Modelo Híbrido de dos niveles para la detección de anomalías con técnicas de Machine Learning en redes IoT

Sebastian Berrios, *Member, Pontificia Universidad Católica de Valparaíso,*
Pamela Hermosilla, *Member, Pontificia Universidad Católica de Valparaíso,*
Herminia Beatriz Parra, *Member, Universidad Católica de Salta,*
Pablo Oñate, Student, *Pontificia Universidad Católica de Valparaíso.*

***Abstract*— La expansión de las redes IoT es una realidad continua, ya que el número de dispositivos conectados a Internet supera a la población mundial. Si bien esto representa un progreso social significativo, también introduce amenazas cibernéticas que pueden afectar la vida diaria de las personas. En respuesta, este estudio tiene como objetivo desarrollar un Sistema de Detección de Intrusos (IDS) utilizando técnicas de aprendizaje automático (ML) para identificar posibles ciberataques en redes IoT. Estas técnicas son expertas en discernir de forma automática y precisa las diferencias clave entre los datos normales y anormales, y su alta generalización les permite detectar ataques previamente desconocidos. Esta investigación incluye un examen de los sistemas IoT y sus protocolos de comunicación, las diversas amenazas a las que se enfrentan, y proporciona una taxonomía de IDS, que describe los métodos de detección y las técnicas de ML más utilizadas. Se propone un modelo de entrenamiento que incorpora diversas técnicas de ML y Deep Learning (DL), que permite evaluar su eficacia en función de los diferentes tipos de anomalías que deben detectarse.**

***Index Terms*—IoT; Ciberataques; Machine Learning; Sistemas de Detección de Intrusos.**

I. Introducción

El uso de redes de internet de las cosas (IoT) crece exponencialmente, siendo cada vez más los dispositivos conectados a internet. El termino IoT hace referencia a los sistemas físicos que reciben y transfieren datos a través de redes inalámbricas con poca intervención humana y esto se debe gracias a la integración de dispositivos informáticos en todo tipo de equipos [1]. El crecimiento de las redes IoT ha conllevado a que sean objetivos de atacantes para que les realicen actividades maliciosas. Los efectos de los ataques cibernéticos se vuelven más destructivos como resultado de que muchas instituciones han experimentado la interrupción de los servicios luego de estos ataques. Debido a esto, se concluye que los dispositivos IoT requerían en estos casos una herramienta más sofisticada para identificar la actividad maliciosa en la infraestructura inteligente. Estos sistemas se conocen como SISTEMAS DE DETECCIÓN DE INTRUSOS (IDS). La complejidad utilizada por los atacantes y el aumento de los ataques de día cero (es un ataque que tiene como objetivo la ejecución de código malicioso explotando vulnerabilidades que son desconocidas para los usuarios y para el fabricante del producto) indican que un IDS basado en anomalías es considerado adecuado para el entorno actual en sistemas IoT, en particular se considera prometedor el uso de herramientas de entrenamiento como la inteligencia artificial [2].

Esta investigación propone un IDS basado en diversas técnicas ML que utiliza un enfoque de selección de características y un clasificador de votos que es un método de conjunto de clasificadores. El marco de detección del IDS basado en ML propuesto, consta de cuatro fases que son: preprocesamiento del conjunto de datos, reducción de la dimensionalidad, entrenamiento de los clasificadores, por último, un reconocimiento de los ataques detectados. Esta publicación surge del trabajo de Tesis de Magister elaborado por Pablo Oñate de la Pontificia Universidad Católica de Valparaíso (PUCV). la cual se encuentra publicada en el reservorio institucional de dicha institución.

En la sección 2 de este artículo, se muestran aspectos metodológicos y características de la problemática de estudio. En la sección 3 se desarrolla el estado del arte, con la inclusión de los documentos más destacados de la revisión bibliográfica que se realizó. En la sección 4 se aborda el marco teórico, desarrollando un análisis exhaustivo de los sistemas de redes IoT. Este análisis abarcará una variedad de aspectos claves, incluyendo los diferentes tipos de redes, las arquitecturas predominantes, los protocolos de comunicación esenciales y las diversas áreas de aplicación de IoT. Posteriormente, en la sección 5 la atención se centrará en la identificación y definición de las amenazas de seguridad, específicamente en una arquitectura IoT de cuatro capas. Esto incluirá el desarrollo de una taxonomía detallada de los Sistemas de Detección de Intrusos (IDS) existentes, así como una clasificación de las técnicas de inteligencia artificial aplicadas en estos sistemas, con especial énfasis en el Aprendizaje Automático (ML). Además, se proporcionará una descripción detallada de los diversos tipos de datasets disponibles hasta la fecha en el ámbito de las redes IoT. En la sección 6, se presentará una solución propuesta, delineando un modelo innovador diseñado para abordar los desafíos identificados en este campo,

mostrando además los resultados logrados para un caso ejemplo. Se finaliza con la sección 7 en el que se destacan las conclusiones de la investigación.

II. Aspectos Metodológicos y Características de la Problemática de Estudio

Antes de abordar la investigación a pleno, es conveniente describir las cuestiones metodológicas y resumir la problemática de estudio.

A. Aspectos Metodológicos

El propósito de este proyecto es desarrollar un modelo para el entrenamiento de un sistema de detección de intrusos (IDS) en el contexto de las redes IoT, con el objetivo de lograr una alta precisión en la detección de ataques. Se emplearán diversas técnicas de inteligencia artificial, específicamente en el ámbito del Machine Learning (ML), para mejorar la detección en las redes IoT.

La primera fase se centrará en una revisión exhaustiva de la literatura y el estado del arte en sistemas IoT, amenazas ciberseguridad, técnicas de ML utilizadas en el área y estudios previos sobre IDS en redes IoT. Esto proporcionará una comprensión integral de los desafíos asociados con las amenazas cibernéticas en redes IoT y culminará en la propuesta de un modelo de entrenamiento.

Durante la fase de desarrollo, se seleccionarán los datasets apropiados, se definirá la técnica de preprocesamiento de datos y se escogerá un método de reducción de características. Se construirá el modelo propuesto, probando diferentes técnicas de ML, y se llevarán a cabo pruebas con los datasets tanto individualmente como en conjunto, creando un nuevo dataset combinado.

En la fase experimental, se evaluarán las técnicas de ML para determinar cuáles se ajustan mejor al problema. Se realizarán diversas pruebas para ajustar parámetros, como la proporción de datos utilizados para el entrenamiento y para las pruebas, y se podrán realizar ajustes al modelo basándose en los resultados obtenidos. Se harán comparaciones con estudios previos similares para establecer estándares de calidad mínimos para la propuesta.

Finalmente, en la fase de resultados, se analizarán en detalle los experimentos realizados. Este análisis permitirá extraer conclusiones sobre la eficacia del modelo propuesto, evaluar cuáles técnicas de ML son más efectivas en situaciones de clasificación binaria (tráfico anómalo o normal), y determinar qué técnicas de ML se adaptan mejor a los desafíos específicos del proyecto.

B. Problemática de Estudio

Los rápidos avances tecnológicos han permitido que existan más de 25 billones de dispositivos conectados alrededor de todo el mundo al internet, llegando a ser tres veces mayor a la población mundial [3]. El internet de las cosas (IoT) ha proporcionado un cambio a la sociedad, alcanzando nuevas capacidades y servicios a las personas, ha permitido hacer realidad ciudades inteligentes, hogares inteligentes, sistemas de transportes inteligentes, abriendo un mundo de oportunidades y funcionalidades para el bien o comodidad de las personas. Los sistemas IoT actuales son vulnerables a la mayoría de las ciber-amenazas, debido a que muchos fabricantes de dispositivos IoT priorizan las funcionalidades técnicas y de bajo costo sobre mecanismos de seguridad [4]. En el 2021, los ataques a dispositivos IoT han incrementado un 1.080 % [5], siendo relevante encontrar estrategias que permitan el análisis del tráfico en sistemas IoT para la detección temprana de estas posibles amenazas utilizando ML. A continuación, se detallarán diferentes conceptos para comprender en plenitud la magnitud y la cobertura del problema. Por tanto, se detallará el entorno de sistemas IoT, los tipos de ciberataques, los sistemas de detección de intrusos (IDS) y técnicas de ML.

III. Estado del Arte

El estudio de la problemática que se describe en este trabajo se abordó desde diferentes puntos o palabras claves, que permitieron hacer una revisión bibliográfica inicial. Aún cuando las secciones que siguen cuentan con la referenciación bibliográfica específica en cada apartado, es conveniente resumir el estado del arte considerando las investigaciones del del período 2019-2023.

Respecto de las cuestiones de ciberseguridad vinculadas a entorno IoT, se tomó en consideración los aportes de [6] en cuanto a la revisión de las amenazas aplicadas a IoT y el aporte de las tecnologías emergentes como Blockchain, Computación en la Niebla (Fog Computing), Computación en el Borde (Edge Computing) y Aprendizaje Automático para mejorar el nivel de seguridad de los entornos IoT. Por otro lado, de la investigación [7] se toma en consideración los aportes referidos a la aplicación de Machine Learning y Deep Learning para colaborar en la identificación de ciberataques dirigidos a los dispositivos IoT. La propuesta de [8] resulta también de interés para el presente trabajo porque plantea estrategias para la identificación de ataques a sistemas ciberfísicos. Por parte, Keshk et alt. [9] define un marco de acción para minimizar los riesgos de acceso a datos privados en contextos en los que se utilizan redes eléctricas inteligentes y en [10] el mismo equipo de investigadores aborda las cuestiones de seguridad en los espacios ciberfísicos desde la Ciencia de Datos. En [11] se estudian las posibles soluciones para asegurar la capa de final del sistema IoT (nodo de borde y dispositivo final), considerando especialmente los microcontroladores. En el artículo [12] se propone un sistema de seguridad para entornos IoT que actúa desde la computación en la niebla y aprovecha otros recursos como las VPN para asegurar el canal de acceso a los dispositivos IoT. La investigación [13] es sumamente interesante, basada en una encuesta de datos destacados, los autores desarrollan una taxonomía de amenazas de red y las características principales de cada tipo de ataque.

