado presentes desde el comienzo de la informática. Si bien John Von Neuman realizó el primer trabajo académico basándose en la teoría de la auto replicación de programas de computadoras en 1949, no fue hasta la década del 70 que aparecieron los primeros casos que pueden ser clasificados como virus. Los virus tienen dos características propias: • Auto replicación: El virus busca distribuir copias de sí mismo utilizando todos los medios posibles.

	• Permanecer encubierto: El virus posee mecanismos para permanecer encubierto, lo cual dificulta su identificación y permite su replicación sin ser detectado. Usualmente requieren de intervención humana para su propagación y básicamente se propagan infectando archivos binarios, archivos de datos o sectores de inicio de discos y dispositivos de almacenamiento. Podemos mencionar tres tipos de virus desde el punto de vista de la infección: • Virus de sector de arranque o de sistema: Infectan parte del sistema operativo y permanecen en memoria, infectando nuevos dispositivos conectados. Cuando el equipo se reinicia tienen mecanismos para volver a cargarse. Este tipo de virus son cada vez menos frecuentes gracias a las políticas de seguridad implementadas por los sistemas operativos. • Virus de ejecutables: Estos virus se insertan en archivos que pueden ser ejecutables por el sistema operativo, originalmente .exe o .com, pero actualmente cualquier archivo que el sistema operativo conozca como ejecutar, por ejemplo, .bat, .js, .vb, .scr entre otros, pueden ser blanco de este tipo de virus. • Virus de macros: Estos son los virus que se ejecutan dentro de una aplicación determinada, como parte de las funcionalidades provistas por la aplicación para enriquecer y automatizar su funcionamiento. Muchos antivirus tienen módulos específicos para identificarlos, y este tipo de virus ha disminuido su accionar en los últimos años.
Gusanos	Son piezas de software independientes, que se auto replican sin un objetivo en particular, dañando los archivos infectados y replicándose hasta

	dejar sin espacio los dispositivos de almacenamiento. Usualmente se distribuyen a través de mail o mensajería, utilizando las redes y fallos de seguridad conocidos para explotarlos. Mientras los virus se cargan en archivos externos, los gusanos tienen sus propios métodos de distribución. Muchos de estos gusanos no tienen intención de daño, solo de distribución, causando daños colaterales en lo referente a carga de tráfico en la red.
Caballos de Troya	Este tipo de *malware* se hace pasar por otro software de utilidad intentando que el usuario lo instale. Usualmente el caballo de troya es un *malware* que se agrega a un software ya existente, cargándose al ejecutarse el mismo y permitiendo que el software host trabaje normalmente mientras el *malware* hace su trabajo una vez cargado en memoria. La carga útil (*payload*) puede ser cualquier cosa, generalmente es algún tipo de puerta trasera o *backdoor* que le permitirá al atacante tener acceso al sistema. Los caballos de troya les proveen a los atacantes información privada del usuario como dirección IP, credenciales, información bancaria, etc. Suelen ser utilizados para instalar *keyloggers*[6] y muchos de los *ransomware* utilizan caballos de troya para ingresar. Son considerados los más peligrosos, especialmente porque suelen ser utilizados para robar información financiera.
Rootkits	Es una colección de herramientas de software especialmente diseñada para permitirle al *malware* obtener información del sistema

[6] Keylogger: Es un programa malicioso para grabar las pulsaciones de teclas del usuario en la computadora con la finalidad de robar contraseñas y otra información confidencial.

	objetivo. Trabaja de manera oculta evitando ser detectado por el usuario y le permite al atacante introducir todo tipo de *malware* en el sistema, trabajando como una puerta trasera. Detectarlos y removerlos es muy difícil ya que muchas veces trabajan al nivel del *kernel* [7] del sistema operativo.
Ransomware	Este tipo de *malware* es el más devastador, especialmente por sus consecuencias económicas. Es actualmente el de mayor crecimiento e innovación y su objetivo primordial es bloquear el acceso a los datos por parte del usuario hasta que el mismo pague una recompensa para su recuperación. Trabaja infectando el sistema desde adentro haciendo inaccesible los datos. Los más simples bloquean el acceso, mientras que los más sofisticados utilizan técnicas avanzadas para el cifrado de los datos para hacer imposible el acceso a los mismos.
Keyloggers	Este *malware* registra toda la información que se tipea en el equipo a través del teclado. Los *keyloggers* almacenan los datos capturados y los envían al atacante, quien extraerá información sensible de los mismos como credenciales, números de tarjetas de crédito, etc.
Híbridos	Actualmente la mayoría de los *malware* tienen características híbridas, combinando troyanos con gusanos y otros tipos. Usualmente se presentan al usuario final como un troyano, pero una vez infectado se propagan lateralmente como gusanos.

[7] Kernel: Programa de computadora que es el núcleo del sistema operativo de una computadora, con un control completo sobre todo en el sistema. En la mayoría de los sistemas, es uno de los primeros programas cargados al inicio.

Grayware	Este término define las aplicaciones que, si bien no caen dentro de la característica de software malicioso, generan un deterioro en el rendimiento de los equipos o traen riesgos de seguridad. Esta categoría contempla desde los que simplemente son programas molestos, hasta los que monitorean el uso de nuestro equipo e informan periódicamente sobre nuestro comportamiento, aunque sea de manera anónima. Se incluyen en este grupo: • Adware: Software que está diseñado para enviarnos avisos comerciales. • Spyware: Software que monitorea nuestro comportamiento. En general nuestra actividad en Internet, y está asociado con un *Adware* que nos enviará avisos basándose en la información provista por el *spyware*.

Tabla 1. Clasificación de malware

Fuente: Autoría propia

2.1.3. Comportamiento

Peiter Zatko, en la reunión de Black Hat de Las Vegas, Nevada en 2011 planteó la naturaleza asimétrica del software moderno y realizó un análisis de más de 9.000 malwares llegando a la conclusión de que, en promedio, estas piezas de software tenían unas 125 líneas de código. Haciendo un análisis en

base a este parámetro identificó que, por ejemplo, una ciber arma como Stuxnet[8] poseía unas 15.000 líneas de código, ósea una tasa de 120:1 respecto de una pieza de *malware* estándar.

En su análisis Peiter muestra que el esfuerzo puesto en la defensa, tanto en herramientas de detección como en mecanismos de protección embebidos en el software objetivo de estas amenazas, es mucho más grande por cada línea de código malicioso; incluso si tomamos como base una suite de *malware* (conjunto de herramientas) en vez de un *malware* sencillo. En la Tabla 2 presentamos una comparación que muestra la tasa de relación de líneas de código relativas entre diferentes elementos de software y un malware tradicional:

Comparación con un malware tradicional	Relación de líneas de código
Stuxnet	120:1
Editor de texto sencillo	500:1
Suite de malware	2.000:1
Herramienta defensiva	100.000:1
Sistema Operativo objetivo	1.000.000:1

Tabla 2. Comparación de líneas de código respecto de un malware tradicional

Fuente: (Sikorski & Honig, 2012)

[8] Stuxnet: Gusano informático malicioso, descubierto por primera vez en 2010, que se cree que ha estado en desarrollo desde al menos 2005. Stuxnet apunta a los sistemas SCADA y se cree que es responsable de causar daños sustanciales al programa nuclear de Irán.

Como menciona Michael Sikorski en su libro *Practical Malware Analysis*, *"No es posible cubrir todos los tipos de malware ya que nuevos malwares están siendo creados constantemente con capacidades casi ilimitadas, pero podemos brindar un buen entendimiento acerca de un orden de cosas a las que prestarle atención"* (Sikorski & Honig, 2012, pág. 231).

2.1.4. Tipologías Enunciadas por Sikorski (Sikorski & Honig, 2012).

Se detallan:

- Downloaders**:** Son *malware* que simplemente se encargan de descargar otra pieza de *malware* desde Internet y ejecutarla en el sistema localmente. Normalmente están empaquetados con un *exploit*[9]. Usualmente utilizan la API de Windows *URLDownloadToFile* seguido de una llamada a *WinExec* para descargar y ejecutar el nuevo *malware* **(Sikorski & Honig, 2012, pág. 232).**

- Launchers**:** También conocidos como *loaders*. Son cualquier pieza de software que instala *malware* para que sea ejecutado de manera subrepticia, inmediatamente o en un futuro. Los *launchers* usualmente contienen el *malware* para el que fueron diseñados a cargar (Sikorski & Honig, 2012, pág. 232).

[9] Exploit: Es un fragmento de software, fragmento de datos o secuencia de comandos y/o acciones, utilizada con el fin de aprovechar una vulnerabilidad de seguridad de un sistema de información para conseguir un comportamiento no deseado del mismo.

- Backdoors: Una puerta trasera o *backdoor* es un tipo de *malware* que le provee al atacante acceso remoto a la máquina de la víctima. Este tipo de *malware* es el más comúnmente encontrado. Existen de muchos tamaños y formas con una amplia variedad de capacidades implementadas. Usualmente, cuando se cuenta con un *backdoor* no es necesario descargar *malware* adicional. Este tipo de *malware* se comunica a través de numerosas formas; un método muy común es a través del puerto 80 utilizando el protocolo HTTP. Ya que este protocolo es el más utilizado para el tráfico saliente de red ofreciéndole al *malware* la posibilidad de mezclar su tráfico con el resto, ayudándolo a pasar desapercibido. Los *backdoors* poseen una amplia serie de funcionalidades comunes, como ser la posibilidad de manipular claves del registro de Windows, capturar pantallas, crear directorios y archivos, buscar archivos dentro del sistema, etc. Es posible determinar las funcionalidades que posee un *malware* de este tipo identificando las funciones de Windows que utiliza e importa. Una de las funcionalidades típicas existentes en los *backdoors* es el *reverse shell*.
- El *reverse Shell:* es una funcionalidad de comunicaciones muy útil y consiste en una conexión originada desde la máquina infectada que provee al atacante acceso a la shell del sistema infectado. Una vez dentro de la *reverse shell,* el atacante puede ejecutar comandos como si estuviera localmente dentro del sistema. Uno de los softwares conocidos para establecer conexiones del tipo *reverse shell* es Netcat. A través de este software, ejecutándolo en dos equipos, es posible establecer una conexión entre ambos. Un ejemplo de su uso es el siguiente:
    - *En la máquina remota se ejecuta:* nc -l -p 80. La opción -l le indica a Netcat que se ejecute escuchando en el puerto indicado por el parámetro -p, que en este caso es el puerto 80.

- *Desde la máquina infectada se ejecuta***:** nc *ip_remota* 80 -e cmd.exe. Esto le indica al Netcat que se conecte a la IP *ip_remota* a través del puerto 80 y luego ejecute el comando cmd.exe localmente (*shell* de comando de Windows), redireccionando su salida a la conexión establecida. De esta manera, en la máquina remota aparecerá el acceso a la *shell* de comandos de Windows.

- RAT (sigla en inglés para Remote Administration Tool**)**: Es utilizado para gestionar una computadora o grupo de computadoras. Son *malware* comúnmente utilizados para lograr objetivos específicos, como robar información o moverse lateralmente dentro de una red (Sikorski & Honig, 2012, pág. 233). El servidor se ejecuta en la máquina infectada, implantado a través de un *malware* y el cliente se ejecuta remotamente en una máquina que hace de unidad de comando y control por el atacante. Los servidores se reportan al cliente periódicamente esperando ser contactados, usualmente a través de puertos conocidos como el 80 o el 443.

- Botnets: Son una cantidad de hosts comprometidos, conocidos como zombis[10], controlados por una única entidad, usualmente conocida como *botnet controller*. El objetivo de una *botnet* es comprometer la mayor cantidad de hosts posible con el objetivo de crear una gran red de zombis que la *botnet* utilizará para distribuir *malware* adicional, realizar *spam* o ejecutar ataques de denegación de servicio distribuida, o DDoS (sigla en inglés para

[10] Zombi: Es una computadora conectada a Internet que ha sido comprometida por un programa de piratas informáticos, virus informáticos o troyanos y puede utilizarse para realizar tareas maliciosas bajo dirección remota.

Distributed Denial of Services). Una *botnet* puede dejar un sitio inaccesible haciendo que todos sus *zombis* lo ataquen simultáneamente.

2.1.4.1. Diferencias entre botnets y RATs

- *Las* botnets *suelen infectar una gran cantidad de hosts, ya que su poderío está en el volumen y capacidad de ataque. Los RATs infectan una cantidad mucho menor de equipos, ya que tiene objetivos específicos y el volumen de infecciones está dado por la superficie de ataque necesaria para lograrlo.*
- *Todas las* botnets *son controladas a la vez, mientras que los RATs se controlan por víctima del ataque. El atacante, a través del RAT interactúa a un mayor nivel de detalle.*
- *Los RATs son utilizados para ataques puntuales, las* botnets *para ataques masivos.*

- Credential stealers: Esta categoría tiene por objetivo el robo de credenciales y está compuesto por tres grupos específicos de funcionalidades:
  - *Programas que esperan a que el usuario realice un* login *para robar sus credenciales.*
  - *Programas que se dedican a descargar información almacenada, ya sean* passwords *o* hashes*, para explotarlas fuera de línea.*
  - *Programas que graban las pulsaciones del teclado.*

Una vez que el *malware* gana acceso al sistema, es común que busque permanecer allí por mucho tiempo. Este comportamiento es conocido como persistencia (Sikorski & Honig, 2012, pág. 241). Si dicho mecanismo es lo suficientemente único, puede ser utilizado como identificador de una pieza de *malware* en particular (*fingerprint* o huella digital característica de una pieza de software).

2.1.5. Métodos de Persistencia

Entre los más comunes podemos mencionar:

- El Registro de Windows: Existen muchas ubicaciones en el Registro de Windows donde se suelen instalar los *malware*. Entre ellas, una de las más utilizadas es *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run*.

  Existen muchas entradas en el Registro de Windows que se pueden utilizar para efectivizar la persistencia y existen herramientas que se encargan de monitorear las mismas como las de Sysinternals.
- Binarios troyanizados**:** Otra manera de obtener persistencia es a través de la troyanización de archivos binarios. Con esta técnica el *malware* modifica un archivo binario para forzar a que se ejecute el *malware* la próxima vez que se corra o cargue el archivo infectado. Los desarrolladores de *malware* suelen troyanizar archivos de uso frecuente en el sistema. Las DLLs son un objetivo muy usual para esto.

- Hacking de orden de carga de DLLs**:** Es una técnica sencilla que les permite a los creadores de *malware* crear DLLs persistentes sin la necesidad de tener una entrada en el Registro de Windows o troyanizar un archivo binario. Esta técnica no siempre requiere un *loader* malicioso dedicado ya que aprovecha la manera en cómo las DLLs son cargadas por Windows.

Para lograr su objetivo de persistencia, usualmente el *malware* requiere contar con los privilegios necesarios. Esto suele obtenerlo a través de una técnica denominada "*escalamiento de privilegios*". La mayoría de los usuarios suele tener privilegios de administrador, lo cual es una buena noticia para los *malware*. Ya que pasan estos privilegios al *malware* que están ejecutando. Las recomendaciones de la

comunidad de seguridad indican no ejecutar tareas como administrador local, para evitar ejecutar malware y otorgarles el control a los mismos. La mayoría de los ataques de escalamiento de privilegios se dan a través de *exploits* conocidos o ataques de *zero-day*[11].

- Rootkit**:** Por último, los *malware* usualmente invierten muchos recursos para borrar sus huellas, ocultando sus procesos en ejecución y sus métodos de persistencia de los usuarios. La herramienta más comúnmente utilizada para esconder la actividad maliciosa se la conoce como *rootkit*. Los *rootkits* pueden venir de diferentes formas, pero la mayoría trabajan modificando el funcionamiento interno del sistema operativo. Estas modificaciones provocan que archivos, procesos, conexiones de red y otros recursos utilizados por el *malware* sean invisibles para otros programas, lo que dificulta su detección para productos antivirus, administradores y analistas de seguridad (Sikorski & Honig, 2012, pág. 247).

2.1.6. Técnicas de Detección Más Utilizadas

Existen muchos mecanismos de detección de *malware*, en general, las técnicas utilizadas se clasifican en tres clases. La primera clase se refiere a las "Técnicas basadas en firmas", que son técnicas que funcionan comparando los hashes de la carga útil o *payload* incluido en los paquetes, contra una base de datos de *malware* conocidos. La desventaja de esta técnica es la velocidad de evolución de los *malware*, muchos incluso tienen módulos que les permiten mutar su código sobre la marcha para evitar ser detectados con estas técnicas. La segunda clase se refiere a las "Técnicas basadas en heurística", las cuales buscan

[11] Ataque de día cero o *zero-day* se refiere a la explotación de una vulnerabilidad aún desconocida por aquellos interesados en mitigarla. Hasta que esta vulnerabilidad es mitigada, los hackers pueden explotarla afectando programas, datos y otros equipos en la red.

ciertos patrones relacionados con comportamiento sospechoso, como el uso o combinación de determinadas llamadas a funciones del sistema operativo típicas de un componente *backdoor*, o funciones de escalamiento de privilegios o acceso determinadas claves del Registro de Windows. Y la tercera clase tiene que ver con las "Técnicas basadas en patrones", las cuales realizan un análisis del tráfico creando patrones de comportamiento "*normales*" y evidenciando cuando el comportamiento actual se desvía del mismo. Dentro de estas técnicas están las basadas en estadísticas y las basadas en algoritmos de IA.

2.2. Internet de las Cosas (Internet of Things)

Las tendencias y proyecciones sobre el desarrollo de la IoT tendrán como consecuencia un incremento sustancial en la interacción con Internet. Esta interacción será dada de manera pasiva con objetos conectados más que de manera activa con el contenido, lo que redundará en un mundo hiperconectado.

La GSMA (*Global System for Mobile Communications Association*), un organismo internacional que representa los intereses de los operadores de redes móviles en todo el mundo ha desarrollado una serie de documentos específicos sobre IoT orientados a delinear recomendaciones de seguridad; y al respecto aclara:

"El auge del Internet de las cosas (IoT, según sus siglas en inglés) está creando nuevos proveedores de servicios que buscan desarrollar productos y servicios nuevos, innovadores y conectados. Los analistas prevén que cientos de miles de nuevos servicios alrededor del IoT, conectarán miles de millones de nuevos dispositivos IoT entre sí en la próxima década.

Este rápido crecimiento del Internet de las cosas representa una gran oportunidad para que todos los participantes en este nuevo ecosistema amplíen sus ofertas de servicios y aumenten su base de clientes.

Los analistas han indicado que los problemas de seguridad son una barrera significativa para el despliegue de muchos servicios nuevos de IoT y, al mismo tiempo, la provisión de conectividad en áreas geográficas cada vez más amplias para una variedad cada vez mayor de servicios de IoT, aumentará la exposición de todo el ecosistema al fraude y ataques malintencionados. Cada vez es más evidente que los atacantes ('hackers') comienzan a mostrar un interés cada vez mayor en esta industria." (Childs, Smith, & Bailey, IoT Security Guidelines Overview Document Version 2.2, 2019, pág. 5)

2.2.1. Su origen y significado

El término IoT, es relativamente reciente. Pero la idea real de los dispositivos conectados ya existía desde los años 70. En aquel entonces, la idea a menudo se llamaba internet embebido o *embedded internet o* computación generalizada o *pervasive computing*. Pero el término real "*Internet de las cosas*" fue acuñado por Kevin Ashton en 1999 durante su trabajo en Procter & Gamble. Ashton, que estaba trabajando en la optimización de la cadena de suministro, quería atraer la atención de la alta gerencia hacia una nueva tecnología interesante denominada RFID. Debido a que Internet era la tendencia más popular en 1999 y porque de alguna manera tenía sentido, llamó a su presentación "*Internet de las cosas*". A pesar de que Kevin captó el interés de algunos ejecutivos de P&G, el término IoT no recibió una atención generalizada durante los siguientes 10 años. (Knud, 2014, pág. 2)

Cuando hablamos de IoT nos encontramos con diversas definiciones, algunas de las cuales citaremos a continuación.

La *Internet Society*, si bien presenta su propia definición, aclara que no existe una única definición para este término, "*Por lo general, el término Internet de las Cosas se refiere a escenarios en los que la conectividad de red y la capacidad de cómputo se extienden a objetos, sensores y artículos de uso diario que habitualmente no se consideran computadoras, permitiendo que estos dispositivos generen, intercambien y consuman datos con una mínima intervención humana. Sin embargo, no existe ninguna definición única y universal*". (Rose, Eldridge, & Chapin, 2015, pág. 6).

La *IEEE* (por su sigla en inglés para *Institute of Electrical and Electronics Engineers*), la organización global que agrupa a ingenieros eléctricos, electrónicos y profesionales de sistemas de información más importante tiene su propia definición, e introduce en la misma el concepto de "*cosa*", *"Internet de las Cosas se refiere a una red auto configurable, adaptativa y compleja que interconecta 'cosas' a Internet mediante el uso de protocolos de comunicación estándar. Las 'cosas' interconectadas tienen representación física o virtual en el mundo digital, capacidad de detección/accionamiento, funciones de programación y son identificables de forma única. La representación de la 'cosa' incluye la identidad, el estado, la ubicación o cualquier otra información comercial, social o privada. Las 'cosas' ofrecen servicios, con o sin intervención humana, mediante la utilización de la identificación única, la captura de datos y la comunicación y capacidad de accionamiento. El servicio se explota mediante el uso de interfaces y está disponible en cualquier lugar, en cualquier momento, y para cualquier cosa, tomando la seguridad en consideración."* (Minerva, Biru, & Rotondi, 2015, pág. 74).