Entre las investigaciones que recurren a la aplicación de la inteligencia artificial para desarrollar sistemas de detección más confiables, se destacan los siguientes trabajos: el [14] analiza en profundidad la problemática de ataques en la computación en la niebla, y propone un sistema de detección de intrusiones basado en la inteligencia artificial; la investigación [15] propone un modelo de detección de anomalías en la red, definiendo algunas métricas de evaluación (precisión, tasa de

detección, falsos positivo y Fi-Score) recurriendo a técnicas de Deep Learning; el trabajo [16] describen la problemática propia de los algoritmos de Machine Learning y Deep Learning, y presentan una mejora al rendimiento de estos algoritmos mediante la implementación de técnicas de selección, aprendizaje datasets y voting que se aplican sobre datasets preprocesados; y esta misma línea se abordan métricas de evaluación en la investigación [17].

Por último, los trabajos [18] y [19] nos permitieron abordar con más sustento la elección de los datasets así como las técnicas a utilizar para el análisis de los resultados obtenidos.

IV. Marco Teórico

A continuación, se realizará un estudio sobre los diferentes entornos que existen en sistemas IoT, también las posibles amenazas en estos entornos. Se explicarán los IDS que se usan para detectar estas amenazas en la actualidad, así como las técnicas de ML utilizadas para desarrollar IDS. Por último, se realiza una comparación de diversos estudios actuales referentes a IDS creados con diferentes técnicas y su rendimiento.

Con su creciente aplicación en campos tan diversos como lo militar, la agricultura, los sistemas de energía, la educación y el comercio, han fomentado la emergencia de una amplia gama de dispositivos y la adopción de múltiples estándares de comunicación y protocolos. Esta diversidad se refleja también en la arquitectura de las redes IoT, que se ha adaptado para soportar una variedad de necesidades y especificaciones. Las arquitecturas de IoT, comúnmente estructuradas en múltiples capas, se diseñan para gestionar eficientemente la interconexión de estos dispositivos variados, asegurando la comunicación fluida y la integración de datos.

Se presentan los principales aspectos de los tipos de redes y arquitecturas asociadas a los sistemas IoT.

A. Tipos de redes

Una red informática se crea cuando dos o más computadoras (o dispositivos) están conectadas para compartir datos y/o recursos. Se estudiarán los diferentes tipos de redes que se pueden encontrar en sistemas IoT, las cuales, se diferencias en base a la distancia y el propósito de su uso.

- Red de área Personal (PAN): Estas redes suelen ser inalámbricas, se establecen bajo demanda o ad-hoc cuando se necesitan para comunicarse entre dos o más dispositivos. Las redes PAN se utilizan entre dispositivos propietarios de dos partes diferentes, o entre dos dispositivos propietarios de una persona, como una PDA y una computadora portátil o un teléfono móvil. Estas redes generalmente se caracterizan como de corto alcance, a menudo limitadas a 10 metros o menos [20, 21].
- Red de área Local (LAN): Una LAN conecta dispositivos de red en una distancia relativamente corta. Un edificio de oficinas, una escuela o un hogar en red generalmente contiene una sola LAN, aunque a veces un edificio contendrá algunas LAN pequeñas) y ocasionalmente una LAN abarcará un grupo de edificios cercanos. Sin embargo, se diferencia de la PAN por su amplio rango de comunicación y la mayor cantidad de dispositivos que pueden estar conectados [20, 21].
- Red de área Amplia (WAM): Este tipo de red conecta varios dispositivos que se ubican a largas distancias. Así pueden comunicarse de manera remota sin importar qué tan lejos se encuentren. El ejemplo más común este tipo de es la Internet, que conecta millones de dispositivos alrededor del mundo. Por eso, debido a su amplio alcance, la gestión de este tipo de red informática es pública y les corresponde a diversos administradores [20, 21].
- Red de área Metropolitana (MAN): El término Red de área metropolitana (MAN) se usa típicamente para describir una red que abarca un área de toda la ciudad o un pueblo. Las MAN son más grandes que las LAN tradicionales y utilizan predominantemente medios de alta velocidad, como cable de fibra óptica, para sus redes troncales. Los MAN son comunes en organizaciones que necesitan conectar varias instalaciones más pequeñas para compartir información [20, 21].
- Red de área Local Inalámbrica (WLAN): Las redes de área local inalámbricas son muy parecidas a las redes LAN, excepto que no requieren cables de red para conectarse entre sí. Las señales de radio e infrarrojas se usan para la comunicación entre máquinas mientras se utiliza una red de área local inalámbrica. Las redes de área local inalámbricas permiten la movilidad de los dispositivos mientras están conectados a la red. Estas redes utilizan el estándar IEEE 802.11 [20, 21].
- Red de área de Campus (CAN): Las redes de área de campus suelen ser una conexión de muchas redes LAN pequeñas que se utilizan a menudo en campus universitarios y edificios de oficinas. Las redes de área de campus permiten compartir archivos fácilmente entre diferentes departamentos, ya que todos los archivos generalmente se comparten en las máquinas servidor de cada red LAN. Este tipo de red ofrece mucha sencillez en la transferencia y descarga de archivos [20, 21].
- Red de área de Almacenamiento (SAN): Las redes de área de almacenamiento se utilizan principalmente como bases de datos de información. Se utilizan específicamente para el almacenamiento de información y la fácil recuperación de datos específicos cuando sea necesario. Las redes de área de almacenamiento suelen ser utilizadas por sitios web que ofrecen servicios de descarga [20, 21].
- Red de área del Sistema (SAN): Estas son redes orientadas a la velocidad que proporcionan conexiones de Internet de alta velocidad a un grupo de dispositivos. Estos se utilizan principalmente para fines de servidor y permiten que otras computadoras se conecten a estas redes de área del sistema [20, 21].
- Red Privada Virtual (VPN): Una red privada virtual (VPN) es una red de comunicación virtual que utiliza la infraestructura de una red física para asociar sistemas informáticos de manera lógica. En este sentido, se puede tratar de todos los tipos de redes expuestos anteriormente. Lo más común es utilizar Internet como medio de transporte, ya que este permite establecer la conexión entre todos los ordenadores a nivel mundial y, al contrario de lo que ocurre con las redes MAN o WAN privadas, está disponible de forma gratuita. La transferencia de datos

tiene lugar dentro de un túnel virtual erigido entre un cliente VPN y un servidor VPN [20, 21].

B. Arquitectura IoT

Una arquitectura IoT se compone de objetos físicos integrados en una red de comunicación y respaldados por equipos informáticos para brindar servicios inteligentes a los usuarios. El sistema IoT debería ser capaz de conectar miles de millones de dispositivos heterogéneos a través de Internet, por lo que se necesita una arquitectura flexible y en capas. Existen numerosas arquitecturas y modelos de referencia propuestos por varios autores y organizaciones, pero aún no se ha consensuado en un modelo de referencia formalmente reconocido [22,23]. Las arquitecturas y modelos de referencia más comunes se explican a continuación.

La arquitectura de 3 capas es el modelo más común y básico que comprende las capas de percepción, red y aplicación. La capa de percepción también se denomina "capa de dispositivo" e incluye dispositivos físicos y sensores. La capa de red también se denomina "capa de transmisión", que debe transmitir de forma segura los datos de telemetría de los sensores a los sistemas de procesamiento y análisis de datos.

La arquitectura de 4 capas es una aplicación de IoT en la que se identifican cuatro capas: (1) capa de detección; (2) capa de red; (3) capa de software intermedio; y (4) capa de aplicación, como se observa en la Figura 1. Cada una de estas capas en una aplicación de IoT utiliza diversas tecnologías que generan una serie de problemas y amenazas de seguridad. Esta arquitectura se diferencia de las de tres capas, por la capa intermedia que permite la conexión entre la capa de red y la capa de aplicación [6]. La capa de aplicación ofrece una gestión global de las aplicaciones utilizando los sistemas de la capa de red [6].

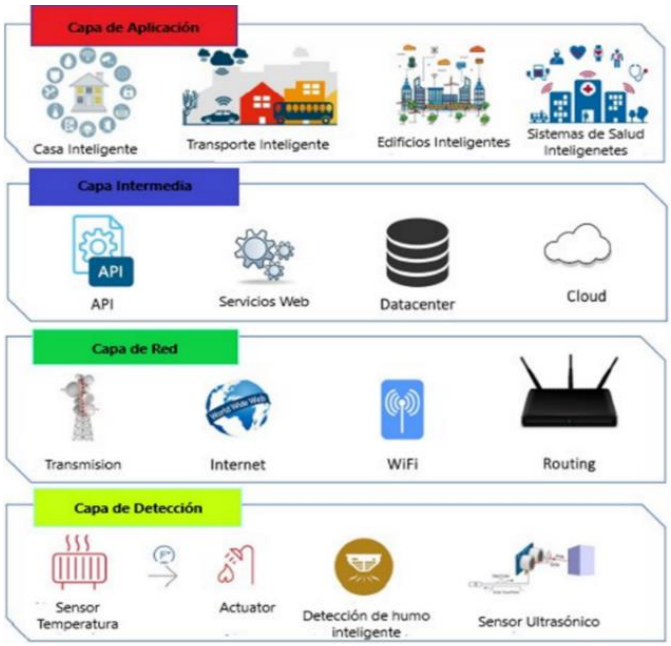


Figura 1: Arquitectura de 4 Capas.

V. Amenazas de Seguridad en las Arquitecturas IoT

En la sección anterior, se definieron diversas arquitecturas en sistemas IoT, a partir de esto utilizaremos la arquitectura de 4 capas para la división de los posibles ataques en redes IoT como se observa en la Figura 2.

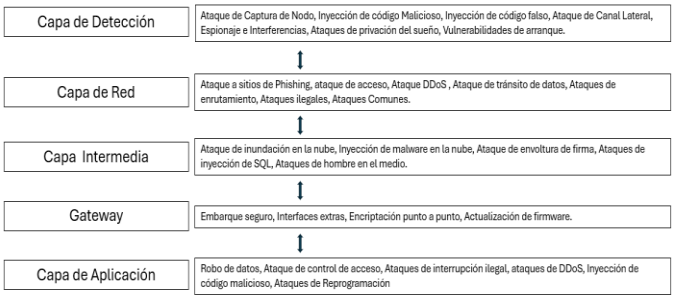


Figura 2: Tipos de Ataques a IoT organizadas por la capa afectada [11].

Cada una de las capas utiliza diversos tipos de tecnologías que causan una serie de problemas y amenazas de seguridad. También se agrega el gateway (puerta de enlace), que no es un tipo de capa, pero es un intermediario entre los dispositivos o servicios entre las capas intermedia y la capa de aplicación.

En esta sección se realizará un análisis de las posibles amenazas de las aplicaciones IoT para la arquitectura de 4 capas anteriormente mencionadas.

A. Amenazas en la capa de Detección

La capa de detección se ocupa principalmente de sensores y actuadores físicos de IoT. Los sensores detectan el fenómeno físico que ocurre a su alrededor y los actuadores realizan una determinada acción en el entorno físico, en función de los datos detectados. Existen una gran diversidad de sensores para detectar diferentes tipos de datos que va desde sensores de cámara hasta sensores de humedad, entre otros, por tanto, se usan diferentes tipos de protocolos cómo son RFID, GPS, WSN, etc.

En la capa de detección de IoT, se enfrentan varios riesgos de seguridad [6,11], exacerbados por la variedad de dispositivos y protocolos que se utilizan. De la investigación [6] se toman los principales riesgos:

- Captura de Nodos: Vulnerabilidad de nodos de baja potencia a ser sustituidos por nodos maliciosos, comprometiendo el sistema IoT. Las aplicaciones IoT se componen de varios nodos de baja potencia, como sensores y actuadores. Estos nodos son vulnerables a una diversidad de ataques, los atacantes pueden intentar capturar o reemplazar el nodo en el sistema IoT por un nodo malicioso. En nuevo nodo puede aparecer como parte del sistema, pero está controlado por el atacante, comprometiendo al sistema completo [6,11].
- Inyección de Código Malicioso: Posibilidad de inyectar código dañino en nodos IoT durante actualizaciones de software, facilitando funciones o accesos no autorizados. El ataque involucra al atacante inyectando algún código malicioso en la memoria del nodo. Por lo general, el firmware o el software de los nodos de IoT se actualizan por medio de tecnologías inalámbricas y esto brinda una puerta de entrada a los atacantes para inyectar código malicioso. Esto puede obligar a los nodos a realizar algunas funciones no deseadas o incluso pueden intentar acceder al sistema IoT completo [6].

- Inyección de Datos Falsos: Uso de nodos comprometidos para introducir datos incorrectos, causando mal funcionamiento o facilitando ataques DDoS. Una vez que se capturó el nodo, el atacante puede usarlo para inyectar datos erróneos en el sistema IoT. Esto puede dar lugar a resultados falsos y provocar un mal funcionamiento de la aplicación IoT. También se puede utilizar este método para provocar un ataque DDoS, DISTRIBUTED DENIAL OF SERVICE (Denegación de Servicio Distribuido). En un ataque DDoS, múltiples sistemas informáticos comprometidos atacan un objetivo, como un servidor, un sitio web u otro recurso de red, y provocan una denegación de servicio para los usuarios del recurso objetivo [6].
- Ataques de Canal Lateral (SCA): Explotación de características como la microarquitectura de procesadores para obtener datos confidenciales, aunque los chips modernos implementan varias contramedidas. Los ataques de canal lateral pueden provocar la filtración de datos confidenciales. Las microarquitecturas de los procesadores, la emanación electromagnética y su consumo de energía revelan información confidencial a los adversarios. Los ataques de canal lateral pueden basarse en consumo de energía, ataques de sincronización o de electromagnéticos. Los chips modernos tienen varias contramedidas para evitar estos ataques de canal lateral mientras implementan los módulos criptográficos [6].
- Espionaje e Interferencias: Riesgo de espionaje y captura de datos en nodos IoT desplegados en entornos abiertos, particularmente durante la transmisión de datos o la autenticación. Las aplicaciones IoT a menudo consisten en varios nodos implementados en entornos abiertos, dando como resultado que estén expuestas a los atacantes en cualquiera de las fases de trabajo, como la de transmisión de datos o la de autenticación [6].
- Ataques de Privación del Sueño: Estrategias para agotar la batería de dispositivos IoT de baja potencia, conduciendo a una denegación de servicio. En este tipo de ataques, se intenta agotar la batería de los dispositivos periféricos IoT de baja potencia. Esto conduce a una denegación del servicio de los nodos en la aplicación IoT debido a una batería descargada. Esto se puede realizar ejecutando bucles infinitos en los dispositivos perimetrales mediante un código malicioso o aumentando artificialmente el consumo de la energía [6].
- Ataques de Arranque: Vulnerabilidades durante el proceso de arranque de dispositivos periféricos, un momento crítico cuando los procesos de seguridad aún no están activos. Los dispositivos perimetrales son vulnerables a varios ataques durante el proceso de arranque. Esto se debe a que los procesos de seguridad incorporados no están habilitados en ese momento. Los atacantes pueden aprovechar esta vulnerabilidad e intentar atacar los dispositivos del nodo cuando se reinician. Es importante tomar las precauciones dado que muchos dispositivos suelen tener poca potencia y, en ocasiones, pasan por ciclos de sueño y vigilia, es esencial asegurar el proceso de arranque en estos [6].

B. Sistema de Detección de Intrusos (IDS)

Un IDS es una aplicación de seguridad informática cuyo objetivo es detectar una gran variedad de violaciones de seguridad, que van desde intentos de intrusión externos hasta penetraciones en el sistema y abusos por parte de personas internas [24]. Las principales funciones de los IDS son monitorear hosts y redes, analizar el comportamiento de los sistemas informáticos, generar alertas y responder a comportamientos sospechosos. Debido a que monitorean las redes y host relacionados, los IDS generalmente se implementan cerca de los nodos de red protegidos, siendo los lugares con mayores tráficos de red [25].

La mayoría de los IDS tienen una estructura común que incluye:

1. Un módulo de recopilación de datos que posiblemente contenga evidencia de un ataque,
2. Un módulo de análisis que detecta ataques después de procesar esos datos, y
3. Un mecanismo para informar un ataque.

En el módulo de recopilación de datos, los datos de entrada de cada parte de los sistemas IoT se pueden recopilar y examinar para encontrar el comportamiento normal de interacción, detectando así el comportamiento malicioso en las primeras etapas. El módulo de análisis se puede implementar utilizando varias técnicas y métodos, sin embargo, los métodos basados en ML son más adecuados y dominantes para para la tarea de aprender comportamientos benignos y anómalos en función de cómo los dispositivos y sistemas de IoT interactúan entre sí en entornos de IoT. Además, los métodos ML pueden predecir nuevos ataques, que a menudo son diferentes de los ataques anteriores, porque los métodos ML pueden predecir de forma inteligente futuros ataques desconocidos aprendiendo datos previamente existentes [8].

Algunos IDS open source que se pueden utilizar de forma gratuita son los siguientes:

- Snort: La primera versión se desarrolló por 1998 y cabe mencionar que en aquel momento no se contempló como IDS puro, pero fue evolucionando hasta ese punto.
- Suricata: Es un sistema de detección de intrusiones avanzado que permite multihilos, aceleración mediante hardware, entre otros.
- Kismet: Es un IDS Wireless enfocado en la red.

C. Métodos de Detección de IDS

Los métodos de detección usados por IDS se pueden dividir en cuatro tipos de metodologías que se explica a continuación.

Las ***técnicas de detección basadas en firmas*** contienen un depósito de firmas de ataques y comparan el tráfico de red o las acciones del sistema con este depósito de firmas. Tan pronto como se encuentra una coincidencia, se genera una alerta de detección. Aunque es suficientemente precisa contra ataques conocidos para los que existen firmas en el repositorio, esta técnica no puede detectar ataques de día cero (nuevos). Incluso si no es efectivo contra mutaciones de un ataque existente [7,9,26].