En cambio la ITU-T (*International Telecommunications Union – Telecommunications Standarization Sector*), una organización global que agrupa empresas de telecomunicaciones y publica estándares y recomendaciones referentes a este ámbito tiene una definición más orientada a una visión de infraestructura de telecomunicaciones, *"Una infraestructura mundial para la sociedad de la información, que permita servicios avanzados mediante la interconexión (física y virtual) de 'cosas' basadas en tecnologías de comunicación y de interoperabilidad de información existentes y en desarrollo. NOTA 1 – Mediante la explotación de las capacidades de identificación, captura de datos, procesamiento y comunicación, el IoT hace pleno uso de las 'cosas' para ofrecer servicios a todo tipo de aplicaciones, a la vez que garantiza el cumplimiento de los requisitos de seguridad y privacidad. NOTA 2 – Desde una perspectiva más amplia, la IoT puede ser percibida como una visión con implicancias tecnológicas y sociales."*. (ITU-T, Recommendation ITU-T Y.2060 - SERIES Y: GLOBAL INFORMATION - INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS - Overview of the Internet of things, 2012, pág. 1)

Por último, citaremos la definición que brinda la GSMA, organización global que agrupa a los operadores de telecomunicaciones móviles y publica estándares y recomendaciones para dicho sector, *"El Internet de las cosas (IoT) describe la coordinación entre múltiples máquinas, dispositivos y aparatos conectados a Internet a través de múltiples redes. Estos dispositivos incluyen objetos cotidianos tales como tabletas y electrónica de consumo y otros dispositivos o máquinas tales como vehículos, monitores y sensores equipados con capacidades de comunicación que les permitan enviar y recibir datos."* (Childs, Smith, & Bailey, IoT Security Guidelines Overview Document Version 2.2, 2019, pág. 8)

Como hemos visto, IoT es el término más popular para describir este mundo interconectado, pero existen términos relacionados que muchas veces pueden confundirse ya que no significan exactamente lo mismo, algunos de los más utilizados se describen a continuación en la Tabla 3:

Término	Definición
M2M (Machine to Machine)	El término *Machine to Machine* ha estado en uso desde hace mucho tiempo, más de una década y media, y es muy bien conocido en el sector de Telecomunicaciones. Las comunicaciones M2M fueron inicialmente pensadas como comunicaciones uno a uno entre máquinas. La explosión de la conectividad móvil facilitó la transmisión de datos hacia un rango mucho más amplio de dispositivos.
Industrial Internet (of Things)	El término Internet Industrial fue propuesto por General Electric y va más allá de M2M ya que no solo se refiere a conexiones entre máquinas sino también incluye interfaces con el hombre.
Web of Things	La Web de las cosas tiene un alcance mucho más limitado que los otros conceptos, ya que solo se centra en la arquitectura de software que integra los objetos de la Internet de las Cosas a la Web.

Internet of Everything (IoE)	IoE tiene como objetivo incluir todo tipo de conexiones que uno pueda imaginar: • M2M: Machine to Machine. • M2P: Machine to People. • P2P: People to People. Además, incluye en su análisis: • Personas • Procesos • Datos • Sitios • Cosas El concepto tiene así el mayor alcance y fue acuñado por Cisco.
Industria 4.0	El término Industria 4.0 que fue fuertemente impulsado por el gobierno alemán es tan limitado como Industrial Internet, ya que solo se enfoca en entornos de fabricación. Sin embargo, tiene el mayor alcance de todos los conceptos. Industria 4.0 describe un conjunto de conceptos para impulsar la próxima revolución industrial. Eso incluye todo tipo de conceptos de conectividad en el contexto industrial. Sin embargo, va más allá e incluye cambios reales en el mundo físico que nos rodea, como las tecnologías de impresión 3D o la introducción de hardware de realidad aumentada.

Tabla 3. Términos relacionados con IoT

Fuente: Autoría propia

Cuando hablamos de "*Cosas*" (*Things*) nos referimos a cualquier objeto capaz de conectarse a Internet, y que poseen las siguientes características:

- Pueden ser físicos o virtuales.
- Tienen capacidades de sensar o actuar.
- Poseen facilidades de programación.
- Son identificables de manera unívoca en la red.

Las características fundamentales de la IoT se enumeran en la siguiente Figura 1:

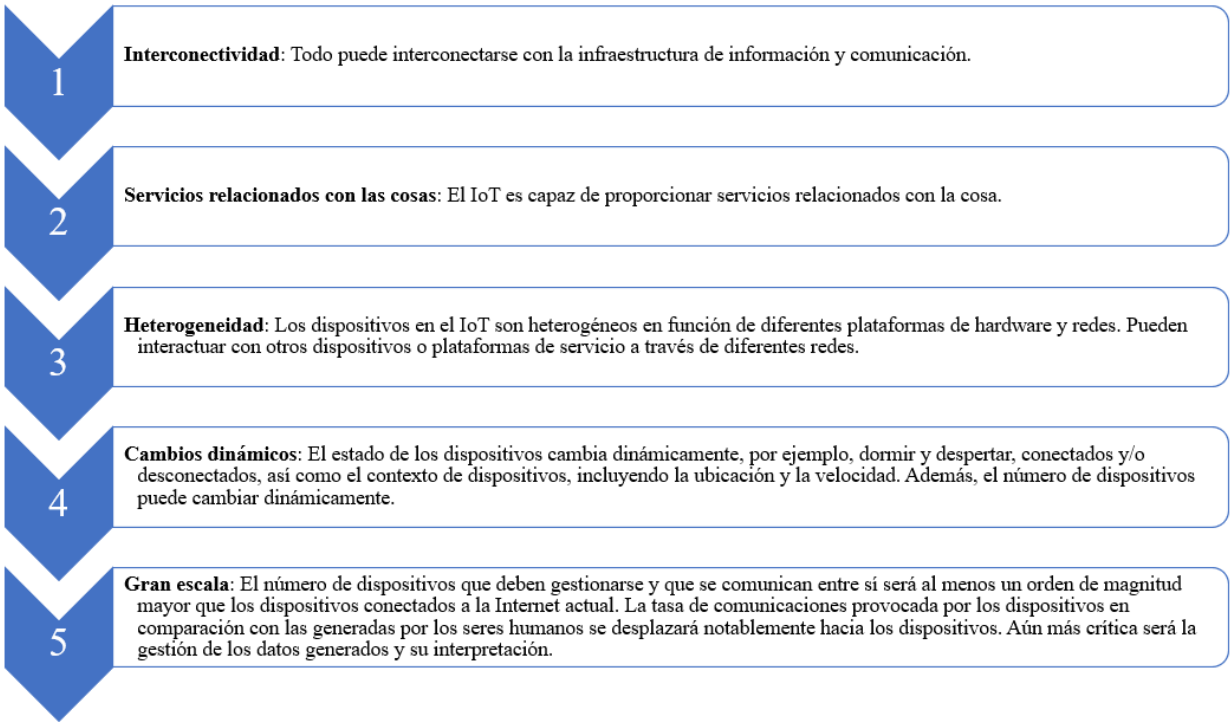

Figura 1. Características fundamentales de las redes IoT

Fuente: Autoría propia

2.2.2. Aplicaciones

Algunas de las aplicaciones más conocidas de la IoT son:

En la Figura 2 se puede ver un resumen de algunas de las aplicaciones más usuales.

Figura 2. Áreas de aplicación de IoT

Fuente: Subex IoT Security (Subex, 2021) -

2.2.3. Arquitecturas de Referencia

Existen distintos organismos y organizaciones que han intentado definir no solo el significado de IoT, como hemos visto en el capítulo anterior, sino también otros aspectos tales como los diferentes componentes dentro del ecosistema de IoT, sus interacciones, protocolos recomendados, etc.

En esta sección mencionaremos la arquitectura de referencia para un ecosistema IoT desde el punto de vista de algunos de los organismos internacionales más importantes. Hemos seleccionado tres ejemplos, uno orientado al aspecto físico de los componentes (ITU-T), uno más amplio orientado a una visión holística de un ecosistema IoT (ISO/IEC) y otro más pragmático y orientado a redes de soporte de comunicaciones celulares (GSMA).

2.2.3.1. ITU-T: Recomendación Y.2060/Y.4000 (06/12) – Visión General de la Internet de las Cosas.

En la Recomendación UIT-T Y.2060 se presenta en términos generales Internet de las cosas (IoT). Se aclara el concepto y el alcance de IoT, se identifican las características fundamentales y los requisitos de alto nivel de IoT y se describe el modelo de referencia IoT. El ecosistema y los modelos empresariales también se incluyen en un Apéndice informativo.

2.2.3.1.1. Tipos de dispositivos IoT

Según la ITU-T, (ITU-T, ITU-T Y.2060 - Overview of Internet of Things, 2012) los dispositivos intervinientes en una Red de IoT se clasifican tal como se muestra en la Figura 3.

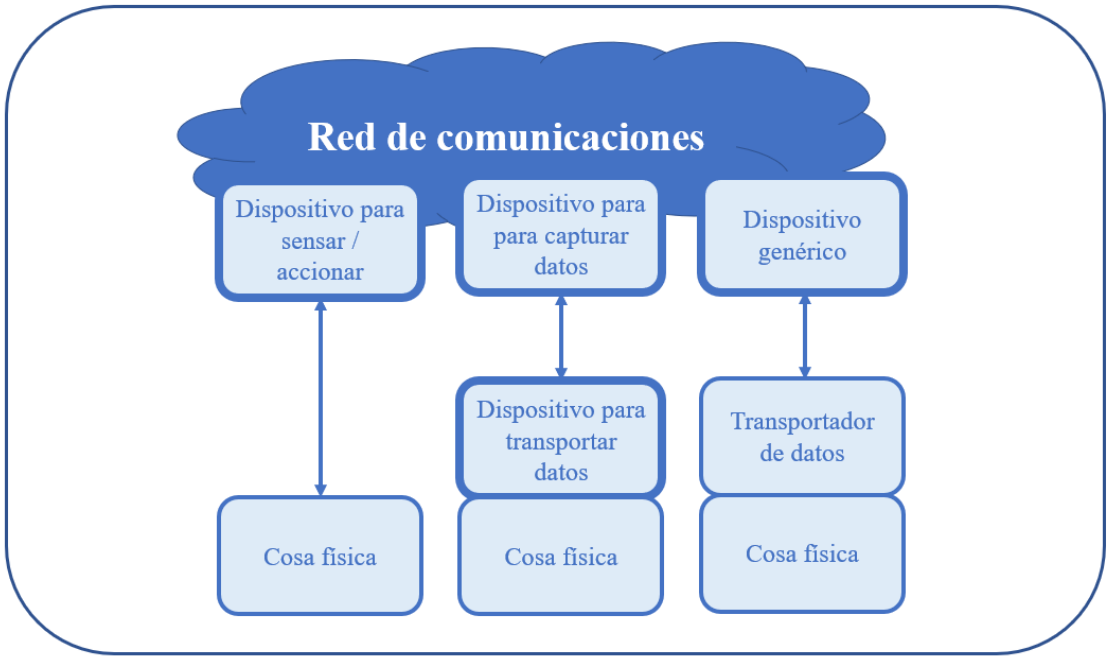

Figura 3. Tipos de dispositivos y su relación con cosas físicas

Fuente: Recomendación ITU-T Y.2060/Y.4000 (06/12)

2.2.3.1.2. Clasificación de dispositivos IoT

La recomendación indica que el requerimiento mínimo para un dispositivo IoT es su capacidad para soportar comunicaciones, y los clasifica en:

- Dispositivos de transporte de datos: Son los dispositivos adjuntos a la cosa física para conectarla con las redes de comunicación indirectamente.

- Dispositivos de captura de datos**:** Se refiere a dispositivos de lectura/escritura con la capacidad de interactuar con las cosas físicas. Esta interacción puede darse a través de un dispositivo de transporte de datos o de un portador de datos conectado a la cosa física.

- Dispositivos de censado y accionado**:** Son dispositivos que pueden medir o sensar información relacionada al entorno que los rodea y convertir estas mediciones en señales electrónicas digitales. También pueden convertir señales electrónicas digitales recibidas desde las redes de información a acciones específicas. En general estos dispositivos se conectan entre sí a través de tecnologías de redes cableadas o inalámbricas y utilizan *gateways* para conectarse con las redes de comunicaciones.

- Dispositivo genérico**:** Es un dispositivo que tiene embebidas capacidades de procesamiento y comunicación y puede conectarse a las redes de comunicación utilizando tecnologías de red cableadas o inalámbricas.

2.2.3.1.3. Modelo de referencia.

A continuación, en la Figura 4 se puede ver el Modelo de Referencia que plantea la ITU-T.

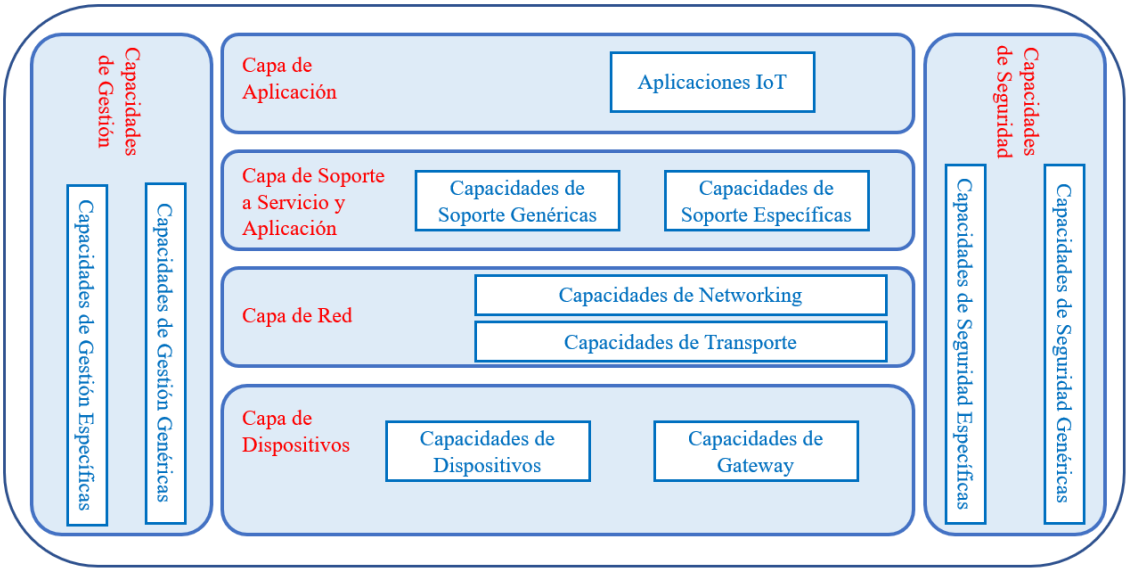

Figura 4. Modelo de Referencia

Fuente: Recomendación ITU-T Y.2060/Y.4000 (06/12)

Las cuatro capas que menciona este modelo de referencia son:

- Capa de Aplicación: Contiene las aplicaciones de IoT.

- Capa de Soporte a Servicio y Aplicación: Esta capa incluye básicamente dos grupos de capacidades:

    - *Capacidades genéricas de soporte*: Son capacidades que pueden ser utilizadas por diferentes aplicaciones de IoT, por ejemplo, procesamiento y almacenamiento de datos.

    - *Capacidades específicas de soporte*: Son capacidades particulares y específicas de las diferentes aplicaciones de IoT.

- Capa de Red: Básicamente consiste en dos grupos:

    - *Capacidades de networking*: Provee funciones relevantes a la conectividad, como funciones de acceso y transporte de los recursos, gestión de movilidad y funciones de autenticación, autorización y contabilización.

    - *Capacidades de transporte*: Enfocada en proveer conectividad para el transporte de información específica relacionada con los servicios y aplicaciones, así como el transporte de la información de gestión y control relacionada con los dispositivos IoT.

- Capa de dispositivos: Se pueden dividir lógicamente en dos grupos de capacidades:

    - *Capacidades de dispositivos***:** Interacción directa o indirecta con la red de comunicaciones, Ad-Hoc *networking*, capacidades de dormir/despertar para ahorrar energía, etc.

    - *Capacidades de gateways***:** Soporte para múltiples interfases (CAN, Wifi, PSTN, 2G/3G, LTE, DSL, etc.), conversión de protocolos, etc.

2.2.3.2. ITU-T: Recomendación Y.4460 (06/19) – Modelos de referencia arquitectónicos de dispositivos para aplicaciones de Internet de las cosas.

En esta recomendación (ITU-T, ITU-T Y.4460 - Architectural reference models of devices for Internet of things applications, 2019) la ITU-T hace mención de los diferentes modelos de referencia arquitectónicos desde el punto de vista del poder de procesamiento y las capacidades de comunicaciones.

La potencia de procesamiento y las capacidades de comunicación definen cómo el dispositivo se comunica e interactúa con otras entidades en una solución de IoT, y según esta recomendación es posible clasificarlos en tres categorías.

2.2.3.2.1. Dispositivos LPLC (Low Processing Low Connectivity).

Se incluyen en esta categoría los dispositivos con baja capacidad de procesamiento y baja capacidad de comunicaciones. Se define en la recomendación como un dispositivo de IoT que solo actúa como una interfaz para la recopilación de datos de elementos físicos o del entorno circundante, y / o realiza operaciones en elementos físicos o el entorno circundante. Este dispositivo no tiene la capacidad de procesamiento suficiente para tomar decisiones o ejecutar algoritmos complejos; tampoco tiene suficientes capacidades de conectividad para conectarse directamente a las redes de comunicación.

Las entidades que participan en la arquitectura LPLC son las siguientes:

- *Sensar / Accionar / Capturar Datos***:** proporciona funciones para leer datos de sensores, escribir datos en actuadores y capturar datos de dispositivos portadores de datos o soportes de datos conectados a objetos físicos.

- *Manejo de Mensajes:* proporciona funciones para enviar y recibir mensajes, utilizando un protocolo de capa de aplicación. También puede proporcionar una máquina de estado para manejar los mensajes entrantes.
- *Acceso al Gateway:* proporciona funciones para la gestión de las comunicaciones con Gateway.

- *Gestión del Hardware:* proporciona funciones para acceder al hardware (sensores y / o actuadores, interfaces de comunicación física, periféricos de hardware como temporizadores, convertidores de analógico a digital, etc.).

En la Figura 5 se puede ver un resumen de las entidades arriba mencionada

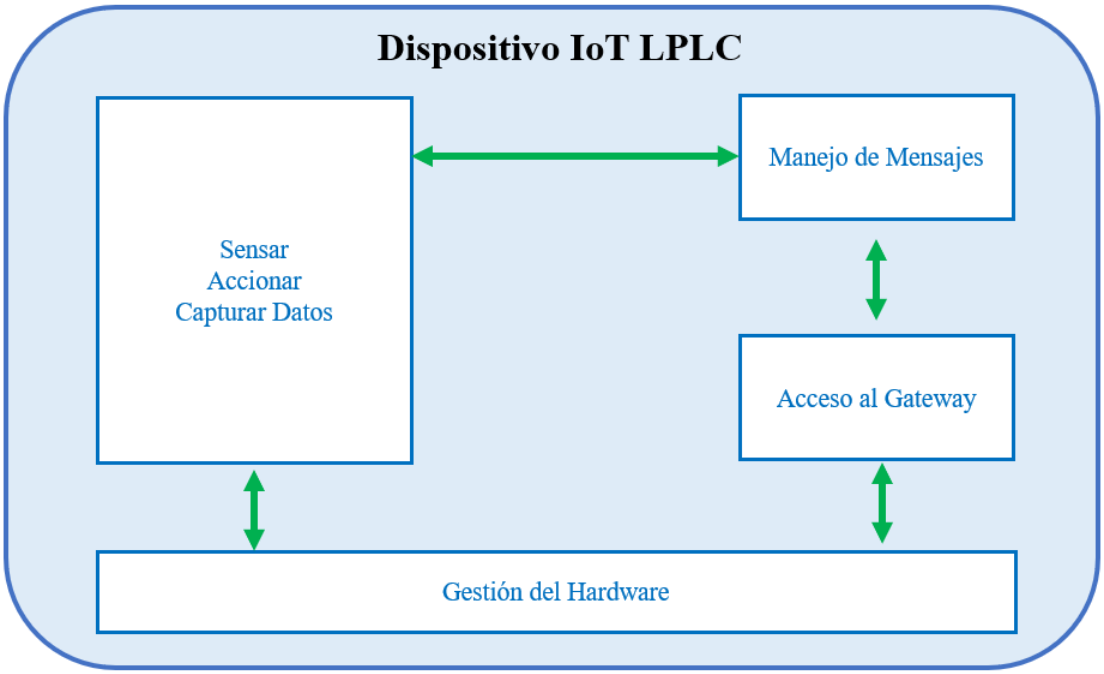

Figura 5. Arquitectura de Referencia de los dispositivos LPLC (Low Processing Low Connectivity)

Fuente: Recomendación ITU-T Y.44600 (06/19)

2.2.3.2.2. Dispositivos LPHC (Low Processing High Connectivity).

En esta categoría se incluyen los dispositivos de baja capacidad de procesamiento y alta capacidad de comunicaciones. Se define en la recomendación como un dispositivo de IoT que solo actúa como una interfaz para la recopilación de datos de elementos físicos o del entorno circundante, y / o realiza operaciones en elementos físicos o el entorno circundante. Este dispositivo tiene suficientes capacidades de conectividad para conectarse directamente a las redes de comunicación.

Los dispositivos LPHC agregan las siguientes entidades a la arquitectura que los contempla:

- *Interfaz de Aplicación / Servicio en la Nube:* proporciona funciones para interactuar con el servicio en la nube de IoT o la aplicación de IoT, enviar y recibir mensajes los mismos, registrar / autenticar el dispositivo, etc.

- *Gestión de Conectividad:* proporciona funciones para la gestión de la conectividad entre el dispositivo y la red de comunicaciones.

A continuación, en la Figura 6, se pueden ver las entidades para los dispositivos LPHC.