Algunas investigaciones, como [27], propusieron medios para superar esta deficiencia de técnicas basadas en firmas mediante el uso de un Sistema Inmune Artificial (AIS). Esta técnica diseñó detectores que se basan en firmas/patrones de ataques utilizando el modelo de células inmunitarias, que pueden detectar si un paquete es normal o malicioso a través de

su clasificación como elemento propio o no propio. El sistema tiene la capacidad de adopción de nuevos patrones a partir del monitoreo continuo del sistema. Sin embargo, la viabilidad de tal técnica de detección en un entorno de IoT con recursos limitados es cuestionable [7].

Los autores de [28] resolvieron las limitaciones de recursos en los IDS basados en firmas mediante la utilización de una máquina Linux independiente con una versión adaptada del IDS de firmas basado en Suricata [28] sin embargo, los autores no proporcionaron ninguna pista sobre la actualización de las firmas de ataque. Los autores en [29] ampliaron el trabajo publicado en [28] al proponer modificaciones en las técnicas de coincidencia de firmas. Otra investigación de [10] abordó las limitaciones de potencia de procesamiento de los sistemas IoT mediante el uso de valores de cambio auxiliares con un algoritmo de detección de patrones múltiples, lo que permite una reducción en la cantidad de operaciones de coincidencia requeridas entre las firmas de ataque y los paquetes de tráfico de red. El sistema utilizó repositorios de firmas del IDS de código abierto (Snort) y del antivirus de código abierto (ClamAV) [7].

Las ***técnicas de detección basadas en anomalías*** se basan en un perfil de comportamiento normal de referencia para el entorno supervisado [26,30]. Esta línea de base normal se usa luego para comparar las acciones del sistema en cualquier momento dado. Cualquier desviación fuera de los límites del umbral permitido se informa generando una alerta sin proporcionar ninguna clasificación para el tipo de ataque detectado. También hay intentos de usar modelos de aprendizaje automático que aprenden eventos normales y de anomalías como modelos de detección de comportamiento. En comparación con las técnicas de detección basadas en firmas, las técnicas de detección basadas en anomalías son más eficaces para descubrir nuevos ataques. Un inconveniente de esta técnica es la dificultad para construir el perfil de referencia de comportamiento normal, lo que da lugar a un aumento de las tasas de falsos positivos [31,32,33]. Las técnicas de detección basadas en anomalías se basan en algoritmos de ML para crear un perfil normal básico de los sistemas supervisados. El uso de tales técnicas de ML en entornos de IoT con recursos y energía limitados sigue siendo un desafío, debido a los altos recursos computacionales necesarios para entrenar y validar las técnicas de ML [2].

Las ***técnicas de detección basadas en especificaciones*** tienen el mismo principio básico de las técnicas de detección basadas en anomalías, donde el comportamiento normal de un sistema se perfila a través de algún medio y se compara con las acciones actuales del sistema para detectar desviaciones fuera de rango. Sin embargo, en las técnicas basadas en anomalías, el comportamiento normal se aprende a través de ML, mientras que, para las técnicas basadas en especificaciones, un experto humano debe especificarlo manualmente a través de un repositorio de reglas y rangos asociados de desviaciones [34]. Esto permite reducir las tasas de falsos positivos en comparación con las técnicas de detección basadas en anomalías [31]. Con la ventaja de no requerir ninguna fase de aprendizaje después de especificar un conjunto de reglas [34], estas técnicas adolecen de falta de adaptabilidad a entornos variados y son propensas a errores en las especificaciones [7,35].

Las ***técnicas de detección basadas en híbridos*** emplean una combinación de las técnicas mencionadas anteriormente para compensar las deficiencias y optimizar las ventajas de detectar ataques nuevos y existentes. Los autores en [36] propusieron SVELTE, que es un IDS para sistemas IoT conectados a IP que utilizan RPL como protocolo de enrutamiento en redes 6LoWPAN. Este IDS se diseñó utilizando un híbrido de técnicas de detección basadas en anomalías y firmas para obtener un equilibrio entre los requisitos de almacenamiento y procesamiento de cada una de estas dos técnicas. Intentaron equilibrar el costo de almacenamiento de la detección basada en firmas y el costo informático de las técnicas basadas en anomalías [7].

D. Taxonomía de ML para Seguridad IoT

En la era actual de la tecnología, la seguridad de las redes de Internet de las Cosas (IoT) se ha convertido en un campo de importancia crítica. La proliferación de dispositivos conectados ha traído consigo nuevos retos de seguridad, impulsando la necesidad de soluciones más avanzadas y adaptativas. En este contexto, las técnicas de Machine Learning y Deep Learning emergen como herramientas poderosas, ofreciendo enfoques novedosos y eficientes para detectar y prevenir amenazas de seguridad en redes IoT. Esta introducción presenta una taxonomía detallada de estas técnicas, destacando cómo se aplican en diferentes escenarios de seguridad de redes IoT. Se exploran diversas metodologías y algoritmos, desde enfoques clásicos de Machine Learning hasta arquitecturas complejas de Deep Learning, proporcionando una visión integral de cómo estas tecnologías están modelando el futuro de la seguridad en el IoT. A través de esta revisión, se busca no solo entender las capacidades actuales de estas técnicas, sino también anticipar futuras direcciones y desafíos en este campo en rápida evolución.

D.1 Técnicas de Machine Learning para IDS

Las técnicas de Machine Learning (ML) en sistemas de detección de intrusiones (IDS) para entornos IoT, destacan tres clasificadores principales:

- Naive Bayes (NB), que usa el teorema de Bayes para clasificar tráfico basándose en observaciones previas, pero con limitaciones en la precisión debido a la falta de consideración de interdependencias entre características [37,38];
- K-Nearest Neighbor (KNN), que clasifica datos usando la distancia euclidiana sin parámetros adicionales, con la eficacia dependiente del número de vecinos más cercanos (k), siendo la selección de k crucial para la precisión [39,40]; y
- Árboles de Decisión (DT), que organizan un árbol basado en las características extraídas, enfrentando desafíos en almacenamiento y complejidad computacional [12,13,41,42].
- Máquinas de Vectores de Soporte(SVM), son modelos de aprendizaje supervisado que se utilizan para realizar tareas

de clasificación y regresión. Se basan en el concepto de encontrar el hiperplano que mejor separa las clases en el espacio de características. Este hiperplano se elige de manera que maximice el margen entre las clases, donde el margen se define como la distancia mínima entre el hiperplano y los puntos más cercanos de cada clase, conocidos como vectores de soporte. [56,57]

- El clasificador de Bosque Aleatorio (RF, por sus siglas en inglés de *Random Forest*) es un método de aprendizaje ensamblado para clasificación (y también puede ser utilizado para tareas de regresión), que opera construyendo múltiples árboles de decisión durante el entrenamiento y outputting la clase que es el modo de las clases (clasificación) o predicción media (regresión) de los árboles individuales. Los Bosques Aleatorios corrigen el hábito de los árboles de decisión de sobreajustar a su conjunto de entrenamiento, proporcionando así una mayor precisión general mediante la agregación de los resultados de múltiples árboles.[57]
- El Clasificador SGD (Descenso de Gradiente Estocástico) es un método de optimización y aprendizaje automático que utiliza el gradiente de la función de pérdida para encontrar el mínimo de esta función, actualizando los parámetros del modelo de forma iterativa. A diferencia del descenso de gradiente tradicional, que calcula el gradiente utilizando todo el conjunto de datos (lo cual puede ser muy costoso computacionalmente para grandes volúmenes de datos), el SGD actualiza los parámetros utilizando solo un subconjunto de los datos (un lote) en cada iteración, lo que resulta en una convergencia más rápida y eficiente.[58]

Se destaca la importancia de los IDS en la detección de amenazas, incluyendo ataques de día cero, y se menciona un modelo híbrido de dos niveles para redes IoT, con enfoque en el balance de datos y selección de características para mejorar el desempeño del modelo. En la solución propuesta que se describe en las siguientes secciones se planea probar el modelo en distintos datasets y escenarios para validar su rendimiento y abordar la detección de anomalías desconocidas.

D.2 Técnicas de Deep Learning para IDS

Los algoritmos DL superan a los algoritmos ML en aplicaciones que involucran grandes conjuntos de datos. DL se vuelve más relevante en las aplicaciones de seguridad de IoT, ya que los entornos de IoT se caracterizan por la producción de grandes cantidades y una variedad de datos. Además, DL es capaz de modelar automáticamente conjuntos de características complejas a partir de los datos de muestra. Otra ventaja de los algoritmos DL es su capacidad para permitir enlaces profundos en redes IoT. Esto permite interacciones automáticas entre sistemas basados en IoT en ausencia de intervención humana para realizar funciones de colaboración asignadas [43].

Debido a su capacidad para extraer representaciones de características jerárquicas en una arquitectura profunda compleja, DL se puede clasificar como una rama de los algoritmos de ML que utiliza múltiples capas no lineales de procesamiento para extraer conjuntos de características. Estos conjuntos de características se utilizan luego para la abstracción y la detección de patrones después de las transformaciones necesarias DL puede usarse en modo generativo con aprendizaje no supervisado, modo discriminativo con aprendizaje supervisado o un enfoque híbrido mediante la combinación de ambos modos [43].

Los diversos algoritmos de Deep Learning (DL) aplicados en sistemas de detección de intrusiones (IDS) para entornos IoT:

- Las Redes Neuronales Recurrentes (RNNs) son eficaces en el procesamiento secuencial de datos, especialmente en la detección de intrusos en la red, con investigaciones previas destacando su utilidad [14,44].
- Las Redes de Memoria a Largo Plazo (LSTM), una variante de RNN, son adecuadas para analizar datos temporales en detección de anomalías [45,46,47].
- Las Redes Neuronales Convolucionales (CNN) son útiles para la extracción de características de datos, aunque su alta demanda computacional plantea desafíos en IoT [15,48,49].
- Los Perceptrones Multicapa (MLP) se utilizan en funciones no lineales para clasificación o regresión.
- Los Autoencoders Profundos (AEs) se aplican en la detección de malware, mostrando mayor precisión que otros métodos [50,51].
- Las Máquinas de Boltzmann Restringidas (RBM) y las Redes de Creencias Profundas (DBN) enfrentan retos en su implementación en IoT por su alta demanda de recursos, pero se han utilizado en sistemas IDS [52,53].
- Las Redes Antagónicas Generativas (GAN) y los Conjuntos de Redes DL (EDLN) también se exploran para mejorar la precisión y el rendimiento en la seguridad de IoT [16,54,55].

VI. Solución Propuesta

En esta sección se describe la modelización de la solución propuesta, los resultados obtenidos de una caso ejemplo y se resumen las principales conclusiones de dichos resultados.

A. Modelo Propuesto

Este modelo se centra en la utilización de técnicas avanzadas de selección de características y la aplicación de un clasificador de votación, un enfoque de múltiples clasificadores que ha demostrado ser efectivo en la mejora de la precisión y eficacia del IDS.

La selección de características se abordará mediante métodos más sofisticados que los tradicionalmente utilizados, buscando optimizar el conjunto de datos para que el modelo de ML pueda identificar patrones de intrusión de manera más efectiva. En cuanto a las técnicas de ML, se explorarán algoritmos más avanzados y posiblemente personalizados para adaptarse mejor a las características únicas de los datos de intrusión, superando así las limitaciones de los métodos convencionales.

Este enfoque renovado no solo busca mejorar la precisión en la detección de intrusiones sino también aumentar la eficiencia operativa del IDS, reduciendo la tasa de falsos positivos y adaptándose dinámicamente a las nuevas amenazas en constante evolución [17].

El método consta de las siguientes seis fases principales:

1. Dataset: Se dividirá el conjunto de datos en entrenamiento y testeo con una proporción del 80% y 20% respectivamente.
2. Reducción de dimensionalidad: Se divide en dos etapas que son preprocesamiento de los datos y la selección de características.
   2.1. Preprocesamiento de los datasets: Consiste en transformar los datos sin procesar en un formato adecuado para el análisis mediante la aplicación de procesar los datasets originales. Por ejemplo, la eliminación de redundancia, la normalización de los datos en un formato más estándar o la eliminación de valores faltantes.
   2.2. Selección de características: para superar el problema de los conjuntos de datos de alta dimensión, se utiliza el enfoque de selección de características para reducir la dimensionalidad de los conjuntos de datos y seleccionar las características más relevantes para la clasificación.
3. Balanceo de datos: Consiste en tres etapas en paralelos que son:
   3.1. Sin balancear: Los datasets están desequilibrados, haciendo que la proporción de datos de cada clase no sea equitativa.
   3.2. SMOTE: Es una técnica de sobre muestreo que permite balancear los datasets y se aumenta la cantidad de datos de la clase minoritaria.
   3.3. RUS: Es una técnica de submuestreo que permite balancear los datasets y se disminuye la cantidad de datos de la clase mayoritaria.

   El uso de estas tres etapas permitirá evaluar el mejor rendimiento de los modelos y su comportamiento según la cantidad de datos usados.
4. Entrenamiento del clasificador: Con el propósito de mejorar la precisión del IDS, se entrenan a varios clasificadores individuales basados en ML y DL y se construye un clasificador en conjunto basados en ellos.
5. Reconocimiento de ataques: el rendimiento de los modelos se prueba utilizando las métricas antes explicadas y la técnica de voting se utiliza para combinar las distribuciones de probabilidad de los modelo base con la regla de votos por mayoría para tomar decisiones de clasificación.
6. Resultados: Basado en las métricas de evaluación se evalúa el rendimiento de los clasificadores.

El modelo propuesto para la clasificación binaria en el contexto de la detección de ataques cibernéticos incorpora una metodología basada en el uso combinado de clasificadores Gradient Boosting (GB) y K-Nearest Neighbors (KNN). Estos clasificadores fueron seleccionados debido a su destacado rendimiento en pruebas experimentales previas, lo que indica su eficacia en la diferenciación precisa entre actividades normales y maliciosas dentro de los sistemas de red. Posteriormente, para mejorar la robustez y la precisión de la detección, se implementa un modelo de votación. Este enfoque de votación consolida los resultados de los clasificadores GB y KNN, permitiendo una decisión final basada en lRFa mayoría, lo que mejora significativamente la capacidad del sistema para reconocer y clasificar correctamente los ataques, minimizando tanto los falsos positivos como los negativos. Este enfoque integrado aprovecha las fortalezas individuales de cada clasificador y proporciona un mecanismo resiliente y adaptativo para la detección de intrusiones en entornos de red dinámicos.

B. Fase Experimentación

Se realizaron las pruebas por cada clasificador por separado, utilizando el modelo de Voting, mencionado en el punto anterior, en la etapa de entrenamiento del clasificador y en la etapa de reconocimiento de ataque se quita la toma de decisiones. Se evaluarán diferentes modelos con las métricas de evaluación, que permitirán cuantificar el rendimiento de los clasificadores, estos son: *accuracy, precision, recall y f1-score*. Para el modelo propuesto, en la clasificación binaria, se utilizan los clasificadores GB y KNN. En la clasificación multiclase se utilizan los clasificadores KNN y RF.

Los datasets utilizados para el entrenamiento de los clasificadores fueron IoTID20, MQTT-IoT-IDS2020 y IoT-DS-2, utilizando un 80 % para el entrenamiento y un 20 % para el testing de los modelos.

- IoTID20: El banco de pruebas para el conjunto de datos IoTID20 [2] es una combinación de dispositivos IoT y estructuras de interconexión. Se implementó en un entorno doméstico inteligente que consiste en un dispositivo doméstico inteligente SKTNGU y una cámara Wi-Fi EZVIZ para generar el conjunto de datos IoTID20. Estos dos dispositivos IoT están conectados a un enrutador Wi-Fi doméstico inteligente. Otros dispositivos conectados al enrutador doméstico inteligente incluyeron computadoras portátiles, tabletas y teléfonos inteligentes. El SKT NGU y la cámara Wi-Fi EZVIZ fueron dispositivos víctimas de IoT y todos los demás dispositivos en el banco de pruebas fueron los dispositivos atacantes.
- Conjunto de datos MQTT-IoT-IDS2020: Fue creado por Hindy et al [18]. Desde la plataforma de red MQTT, este conjunto de datos consiste en tráfico de red regular y ataques de fuerza bruta. La red consta de 12 sensores MQTT, un intermediario, un dispositivo para replicar un flujo de cámara y un atacante. El conjunto de datos incluye los ataques y escenarios MQTT más populares para analizar dispositivos IoT del mundo real. El conjunto de datos MQTT-IoT-IDS2020 contiene cuatro categorías de ataque [19].
- Conjunto de datos IoT-DS-2: Incluye 15 clases de ataque y 1 clase normal. Se puede acceder a la recopilación de datos IoT-DS-2 en [19].

C. Tratamiento de los Dataset

Esta fase se inicia con el preprocesamiento de los datasets, las instancias duplicadas se eliminarán de todos los conjuntos de datos. Después, se normalizarán cada una de las características utilizando el escalado de variables (Escala MinMax). De este proceso resultan datos que se encuentran en rango específico (0, 1), lo cual, elimina valores extremos y acelera significativamente los cálculos.

Después de realizar la normalización, se realiza un reemplazo de los valores Missing Values (Valores no identificados) con 0, en caso de que el proceso previo, haya creado valores inconsistentes. Por último, para las columnas no numéricas (Label y Cat) se reemplazarán su valores de cadena

de caracteres (string) a números enteros (number). En el caso de la columna Label las anomalías están representada por un valor de 1, mientras que un valor normal está representado por 0 en la clasificación binaria.

Para balancear los datasets se utiliza la técnica SMOTE, la cual realiza un sobremuestreo de minorías sintéticas (Synthetic Minority Over-sampling TEchnique - SMOTE) [19] permite aumentar el número de muestras tomando como referencia la clase minoritaria, generando nuevas instancias de las clases. SMOTE no cambia la cantidad de datos de la clase mayoritaria. La creación de las nuevas instancias no son un duplicado de las ya existentes, lo que hace el algoritmo es tomar una muestra al azar del espacio de características de la clase y de sus vecinos más cercanos. Después, se crea un ejemplo sintético en un punto seleccionado al azar entre las dos muestras en el espacio de características. Finalmente, permitiendo el aumento de las muestras para cada clase y hace que las muestras sean más generales.

D. Entorno Experimental

Se utilizaron tres entornos distintos para la parte experimental, los cuales, tienen distintas características computacionales.