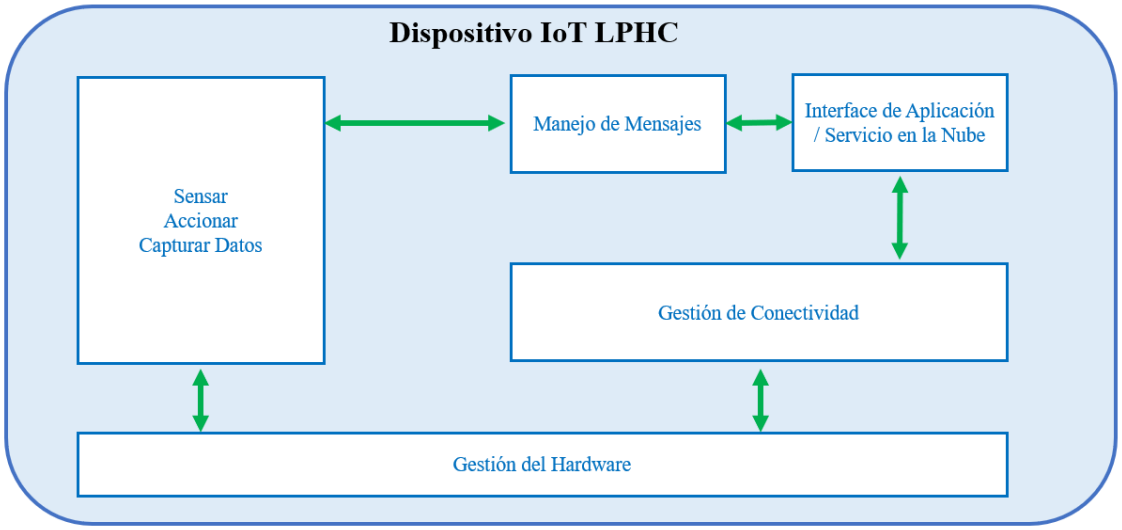

Figura 6. Arquitectura de Referencia de los dispositivos LPHC (Low Processing High Connectivity)

Fuente: Recomendación ITU-T Y.44600 (06/19)

2.2.3.2.3. Dispositivos HPHC (High Processing High Connectivity).

Esta última categoría contempla los dispositivos de alta capacidad de procesamiento y alta capacidad de comunicaciones. Se lo define como un dispositivo de IoT que no solo tiene altas capacidades de conectividad, lo que le permite conectarse directamente a aplicaciones y servicios en la nube, sino que también tiene capacidades de procesamiento lo suficientemente altas como para tomar decisiones y ejecutar algoritmos complejos (por ejemplo, algoritmos relacionados con IA). Los dispositivos son autónomos. Toman decisiones sobre sus propias funciones y también pueden coordinar otros dispositivos.

Los dispositivos HPHC agregan las siguientes entidades a la arquitectura que los contempla, la cual se puede visualizar en la Figura 7:

- *Motor de Ejecución de Aplicaciones*. proporciona funciones para instalar, eliminar, actualizar y ejecutar aplicaciones en los dispositivos. También proporciona a las aplicaciones acceso a otras entidades funcionales.

- *Gestión de Dispositivos.* proporciona funciones para gestionar el dispositivo y otros conectados al mismo.

- *Intercambio de Información.* proporciona funciones tales como interacción de dispositivo a dispositivo (intercambio de datos entre dispositivos), descubrimiento de servicios, supervisión de servicios e interoperabilidad de descubrimiento de servicios.

- *Analítica de Datos.* proporciona funciones para el procesamiento de datos y la decisión autónoma mediante la ejecución de algoritmos de análisis e IA.
- *Almacenamiento de Datos.* proporciona funciones de almacenamiento y recuperación de datos.

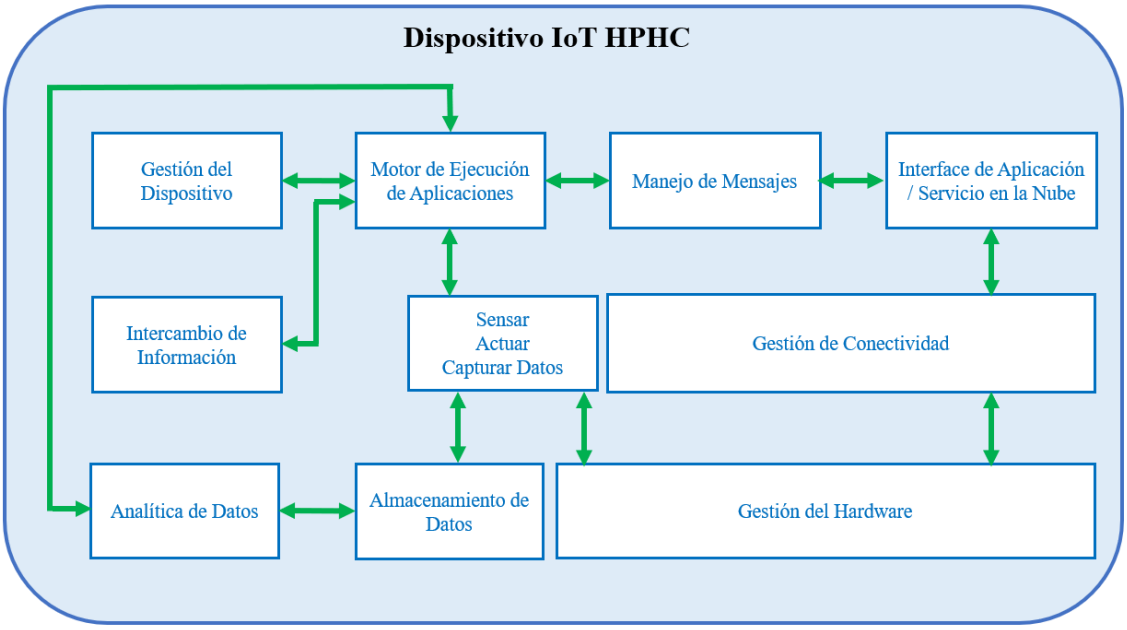

Figura 7. Arquitectura de Referencia de los dispositivos HPHC (High Processing High Connectivity)

Fuente: Recomendación ITU-T Y.44600 (06/19)

Los dispositivos sin capacidad de procesamiento no están contemplados en esta recomendación, los mismos se consideran en la recomendación ITU-T Y.4108/Y.2213 (ITU-T, ITU-T - Y.4108/Y.2213 - NGN service requirements and capabilities for network aspects of applications and services using tag-based identification , 2009).

2.2.3.3. ISO/IEC: 30141:2018 – Internet de las Cosas (IoT) – Arquitectura de Referencia

A diferencia de la recomendación de la ITU-T, que está más centrada en la comunicación de nivel de dispositivos. Esta norma intenta dar una visión más completa del ecosistema de IoT, dividiéndolo en dominios, según se ve en la Figura 8 a continuación:

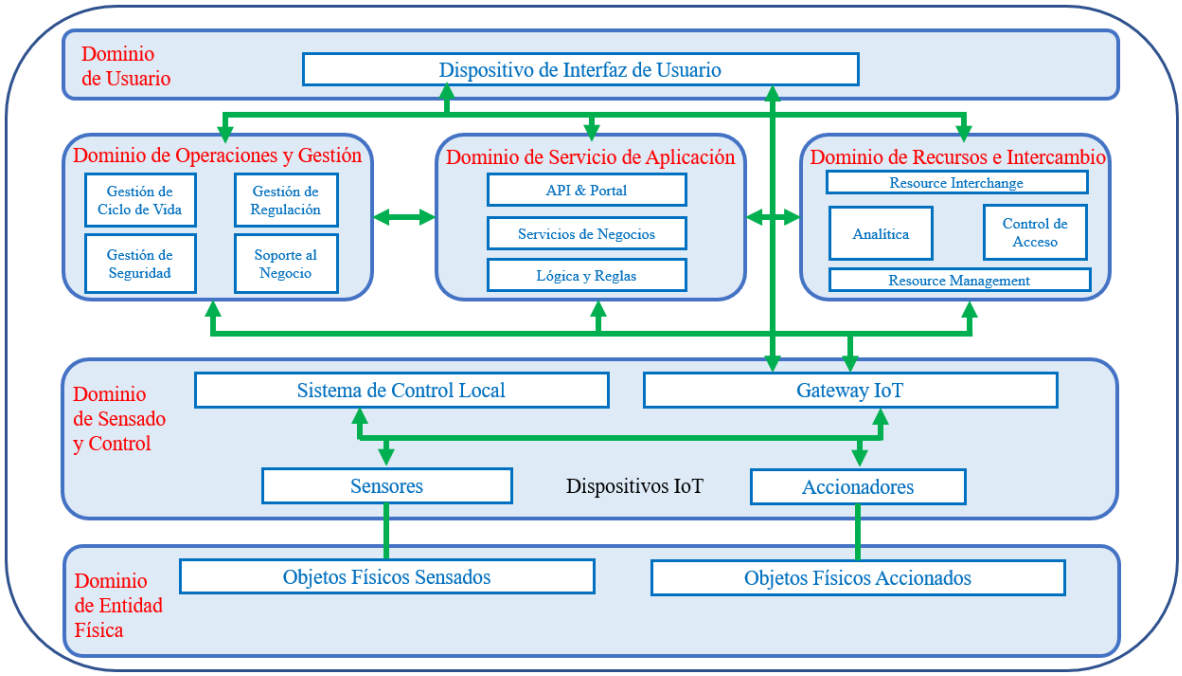

Figura 8. Dominios de la arquitectura de referencia

Fuente: Recomendación ISO/IEC 30141:2018

La norma divide al ecosistema de IoT en seis dominios, a saber:

- Dominio de entidad física**.** Son todos los objetos físicos sujetos de ser sensados y controlados por un sistema IoT.

- Dominio de sensado y control. Consiste en los dispositivos IoT, sensores y actuadores, utilizados para medir el estado y características de los objetos físicos y sobre los cuales se ejecutan acciones de control. Es un dominio esencial en el ecosistema IoT y provee información crítica acerca del entorno al resto de los dominios. Los dispositivos IoT requieren transmitir y recibir datos desde y hacia servicios que pueden ser locales o remotos. Estas comunicaciones las realizan a través de los *gateways* IoT.

- Dominio de operaciones y gestión**.** Representa el conjunto de funciones responsables del aprovisionamiento, la gestión, el monitoreo y la optimización del rendimiento operativo de los sistemas en tiempo real. Los operadores y gerentes de sistemas son los actores principales en este dominio.

- Dominio de recursos e intercambio**.** Interactúa con entidades, aplicaciones, servicios y sistemas externos al sistema IoT en términos de recursos.

- Dominio de servicio de aplicación**.** Ofrece servicios orientados al negocio para los usuarios del sistema IoT. Incluye a todos los proveedores de servicios del sistema IoT. Este dominio interactúa también con el Dominio de Sensado y Control para obtener información de los dispositivos físicos que forman parte del sistema, y también interactúan con el Dominio Recursos e Intercambio para comunicarse con otros sistemas externos y plataformas de IoT.

- Dominio de usuario**.** Está formado por las partes interesadas y actores principales del sistema IoT. Pudiendo ser tanto personas individuales como grupos de personas u organizaciones.

2.2.3.4. GSMA - Descripción General de los Lineamientos de Seguridad IoT de la GSMA Versión 2.2 febrero de 2020

En la Figura 9 que observamos a continuación, se puede visualizar el modelo de IoT utilizado por la GSMA para representar la arquitectura soportada por tecnologías celulares sobre la cual basa su marco

de gestión de riesgos: (Childs, Smith, & Bailey, IoT Security Guidelines Overview Document Version 2.2, 2019, pág. 16)

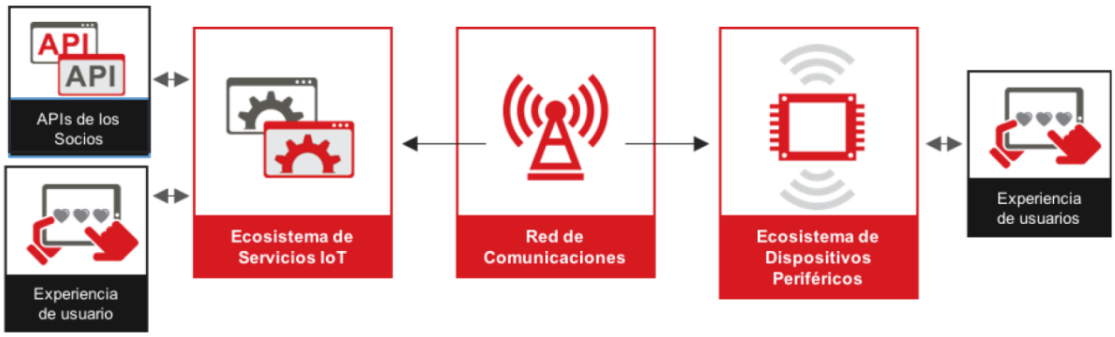

Figura 9. Modelo estándar de IoT

Fuente: GSMA - Descripción General de los Lineamientos de Seguridad IoT de la GSMA Versión 2.2

En el modelo de la GSMA se hace referencia a dos ecosistemas interconectados por la red de comunicaciones, el ecosistema de servicios y el ecosistema de dispositivos periféricos.

- Ecosistema de servicios. Está representado por el conjunto de servicios, plataformas, protocolos y otras tecnologías necesarias para proporcionar las capacidades requeridas y recopilar datos de dispositivos periféricos desplegados sobre el terreno. Típicamente recoge los datos de los dispositivos periféricos y los almacena en su entorno de servidor.

- Ecosistema de dispositivos periféricos. Se dividen en tres grupos.

    - *Dispositivo periférico ligero.* Es típicamente un sensor o dispositivo físico simple. Su objetivo es servir un propósito físico específico (comando) y proporcionar mediciones al ecosistema de servicios IoT o al usuario final. Usualmente utilizando una CPU económica (probablemente un microcontrolador de 8 bits) y una red de

área personal, PAN (sigla en inglés para *Personal Area Network*) o un protocolo de red capilar como BLE (sigla en inglés para *Bluetooth Low Energy*), Thread o Zigbee. Suelen ser dispositivos de baja potencia que pueden operar con una batería, una pila de tipo botón, con energía solar o batería de litio. Se conectan al ecosistema de servicios de IoT a través del Hub y un *router* del lado del cliente, como se ve en la configuración #3 en el siguiente gráfico.

Entre los dispositivos típicos de esta categoría podemos mencionar:

- Weareables *o dispositivos personales.*
- *Sensores para servicios de seguridad en el hogar.*
- *Balizas/sensores de proximidad.*
- *Dispositivos para redes capilares no celulares*

- *Dispositivo periférico complejo.* Este modelo de periférico generalmente tiene una conexión persistente al servidor de *back-end* a través de un enlace tipo WAN (configuración #1 en la Figura 10), como, por ejemplo, a través de la red celular o con una conexión local tipo WIFI o Ethernet a través un Hub de cliente final (configuración #2 en la Figura 10). El dispositivo puede tener un procesador rudimentario, o incluso un microcontrolador de 8 bits, aunque la unidad de procesamiento suele ser más robusta que los dispositivos periféricos ligeros, ya que tienen una alimentación constante a la red eléctrica, e incluso pueden contar con baterías recargables para alimentarse en caso de corte y energía. Algunos de estos dispositivos requieren más capacidad de procesamiento por ejemplo para procesamiento de audio o video en tiempo real. Ejemplos de estos dispositivos son:
  - *Sistemas de iluminación conectados.*
  - *Electrodomésticos como heladeras, lavarropas conectados.*

- *Sistemas de control industrial (SCADA).*
- *Dispositivos de supervisión y seguimiento para automóviles conectados.*

- *Gateway o Hub*. Es un dispositivo, generalmente conectado a una fuente de alimentación dedicada que gestiona la comunicación entre los Dispositivos Periféricos ligeros y los sistemas back-end que los gestionan. Se encarga de administrar las comunicaciones WAN de larga distancia, usualmente con tecnologías celulares (incluyendo LPWA), por satélite, UTP, fibra o Ethernet. Acepta comandos de los sistemas back-end que forman parte del Ecosistema de Servicios y los traduce a mensajes aptos hacia los dispositivos periféricos ligeros. También puede ejecutar tareas críticas como:
  - Descubrimiento de dispositivos.
  - Despliegue del controlador de red.
  - Gestión.
  - Supervisión en tiempo real.
  - Autenticación y seguridad.

  Usualmente hay dos tipos de *gateways*:
  - Gateways de servicios de IoT: Puede ser propiedad del usuario final o no, pero es administrada por el proveedor de servicios IoT y se utiliza como concentrador de todos los dispositivos periféricos ligeros para conectarlos al ecosistema de servicios de IoT, ya sea a través de una conexión cableada, celular o una puerta de enlace CPE.
  - Gateway CPE: El operador de red la proporciona, normalmente un router de banda ancha conectado a internet por cable o red celular. Esto puede utilizarse en entornos residenciales o empresariales. En esta configuración

el gateway se administra y configura desde los sistemas del operador de red.

En la Figura 10 a continuación, se pueden ver algunas configuraciones típicas:

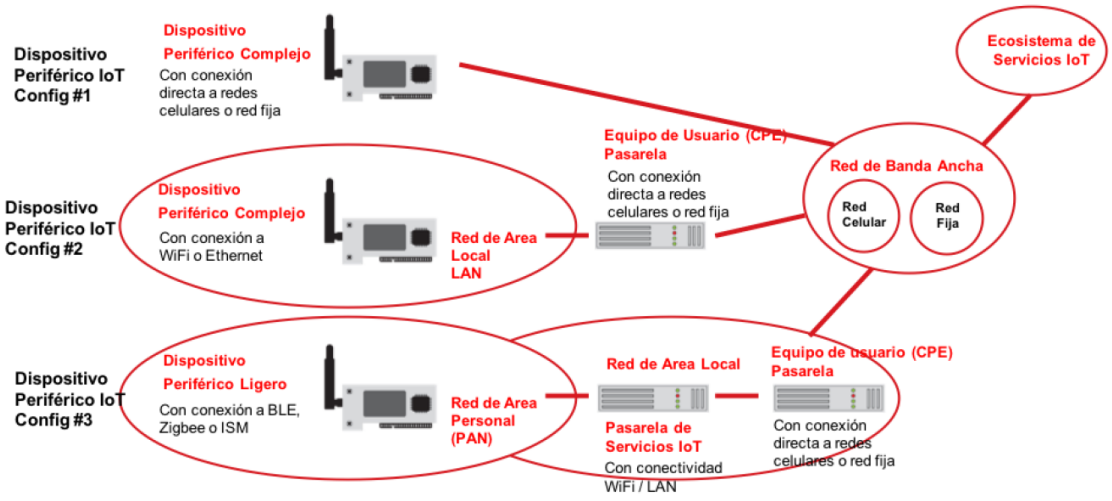

Figura 10. Ejemplo de configuración de ecosistema de dispositivos periféricos

Fuente: GSMA - Descripción General de los Lineamientos de Seguridad IoT de la GSMA Versión 2

2.2.4. Protocolos Utilizados

Existen diversas clasificaciones de redes, según su topología. Las mismas se detallan en la Figura 11:

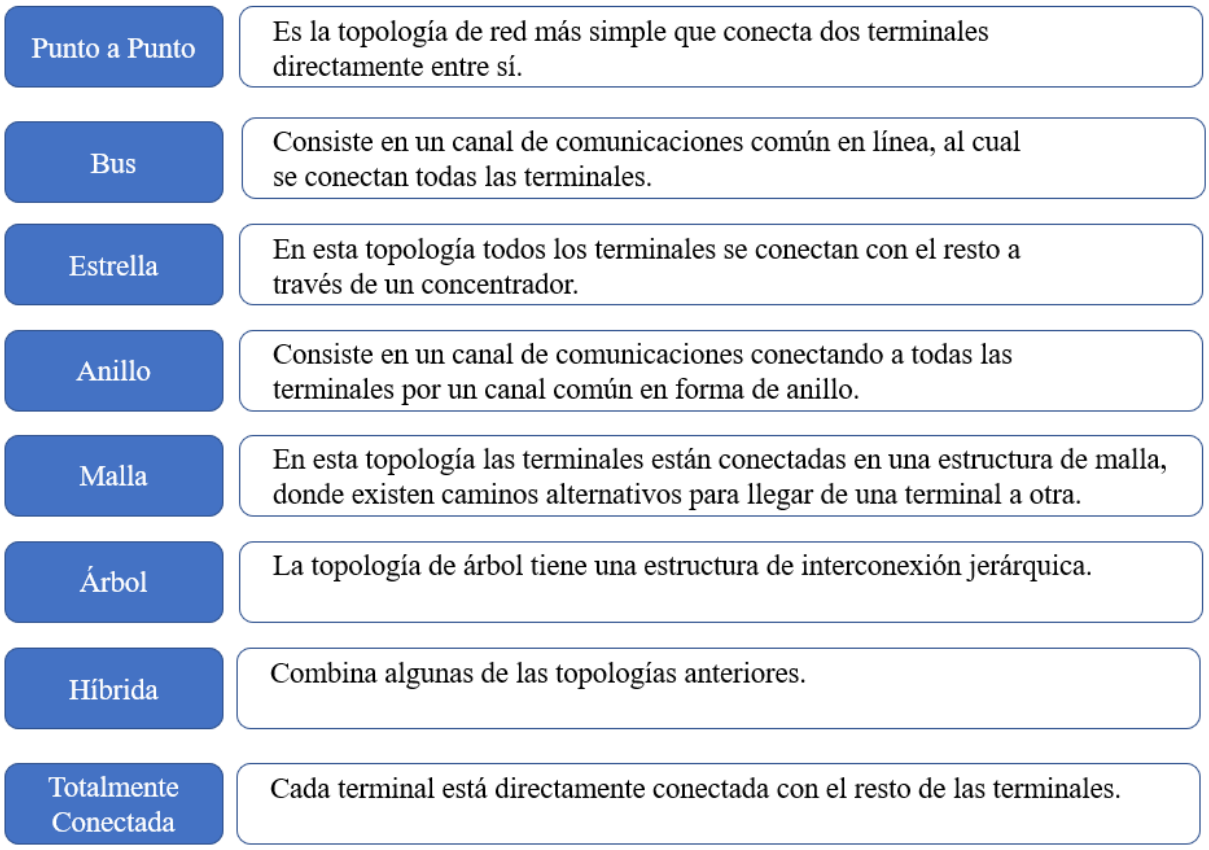

Figura 11. Topologías de redes

Fuente: Autoría propia

Otra clasificación de las redes puede ser según el tipo de conectividad utilizada para la transmisión de datos (cableada, inalámbrica) y también, entre otras clasificaciones, según su ámbito de aplicación.

Tradicionalmente existían por un lado las redes de Tecnología de la Información, conocidas como redes IT (sigla en inglés para *Information Technology*), utilizadas por las típicas redes corporativas empresariales o gubernamentales, y por otro lado las redes de Tecnología de la Operación o redes OT (sigla en inglés para *Operation Technology*), las cuales eran utilizadas en ámbitos industriales, soportando procesos de producción, infraestructuras críticas y *utilities*.