En la parte de preprocesamiento de los datos, se utilizó un computador personal con las siguientes capacidades:

- Procesador (CPU): Intel I5-7200U
- Núcleos: 2
- RAM: 12 Gigabytes
- Sistema Operativo: Windows 10

En la etapa experimental se utilizó un servidor externo con las siguientes características:

- Núcleos: 8
- RAM: 15,19 Gigabytes
- Sistema Operativo: Centos

Se utilizó el lenguaje de programación Python 3.8.

E. Resultados

En los resultados ***Clasificación Binaria Balanceada*** (ver Tabla 1), los mejores clasificadores son RF, GB y KNN los que obtienen un f1 score sobre el 99% en todas sus métricas, siendo muy confiables para el dataset IoTID20.

En el dataset MQTT-IoT-IDS2020 los resultados demuestran que el mejor clasificador por lejos es GB con un 98.48% de f1 score, siendo el modelo más confiable. En el dataset IoT-DS-2 los resultados, muestran que la mayoría de los modelos han dado muy buenos resultados, siendo el peor NB con un 86,55% y el mejor GB con un 99.98% de f1 score.

Las demás métricas nos demuestran que los modelos realmente no sufrieron un mayor sobreajuste, dando unos modelos muy confiables y eficientes, mostrando lo eficaz del balanceo de datos con una gran cantidad de datos.

El modelo propuesto con voting obtiene un f1 score promedio en los tres datasets de 95.69%, siendo un modelo confiable con un mayor f1 score del 99.53%. Pero a pesar de ser buen modelo GB tiene un mejor rendimiento en general que el modelo propuesto, dando como resultado un f1 score de 99.7%, 98.48% y 99.98%, siendo más confiable para su uso en clasificación binaria balanceada

Clasificador	Data	Accuracy	Precision	Recall	F1 score	Tiempo [s]
ADABoost	IoTID20	**0.9504**	**0.9505**	**0.9504**	**0.9504**	523.51
ADABoost	MQTT	0.7966	0.8554	0.7964	0.7877	1926.77
ADABoost	IoTDS2	**0.9324**	**0.9399**	**0.9324**	**0.9322**	1252.06
DT	IoTID20	0.8043	**0.9367**	0.6519	0.7688	9.95
DT	MQTT	0.7698	**0.9996**	0.5395	0.7008	36.43
DT	IoTDS2	**0.9258**	**0.9823**	0.8672	**0.9211**	24.41
GB	IoTID20	**0.9970**	**0.9965**	**0.9974**	**0.9970**	113.77
GB	MQTT	**0.9850**	**0.9995**	**0.9705**	**0.9848**	263.49
GB	IoTDS2	**0.9998**	**0.9999**	**0.9998**	**0.9998**	624.41
KNN	IoTID20	**0.9947**	**0.9957**	**0.9935**	**0.9946**	0.41
KNN	MQTT	0.8636	0.8935	0.8253	0.8581	1.39
KNN	IoTDS2	**0.9631**	**0.9973**	**0.9287**	**0.9618**	1.17
NB	IoTID20	0.5756	**0.9362**	0.1607	0.2744	0.95
NB	MQTT	0.7853	**0.9847**	0.5794	0.7296	5.35
NB	IoTDS2	0.8786	**0.968**	0.7822	0.8655	2.50
RF	IoTID20	**0.9971**	**0.9955**	**0.9987**	**0.9971**	2572.12
RF	MQTT	0.7939	**0.9836**	0.5974	0.7433	13692.35
RF	IoTDS2	**0.9347**	**0.9998**	0.8695	**0.9301**	5419.03
SGD	IoTID20	0.5893	0.5518	**0.9429**	0.6962	3.10
SGD	MQTT	0.6923	**0.9700**	0.3964	0.5628	25.37
SGD	IoTDS2	0.8792	**0.9722**	0.7805	0.8659	42.38
Voting	IoTID20	**0.9953**	**0.9953**	**0.9953**	**0.9953**	19.72
Voting	MQTT	**0.9118**	**0.9250**	**0.9117**	**0.9111**	74.48
Voting	IoTDS2	**0.9643**	**0.9667**	**0.9643**	**0.9643**	112.35

Tabla 1: Resultados Clasificación Binaria Balanceada

Los resultados de ***Clasificación de multiclase*** (ver Tabla 2) se muestran que los modelos RF y KNN tienen el mejor rendimiento con un 97% de F1 Score pero solo en el dataset IoTID20, pero al calcular el promedio del rendimiento de estos clasificadores da 75,91% y 83,28, obteniendo KNN un mejor rendimiento general.

Clasificador	Data	Accuracy	Precision	Recall	F1 score	Tiempo [s]
ADABoost	IoTID20	0.7297	0.7378	0.7296	0.7298	582.46
ADABoost	MQTT	0.3616	0.4114	0.3614	0.3063	2074.39
ADABoost	IoTDS2	0.1527	0.2518	0.1525	0.0840	3410.84
DT	IoTID20	0.4755	0.4879	0.4749	0.3907	10.60
DT	MQTT	0.4978	0.4528	0.4979	0.4268	38.44
DT	IoTDS2	0.2352	0.0707	0.2348	0.1043	72.45
GB	IoTID20	0.6327	0.7451	0.6325	0.6313	13376.06
GB	MQTT	0.4392	0.6719	0.4396	0.3968	20117.72
GB	IoTDS2	0.3451	0.3727	0.3449	0.3094	9592.04
KNN	IoTID20	**0.9785**	**0.9788**	**0.9785**	**0.9785**	0.40
KNN	MQTT	0.6947	0.7589	0.6947	0.7009	1.52
KNN	IoTDS2	0.8104	0.8385	0.8103	0.8189	1.58
NB	IoTID20	0.5840	0.6731	0.5841	0.5784	1.10
NB	MQTT	0.4986	0.6409	0.4986	0.4273	4.68
NB	IoTDS2	0.4835	0.5808	0.4830	0.4391	7.71
RF	IoTID20	**0.9799**	**0.9804**	**0.9798**	**0.9799**	2485.16
RF	MQTT	0.6489	0.7361	0.6488	0.6489	9685.09
RF	IoTDS2	0.6509	0.9351	0.6504	0.6485	13326
SGD	IoTID20	0.4130	0.5472	0.4132	0.2910	76.51
SGD	MQTT	0.4058	0.4659	0.4059	0.3408	322.45
SGD	IoTDS2	0.4027	0.3662	0.4024	0.3183	1453.06
Voting	IoTID20	**0.9617**	**0.9633**	**0.9617**	**0.9616**	2524.21
Voting	MQTT	0.6876	0.7500	0.6876	0.6928	10507.97
Voting	IoTDS2	0.7000	0.8493	0.6996	0.6921	13012.95

Tabla 2: Resultados Clasificación de multiclase

Los resultados de cada uno de los modelos disminuyo drásticamente, siendo posible un error de configuración con SMOTE. El rendimiento de los modelos definitivamente no es óptimo, por tanto, es necesario replantear la técnica de balanceo utilizada o buscar otra configuración de los parámetros de los clasificadores.

F. Síntesis de los Resultados

Al realizar el análisis de resultados de la primera etapa experimental al clasificar los datos no balanceados con clase

binaria, se puede deducir que los modelos tienen un alto rendimiento, pero en algunos casos sufren de sobreajustes por el desbalanceo de clases. Los dos mejores modelos en base a sus métricas son GB y KNN, que son utilizados como clasificadores del modelo propuesto, pero el mejor modelo fue GB, obteniendo un mejor rendimiento en los tres datasets. Al clasificar los datos balanceados con SMOTE con clase binaria, el rendimiento de los clasificadores bajo en comparación a la clasificación binaria sin el balanceo de datos.

Los clasificadores que más destacaron fueron GB y KNN, que son utilizados como clasificadores del modelo propuesto con voting, pero se repite que GB sobresale a los demás en la clasificación binaria. Al clasificar los datos no balanceados multiclase, el rendimiento de los modelos empezó a disminuir, una de las causas puede ser el desbalanceo de las clases, habiendo clases que están muy poco representadas en los datasets. A pesar de esto, los clasificadores RF y KNN destacaron, pero solo en el dataset IoTID20, siendo utilizados como clasificadores del modelo propuesto con voting, pero en general KNN tiene un mejor rendimiento. Al clasificar datos balanceados con SMOTE multiclase, el rendimiento en general de los clasificadores disminuyó drásticamente. Esto puede deberse a una mala configuración de los parámetros en los modelos para una clasificación multiclase o a la técnica de balanceo SMOTE que hizo que se solaparan ciertos tipos de datos, haciendo que los clasificadores tengan un bajo rendimiento al no clasificar correctamente. El modelo que destacó fue KNN teniendo el mejor rendimiento de todos.

VII. Conclusiones

El crecimiento exponencial de las redes IoT ha impulsado significativos avances en diversos sectores de la sociedad, integrándose como un componente esencial en nuestro presente y futuro. Sin embargo, esta integración masiva también introduce múltiples vectores de amenazas, demandando soluciones innovadoras para garantizar la seguridad de estas redes. Los Sistemas de Detección de Intrusos (IDS) basados en técnicas de Aprendizaje Automático (ML) y Aprendizaje Profundo (DL) emergen como soluciones prometedoras, dada su capacidad para no solo detectar amenazas conocidas sino también adaptarse y reconocer nuevas vulnerabilidades, como los ataques de día cero.