Las redes IT eran diferenciadas de las redes OT, en que las primeras eran principalmente aplicaciones que funcionaban sobre bases de datos basadas en servidores web y de aplicaciones, mientras

que las segundas eran usualmente sistemas SCADA (sigla en inglés para *Supervisory Control and Data Acquisition*) basados en sistemas de hardware y software propietarios.

Actualmente existe una tendencia a una convergencia entre ambas redes, especialmente desde el ámbito OT los sistemas empiezan a utilizar sistemas operativos y hardware estandarizado, incluso muchos sistemas OT ya corren en sistemas virtualizados.

Las redes IT tienen su centro en la gestión de la información mientras que en las redes OT su foco está orientado a la gestión de procesos industriales y, finalmente tenemos a las redes IoT las cuales tienen su principal foco en la conectividad. Esta característica típica de las redes IoT nos trae una gran cantidad de protocolos y mecanismos de operación que conviven y hacen más dificultosa la gestión de la seguridad sobre este tipo de redes.

Si bien hoy en día existen diversas tecnologías para soportar las redes IoT, las comunicaciones M2M (sigla en inglés para *Machine to Machine*) son utilizadas desde hace muchos años por distintas industrias como bancos y energía, siendo algunas de sus aplicaciones típicas el soporte para medidores inteligentes en el sector energético, detección de pérdidas en tuberías, procesamiento de tarjetas de crédito, etc.

Muchas redes M2M se basaron en el standard IEEE 802.15.4 creado para aplicaciones de monitoreo y control de baja velocidad de transferencia de datos y bajo consumo de energía. Entre estos protocolos podemos mencionar ZigBee, WirelessHART y MiWI, los cuales, junto con tecnologías como WiFi, *Bluetooth* y NFC (sigla en inglés para *Near Field Communication*) continúan dándole soporte a las comunicaciones M2M.

Como mencionamos previamente, las redes IoT están caracterizadas por dos requerimientos:

- Ser mayormente inalámbricas.
- Requerir baja potencia para su operación.

Estos dos elementos nos dejan una tercera variable que tiene que ver con el alcance, la cual nos definirá el protocolo a ser utilizado.

Respecto del alcance podemos definir 3 tipos de redes tal como se puede ver en la Figura 12:

1. Redes IoT de corto alcance: LoWPAN - *Low-Power Wireless Personal Area Networks*.
2. Redes IoT de medio alcance: LPWLAN – *Low-Power Wireless Local Area Network*.
3. Redes IoT de largo alcance: LPWAN – *Low-Power Wide Area Network*.

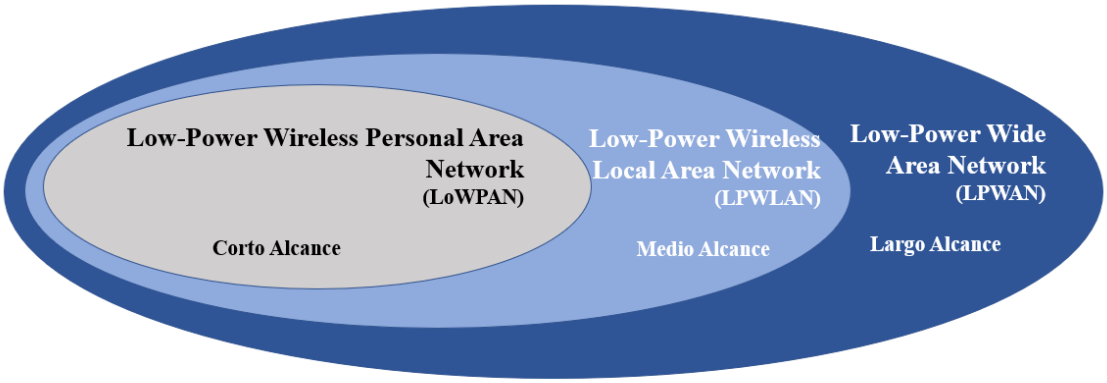

Figura 12. Tipos de Redes de baja potencia según su alcance

Fuente: Autoría propia

A continuación, en la Figura 13, se ve un gráfico donde se agrupan los protocolos según el tipo de red clasificadas por velocidad y alcance.

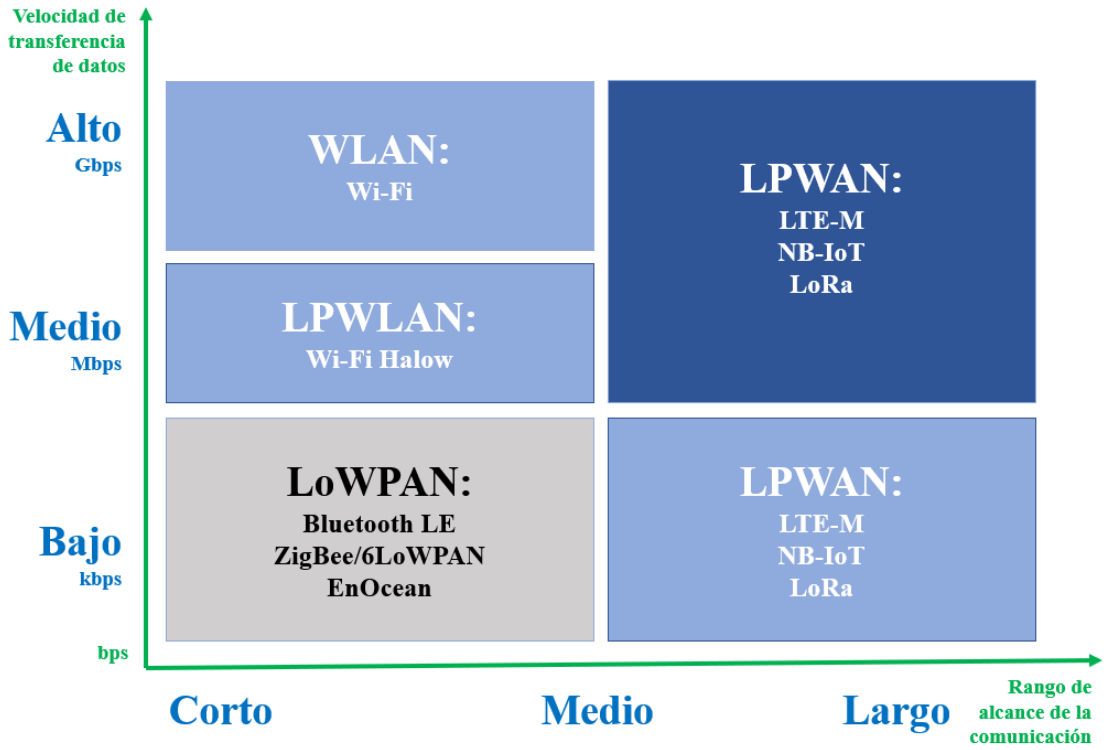

Figura 13. Agrupamiento de Tipos de Redes según alcance y tasa de transferencia

Fuente: Autoría propia

A continuación, en la Figura 14, se incluye un gráfico donde el lector puede visualizar los protocolos IoT y su correlación con el Modelo OSI.

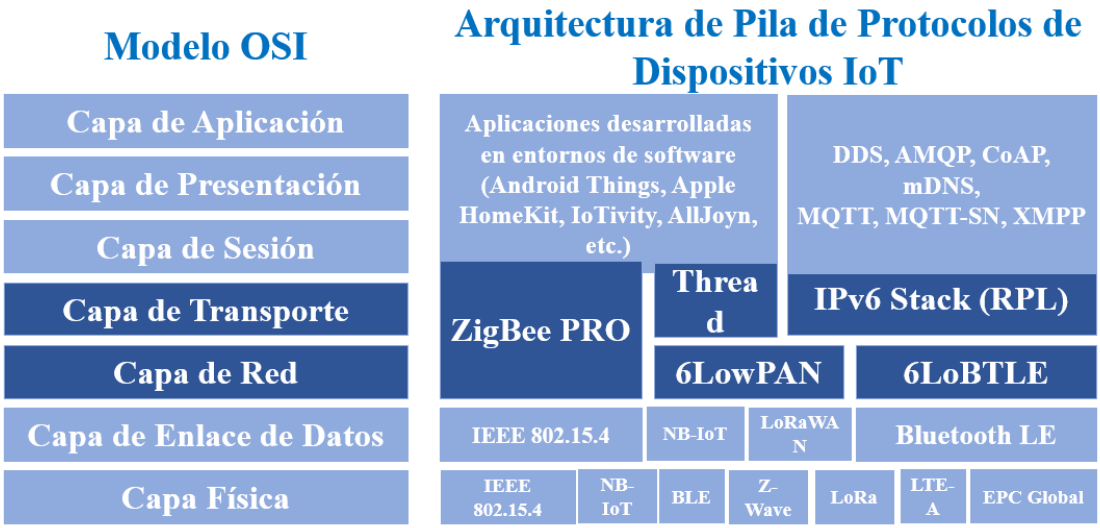

Figura 14. Protocolos de dispositivos IoT y su correlación con el Modelo OSI

Fuente: Autoría propia

A continuación, describiremos los protocolos más utilizados en las diferentes capas de la arquitectura de IoT.

Dentro de los protocolos de infraestructura, los que actúan sobre las capas OSI 1 a 4 (capa física, de enlace de datos, de red y de transporte) podemos mencionar los que se incluyen en la Tabla 4:

Protocolo	Descripción
6LoWPAN	Los componentes de una red LoWPAN (Low Power Personal Area Network) están conectados a la red IPv6 a través de un router de frontera 6LoWPAN. Entre sus funciones podemos mencionar: • Permite el intercambio de datos entre dispositivos 6LoWPAN e Internet u otras redes IPv6. • Permite el intercambio de datos entre otros dispositivos que forman parte de la red 6LoWPAN. • Ayuda a generar y mantener la red 6LoWPAN.
BLE (Bluetooth Low Energy)	Comenzó como parte de la especificación principal de Bluetooth 4.0. Utiliza ondas de radio de corto alcance y funciona por períodos prolongados. El rango de cobertura es de aproximadamente 100 metros, 10 veces más que el alcance de Bluetooth convencional y su latencia es 15 veces menor. BLE opera con rangos de potencia entre 0,01 mW y 10 mW.
EnOcean	Definido por el estándar ISO/IEC 14543-3-10, es una tecnología inalámbrica de recolección de energía utilizada principalmente en sistemas de automatización de edificios, y también se aplica a otras aplicaciones en la industria, transporte, logística y hogares inteligentes. Los módulos basados en esta tecnología combinan micro convertidores de energía con electrónica de ultra baja potencia y permiten comunicaciones inalámbricas entre sensores inalámbricos sin batería, interruptores, controladores y puertas de enlace.

EPC (Electronic Product Code) Global	Los dispositivos RFID (*Radio Frequency Identification*) son microchips utilizados para identificar objetos de manera automática. El EPC es un identificador único almacenado en un tag RFID que ayuda a identificar y realizar el seguimiento de objetos en un escenario de cadena de suministros. EPC Global es la organización que desarrolla y mantiene estándares relacionados con RFID y EPC. Esta tecnología es importante para los dispositivos IoT ya que permite el soporte para el descubrimiento de dispositivos en una red IoT.
IEEE 802.15.4	Este protocolo fue diseñado para especificar una subcapa para el control de acceso medio (MAC) y capa física, principalmente para redes inalámbricas de área personal de baja velocidad. Los beneficios que ofrece este protocolo son: • Bajo consumo de potencia. • Baja tasa de transferencia. • Bajo costo. • Gran capacidad de procesamiento de mensajes. • Muy adecuado para su uso en sistemas IoT como protocolo de comunicación. Es un protocolo ideal para comunicaciones seguras ya que provee altos niveles de seguridad, cifrado y servicio de autenticación.

IEEE 802.11ah	Es uno de los estándares más discutidos y explotados en sus funcionalidades y aplicaciones. Su diseño tiene como impulso la demanda de altas tasas de transferencia de datos. Con el surgimiento de las comunicaciones de máquina a máquina (M2M), fue necesario ajustar el estándar IEEE 802.11 que fue diseñado principalmente para la comunicación a computadora. La comunicación M2M exige características distintas, como un rango de transmisión superior a 1 km, velocidades de transmisión superiores a 100 Kbps y bajo consumo de energía. También es necesario tener una red que admita un gran número de nodos, operando bajo una política de bajo consumo de energía. En respuesta a estos requerimientos es que se desarrolló el estándar 802.1ah.
Lo-Ra (Long Range)	Es una tecnología WAN de bajo consumo que trabaja en la capa física del modelo OSI. Está basada en técnicas de modulación de espectro expandido. LoRaWAN (Long Range Wide Area Network) cubre las capas superiores y es mantenido por la LoRa Alliance.
LTE-M (LTE-Machine Type communications)	Estándar de tecnología LPWAN. Es una tecnología de área amplia de baja potencia que admite IoT a través de una menor complejidad del dispositivo y proporciona una cobertura extendida, al tiempo que permite la reutilización de la base instalada LTE. Esto permite una vida útil de la batería de hasta 10 años o más para una amplia gama de casos de uso.
NB-IoT (Narrow Band Internet of Things)	Es una tecnología de radio LPWAN estándar desarrollada por el 3GPP. Está enfocada especialmente para cobertura interior con bajo costo, bajo consumo y alta densidad de conexión. Está basado en el estándar LTE.

NFC (Near Field Communications)	Basado en el estándar ISO/IEC 18092: 2004, es un estándar de conectividad inalámbrica de corto alcance que utiliza la inducción de campo magnético para permitir la comunicación entre dispositivos cuando se tocan o se acercan unos pocos centímetros el uno del otro. Utiliza dispositivos inductivos acoplados a una frecuencia central de 13.56 MHz. La velocidad de datos es de hasta 424 kbps y el rango es de unos pocos metros en comparación con las redes inalámbricas de sensores.
RPL	Protocolo de enrutamiento para redes de baja potencia y con pérdida. Es un protocolo de IPv6. Es utilizado en redes WPAN (Wireless Personal Area Networks), líneas de comunicación de baja potencia o PLC (Low-Power Line Communications) y redes de sensores inalámbricos o WSN (Wireless Sensors Networks). Estas redes tienen algunas características comunes como ser: • Capacidad para optimizar y ahorrar energía. • Capacidad de soportar patrones de tráfico adicionales a las comunicaciones unicast. • Capacidad para ejecutar protocolos de enrutamiento sobre capas de enlace con tamaños de trama restringidos.

SigFox	Es una tecnología que trae una nueva red y estrategia de información a IoT. SigFox posee un modelo de negocio de operador de red y es utilizado por aplicaciones que requieren velocidades de transmisión de datos bajas. A diferencia de los sistemas celulares móviles, un terminal Sigfox no está conectado a una sola estación base. Con una implementación adecuada, el mensaje enviado por un terminal Sigfox puede ser recibido por varias estaciones base. Esta característica se denomina diversidad espacial que, junto con la diversidad de repeticiones en el tiempo y la frecuencia, son puntos fuertes en la tecnología de concepción del protocolo Sigfox.
Thread	• Es mantenido por la Thread Alliance. Basado en IPv6 6LoWPAN. Trabaja con una topología de red en malla de bajo consumo para dispositivos IoT.
Wireless HART (Highway Addressable Remote Transducer Protocol)	Es una variación del diseño de la IEEE 802.15.4 para funcionar esencialmente como una red inalámbrica centralizada. Posee las mismas especificaciones que la IEEE 802.15.4 pero tiene su propia capa de acceso al medio utilizando una técnica basada en TDMA (Time División Multiplex Access).

Z-Wave	Es un protocolo de comunicaciones inalámbricas de baja potencia principalmente utilizado para redes hogares o HAN (sigla en inglés para Home Area Networks). Es un protocolo muy utilizado por aplicaciones de control remoto de hogares inteligentes o pequeños comercios. Utiliza una topología de red de tipo malla de baja potencia. Cada nodo o dispositivo que forma parte de esa red tiene la capacidad de enviar y recibir comandos a través de paredes o pisos y también utiliza nodos intermedios para enrutar datos alrededor de obstáculos.
ZigBee	Es mantenido por la ZigBee Alliance. Tiene una topología de red descentralizada y tiene la capacidad de que los nodos encuentren y utilicen rutas alternativas en caso de encontrar una falla en la ruta predeterminada, lo cual lo hace un protocolo robusto. Las siguientes características lo posicionan como un protocolo muy adecuado para aplicaciones IoT: • Bajo consumo de energía. • Bajo costo. • Soporte para un gran número de nodos de red

Tabla 4. Protocolos de infraestructura IoT

Fuente: Autoría propia

A continuación, en la Tabla 5, se pueden visualizar los protocolos de corto alcance más usualmente utilizados en las redes IoT; y en la Tabla 6 los de largo alcance.

Protocolo	Estándar	Topología	Tasa de TX	Alcance	Frecuencia
BlueTooth LE	IEEE 802.15.1	Estrella	< 2 Mb/s	70 mts	2.4GHZ
EnOcean	ISO/IEC 14543-3-10:2020	Estrella	125 Kb/s	30mts (indoor) 300mts (outdoor)	EU & China: 868MHZ NA: 902MHZ Japan: 928MHZ
IEEE 802.11ah	IEEE 802.11ah	Estrella	100 Kb/s	1 km	< 1GHZ
IEEE 802.15.4	IEEE 802.15.4	Malla	20-250 Kb/s	10-100 mts	Global: 2.4GHZ EU: 868GHZ NA: 915GHZ
NFC	ISO/IEC 14443, 18092, JIS X6319-4	Peer-to-Peer	106/848 Kb/s	0.1 mts	13.56MHZ

RFID	ISO/IEC 18000,29167, 20248, JTC 1/SC 31	Punto a Punto Punto a Multipunto	500 Kb/s	0.1 - 5 mts	Global: 6MHZ ISM: 13.5MHZ, 433MHZ ISM EU: 633-870MHZ ISM NA: 902-928MHZ ISM: 2.4GHZ UWB: 5-27GHZ
Thread	IEEE 802.15.4	Malla	250 Kb/s	100 mts	Global: 2.4GHZ
Wireless HART	IEEE 802.15.4 PHY HART MAC	Estrella Cluster Malla	250 Kb/s	10-600 mts	Global: 2.4GHZ

ZigBee	IEEE 802.15.4	Estrella Árbol Malla	<250 Kb/s	80-200 mts	Global: 2.4GHZ EU: 868GHZ NA: 915GHZ
Z-Wave	Basado en ITU G.9959	Malla	9.6-100 Kb/s	100 mts	EU: 868GHZ NA: 915GHZ

Tabla 5. Protocolos de corto alcance

Fuente: Autoría propia

Protocolo	Estándar	Topología	Tasa de TX	Alcance	Frecuencia
LoRaWAN	LoRaWAN	Estrella de Estrellas	1 Mb/s	5 km Urbano 15 Km Rural	EU: 868MHZ US: 433/915MHZ AS: 430MHZ
LTE-M	3GPP	Estrella	<1 Mb/s	<100 km	Bandas de LTE
NB-IoT	3GPP	Estrella	DL 234.7 Kb/s UL 204.8 Kb/s	20 km	Bandas de LTE
SigFox	SigFox	Estrella	DL 100 b/s UL 600 b/s	15 km	EU: 868MHZ US: 902MHZ

Tabla 6. Protocolos de largo alcance

Fuente: Autoría propia

A continuación, hacemos referencia a algunos de los protocolos de aplicación más comúnmente utilizados:

2.2.4.1. AMQP (Advanced Message Queuing Protocol).

Es un protocolo de capa de aplicación estándar abierto (RFC 7252) para middleware orientado a mensajes. Las características definitorias de AMQP son la orientación de mensajes, las colas y el enrutamiento (incluido el punto a punto), la publicación y suscripción, la confiabilidad y seguridad.

2.2.4.2. CoAP (Constrained Application Protocol).

Protocolo destinado para su uso en dispositivos de recursos limitado, como los nodos WSN (*Wireless Sensor Network*). Está diseñado para traducir fácilmente a HTTP para integración simplificada con la web, también soporta multidifusión, es de bajo costo y simple.

2.2.4.3. DDS (Data Distribution Service for Real-Time Systems).

Es un protocolo de middleware y una API estándar para conectividad centrada en datos. Está mantenido por el Object Management Group. Integra los componentes de un sistema entre sí, brinda conectividad de datos de baja latencia, confiabilidad y una arquitectura escalable.

2.2.4.4. MDNS (Multicast Domain Name System).

Es un protocolo para descubrimiento. Resuelve nombres de hosts a direcciones IP dentro de redes pequeñas que no cuentan con un Servidor de Nombres de Dominio.

2.2.4.5. MQTT (Message Queuing Telemetry Transport).

Permite un modelo de mensajería de publicación / suscripción ligera. Es útil para conexiones con ubicaciones remotas donde se requiere una transferencia pequeña o el ancho de banda de la red es restringido.

2.2.4.6. MQTT-SN (MQTT For Sensor Networks).

Un protocolo de publicación / suscripción abierto y ligero diseñado específicamente para aplicaciones M2M y móviles.

2.2.4.7. XMPP (Extensible Messaging and Presence Protocol).

Tecnología abierta para la comunicación en tiempo real, que soporta una amplia gama de aplicaciones que incluyen mensajería instantánea, chat múltiple, llamadas de voz y video, colaboración, middleware ligero, sindicación de contenido y enrutamiento generalizado de datos XML.

2.2.5. Técnicas de Detección de Ataques de Botnets

Los dispositivos IoT fueron diseñados para cumplir una funcionalidad determinada, específica y acotada y tener un muy bajo consumo. Estas características hacen muy difícil la instalación de agentes de detección de *malware* dentro de los mismos dispositivos, convirtiendo al análisis y monitoreo del tráfico que se cursa en la red en una herramienta importante para su mitigación.

Las *botnets* pueden infectar tanto a redes IT, OT o IoT, pero las particularidades propias de cada una de estas redes requieren técnicas específicas para cada una de ellas.

La detección de *botnets* ha sido un desafío en el área de Seguridad de Redes desde hace tiempo. Existen muchas técnicas y métodos que han sido utilizados para detectar este tipo de *malware*. Respecto de

las técnicas podemos mencionar las basadas en firmas, heurística o comportamiento. Existen métodos estáticos que se basan en analizar el código, identificando la utilización de comandos usualmente invocados por este tipo de *malware*; y los dinámicos que se basan en el análisis de su comportamiento en un entorno controlado. También existen técnicas que en vez de analizar el código analizan su interacción con el entorno realizando análisis del tráfico de red, entre el origen y destino de los paquetes, los puertos utilizados, los tamaños de paquetes, la interacción con los DNS, etc.

Para obtener información sobre las amenazas se suelen utilizar las *honeypots*, que son ambientes que simulan un entorno vulnerable expuesto a los atacantes, donde se espera recibir ataques para conocer sus mecanismos y de esta manera poder diseñar estrategias para mitigarlos en los entornos reales.

La detección basada en firmas se basa en patrones conocidos denominados firmas, las cuales se utilizan para identificar dentro del tráfico analizado *malware* conocidos. La desventaja de este método es que no se adapta al típico comportamiento cambiante de este tipo de ataques.

Los *honeypots* son herramientas utilizadas para colectar información acerca del comportamiento de *bots*. Esto se suele denominar Inteligencia de Amenazas y es una herramienta de utilidad al momento de mantener actualizada la información acerca de nuevas tendencias y comportamientos respecto de los ciberataques. Los *botmasters*, quienes gestionan las *botnets*, suelen ser muy cuidadosos y poseen técnicas para detectar estos *honeypots*, lo cual implica un gran desafío a quien los configura ya que deben tener características muy similares a las redes que pretenden simular para lograr engañar a los atacantes y obtener la información de comportamiento que se busca.

El monitoreo de tráfico de DNS es un método que inicialmente era utilizado para analizar el tráfico de DNS generado por los *botnets*, pero actualmente ya no se utiliza debido a que estos *malware* están diseñados para minimizar este tipo de tráfico.

La detección basada en análisis de comportamiento provee un enfoque efectivo al momento de la detección de tráfico relacionado con *botnets*, basándose en la búsqueda de patrones anómalos en el mismo. El análisis de comportamiento puede ser llevado a cabo a través del monitoreo estadístico del tráfico como

explica Baieli en su trabajo (Baieli, 2017). Otra manera de encarar el análisis de comportamiento de tráfico es a través de la aplicación de técnicas de IA empleando básicamente dos técnicas para la detección.

La primera se refiere a las técnicas de Aprendizaje Automático o *Machine Learning*, con modelos denominados "*supervisados*". Estos modelos se crean obteniendo información de *honeypots* o de tráfico previamente "*etiquetado*", es decir, que se conoce de antemano que es tráfico malicioso. El modelo se alimenta de este tráfico y se lo entrena para que elabore su aprendizaje acerca de cómo es el comportamiento que se quiere identificar. Una vez entrenado el modelo, se lo testea con tráfico también conocido de antemano y se valida la tasa de detección sobre el mismo (cantidad de detecciones / total de casos reales). Es un proceso iterativo donde luego de que el modelo es entrenado y testeado y habiendo obtenido la tasa de efectividad deseada, se lo puede poner operativo en producción para analizar tráfico nuevo en la red con un alto grado de confianza en su funcionamiento.

La segunda se refiere a las técnicas relacionadas con Aprendizaje Profundo o *Deep Learning*; son modelos "*no supervisados*", ósea, no se conoce de antemano si el tráfico corresponde o no a comportamientos maliciosos. Básicamente lo que se hace con estos modelos es realizar un análisis del tráfico en condiciones normales, generando un perfil de comportamiento, el cual será utilizado de línea base para el análisis de tráfico en el futuro. Cualquier desvío en un porcentaje definido por el analista, en una o varias de las características de tráfico analizadas, automáticamente alertará que se ha encontrado una anomalía.

2.3. Inteligencia Artificial

2.3.1. Introducción

La IA es una rama de las Ciencias Informáticas; y si bien existen diversas definiciones podríamos intentar definirla de manera inicial como la habilidad de las computadoras de imitar actividades que

usualmente requieren de inteligencia humana. Si profundizamos un poco más en la definición podríamos decir que la IA es la capacidad de las computadoras de aprender de los datos a través del uso de algoritmos y utilizar dicho aprendizaje para tomar decisiones tal como lo haría un humano.

La IA se ocupa del diseño y la implementación de sistemas informáticos capaces de resolver problemas que suelen requerir inteligencia humana, pero con dos grandes ventajas; la primera es la capacidad de procesar grandes volúmenes de datos en muy poco tiempo y la segunda, con una tasa de error mucho menor.

Desde el artículo fundamental de Alan Turing en 1950 (Turing, 1950) sobre la posibilidad de programar una computadora electrónica para que se comporte de manera inteligente, la IA ha experimentado durante las últimas décadas un rápido crecimiento en investigación y desarrollo. El éxito actual de la IA proviene, entre otros factores, del diseño de nuevas arquitecturas de sistemas capaces de utilizar todo el conocimiento (incluyendo la experiencia humana) disponible en un dominio determinado para mejorar sus propios resultados.

A continuación, podemos mencionar como algunas de las capacidades de la IA:

1. Predecir y adaptarse a través de la utilización de algoritmos que descubren patrones analizando grandes volúmenes de información.

2. Tomar decisiones de manera autónoma, pudiendo aumentar la inteligencia humana brindando información y mejorando la productividad.

3. Aprender de manera continua utilizando algoritmos para construir modelos analíticos. A partir de esos algoritmos, la tecnología de IA descubrirá cómo realizar tareas a través de sucesivas iteraciones de prueba y error.

4. Visión de futuro; la IA es una herramienta que permite a las personas reconsiderar cómo analizamos los datos e integramos la información para luego utilizar estos conocimientos para tomar mejores decisiones.

A medida que esta tecnología crece va teniendo un impacto importante en nuestras vidas.

2.3.2. Antecedentes

En esta sección presentaremos algunos de los antecedentes más sobresalientes relacionados con el nacimiento y desarrollo de las técnicas de IA, los mismo se detallan en la Tabla 7 y la Figura 15, a continuación.

Año	Evento
1943	El trabajo de Warren McCullogh y Walter Pitts (McCullogh & Pitts, 1943) "*A logical calculus of the ideas immanent in nervous activity*" es el primer estudio sobre redes neuronales presentando el primer modelo computacional de una neurona biológica y describiendo los componentes básicos de una neurona artificial.
1949	Donald Hebb publica su libro "*The organization of behaviour: A neuropsychological theory*" (Hebb, 1949), el cual sirvió como punto fundacional para los algoritmos de aprendizaje de redes neuronales. En dicho trabajo se enuncia el aprendizaje por refuerzo el cual será utilizado como base de los sistemas de aprendizaje por refuerzo de redes neuronales, también conocido como aprendizaje no supervisado.
1950	Alan Turing publica su famoso artículo "*Computing Machinery and Intelligence*", donde describe un método para que los humanos podamos testear programas de IA al cual se lo denominó como el "*Test o Prueba de Turing*". Esta prueba consiste en una evaluación sobre la capacidad de una máquina para exhibir un comportamiento inteligente similar al de un ser humano o indistinguible de este.
1956	John McCarthy acuña el término "*Inteligencia Artificial*" en la conferencia de Dartmouth, la primera conferencia dedicada a la IA. En dicha conferencia, organizada por Marvin Minsky, John McCarthy y Claude Shanon se adoptó como punto de partida

	que "*Cualquier aspecto del aprendizaje u otra característica de la inteligencia puede en principio ser descrita con precisión de tal forma que se puede construir una máquina que la simule*". En la misma conferencia de Dartmouth se presentó el primer programa de IA, el "*Logic Theorist"*, escrito por Allen Newell, J.C. Shaw y Herbert Simon. Fue el primer programa diseñado deliberadamente para realizar razonamiento automatizado y se lo conoce como "*el primer programa de Inteligencia Artificial*". Eventualmente probaría 38 de los primeros 52 teoremas en los *Principia Mathematica* de Whitehead y Russell.
1958	John McCarthy inventa el lenguaje de programación LISP mientras se desempeñaba en el Laboratorio de Inteligencia Artificial del MIT, el cual había fundado un año antes junto a Marvin Minsky.
1959	Desarrollo del *General Problem Solver*, escrito por Allen Newell, J.C. Shaw y Herbert Simon, con la intención de construir una máquina capaz de resolver problemas de carácter general. Fue el primer programa de computadora en el que se separó el conocimiento de los problemas de su estrategia sobre cómo resolverlos y fue implementado en el lenguaje de programación IPL (*Information Processing Language*).
1960	Bernard Widrow y Marcian E. Hoff publican su trabajo *Adaptative Switching Circuits* (Widrow & Hoff, 1960), donde presentan ADALINE (*Adaptive Linear Elements*) utilizando el algoritmo LMS (*Least Mean Square*) como sistema de aprendizaje supervisado basado en perceptrones.
1962	Rosenblatt publica su libro *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms* (Rosenblatt, 1962), centrado en lo que luego se

	denominará Aprendizaje Supervisado. Rosenblatt plantea un mecanismo para reducir la diferencia entre la señal producida por la red neuronal y la señal esperada, a través de sucesivas iteraciones corrigiendo los pesos sinápticos. A este tipo de redes, con la regla de aprendizaje supervisado se las denomina "*perceptrones*".
1963	Edward A. Feigenbaum y Julian Feldman publicaron la primera colección de artículos sobre IA *Computers and Thought* (Feigenbaum & Feldman, 1963).
1969	Marvin Misnky y Seymour Papert publicaron el libro *Perceptrons. An Introduction to Computational Geometry* (Minsky & Papert, Percetrons. An Introduction to Computational Geometry, 1969), donde mostraron qué clases de funciones no eran computables por los perceptrones, paralizando por un par de décadas el avance en el conexionismo.
1972	Alain Colmerauer junto a Philippe Roussel desarrollaron el lenguaje PROLOG (*PROgrammation en LOGique*), cuyo objetivo inicial fue el de interpretar el lenguaje natural, en este caso el francés.
1975	Edward Shortliffe desarrolló en la Universidad de Stanford el sistema experto MYCIN, el cual tenía como objetivo el diagnóstico de enfermedades infecciosas de la sangre. También el sistema experto podía recetar los medicamentos personalizados para cada paciente. Marvin Minsky publicó su trabajo *A Framework for Representing Knowledge* (Minsky, A Framework for Representing Knowledge, 1974) desarrolla los marcos como forma de representación del conocimiento.
1986	Rumelhart, McClelland y el grupo PDP (*Parallel Distributed Processing Research Group*) desarrollaron el perceptrón multicapa y el algoritmo de aprendizaje

	por retro propagación (*Back Propagation*) del error, dando un renacimiento al conexionismo.

Tabla 7. Antecedentes sobresalientes relacionados con el nacimiento y desarrollo de la IA

Fuente: Autoría propia

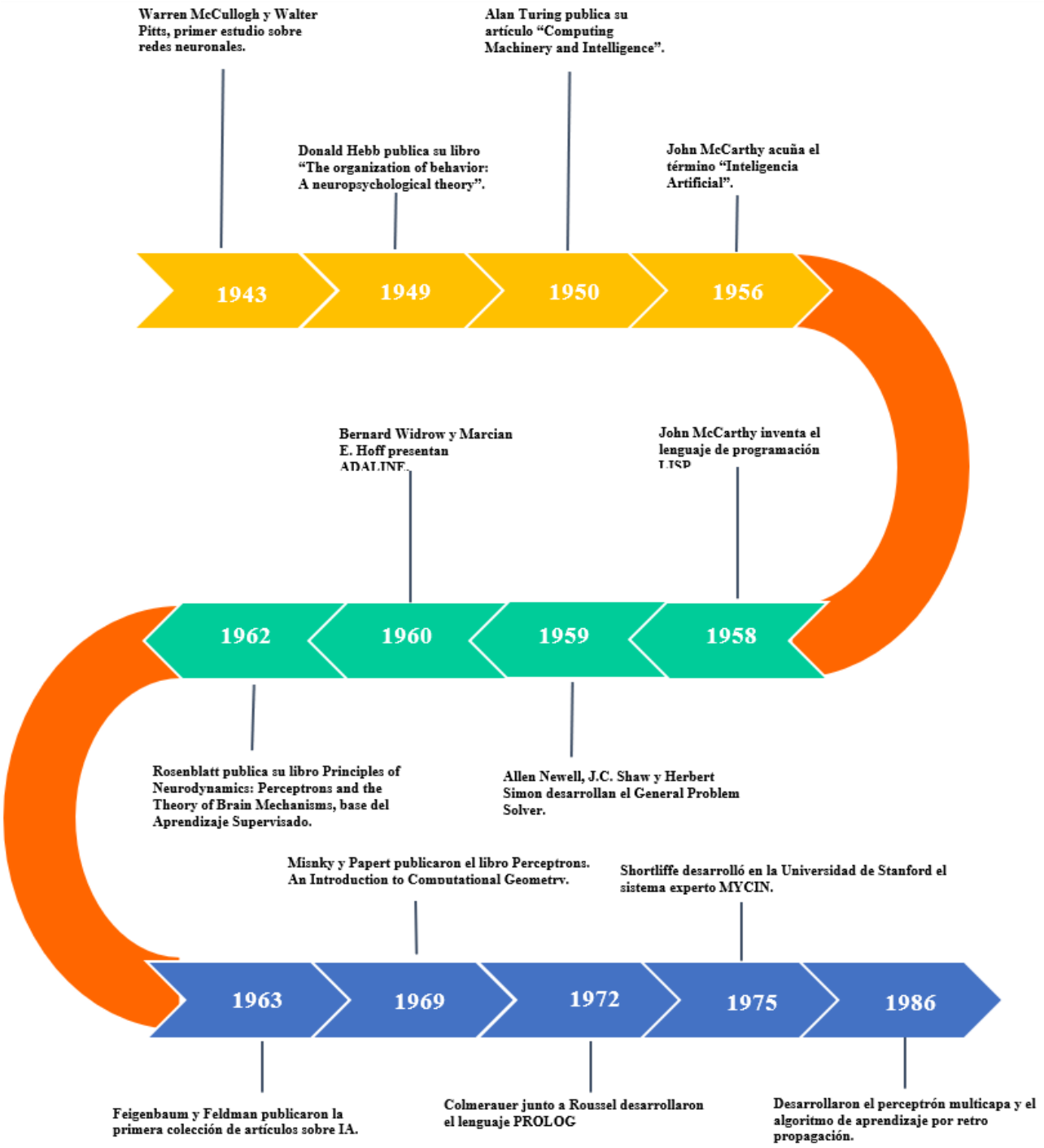

Figura 15. Línea de tiempo de los antecedentes relacionados con la IA

Fuente: Autoría propia

2.3.3. Clasificación de la IA

Cuando hablamos de tipos de IA, usualmente se utiliza una clasificación basada en dos factores; el primero basado en la tecnología utilizada por el sistema para resolver el problema, y el segundo la funcionalidad, las funciones y comportamiento utilizados.

2.3.3.1. IA de Tipo 1.

Está dada por la tecnología o capacidades y comprende tres categorías, tal como se muestra en la Figura 16. La Inteligencia Artificial Débil o ANI (*Artificial Narrow Intelligence* que es la forma más común de IA que se encuentra hoy en el mercado. Estos sistemas están diseñados para resolver un solo problema y podrían ejecutar una sola tarea realmente bien. Por definición, tienen capacidades limitadas, como recomendar un producto para un usuario de comercio electrónico o predecir el clima. Este es el único tipo de IA que existe en la actualidad. Son capaces de acercarse al funcionamiento humano en contextos muy específicos e incluso superarlos en muchos casos, pero solo lograr la excelencia en ambientes controlados y con un número acotado de parámetros. La segunda categoría se denomina Inteligencia Artificial General o AGI (*Artificial General Intelligence*, y es un concepto teórico, que se define como IA que tiene una función cognitiva a nivel humano en una amplia variedad de dominios, como el procesamiento del lenguaje, el procesamiento de imágenes, el funcionamiento computacional, el razonamiento, etc. Estamos lejos de construir un sistema AGI, el cual debería estar formado por miles de sistemas ANI que trabajan en conjunto, comunicándose entre sí para imitar el razonamiento humano. La tercera categoría de esta clasificación corresponde a la Superinteligencia Artificial o ASI (*Artificial Super Intelligence*), que, si bien puede parecer ciencia ficción, la ASI se ve como la progresión lógica de AGI. Un sistema de Superinteligencia Artificial (ASI) podría superar todas las capacidades humanas. Esto incluiría la toma de decisiones racionales e incluso cosas como hacer mejor arte y construir relaciones emocionales. Si bien la brecha entre AGI y ASI

sería relativamente estrecha, el largo viaje que tenemos por delante hacia AGI en sí hace que esto parezca un concepto futurista.

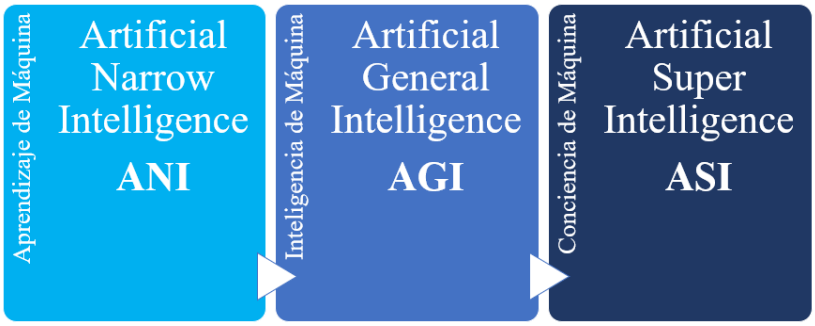

Figura 16. IA – Clasificación Tipo 1

Fuente: (Great Learning, 2021)

2.3.3.2. IA de Tipo 2**.**

Estas categorías están dadas en base a la funcionalidad. Son 4 grupos, de los cuales podríamos decir que hemos superado el primer grupo y estamos transitando por el segundo, mientras que los últimos dos son, por ahora teóricos. El primer grupo de esta categoría son las “Máquinas Puramente Reactivas” y son básicas en el sentido de que no almacenan "*recuerdos*" ni utilizan experiencias pasadas para determinar acciones futuras. Simplemente perciben el mundo y reaccionan a él. Deep Blue de IBM, que derrotó al gran maestro de ajedrez Kaspárov, es una máquina reactiva que ve las piezas en un tablero de ajedrez y reacciona a ellas. No puede referirse a ninguna de sus experiencias anteriores y no puede mejorar con la práctica. No pueden usar sus experiencias previas para administrar las tareas actuales que se les asignan. En resumen, estos se encuentran entre los tipos de sistemas de IA sin el poder de "*aprender*" cosas e implementar lo que aprenden en sus acciones futuras. El segundo grupo corresponde a las “Máquinas de memoria limitada”, las cuales combinan las habilidades para reaccionar y aprender de los datos anteriores. Estas máquinas pueden retener datos durante un breve período de tiempo, pero no pueden agregarlos a una base de datos

de experiencias o conocimiento. Muchos automóviles autónomos utilizan la tecnología de memoria limitada almacenando datos como la velocidad reciente de los automóviles cercanos, la distancia de dichos automóviles, el límite de velocidad y otra información que puede ayudarlos a navegar. Los sistemas de IA actuales, incluidos los que utilizan *Deep Learning*, están "*entrenados*" para reaccionar y aprender. Los sistemas de IA con memoria limitada tienen una gran cantidad de datos de entrenamiento lo que les permite 'aprender' de experiencias anteriores y adoptar lo que han aprendido para tomar decisiones en el futuro. Existen muchos modelos que apuntan a actualizar el conocimiento adquirido en el pasado para optimizar el comportamiento actual. Entre ellos se pueden mencionar el Aprendizaje por Refuerzo, LSTMs (*Long Short Term Memory*) e E-GAN (*Evolutionary Generative Adversarial Networks*). La IA de máquinas de memoria limitada trabaja básicamente de dos maneras para lograr este objetivo;

- El equipo continuamente entrena el modelo con datos nuevos
- El entorno del modelo se crea de manera que el mismo es automáticamente entrenado y renovado en base a su comportamiento y uso.

2.3.3.3. Teoría de la mente**.**

Los investigadores de la Teoría de la Mente esperan construir computadoras que imiten nuestros modelos mentales, formando representaciones sobre el mundo y sobre otros agentes y entidades en él. Uno de los objetivos de estos investigadores es construir computadoras que se relacionen con los humanos y perciban la inteligencia humana y cómo las emociones de las personas se ven afectadas por los eventos y el medio ambiente. Estos modelos están solo en sus fases iniciales. Los modelos actuales tienen una relación unidireccional con IA, por ejemplo, Alexa y Siri reaccionan ante cada comando. Si uno le grita enojado a Google Maps para que lo lleve en otra dirección, no ofrece apoyo emocional. Los campos de estudio que abordan este tema incluyen la Inteligencia Emocional Artificial y los desarrollos en la teoría de la Toma de Decisiones. El cuarto grupo se denomina "Conciencia de sí mismo". Las máquinas conscientes de sí mismas

son cosa de ciencia ficción, aunque muchos entusiastas de la IA creen que son el objetivo final del desarrollo de la IA. Si una máquina puede funcionar como lo hace una persona, por ejemplo, preservándose a sí misma, prediciendo sus propias necesidades y demandas y relacionándose con los demás como iguales, entonces una máquina puede volverse verdaderamente consciente de sí misma. Una inteligencia consciente de sí misma más allá de la humana tiene una inteligencia independiente y, probablemente, la gente tendrá que negociar los términos con la entidad que creó. Lo que sucede, bueno o malo, es una incógnita.

A continuación, en la Figura 17 se puede ver la relación entre IA, *Machine Learning* y *Deep Learning*, y en la Figura 18 algunas de las aplicaciones más utilizadas de las técnicas de *Machine Learning*.