El documento establece una metodología de investigación rigurosa, reportando un avance completo en la revisión del estado del arte. Se exploran las distintas arquitecturas IoT, destacando la ausencia de un estándar universal y la diversidad de protocolos y tecnologías que facilitan la interconexión de dispositivos. Se identifican las amenazas prevalentes en un marco de arquitectura IoT de cuatro capas y se profundiza en la taxonomía de los IDS, incluyendo una revisión de las técnicas de ML y DL aplicables, los conjuntos de datos relevantes para IoT y un análisis de investigaciones previas que abordan la detección de intrusos en este contexto.

El enfoque inicial de votación no cumplió las expectativas, pero orientó la formulación de una metodología efectiva evaluada mediante los conjuntos de datos IoTID20, MQTT-IoT-IDS2020, e IoT-DS-2. Se propone un modelo híbrido bifásico: la primera fase emplea un clasificador de Gradiente Boosting (GB) para filtrar actividades normales de las anómalas, alcanzando altas puntuaciones F1 en los conjuntos de datos probados. Las anomalías detectadas avanzan a una segunda fase para una clasificación detallada, utilizando técnicas avanzadas como RFECV para la selección de características y SMOTE para el equilibrio de datos, culminando en la clasificación mediante KNN. Este modelo híbrido demuestra un desempeño sobresaliente en la detección y clasificación de anomalías.

La consistencia en el desempeño entre los conjuntos de datos sugiere la robustez del modelo, aunque se identificaron limitaciones en la calidad de los datos de MQTT-IoT-IDS2020. La eficacia del sobremuestreo (SMOTE) en la clasificación multiclase contrasta con el submuestreo (RUS), que, pese a simplificar el entrenamiento, compromete la precisión del modelo. Investigaciones futuras se enfocarán en validar el modelo con conjuntos de datos adicionales, explorar otras técnicas de equilibrio de datos, integrar métodos de selección de características y modelos de clasificación alternativos, y finalmente, implementar y verificar el modelo híbrido en escenarios reales para abordar la detección de anomalías desconocidas de manera efectiva.

Como futuras líneas de investigación, se propone la validación del modelo propuestos con otros casos de uso pertenecientes a contexto de uso más sensible, como es el caso de los sistemas ciberfísicos aplicados al área de la salud. También sería de mucho interés, el análisis detallado de los algoritmos ML y DL utilizados, a fin de mejorar su perfomance y efectividad.

VIII. Referencias

[1] Flores F. and Cossio E. Aplicaciones, enfoques y tendencias del internet de las cosas (IoT): revisión sistemática de la literatura. Academia Journals, 2021. http://ciateq.repositorioinstitucional.mx/jspui/handle/1020/543

[2] Ullah Imtiaz and Mahmoud Qusay. A scheme for generating a dataset for anomalous activity detection in iot networks. pages 508–520, 2020. Department of Electrical, Computer and Software Engineering, Ontario Tech University, Oshawa, ON L1G 0C5, Canada.

[3] Hasan Alkahtani and Theyazn H. H. Aldhyani. Intrusion Detection System to Advance Internet of Things Infrastructure-Based Deep Learning Algorithms. Complexity, 2021(Article ID 5579851):18, 2021. https://doi.org/10.1155/2021/5579851.

[4] Yang Li, Manias, Dimitrios Michael, and Shami Abdallah. Pwpae: An ensemble framework for concept drift adaptation in IoT data streams. 2021 IEEE Global Communications Conference (GLOBECOM), page 6, 2021. http://dx.doi.org/10.1109/GLOBECOM46510.2021.9685338.

[5] V. Ishanoglu, C. Covarrubias, R. Rivera, H. Espinoza, and J. Millán. Panorama de las Smart Cities y la ciberseguridad. CSIRT,13:32,2021. https://www.csirt.gob.cl/media/2022/01/CSIRT-CiberSucesos- 2021-11-12.pdf.

[6] Doohwan Oh, Deokho Kim, and Won Woo Ro. A malicious pattern detection engine for embedded security systems in the internet of things. Sensors (Basel, Switzerland), 14:24188–24211, 12 2014.

[7] Marwa Keshk, Nour Moustafa, Elena Sitnikova, and Benjamin Turn- bull. Privacy-preserving big data analytics for cyber-physical systems. Wireless Networks, 28, 04 2022.

[8] Robert Mitchell and Ing-Ray Chen. A survey of intrusion detection techniques for cyber-physical systems. ACM Computing Surveys (CSUR), 46, 03 2014.

[9] Debar, H. (2000). An introduction to intrusion-detection systems. Proceedings of Connect, 2000.

[10] Anoop and Sunil Rai. A COMPARATIVE STUDY OF DIFFERENT TYPES OF NETWORKS IJIRT, 1:5, 2014. https://ijirt.org/master/publishedpaper/IJIRT101024_PAPER.pdf.

[11] M. Benaiah Deva Kumar and B. Deepa. Computer Networking: A Survey. IJTRD, 2(5):5, 2015. https://www.researchgate.net/publication/317101504_ComputerNetworking_A_Survey

[12] Ashraf Javed, Moustafa Nour, Khurshid Hasnat, Debie Essam, Haider Waqas, and Wahab Abdul. A review of intrusion detection systems using machine and deep learning in Internet of things: Challenges, solutions and future directions. Electronics, 9, 07 2020.

[13] Karen Scarfone and Peter Mell. Guide to Intrusion Detection and Prevention Systems (IDPS), volume 800. 01 2007.Nist National Institute os Standards and Technology.

[14] João P. Amaral, Luís M. Oliveira, Joel J. P. C. Rodrigues, Guangjie Han, and Lei Shu. Policy and network-based intrusion detection system for ipv6-enabled wireless sensor networks. In 2014 IEEE International Conference on Communications (ICC), pages 1796–1801, 2014.

[15] Sethi Pallavi and Sarangi Smruti. Internet of things: Architectures, protocols, and applications. Journal of Electrical and Computer Engineering,2017:1–25, 01 2017.

[16] Ashraf Javed, Moustafa Nour, Khurshid Hasnat, Debie Essam, Haider Waqas, and Wahab Abdul. A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions. Electronics, 9, 07 2020.

[17] Ismail Butun, Salvatore Morgera, and Ravi Sankar. A survey of intrusion detection systems in wireless sensor networks. IEEE Communications Surveys Tutorials, PP:266 – 282, 05 2013.

[18] Shahid Raza, Linus Wallgren, and Thiemo Voigt. Svelte: Real-time intrusion detection in the internet of things. Ad Hoc Networks, 11, 05 2013.

[19] G. D'Agostini. A multidimensional unfolding method based on bayes' theorem. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 362(2):487–498, 1995.

[20] Andrew Ng and Michael Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, Advances in Neural Information Processing Systems, volume 14. MIT Press, 2001.

[21] Vikas Hassija, Vinay Chamola, Vikas Saxena, Divyansh Jain, Pranav Goyal, and Biplab Sikdar. A survey on IoT security: Application areas, security threats, and solution architectures. IEEE Access, PP:1–1, 06 2019.

[22] P. Soucy and G.W. Mineau. A simple KNN algorithm for text categorization. In Proceedings 2001 IEEE International Conference on Data Mining, pages 647–648, 2001.

[23] Xue-wen Chen and Jong Cheol Jeong. Enhanced recursive feature elimination. In Sixth International Conference on Machine Learning and Applications (ICMLA 2007), pages 429–435, 2007.

[24] Sotiris Kotsiantis. Supervised machine learning: A review of classification techniques. Informatica (Slovenia), 31:249–268, 01 2007.

[25] Wenliang Du and Zhijun Zhan. Building decision tree classifier on private data. 01 2003.

[26] J. R. Quinlan. Induction of decision trees. Mach. Learn., 1(1):81–106, March 1986.

[27] Salem Alharbi, Peter Rodriguez, Rajaputhri Maharaja, Prashant Iyer, Nivethitha Subaschandrabose, and Zilong Ye. Secure the internet of things with challenge response authentication in fog computing. In 2017 IEEE 36th International Performance Computing and Commu- nications Conference (IPCCC), pages 1–2, 2017.

[28] Hanan Hindy, David Brosset, Ethan Bayne, Amar Kumar Seeam, Christos Tachtatzis, Robert Atkinson, and Xavier Bellekens. A taxonomy of network threats and the effect of current datasets on intrusión detection systems. IEEE Access, 8:104650–104675, 2020.

[29] Pablo Torres, Carlos Catania, Sebastian Garcia, and Carlos Garcia Garino. An analysis of recurrent neural networks for botnet detection behavior. In 2016 IEEE Biennial Congress of Argentina (ARGEN- CON), pages 1–6, 2016.

[30] Muder Almi'ani, Alia Abughazleh, Amer Al-rahayfeh, Saleh Atiewi, and Abdul Razaque. Deep recurrent neural network for IoT intrusion detection system. Simulation Modelling Practice and Theory, 101:102031, 11 2019.

[31] Tian Guo, Zhao Xu, Xin Yao, Haifeng Chen, Karl Aberer, and Koichi Funaya. Robust online time series prediction with recurrent neural networks. In 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pages 816–825, 2016.

[32] Yao Qin, Dongjin Song, Haifeng Cheng, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. 04 2017.

[33] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short-term memory networks for anomaly detection in time series. 04 2015.

[34] Niall McLaughlin, Jesus Martinez del Rincon, BooJoong Kang, Suleiman Yerima, Paul Miller, Sakir Sezer, Yeganeh Safaei, Erik Trickel, Ziming Zhao, Adam Doupé, and Gail Joon Ahn. Deep android malware detection. In Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy, CODASPY '17, page 301–308, New York, NY, USA, 2017. Association for Computing Machinery.