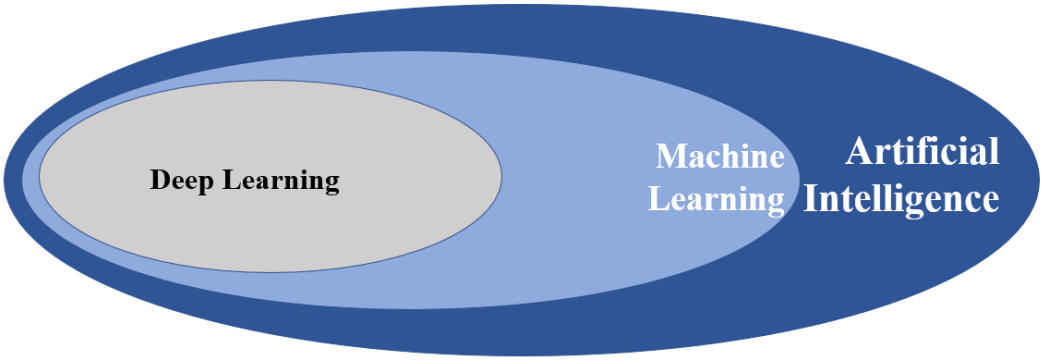

Figura 17. IA – Machine Learning y Deep Learning

Fuente: Autoría propia

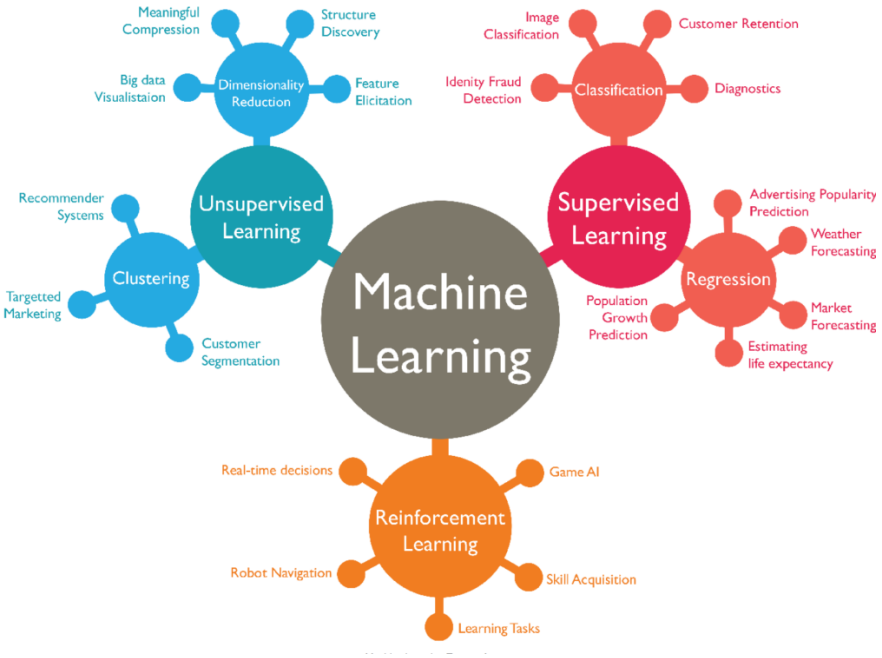

Figura 18. IA – Técnicas de Machine Learning

Fuente: Sitio Towards Data Science (Khadka, 2017)

2.3.4. Usos de la IA y futuro

La IA tiene el potencial real de transformar muchas industrias, con una amplia gama de posibles casos de uso. Lo que todas estas industrias y casos de uso diferentes tienen en común es que todos se basan en datos. A continuación, mencionaremos algunos de los casos de uso más utilizados hoy en día en diferentes industrias.

- Salud: En la Administración a través del soporte en las tareas diarias (transcripciones médicas, procesamiento de lenguaje natural, automatización de tareas, etc.), a través de la Telemedicina para casos que no requieren emergencias a través de la evaluación y diagnóstico preliminares, con el Diagnóstico Asistido a través de la visión computarizada

y la utilización de redes neuronales convolucionales capaz de identificar tumores y crecimiento de signos malignos para el paciente más rápidamente y con una tasa de error menor. También utilizando cirugías asistidas por robots o el monitoreo de signos vitales en tiempo real proporcionando recomendaciones de acciones basándose en el análisis de múltiples parámetros.

- Comercio Electrónico: La mayoría de las empresas de comercio electrónico han implementado técnicas de IA en diferentes casos de uso, como la recomendación de productos, Chatbots para dar ayuda en línea o capturar problemas comunes y liberar los canales que requieran recursos humanos, filtrado de Spams y noticias falsas a través de la utilización de procesamiento de lenguaje natural, detección proactiva de sentimiento del cliente a través del monitoreo de los comentarios en redes sociales, optimización de búsquedas de usuarios basándose en cientos o miles de parámetros relacionados con consumos, comportamiento de navegación, estadísticas, etc.. También la utilización en el soporte de la cadena de aprovisionamiento a través de la predicción de demanda por región.

- Recursos Humanos: Mejorando la cultura de trabajo a través de la asignación de los recursos a diferentes proyectos dependiendo de sus preferencias, o en el reclutamiento, realizando análisis de miles de C.V. y utilizando procesamiento de lenguaje natural para extraer contenidos o capacidades requeridas para cada posición.

- Robótica: Si bien la robótica se fue desarrollando, incluso antes de la revolución e la IA, esta última ha impulsado esta industria mejorándola sustancialmente, en especial en la

eficiencia de algunas aplicaciones como el ensamblaje, el servicio al cliente, el empaquetamiento, etc.

- Vehículos autónomos: Esta área se ha desarrollado ampliamente en los últimos años y promete reducir considerablemente los accidentes viales y los costos de transporte.

- Fraude y Ciberseguridad: Estas áreas han hecho un uso intensivo y se encuentran en constante desarrollo nuevas técnicas para utilizar la IA para la prevención de fraudes y ciberataques. Algunos de estos ejemplos son el análisis de patrones de comportamiento, los controles biométricos y la interacción con elementos de red.

La IA se ha convertido en la próxima gran novedad en el campo de la tecnología y las organizaciones de todo el mundo están presentando innovaciones revolucionarias en IA y aprendizaje automático. No solo está impactando el futuro de cada industria y cada ser humano, sino que también ha actuado como el principal impulsor de tecnologías emergentes como *big data*, robótica e IoT.

Se estima que la IA creará 133 millones de nuevos puestos de trabajo para el año 2022.

En la Figura 19 se muestran algunas de las aplicaciones más comunes de la IA.

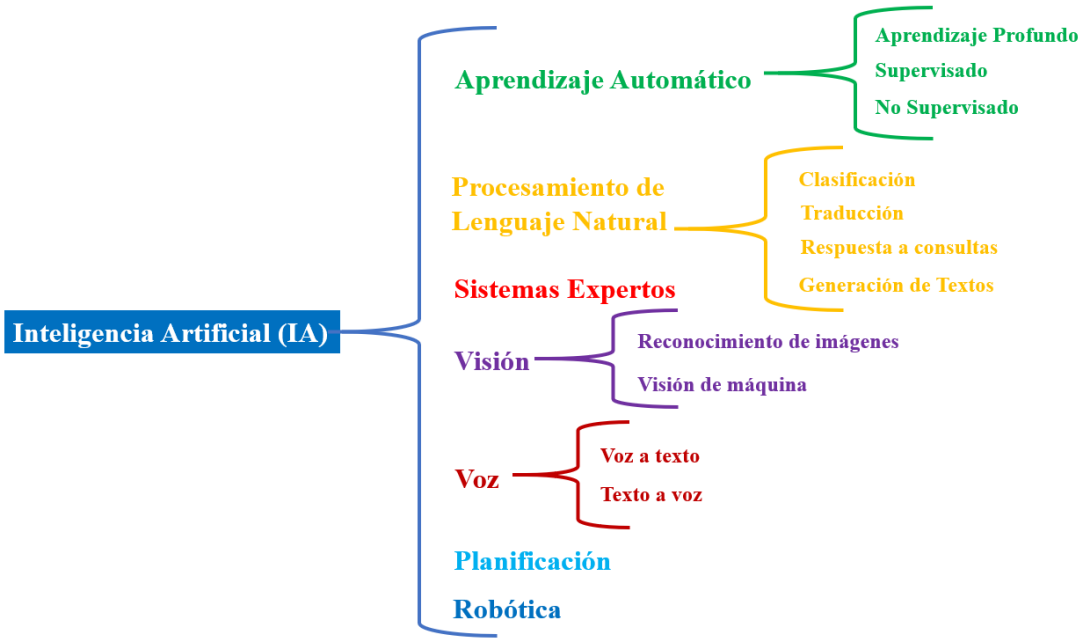

Figura 19. Ejemplos de aplicación de la IA

Fuente: Autoría propia

2.4. Importancia para la Ciberdefensa

2.4.1. Implicancia de los Malware en la Ciberdefensa

Stuxnet no fue solo un nuevo virus o gusano, sino que inició una nueva era de malware. Este virus cambió el significado del *malware* y sus objetivos. Uno estaba acostumbrado a escuchar acerca de virus molestos o que robaban datos bancarios o de tarjetas de crédito, pero fue la primera vez que escuchamos sobre daños a edificios, maquinas o a muertes de personas causadas por un virus. Eso fue Stuxnet (Thabet, 2011, pág. 2).

Stuxnet utilizaba hasta 4 vulnerabilidades *zero-day* en Windows y utilizaba varias técnicas para evitar ser detectado por los antivirus y logró dañar los reactores nucleares iraníes y sus máquinas, infectando

los Controladores Lógicos Programables o PLC (sigla en inglés para *Programmable Logic Controller*) que los controlaban.

Este gusano fue creado con el objetivo de sabotear el Programa nuclear iraní, utilizando, una vez instalado en una PC, *passwords* por defecto de Siemens para ganar acceso a los sistemas de controlan los PLC. Una vez infectado el sistema, el *malware* operaba en dos etapas, primero enviaba la configuración de los equipos a un Centro de Comando y Control, y luego le permitía al atacante modificar la misma de manera de definir como se quería que se comporte el equipo supervisado. De esta manera afectó a las centrifugadoras cambiando constantemente la velocidad para reducir significativamente su vida útil.

Si bien el caso de Stuxnet no fue el primero, marcó un punto de inflexión en la valoración de estas ciberarmas y su impacto dentro del quinto dominio.

A continuación, en la Tabla 8, se mencionan algunos de los casos más recientes relacionados con ciberataques a infraestructuras críticas:

Fecha	Descripción
20-Oct-2019	Las autoridades de la India informaron que una de sus centrales nucleares había sido víctima de un ciberataque avanzado (Engineering & Technology, 2019). El ataque de *malware* en la Planta de Energía Nuclear de Kudankulam (KKNPP), identificado por primera vez el 4 de septiembre, se ha atribuido al actor estatal con vinculaciones con el régimen de Pyongyang conocido como Lazarus[12]. Si bien el *malware* no se dirigió a los sistemas de control críticos de la central, quedando su afectación limitada a una red de administración, el ataque pone en evidencia la fragilidad de las infraestructuras críticas a ataques de esta naturaleza. "*El malware utilizado en el ataque a KKNPP, Dtrack (que también se usó para propagar los ataques de ransomware WannaCry en 2017 y cuyo nombre ha sido asignado por Kaspersky), es una herramienta de monitorización y recopilación de datos del sistema infectado que permite escanear redes y sistemas en busca de posibles vulnerabilidades que puedan ser explotadas. Este malware implant saltó a la fama en septiembre de este año tras haberse detectado su presencia en un vector de ataque a cajeros automáticos indios.* *Así pues, los atacantes pudieron establecer una puerta trasera en la red del KKNPP, siendo un paso previo a un ataque posteriores, ya que garantizaría la presencia (persistente) en las redes de la central. Una vez desplegado, Dtrack permite aprovechar rápidamente vulnerabilidades o puntos ciegos en las defensas de seguridad, como los puertos no seguros; sistemas sin parchear o desactualizados; o nuevos dispositivos IoT no administrados. Todos estos plantean riesgos significativos de ciberseguridad en el sector de servicios públicos*." (Hernández Lorente, 2019, pág. 8).

[12] Grupo Lazarus: El Grupo Lazarus es un grupo que se dedica a cometer delitos informáticos compuesto por un número desconocido de individuos. Si bien no se sabe mucho sobre el Grupo Lazarus, los investigadores les han atribuido muchos ataques cibernéticos en la última década.

1-Dic-2020	El grupo *Unidentified TEAM* relacionado con el gobierno iraní publicó un video sobre una brecha de seguridad en un sistema de gestión de un reservorio de agua israelí que contaba con acceso a internet sin medidas de seguridad apropiadas (Kovacs, 2020). Si bien el ataque no produjo consecuencias graves, el mismo grupo se adjudicó un ataque al sitio educativo gubernamental de Texas, ambos como reacción por el asesinato del científico nuclear iraní Mohsen Fakhrizadeh, del cual responsabilizan a los gobiernos de Estados Unidos e Israel.
5-Feb-2021	Atacantes no identificados accedieron al sistema SCADA de una planta de tratamiento de agua potable de Florida, USA. Los atacantes aprovecharon vulnerabilidades de Windows 7, *passwords* débiles y el uso compartido de la aplicación de acceso remoto *TeamViewe*r para acceder remotamente al sistema de tratamiento de agua y cambiar la composición de la misma. El cambio en la composición fue detectado por otros controles posteriores y se evitó que sea distribuida a los ciudadanos. Se cree que este ataque pudo ser ejecutado por un empleado o exempleado disconforme (Goodin, 2021).
28-Feb-2021	Se publicó un informe de la firma Recorded Future's Insikt Group (Insikt Group, 2021), donde indicaba que las tácticas, técnicas y procedimientos utilizados por el grupo *RedEcho*, detrás de los sucesivos ataques al sector energético indio durante el último año, permitían asociarlos a los utilizados por grupos como *APT41* y *Tonto Team*, con relaciones confirmadas con el gobierno chino. En el informe se menciona que luego de los enfrentamientos fronterizos de mayo de 2020, el ministro de relaciones exteriores de India indicó que la confianza entre ambos países había sido deteriorada, dando lugar a un fuerte incremento de operaciones cibernéticas para brindar a los países una potente capacidad asimétrica para realizar espionaje y posicionamiento previo dentro de las redes por razones potencialmente disruptivas.

7-May-2021	Se produjo un ciberataque de *ransomware* al oleoducto *Colonial* por el grupo *Darkside*. Los atacantes habrían obtenido acceso tanto a los sistemas de TI como a los sistemas de OT de la empresa de oleoductos objetivo. El impacto de este ataque, afectó al 45% de provisión de combustibles refinados que se envía de Texas al puerto de Nueva York, obligando a la Administración Federal de Seguridad de Auto transportistas a emitir un comunicado de emergencia para permitir el transporte terrestre de combustibles, para evitar el desabastecimiento (Sharwood, 2021).
3-Ago-2021	Cybereason, un sitio especializado en ciberseguridad, dio a conocer detalles de un triple ataque de grupos cercanos al ejército chino contra proveedores de redes celulares en el sudeste asiático. Yonatan Striem-Amit, director de tecnología y cofundador de Cybereason, dijo en una entrevista a SecurityWeek: "Descubrimos y tenemos evidencia de que los grupos avanzados chinos han estado usando los días cero de hafnio desde al menos 2017". La motivación para estos objetivos tiene dos aspectos principales, el primer evadir los controles que interponen las empresas de telecomunicaciones para evitar lo ciberataques y por otro lado ejecutar tareas de espionaje al acceder a los detalles de las comunicaciones (Cybereason Nocturnus, 2021).

Tabla 8. Ejemplos de aplicación ciberataques a infraestructuras críticas

Fuente: Autoría propia

El crecimiento en esta modalidad de ataques y las graves consecuencias de los mismos remarcan la importancia de la detección temprana de este tipo de ciber ataques es crucial para efectivizar una política de Ciberdefensa sólida.

2.4.2. Importancia de las Redes IoT en la Ciberdefensa

Tal como se ha mencionado anteriormente, la convergencia entre redes OT y redes IT, ha llevado las amenazas conocidas de los sistemas operativos y aplicaciones comúnmente utilizados en ambientes empresariales, a los ambientes operacionales e industriales, que históricamente solían utilizar soluciones específicas y muy poco difundidas. Esto ha ampliado la superficie de ataque sobre las redes informáticas que dan soporte a las infraestructuras críticas.

Adicionalmente a esto, la incorporación de sistemas de medición y monitoreo de gestión remota ha terminado por conectar estas redes a Internet brindando una alternativa de entrada para los atacantes, la cual no estaba disponible en este tipo de infraestructuras en el pasado.

Estas dos características, sumadas al avance tecnológico de las ciber amenazas, que son cada vez más complejas, han impulsado el desarrollo creciente de ciber armas focalizadas en infraestructuras críticas para ser utilizadas en la guerra que se libra en el Quinto Dominio.

Entre las infraestructuras críticas cuyos los sistemas, redes y activos, ya sean físicos o virtuales, son considerados vitales para una nación podemos mencionar las bases instalaciones de investigación y desarrollo, infraestructura naval, fuerza aérea, entre otras, de la Defensa; redes, hardware, software, interconexión a Internet, instalaciones gubernamentales como el parlamento, embajadas, educativas, juzgados; el sector energético de generación, distribución y transporte, servicios públicos, gasoductos, refinerías de gas natural; las centrales nucleares, centros de investigación y ensayos, residuos; el suministro de agua potable, riego, tratamiento de aguas residuales; el sector de transporte, ya sea terrestre (carreteras, ferroviario, postal), aéreo o marítimo; las comunicaciones, satélites, tele puertos, tendidos de fibra; los servicios de emergencias como las fuerzas de seguridad, bomberos, servicios médicos; la atención sanitaria como ser los hospitales, bancos de sangre, reservas de medicamentos, farmacia; los servicios financieros como bancos, bolsas de valores; las infraestructuras de fabricación crítica como metales, maquinarias, transporte; el sector alimenticio; las represas, las instalaciones comerciales como hoteles, estadios, centros

comerciales y el sector químico como las petroquímicas, industrias de fertilizantes, etc. En la Figura 20 se muestra un diagrama con ejemplos de infraestructuras críticas.

Figura 20. Ejemplos de infraestructuras críticas

Fuente: Autoría propia

A modo de ejemplo de la importancia de estas amenazas en la Ciberdefensa, como consecuencia de los incidentes identificados este último año, la Administración de Seguridad en el Transporte del Departamento de Seguridad Nacional (TSA) de Estados Unidos, ha publicado recientemente nuevos requerimientos de Ciberseguridad para dueños y operadores de oleoductos críticos. Citando al secretario de Seguridad Nacional Alejandro N. Mayorkas:

"El panorama de la ciberseguridad está en constante evolución y debemos adaptarnos para abordar las amenazas nuevas y emergentes," ..." El reciente ataque de ransomware en un importante

oleoducto demuestra que la ciberseguridad de los sistemas de oleoductos es fundamental para nuestra seguridad nacional. El DHS continuará trabajando en estrecha colaboración con nuestros socios del sector privado para respaldar sus operaciones y aumentar la resistencia de la infraestructura crítica de nuestra nación." (Homeland Security, 2021).

La Directiva de Seguridad requerirá que los propietarios y operadores de oleoductos críticos informen los incidentes de ciberseguridad potenciales y confirmados a la Agencia de Seguridad de Infraestructura y Ciberseguridad (CISA) del DHS y que designen un Coordinador de Ciberseguridad, que estará disponible las 24 horas del día, los siete días de la semana. También requerirá que los propietarios y operadores de oleoductos críticos revisen sus prácticas actuales, así como que identifiquen las brechas y las medidas de recuperación relacionadas para abordar los riesgos cibernéticos e informar los resultados a la TSA y CISA.

3. Estado del Arte y Trabajos Relacionados

La IoT facilita la integración masiva de dispositivos en las redes con la finalidad de proveer datos para ser consumidos por las múltiples aplicaciones que aparecen a diario y que van desde ciudades inteligentes, infraestructura, agricultura inteligente, servicios de salud, autos conectados, etc. Asegurar las redes IoT es un desafío que debemos afrontar para proteger nuestro entorno y nuestra propia seguridad física en muchos casos.

En esta sección mencionaremos las tendencias en cuanto a trabajos realizados para proteger estas redes, especialmente haciendo uso de técnicas de IA.

A continuación, en la Tabla 9 se muestran algunos de los ataques más críticos en las redes IoT:

Tipo de Ataque	Descripción
Denial of Service (DoS)	Denegación de servicios. El atacante sobrecarga el servidor con peticiones redundantes con el objetivo de evitar que el dispositivo IoT pueda dar respuesta a peticiones de otros clientes.
Man-in-the-Middle	También conocido como MitM, es un ataque de intermediario, donde el atacante interviene una comunicación entre dispositivos IoT haciéndole crear a ambas partes que están comunicándose directamente.
Spoofing	El atacante se hace pasar por un dispositivo IoT legítimo, obteniendo acceso ilegal y desde ahí ejecuta ataques de DoS y MitM para hacer caer los servicios del servidor comprometido.
Jamming	Es un tipo de ataque de DoS, donde los nodos comprometidos emiten señales falsas con la finalidad de interferir las comunicaciones legítimas.
Eavesdropping	Se da cuando un intruso en la red roba, modifica y elimina información transmitida en la red.
Software Attacks	Cuando software malicioso o Malware, como virus, gusanos, troyanos, etc. roban información sensible de la red o causan daño en la misma produciendo degradación de los sistemas que componen la red IoT.

Tabla 9. Ataques más críticos a redes IoT

Fuente: Autoría propia

Existen diferentes técnicas basadas en *Machine Learning*, que permiten dar soluciones a la detección de este tipo de ataques. A continuación, se describen los mencionados en el trabajo (Malik & Chauhan, 2020).

3.1. Técnicas de Aprendizaje Supervisado o Supervised Learning

Requieren el entrenamiento del modelo utilizando datos etiquetados. Los algoritmos más utilizados en esta categoría son:

- Logistic Regression (LR)
- Support Vector Machine (SVM)
- K-Nearest Neighbor (K-NN)
- Naïve Bayes (NB)
- Decision Tree (DT)
- Random Forest (RF)
- Deep Neural Network (DNN)
- Neural network

3.2. Técnicas de Aprendizaje no Supervisado o Unsupervised Learning

Utilizan datos sin etiquetar y se basan en el agrupamiento de estos en base a características o comportamientos similares. Entre los algoritmos más conocidos podemos mencionar:

- Singular Value Decomposition
- Multivariate Correlation Analysis
- K-means clustering

3.3. Técnicas de Aprendizaje por Refuerzo o Reinforcement Learning (RL)

Estas técnicas aprenden observando el entorno para auto entrenarse. Algunos de los algoritmos que forman parte de esta técnica son:

- Q-Learning
- Deep Q Learning
- Post-decision state
- Dyna-Q

3.4. Técnicas de Aprendizaje Profundo o Deep Learning (DL)

Estas técnicas, inspiradas en redes neuronales artificiales, se componen de redes de neuronas, las cuales se conectan a través conexiones activadas por funciones de peso, las cuales activan las conexiones con otras neuronas dependiendo del estímulo recibido en la entrada a la misma. Las redes neuronales se forman componiendo múltiples capas. Algunos de los algoritmos más utilizados en estas técnicas son:

- Recurrent Neural Network (RNN)
- Generative Adversarial networks (GAN)
- Convolutional Neural Network (CNN)

A continuación, en la Tabla 10, se resume algunos de los trabajos realizados en los últimos años utilizando algunas de estas técnicas para la detección de ataques en redes de IoT (Malik & Chauhan, 2020):

Autor y Año	Tipo de Ataque	Algoritmo Utilizado
(Alam & Vuong, 2013, págs. 663-669)	Android Malware	Random Forest (RF)
(Kim, Ham, Kim, & Choi, 2014)	Android Malware	Support Vector Machine
(Haddadpajouh, Javidan, Khayami, & Dehghantanha, 2016)	User2Remote Remote2Local	Naïve Bayes K-Nearest Neighbor
(Cañedo & Skjellum, 2016)	Man in the Middle	Artificial Neural Network
(Diro & Chilamkurti , 2018)	DoS Probe R2R U2R	Deep Neural Network
(Bohadana, y otros, 2018)	IoT Botnets	Deep Autoencoders
(ChIoannou & Vassiliou, 2019)	Selective forward Blackhole Sinkhole	Support Vector Machine

(Hasan, Islam, Zarif, & Hashem, 2019)	DoS Malicious Control Data Type Probing Spying Scan Wrong Setup Malicious Operation	Logistic Regression Decision Tree Support Vector Machine Artificial Neural Network Random Forest
(Zaca, Kharroub, & Abualsaud, 2020)	IoT Malware	Convolutional Neural Networks
(Susanto, Stiawan, Arifin, Yazid, & Budiarto, 2020)	IoT Botnet Malware	AdaBoost Decision Tree Random Forest Naïve Bayes.

Tabla 10. Resumen de trabajos relacionados con técnicas de detección de ataques en redes IoT

Fuente: Securing the Internet of Things using Machine Learning: A Review (Malik & Chauhan, 2020)

4. Ejemplo Práctico de Implementación de un Caso de Uso

4.1. Desafíos

Con el crecimiento de los ataques a redes de computadores, una solución que se ha impuesto es la implementación de sistemas de detección de intrusiones de redes (NIDS de su sigla en inglés). Los NIDS son dispositivos de hardware o software que, desplegados en puntos estratégicos de la red, permiten monitorear el tráfico en busca de actividades maliciosas. Cuando una actividad sospechosa es identificada se envía una alerta al administrador de la red para que tome las acciones necesarias para mitigar el riesgo potencial de dicha actividad.

Usualmente estos dispositivos se conectan en un único punto de la red, por ejemplo, el Gateway de acceso a Internet, lo cual permite controlar lo que entra y sale de la red, pero no permite conocer lo que está pasando dentro de la misma. Para poder conocer las actividades dentro de la red es necesario un NIDS distribuido, que permita desplegarse en diferentes elementos de red dentro de la red y así tener un control del tráfico que se cursa en la misma.

En los últimos años las redes neuronales han tomado un papel importante como soluciones para sistemas de detección de intrusiones de red dada su capacidad para aprender patrones de comportamientos complejos permitiéndoles discriminar entre tráfico normal y tráfico anómalo dentro de una red. Sin embargo, uno de los puntos en contra está relacionado con la demanda intensiva de recursos que necesitan las mismas, tanto para entrenar los modelos como para su ejecución. Por otro lado, la mayoría de las soluciones basadas en redes neuronales se basan en modelos supervisados, lo que implica la participación de expertos para la clasificación de los conjuntos de datos de entrenamiento, lo cual lo hace muy costoso.

Para extender una estrategia de despliegue distribuida es necesario que la cantidad de NIDS pueda escalar al mismo nivel que la red, y de manera económicamente sustentable. Una opción de que esto sea

factible es realizar el despliegue de los NIDS directamente dentro de enrutadores de bajo costo y esto no es posible con modelos de clasificación como los previamente descriptos por los siguientes motivos:

- Procesamiento fuera de línea: Para entrenar los modelos supervisados es necesario contar con los conjuntos de datos de entrenamiento localmente, lo cual es imposible ya que el volumen de datos es muy alto y el entrenamiento se debería efectuar fuera de línea y luego actualizar el modelo en cada instancia. Esto generaría una demora en la sincronización de los elementos de red y una sobrecarga del tráfico, a la vez que haría muy dificultoso el escalamiento.
- Aprendizaje supervisado: El etiquetado del tráfico toma tiempo y es costoso y la normalidad o no del mismo es altamente dependiente de la localización dentro de la red.
- Alta complejidad: La complejidad de las redes neuronales crece exponencialmente con el número de neuronas, lo cual restringe la cantidad de características que pueden ser utilizadas por el impacto en la necesidad de recursos.

En respuesta a las restricciones arriba mencionadas se concluye que un NIDS basado en redes neuronales que sea desplegado y entrenado de manera distribuida, deberá contar con las siguientes características:

- Procesamiento en línea: Luego de que el modelo se entrena o se ejecuta al procesar una instancia, la misma se descarta.
- Aprendizaje no supervisado: No se utilizan etiquetas para determinar si el tráfico es normal o anormal.
- Baja complejidad: La tasa de procesamiento de paquetes debe ser mayor que la tasa de recepción de estos. Se debe garantizar que no existan paquetes esperando en la cola para ser procesados.

4.2. Kitsune: detección de intrusiones en redes

Para cumplir con los requerimientos mencionados en el apartado anterior, presentaremos un NIDS, un modelo de detección de intrusiones de red en línea, basado en redes neuronales, no supervisado y de bajo consumo de recursos denominado Kitsune (Mirsky, Doitshman, Elovici, & Shabtai, 2018). Kitsune está compuesto por un conjunto de pequeñas redes neuronales, denominadas autoencoders, las cuales son entrenadas para reconstruir patrones de tráfico.

En la Figura 21 se puede ver un esquema del algoritmo de detección de anomalías de Kitsune, denominado KitNet. Las características que identifican una instancia se agrupan y cada grupo de características se envían a la primera capa de neuronas de cada autoencoder que componen el conjunto. Cada autoencoder intenta reconstruir la instancia original y computa la reconstrucción utilizando el Error Cuadrático Medio como medida de valoración. Estos valores se envían a la capa de salida, la cual toma los errores cuadráticos medios de cada autoencoder y los valora, generando una puntuación, la cual nos indicara qué tan alejada esta la reconstrucción del valor inicial y dependiendo de esto es que se detectará una anomalía.

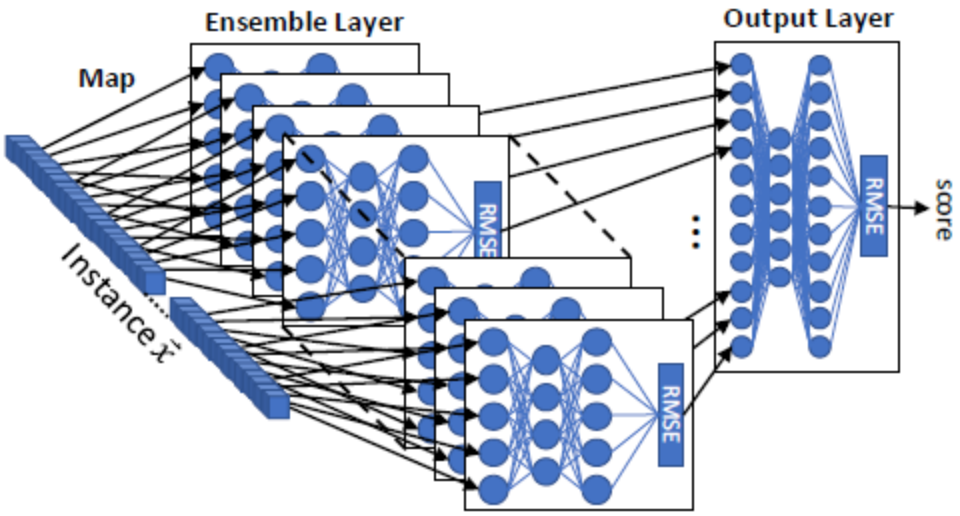

Figura 21. Algoritmo de detección de anomalías Kitnet

Fuente: (Mirsky, Doitshman, Elovici, & Shabtai, 2018)

Los autoencoders se han utilizado para diferentes actividades de aprendizaje automático, como la generación de nuevo contenido, la eliminación de ruido en imágenes, etc.

En general, un autoencoder entrenado en un conjunto de datos X tiene la posibilidad de reconstruir instancias desconocidas de la misma distribución que X, por lo tanto, podemos esperar que al intentar reconstruir una instancia que no posee la misma distribución, el error sea alto.

En la Ecuación 1 podemos ver cómo sería la estimación del error entre una instancia ***v*** y el resultado ***y***, donde ***n*** es la dimensionalidad del vector de características de entrada.

$$\mathrm{RMSE}\left(\vec{x},\vec{y}\right)=\sqrt{\frac{\sum_{i=1}^{n}\left(x_i-y_i\right)^2}{n}}$$

Ecuación 1. Cálculo del error de reconstrucción del autoencoder para el vector v

Fuente: (Mirsky, Doitshman, Elovici, & Shabtai, 2018)

4.2.1. Detección de anomalías

Sea ***U*** el umbral de anomalía, con un valor inicial de -1 y ***S*** un parámetro de sensibilidad, definido entre [1, +∞), podemos esperar que un autoencoder detecte una anomalía siguiendo los siguientes pasos:

Para la fase de entrenamiento: para cada instancia de ***x*** en el conjunto de datos de ***X.***

1. Ejecutar el autoencoder con el vector de características de ***x***, obteniendo el error asociado ***s.***
2. Si ***s*** > que ***U*** entonces, a ***U*** se le asigna el valor ***s.***
3. Se actualizan los parámetros del modelo en base a lo aprendido de la instancia x_i

Para la fase de ejecución: cuando una instancia desconocida x llega.

1. Se ejecuta el modelo y se estima el error asociado ***s.***
2. Se determina si es anómalo o no evaluando si (***s*** >= ***U*********S***) y en caso de cumplirse dicha condición se genera la alerta.

4.2.2. Componentes de software

Kitsune está compuesto por 4 módulos principales:

- Packet Capturer: este componente es el encargado de capturar los paquetes de red y se utilizan librerías externas para tal fin como tshark [13].
- Packet Parser: este módulo es responsable de realizar el parsing de los paquetes capturados por el Packet Capturer y también utiliza librerías externas como Packet++[14].

[13]Se puede consultar el manual de tshark en: https://www.wireshark.org/docs/man-pages/tshark.html

[14]El proyecto Packet++ se puede encontrar en GitHub: https://pcapplusplus.github.io/

- Feature Extractor (FE): este componente es el encargado de extraer ***n*** características del paquete ***P*** recibido, creando un vector de ***x*** de características, que representaran a la instancia ***P*** y al canal utilizado.
- Feature Mapper (FM): en este módulo, tomando como entrada el vector de características ***x*** se crea un conjunto de instancias más pequeñas ***v***, las cuales se agruparán en base a la relación de las características entre sí, y cada subgrupo se enviará a un autoencoder. Es también responsabilidad de este módulo aprender las relaciones entre ***x*** y ***v***.
- Anomaly Detector (AD): este componente será responsable de detectar paquetes anormales en base a la representación de estos dada por ***v***.

Kitsune tiene un parámetro de entrada m, el cual indica la mayor cantidad de características para cada autoencoder; a mayor cantidad de parámetros más eficiente será la detección, pero mayor tiempo y recursos consumirá, y viceversa; a menor cantidad de parámetros, más rápido se ejecutará el modelo, pero también menor será su efectividad.

Los módulos FM y AD tienen dos modos de ejecución, modo entrenamiento y modo ejecución, pasando del modo de entrenamiento al de ejecución luego de un tiempo definido de antemano por el usuario. En el modo entrenamiento se actualizan los parámetros internos del modelo y no se genera ninguna salida, mientras que en el modo de ejecución no se actualiza ningún parámetro interno, generándose la salida correspondiente a cada módulo.

Resumiendo, el proceso de funcionamiento de Kitsune sería el siguiente: el Packet Capturer captura los paquetes de red y los envía en formato binario al Packet Parser, el cual los decodifica y envía la información al módulo FE (Feature Extractor); el FE utiliza esta información para extraer más de 100 estadísticas (por ejemplo, fecha y hora de recepción del paquete, tamaño del mismo, direcciones IP de origen y destino, protocolo utilizado, canales utilizados, etc.), las cuales formarán el vector ***x*** que representará al paquete de datos bajo análisis. El vector ***v*** es enviado al módulo FM; si está en modo

entrenamiento utilizará ***x*** para aprender las relaciones entre las características y su correspondiente agrupamiento, creando grupos de un máximo de ***m*** elementos. Una vez finalizado el entrenamiento este mapeo será pasado al módulo AD para que se cree la arquitectura de ensamblado de autoencoders. En caso de estar en modo de ejecución, el mapeo aprendido se utiliza para dividir ***x*** en instancias menores ***v***, las cuales se envían al módulo AD. Por último, el módulo AD recibe ***v***; en caso de estar ejecutándose en modo de entrenamiento utiliza ***v*** para estimar el valor máximo de error ***U***, almacenando el mismo, el cual será utilizado como parte de la condición para detectar anomalías. En caso de estar en modo ejecución, utiliza ***v*** y lo ejecuta a través de todas las capas, obteniendo el error, y si el mismo es mayor que ***U*S*** (parámetro de sensibilidad definido por el usuario), entonces una alerta es generada. En la Figura 22 se puede ver un diagrama de la arquitectura funcional de Kitsune.

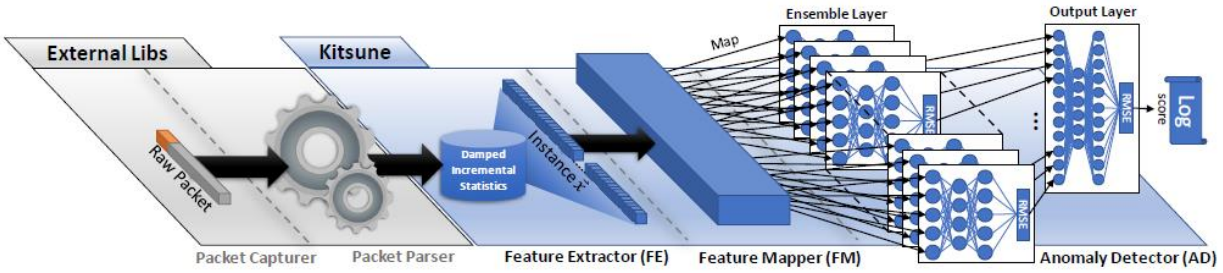

Figura 22. Arquitectura de Kitsune

Fuente: (Mirsky, Doitshman, Elovici, & Shabtai, 2018)

4.2.3. Evaluación de desempeño

En esta sección describiremos la evaluación de Kitsune descripta en el documento donde se presentó inicialmente esta herramienta (Mirsky, Doitshman, Elovici, & Shabtai, 2018), en donde describiremos los sets de datos utilizados, la arquitectura desplegada, los tipos de ataques y los resultados obtenidos y su comparación con otras herramientas.

4.2.3.1. Set de Datos

En el trabajo realizado se evaluaron las capacidades Kitsune para detectar ataques en una red de cámaras IP de vigilancia, desplegadas en dos sitios, cada uno con 4 cámaras, conectados ambos sitios al DVR a través de una red privada virtual. Los usuarios del sistema de monitoreo se pueden conectar al DVR para ver el streaming de video a través de un túnel utilizando también una red privade virtual. La arquitectura mencionada se puede ver en la Figura 23.

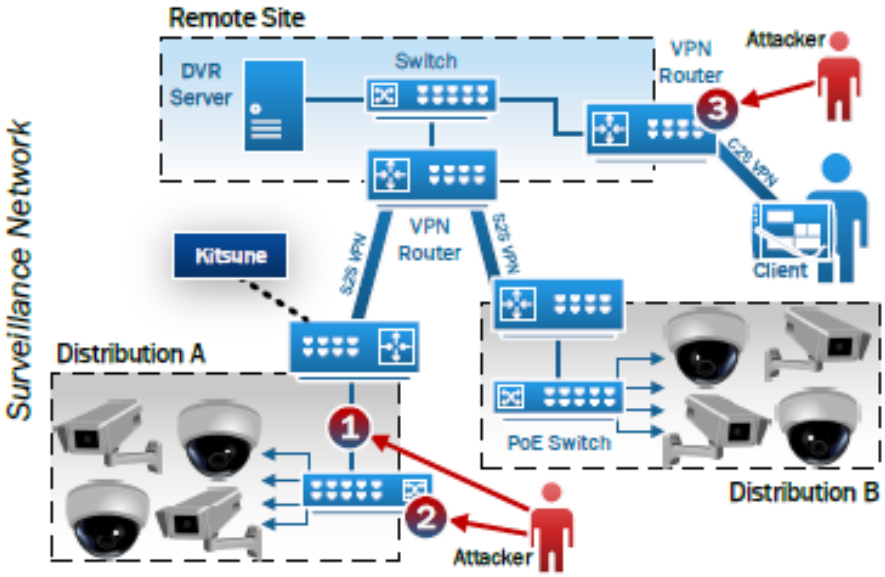

Figura 23. Arquitectura red de video vigilancia

Fuente: (Mirsky, Doitshman, Elovici, & Shabtai, 2018)

Si bien existen diferentes tipos de ataques que pueden perpetrarse sobre una red de video vigilancia, los más críticos son los que afectan la disponibilidad o la integridad de la información que suministra la misma. Para esta evaluación el autor decidió evaluar la detección sobre ataques tipo SYN Flood (afectando la disponibilidad del servicio) y MiTM (afectando la integridad a través de la captura e inyección de tráfico en tiempo real).

El segundo escenario de evaluación de Kitsune se realizó en una red Wi-Fi en donde se desplegaron 9 dispositivos IoT (un termostato, un monitor de bebés, una cámara web, dos timbres electrónicos y 4 cámaras de seguridad de bajo costo) y 3 computadoras. En la Figura 24 se puede ver la arquitectura IoT.

Figura 24. Arquitectura red IoT

Fuente: (Mirsky, Doitshman, Elovici, & Shabtai, 2018)

Para la evaluación de Kitsune en esta red el autor decidió infectar una de las cámaras con el malware Mirai.

En la Tabla 11 se puede ver un resumen de los sets de datos de los ataques utilizados en esta evaluación. La columna Violación indica la violación de seguridad por parte del atacante, confidencialidad (C), integridad (I) o disponibilidad (A).

Attack Type	Attack Name	Tool	Description: *The attacker…*	Violation	Vector	# Packets	Train [min.]	Execute [min.]
Recon.	**OS Scan**	Nmap	*…scans the network for hosts, and their operating systems, to reveal possible vulnerabilities.*	C	1	1,697,851	33.3	18.9
	Fuzzing	SFuzz	*…searches for vulnerabilities in the camera's web servers by sending random commands to their cgis.*	C	3	2,244,139	33.3	52.2
Man in the Middle	**Video Injection**	Video Jack	*…injects a recorded video clip into a live video stream.*	C, I	1	2,472,401	14.2	19.2
	ARP MitM	Ettercap	*…intercepts all LAN traffic via an ARP poisoning attack.*	C	1	2,504,267	8.05	20.1
	Active Wiretap	Raspberry PI 3B	*…intercepts all LAN traffic via active wiretap (network bridge) covertly installed on an exposed cable.*	C	2	4,554,925	20.8	74.8
Denial of Service	**SSDP Flood**	Saddam	*…overloads the DVR by causing cameras to spam the server with UPnP advertisements.*	A	1	4,077,266	14.4	26.4
	SYN DoS	Hping3	*…disables a camera's video stream by overloading its web server.*	A	1	2,771,276	18.7	34.1
	SSL Renegotiation	THC	*…disables a camera's video stream by sending many SSL renegotiation packets to the camera.*	A	1	6,084,492	10.7	54.9
Botnet Malware	**Mirai**	Telnet	*…infects IoT with the Mirai malware by exploiting default credentials, and then scans for new vulnerable victims network.*	C, I	X	764,137	52.0	66.9

Tabla 11. Sets de datos utilizados para la evaluación de Kitsune.

Fuente: (Mirsky, Doitshman, Elovici, & Shabtai, 2018)

4.2.3.2. Configuración de las pruebas

Como parte del alcance de la evaluación de desempeño el autor incluyó otras herramientas de detección para comparar los resultados de estas con los obtenidos por Kitsune. Es por esta razón que se han incluido algoritmos para detección de anomalías offline como el Isolation Forest (IF) y Gaussian Mixture Models (GMM), que poseen en general un mejor desempeño en comparación con las herramientas online ya que pueden realizar varias pasadas sobre los datos. También se incluyeron herramientas online para tener un valor de desempeño esperado mínimo, como los algoritmos Incremental GMM y pcStream2 y la herramienta NIDS Suricata.

A continuación, se enumeran las herramientas utilizadas para realizar la evaluación de desempeño y la comparación relativa entre estas:

- Herramientas de detección online:

- Suricata15: Un NIDS basado en firmas.
- Incremental GMM
- pcStream2
- Kitsune con parámetro m=1.
- Kitsune con parámetro m=10.

- Herramientas de detección offline:
  - Isolation Forest
  - GMM

Los diferentes algoritmos fueron entrenados con el primer millón de paquetes, y la cantidad de paquetes para cada tipo de ataque se puede ver en la Tabla 11.

4.2.3.3. Métricas utilizadas para la evaluación

A continuación, se describen las diferentes métricas utilizadas para la evaluación del desempeño de los diferentes modelos. Antes de definirlas explicamos los indicadores sobre los cuales se basa cada métrica, los cuales se pueden consultar en la Figura 25.