[35] Sudeendra Kumar K, Sauvagya Sahoo, Abhishek Mahapatra, Ayas Kanta Swain, and K.K. Mahapatra. Security enhancements to system on chip devices for iot perception layer. In 2017 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS), pages 151–156, 2017.

[36] D.E. Denning. An intrusion-detection model. IEEE Transactions on Software Engineering, SE-13(2):222–232, 1987.

[37] Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye, and Yiqiang Sheng. Malware traffic classification using convolutional neural network for representation learning. In 2017 International Conference on Information Networking (ICOIN), pages 712–717, 2017.

[38] Hongyu Liu and Bo Lang. Machine learning and deep learning methods for intrusion detection systems: A survey. Applied Sciences, 9:4396, 10 2019.Mohammed Ali Al-Garadi, Amr Mohamed, Abdulla Khalid Al-Ali, Xiaojiang Du, Ihsan Ali, and Mohsen Guizani. A survey of machine and deep learning methods for Internet of things (IoT) security. IEEE Communications Surveys Tutorials, 22(3):1646–1685, 2020.

[39] Sahil Garg, Kuljeet Kaur, Neeraj Kumar, Georges Kaddoum, Albert Y. Zomaya, and Rajiv Ranjan. A hybrid deep learning-based model for anomaly detection in cloud datacenter networks. IEEE Transactions on Network and Service Management, 16(3):924–935, 2019.

[40] Mahmood Yousefi-Azar, Vijay Varadharajan, Len Hamey, and Uday Tupakula. Autoencoder-based feature learning for cyber security applications. In 2017 International Joint Conference on Neural Networks (IJCNN), pages 3854–3861, 2017.

[41] Yisroel Mirsky, Tomer Doitshman, Yuval Elovici, and Asaf Shabtai. Kitsune: An ensemble of autoencoders for online network intrusion detection. CoRR, abs/1802.09089, 2018.

[42] Mayuranathan Mani, Murugan Mahalingam, and V. Dhanakoti. Best features-based intrusion detection system by rbm model for detecting ddos in cloud environment. Journal of Ambient Intelligence and Humanized Computing, 12, 03 2021.

[43] Ugo Fiore, Francesco Palmieri, Aniello Castiglione, and Alfredo Santis. Network anomaly detection with the restricted boltzmann machine. Neurocomputing, 122:13–23, 12 2013.

[44] Marwa Keshk, Elena Sitnikova, Nour Moustafa, Jiankun Hu, and Ibrahim Khalil. An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems. IEEE Transactions on Sustainable Computing, 6(1):66–79, 2021.

[45] Yuanfang Chen, Yan Zhang, and Sabita Maharjan. Deep learning for secure mobile edge computing. CoRR, abs/1709.08025, 2017.

[46] Yuancheng Li, Rong Ma, and Runhai Jiao. A hybrid malicious code detection method based on deep learning. International Journal of Software Engineering and Its Applications, 9:205–216, 05 2015.

[47] Robert E. Hiromoto, Michael Haney, and Aleksandar Vakanski. A secure architecture for IoT with supply chain risk management. In 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), volume 1, pages 431–435, 2017.

[48] Yuyang Zhou, Guang Cheng, Shanqing Jiang, and Mian Dai. Building an efficient intrusion detection system based on feature selection and ensemble classifier. Computer Networks, 174, 06 2020.

[49] Hanan Hindy, Ethan Bayne, Miroslav Bures, Robert Atkinson, Christos Tachtatzis, and Xavier Bellekens. Machine Learning Based IoT Intrusion Detection System: An MQTT Case Study (MQTT-IoTIDS2020 Dataset), pages 73–84. 01 2021.

[50] I. Ullah and Q.H Mahmoud. In IoT Intrusion Detection Datasets., 2021. Disponible online: https://sites.google.com/view/iotdataset1 (Accedido el 6 de Julio del 2023).

[51] Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. CoRR, abs/1106.1813, 2011.

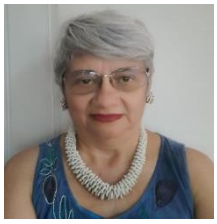
[52] Marwa Keshk, Benjamin Turnbull, Nour Moustafa, Dinusha Vatsalan, and Kim-Kwang Raymond Choo. A privacy-preserving-framework-

based blockchain and deep learning for protecting smart power networks. IEEE Transactions on Industrial Informatics, 16(8):5110–5118, 2020.
[53] Caiming Liu, Jin Yang, Run Chen, Yan Zhang, and Jinquan Zeng. Research on immunity-based intrusion detection technology for the internet of things. In 2011 Seventh International Conference on Natural Computation, volume 1, pages 212–216, 2011.
[54] Prabhakaran Kasinathan, Claudio Pastrone, Maurizio A. Spirito, and Mark Vinkovits. Denial-of-service detection in 6lowpan based internet of things. In 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pages 600–607, 2013.
[55] Prabhakaran Kasinathan, Gianfranco Costamagna, Hussein Khaleel, Claudio Pastrone, and Maurizio Spirito. Demo: An ids framework for internet of things empowered by 6lowpan. pages 1337–1340, 11 2013.
[56] Arenas-Hoyos, S.A., Bernal-Noreña, Á. Support vector machines implementation over integers modulo-M and Residue Number System.
[57] Longfei Sun;Yanhui Liu;Yuanjian Wang;Qinghao Dong;Wanjie Zhao, Analysis of characteristic index and prediction of river bottom tearing scour in the Yellow River.
[58] S. Gomez, M. Martinez, y J. Lopez, "Eficiencia del clasificador SGD en grandes conjuntos de datos," *Journal of Machine Learning Research*, vol. 15, no. 4, pp. 1123-1145, Jul. 2023.

IX. BIOGRAFÍAS

Dr.(c)Sebastián Berrios nacido el 7 diciembre de 1986.Actualmente candidato a doctor en Ingeniería informática en la Pontificia Universidad Católica de Valparaíso. Graduado de Ingeniería Civil en Computación e Informática en la Universidad De Las Américas. Magíster en Ciencias de la Ingeniería y Magíster en Ingeniería en Informática en la Pontificia Universidad Católica de Valparaíso. Actualmente docente del área de Ciberseguridad y Administrador de TI de la escuela de ingeniaría informática de la Pontificia Universidad Católica de Valparaíso. Cursando un Diplomado de Inclusión en Educación en la Pontificia Universidad Católica de Valparaíso, con una duración de 140 horas. Además, se cursó un Diplomado en Seguridad de la Información de 132 horas en la Universidad de Chile y un Diplomado en Ciberseguridad de 96 horas en la misma universidad. También se obtuvo un Diplomado en Ciberseguridad de 100 horas en el Instituto Profesional IACC.

Pamela Hermosilla nacida en Valparaíso, Chile, el 8 de octubre de 1974. Graduada de Universidad Téecnica Federico Santa Naría (UTFM Ingeniero Civil en Informática (UTFSM), Diplomada en Comercio Electrónico y Logística Empresarial (UTFSM), Auditor Interno ISO 9001 (Brain & Cia Consultores), MBA of Chief Information Officer CIO (Abet Open University), Diplomada en Docencia Universitaria de la Pontificia Universidad Católica de Valparaíso (PUCV), Diplomada en Formación Virtual Universitaria (PUCV), Symposium for Entrepreneurship Educators (Luksic Scholars – Babson College). Asesorías: miembro del Consejo Público Privado de la red Fortalece Pyme, Valparaíso - Corfo, integrante del Boardd de Directores, de la incubadora Chrisalys PUCV. Desarrollo profesional en áreas de Aseguramiento de calidad en gestión de proyectos, Planificación estratégica organizacional, Rediseño curricular basado en competencias, Habilidades de Innovación y emprendimiento en estudiantes de ingeniería, Gamificación en el proceso de enseñanza y aprendizaje.



Herminia Beatriz Parra de Gallo, Especialista en Informática Forense, Dra. en Ingeniería Mención Sistemas de Información, Master en Administración de Negocios, Ingeniería en Computación. Con una extensa carrera académica como docente e investigadora en la UCASAL reviste la categoría de INVESTIGADOR INDEPENDIENTE “B” (CI-UCASAL), y es Directora del Grupo de I+D+i de Forensia Digital de esa institución, grupo abocado al desarrollo de proyectos de i+d interinstitucionales con universidades nacionales y latinoamericanas. Es además, Directora del Instituto de Estudios Interdisciplinarios de Ingeniería de la Facultad de Ingeniería de la UCASAL y Directora de la carrera de Especialización en Administración de Bases de Datos. Participa en la formación de recursos humanos impartiendo cursos de posgrado y como docente de carreras de grado y posgrado en UFASTA (Argentina), UTN-FRSF (Argentina), UTN-FRC (Argentina), UNIVA (México), UNISANGIL (Colombia), Universidad Católica de Colombia (Colombia), UniTECH (España). Perito Informático de Parte. Consultora en Proyectos Tecnológicos Críticos.Sus temáticas de interés son: Forensia Digital, Educación en Ingeniería y Ética en la Ingeniería.

Pablo Oñate Ingeniero civil informático con el grado de magister en ingeniería informática en la Pontificia Universidad Católica de Valparaíso, cuenta con amplio conocimiento de machine learning y matemática estadística para el análisis de datos y creación de modelos predictivos.

.