A saber:

- True Positive (TP): Cantidad de ataques detectados.
- False Positive (FP): Casos de tráfico normal identificado como ataques.
- True Negative (TN): Casos de tráfico normal identificado como tal.
- False Negative (FN): Casos de ataques identificados como tráfico normal.

[15] Sistema de detección de intrusiones de red de código abierto. https://suricata.io/

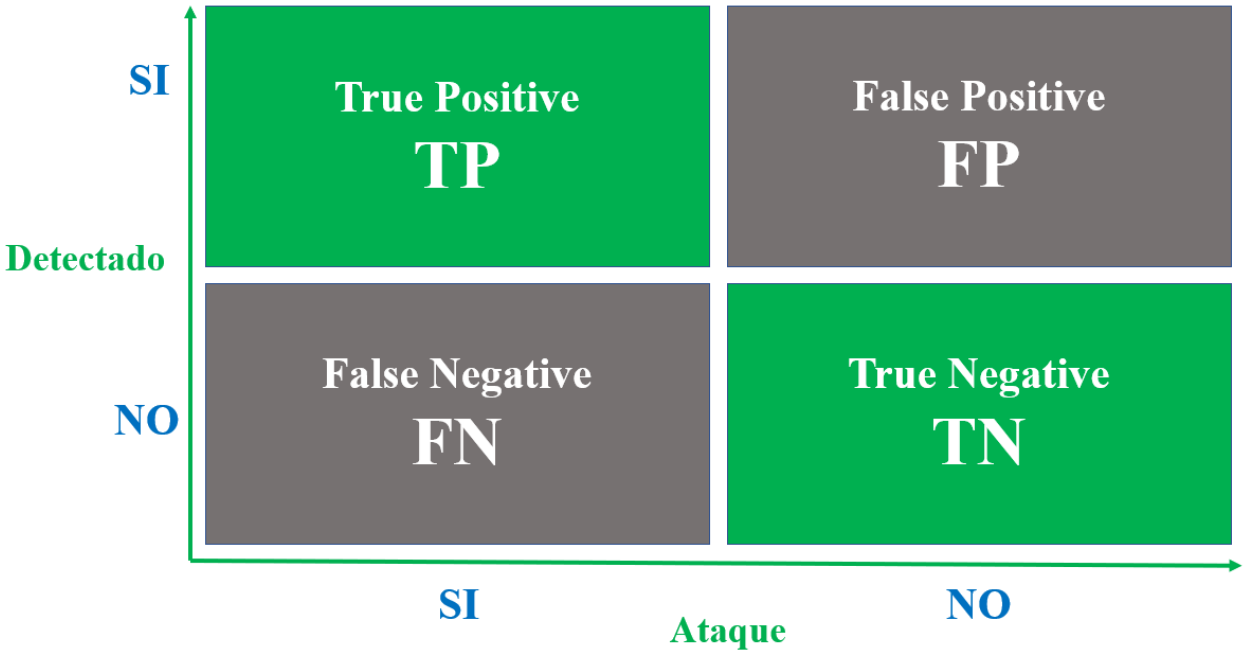

Figura 25. Indicadores de detección.

Fuente: Autoría propia.

Basándose en los indicadores previamente descriptos se definen las siguientes métricas de evaluación:

- TPR (True Positive Rate): $\frac{\text{TP}}{(\text{TP} + \text{FN})}$

  Representa la medida sobre la cantidad que el modelo va a detectar, midiendo los casos correctamente detectados, sobre el total de casos que debería de haber detectado. Este indicador también se lo conoce como Recall o Exhaustividad en la analítica de datos.

- FNR (False Negative Rate): $\frac{\text{FN}}{(\text{FN} + \text{TP})}$

  Representa el porcentaje de casos que no está pudendo ser identificado.

- FPR (False Positive Rate): $\frac{FP}{(FP + TN)}$

Representa el porcentaje de casos que está siendo identificado erróneamente como ataques, ósea, el nivel de falsas alarmas. Este parámetro también se lo conoce como Especificidad en analítica de datos.

- AUC: área bajo la curva ROC. Es una medida de rendimiento para los problemas de clasificación. Indica la capacidad del modelo de distinguir entre clases. Cuanto mayor sea el AUC, mejor será el modelo para predecir las clases.
- EER (Equal Error Rate): Es una medida que busca el equilibrio en un algoritmo entre su FNR y su FPR. Se calcula como el valor de FNR y FPR cuando son mínimos e iguales.

4.2.3.4. Resultados obtenidos

En la Tabla 12 se pueden observar los resultados obtenidos para las métricas definidas, con las diferentes herramientas y algoritmos bajo análisis.

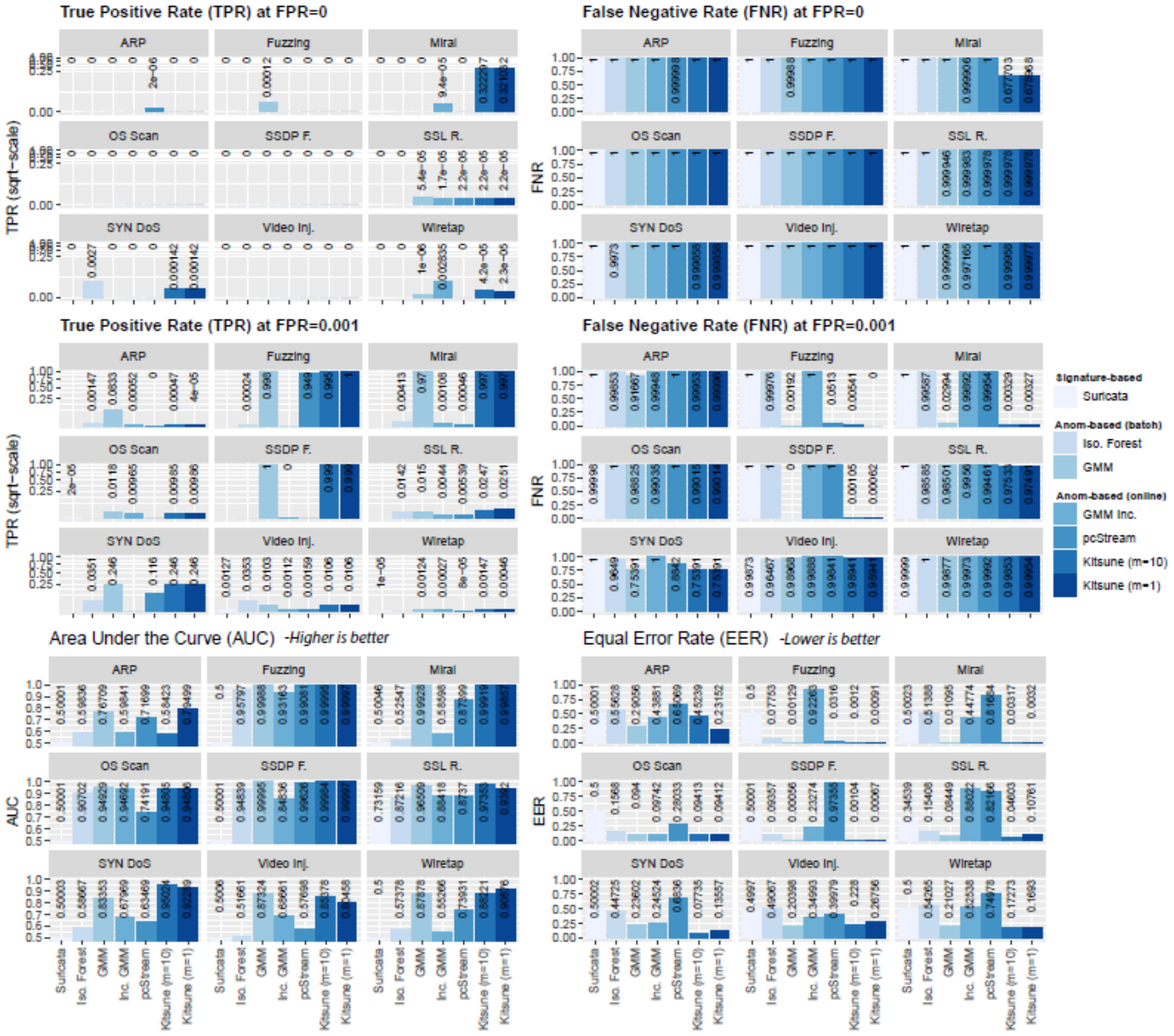

Tabla 12. Resultados obtenidos.

Fuente: (Mirsky, Doitshman, Elovici, & Shabtai, 2018)

De lo presentado se concluye que, comparando el rendimiento de los algoritmos en línea, Incremental GMM y pcStream 2, Kitsune, en general, muestra un desempeño superior a ambos algoritmos en términos de AUC y EER.

La fila superior de la Tabla 12 muestra el número máximo de True Positives que cada algoritmo pudo obtener, cuando se estableció el umbral de modo que no hubiera falsos positivos. Los resultados muestran que Kitsune detecta ataques en los conjuntos de datos mejor que los otros algoritmos, y más que el GMM en la mayoría de los casos.

Una de las mayores ventajas de Kitsune es su rendimiento en tiempo de ejecución. El conjunto de pequeños autoencoders de KitNET es más eficiente que usar un solo autoencoder. Esto se debe a que el conjunto reduce el número total de operaciones necesarias para procesar cada instancia.

5. Conclusiones

La revolución de internet está dando lugar a un mundo digital interconectado lleno de oportunidades de innovación, creatividad y nuevas formas de comunicación social. Para mantener los sistemas de información y las redes de comunicación seguras es esencial comprender el panorama de amenazas concerniente al IoT ya que es extremadamente amplio el campo de posibles amenazas y donde la Inteligencia Artificial juega un rol clave.

El objetivo del presente trabajo fue analizar y estudiar la importancia para la Ciberdefensa del monitoreo de las redes de IoT con tecnología capaz de adaptarse a los constantes cambios en las técnicas utilizadas para los ciberataques. Para tal fin en el marco teórico se ha desarrollado una introducción a la arquitectura de este tipo de redes, las diferentes amenazas sobre las mismas, los métodos de detección usualmente utilizados y una introducción a las técnicas de Inteligencia Artificial.

Asimismo, se planteó en el presente trabajo la relevancia que esta temática tiene para la Ciberdefensa, enumerando algunos casos recientes de ciberataques a infraestructuras críticas y sus consecuencias en los estados afectados.

Adicionalmente, se presentó un caso de uso práctico basado en un trabajo realizado en la Universidad de Ben Gurión, el cual desarrolla un software especialmente diseñado para detectar intrusiones en redes, de despliegue plug-and-play, con baja utilización de recursos, de ejecución distribuida y monitoreo en línea. Éste utiliza técnicas de aprendizaje automático, el cual permite vislumbrar la eficiencia de esta herramienta comparándola con otras técnicas disponibles. Sus ventajas en cuanto a desempeño, utilización de recursos y adaptabilidad posicionan a este tipo de soluciones como las más aptas para redes IoT, y por ende de importancia para el ámbito de la Ciberdefensa.

Asimismo podemos ver que por la constante evolución de la tecnología, es muy difícil conocer cuál será el alcance del avance del IoT en los servicios del futuro; sin embargo, lo que hemos visto a lo largo de

este trabajo es que puede llevar a una gran cantidad de problemas de ciberseguridad y privacidad de la información de los usuarios que podrán afectarles.

Finalmente en próximos trabajos se buscará plantear otras soluciones que filtren este tipo de tráfico anómalo durante el período de entrenamiento del modelo debido a que esta solución durante el período de aprendizaje asume que el tráfico no está contaminado.

6. Referencias bibliográficas

(s.f.).

Adam, M., & Vuong, S. (2013). Random forest classification for detecting android malware. *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, 663-669.

Alam, M., & Vuong, S. (2013). Random Forest Classification for Detecting Android Malware. *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing.* Beijing: IEEE.

Baieli, C. H. (2017). Análisis de flujos de redes para detectar patrones compatibles con ciberataques. San Luis, San Luis, Argentina: Universidad Nacional de San Luis.

Balliu, M., Bastys, J., & Sabelfeld, A. (Octubre de 2018). Securing IoT Apps. *IEEE Security & Privacy*, 22-29.

Bohadana, M., Meidan, Y., Mathov, Y., Mirsky, Y., Breitenbacher, D., Shabtai, A., & Elovici, Y. (2018). N-BaIoT: Network-based Detection of IoT Botnet Attacks Using Deep Autoencoders. *IEEE PERVASIVE COMPUTING, 13*(9).

Cambridge. (2021). *Cambridge Dictionary*. Obtenido de https://dictionary.cambridge.org/dictionary/english/

Cañedo, J., & Skjellum, A. (2016). Using machine learning to secure IoT systems. (págs. 2016 14th Annual Conference on Privacy, Security and Trust (PST)). Auckland: IEEE.

Cano, J. C., Bernios, V., García, B., & Toh, C. (Diciembre de 2018). Evolution of IoT: An Industry Perspective. *IEEE Internet of Things Magazine*, 12-17.

Childs, R., Smith, I., & Bailey, D. (2017). *CLP.13-Lineamientos de Seguridad IoT para el ecosiswtema de Dispositivos Periféricos IoT v2.0.* GSMA.

Childs, R., Smith, I., & Bailey, D. (2019). *IoT Security Guidelines Overview Document Version 2.2.* GSMA.

ChIoannou, C., & Vassiliou, V. (2019). Classifying Security Attacks in IoT Networks Using Supervised Learning. *15th International Conference on Distributed Computing in Sensor Systems (DCOSS) 2019* (págs. 652-658). Nicosia: Department of Computer Science, University of Cyprus and RISE - Research Center on Interactive Media, Smart Systems and Emerging Technologies.

Clarke , R. A., & Knake, R. K. (2019). *The Fifth Domain, Defending Our Country, and Ourselves in the Age of Cyber Threats.* New York: Penguin Press.

Collins. (2021). *Collins Dictionary*. Obtenido de https://www.collinsdictionary.com/dictionary/english

Cybereason Nocturnus. (3 de Agosto de 2021). *Cybereason*. Obtenido de https://www.cybereason.com/blog/deadringer-exposing-chinese-threat-actors-targeting-major-telcos

Dalal, K. R. (2020). Analysing the Role of Supervised and Unsupervised Machine Learning in IoT. *Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020).* Coimbatore: IEEE.

De Spiegeleire, S., Maas, M., & Sweijs, T. (2017). *Artificial Intelligence and the Future of Defense: Strategic implications for Small and Medium-sized force providers.* The Hague: The Hague Centre for Strategic Studies (HCSS). Obtenido de https://hcss.nl/sites/default/files/files/reports/Artificial%20Intelligence%20and%20the%20Future%20of%20Defense.pdf

Dewar, D. R. (2017). *Active Cyber Defense.* Zürich: Center for Security Studies (CSS). Obtenido de https://css.ethz.ch/content/dam/ethz/special-interest/gess/cis/center-for-securities-studies/pdfs/Cyber-Reports-2017-03.pdf

Diro, A. A., & Chilamkurti , N. (Mayo de 2018). Distributed Attack Detection Scheme using Deep Learning Approach for Internet of Things. *Future Gener. Comput. Syst.*, 761-768.

Engineering & Technology. (31 de Octubre de 2019). *E&T*. Obtenido de https://eandt.theiet.org/content/articles/2019/10/cyber-attack-on-india-s-largest-nuclear-power-plant-confirmed/

Feigenbaum, E. A., & Feldman, J. (1963). *Computers and Thought.* California: McGraw-Hill.

Goodin, D. (10 de Febrero de 2021). *Ars Technica*. Obtenido de https://arstechnica.com/information-technology/2021/02/breached-water-plant-employees-used-the-same-teamviewer-password-and-no-firewall/

Great Learning. (11 de Febrero de 2021). *What is Artificial Intelligence? How does AI work, Types and Future of it?* Obtenido de https://www.mygreatlearning.com/blog/what-is-artificial-intelligence/

Haddadpajouh, H., Javidan, R., Khayami, R., & Dehghantanha, A. (2016). A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks. *IEEE Transactions on Emerging Topics in Computing*, 99.

Hasan, M., Islam, M., Zarif, M. I., & Hashem, M. (2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things*.

Hebb, D. (1949). *Organization of Behavior: A Neuropsychological Theory.* New York: John Wiley and Sons, Inc.

Hernández Lorente, A. (2019). Análisis de actualidad: Ciberataques octubre 2019. *THIBER DIGEST - Informe mensual de ciberseguridad Nro. 14*, 6-10.

Hernández Sampieri, R., & Mendoza Torres, C. P. (2018). *Metodología de la Investigación: Las rutas cuantitativca, cualitativa y mixta.* Ciudad de México: Mc Graw-Hill.

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. (2014). *Metodología de la Investigación 6ta Edición.* México DF: Mc Graw Hill Education.

Homeland Security. (27 de Mayo de 2021). *Official website of the Department of Homeland Security*. Obtenido de https://www.dhs.gov/news/2021/05/27/dhs-announces-new-cybersecurity-requirements-critical-pipeline-owners-and-operators

Insikt Group. (28 de Febrero de 2021). *Recorded Future*. Obtenido de https://www.recordedfuture.com/redecho-targeting-indian-power-sector/

ITU-T. (2009). *ITU-T - Y.4108/Y.2213 - NGN service requirements and capabilities for network aspects of applications and services using tag-based identification* . Geneva: ITU-T.

ITU-T. (2012). *ITU-T Y.2060 - Overview of Internet of Things.* Geneva: ITU-T.

ITU-T. (2012). *Recommendation ITU-T Y.2060 - SERIES Y: GLOBAL INFORMATION - INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS AND NEXT-GENERATION NETWORKS - Overview of the Internet of things.* ITU-T.

ITU-T. (2019). *ITU-T Y.4460 - Architectural reference models of devices for Internet of things applications.* Geneva: ITU-T.

Kelly, D., Hahn, T., Szakal, A., & Murphy, J. (2017). *Five Indisputable Facts about IoT Security.* Obtenido de Security Intelligence Magazine: https://securityintelligence.com/series/five-indisputable-facts-about-iot-security/

Khadka, R. (7 de Septiembre de 2017). *Towards Data Science*. Obtenido de https://towardsdatascience.com/machine-learning-types-2-c1291d4f04b1

Kim, H.-H., Ham, H.-S., Kim, M.-S., & Choi, M.-J. (2014). Linear SVM-Based Android Malware Detection for Reliable IoT Services. *Journal of Applied Mathematics*.

Knud, L. L. (2014). *IoT Analytics.* Obtenido de https://iot-analytics.com/internet-of-things-definition/

Kovacs, E. (4 de Diciembre de 2020). *Security Week*. Obtenido de https://www.securityweek.com/iranian-hackers-access-unprotected-ics-israeli-water-facility

Lagraa, S., Francois, J., Lahmadi, A., Miner, M., Hammerschmidt, C., & State, R. (2017). BotGM: Unsupervised Graph Mining to Detect Botnets in Traffic Flows. *Centre pour la Communication Scientifique Directe*. Obtenido de https://hal.inria.fr/hal-01636480

Lewis, K. (November de 2016). *IBM IoT Blogs.* Obtenido de IBM: https://www.ibm.com/blogs/internet-of-things/security-iot/

Malik, S., & Chauhan, R. (2020). Securing the Internet of Things using Machine Learning: A Review. *2020 IEEE International Conference on Convergence to Digital World*. Mumbai: IEEE.

McCullogh, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. (U. o. Chicago, Ed.) *Bulletin of Mathematical Biophisycs, 5*, 115-133.

Meriam-Webster. (2021). *Merriam-Webster*. Obtenido de https://www.merriam-webster.com/

Minerva, R., Biru, A., & Rotondi, D. (2015). *Towards a definition of the Internet of Things (IoT).* IEEE.

Minsky, M. (1974). *A Framework for Representing Knowledge.* MIT, AI LAboratory. MIT.

Minsky, M., & Papert, S. (1969). *Percetrons. An Introduction to Computational Geometry.* Cambridge: MIT Press.

Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). *Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection.* Ben-Gurion University of the Negev.

RAE. (2021). *Real Academia Española*. Obtenido de https://dle.rae.es/

Rafal, K., & Choras, M. (2017). Pattern Extraction Algorithm for NetFlow-Based Botnet Activities Detection. *Hindawi Security and Communication Networks*. Obtenido de https://www.hindawi.com/journals/scn/2017/6047053/abs/

Ranjitha R. (2019). IoT: Architecture, Protocols and Devices. *eForensics Magazine*, 36-47.

Rose, K., Eldridge, S., & Chapin, L. (2015). *La Internet de las cosas - Una breve reseña.* Internet Society.

Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* New York: Spartan Books.

Rothman, D. (2018). *Artificial Intelligence by Example.* Birmingham: Packt Publishing Ltd.

Sharwood, S. (10 de Mayo de 2021). *The Register*. Obtenido de https://www.theregister.com/2021/05/10/colonial_pipeline_ransomware/

Sikorski, M., & Honig, A. (2012). *Practical Malweare Analysis.* San Francisco: No Starch Press.

Subex. (2021). *Subex IoT Security*. Obtenido de https://www.subex.com/iot-security/

Susanto, Stiawan, D., Arifin, A., Yazid, M., & Budiarto, R. (2020). IoT Botnet Malware Classification Using Weka Tool and Scikit-learn Machine Learning. En EECSI (Ed.). EECSI.

Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining.* Boston: Pearson - Addison Wesley.

Thabet, A. (2011). *Stuxnet Malware Analysis.*

Turing, A. M. (Octubre de 1950). Computing Machinery and Intelligence. *Mind, New Series, 59*(236), 433-460.

Wagner, C., Francois, J., State, R., & Engel, T. (2011). *Machine Learning Approach for IP-Flow Record.* Luxembourg: University of Luxembourg. Obtenido de http://dl.ifip.org/db/conf/networking/networking2011-1/WagnerFSE11.pdf

Widrow, B., & Hoff, M. E. (1960). Adaptative Switching Circuits. *IRE WESCON Convention Record*, (págs. 96-104).

Zaca, A., Kharroub, S., & Abualsaud, K. (2020). *Lightweight IoT Malware Detection Solution Using CNN Classification.* Qatar University, Department of Computer Engineering and Computer Science. Doha: Qatar University.