PROYECTO DE GESTIÓN DE REDES

Gestión de redes IoT

Alejandro Muñoz García
Miguel García Menchén
Iván López Sánchez

Marzo 2025

Índice

Lista de Figuras

Lista de Tablas

1 Introducción

El Internet de las Cosas o Internet of Things (IoT) ha provocado una revolución en los últimos tiempos en la manera con la que los seres humanos interactúan con la tecnología, permitiendo la interconexión de diferentes tipos de dispositivos para recopilar información, analizarla y poder compartir distintos datos en tiempo real. IoT está caracterizado por el uso de sensores y plataformas en la nube que se encargan de facilitar la comunicación entre dispositivos, sin que tengan que intervenir personas en dicho proceso.

En el ámbito de la salud, la aplicación del Internet de las Cosas ha dado lugar al llamado IoMT, Internet de las Cosas Médicas, el cual es una red de distintos dispositivos especializados que están interconectados para poder recopilar datos médicos en tiempo real que sean capaces de mejorar la atención médica.

También, cabe destacar que este constante crecimiento del IoMT ha hecho que traiga consigo importantes desafíos en materia de seguridad, ya que, al estar interconectados unos dispositivos con otros, estos se convierten en objetos un tanto vulnerables a posibles ciberataques. Entre los principales ataques, podemos encontrar ataques de denegación de servicio (DoS), malwares, manipulación de los datos y el posible acceso a información de total privacidad. Por lo que la seguridad en estos dispositivos se ha vuelto un aspecto crítico y a la orden del día.

Para abordar este tipo de riesgos, se han ido desarrollando diferentes métodos, así como protocolos de seguridad, cifrado de conexión para la transmisión de los datos y la introducción de la inteligencia artificial para poder ser capaces de determinar ciertos tipos de anomalías.

Por último, los modelos de aprendizaje autómatico que aplican técnicas de *Machine Learning* juegan un papel fundamental, ya que pueden ser utilizados para clasificar tráfico en la red, analizar diferentes patrones del tráfico en la red y con ello ser capaces de detectar comportamientos atípicos de cara a interferir en tiempo real ante ataques.

2 Estado del Arte

2.1 Entorno y dispositivos IoMT

En los últimos tiempos, el Internet de las Cosas (IoT) ha tenido un desarrollo muy significativo en diversos ámbitos de la industria, transformando de manera notable la manera en que los usuarios y los dispositivos intercambian información entre sí. Esta revolucionaria tecnología ha supuesto un impacto significativo en diversas áreas que van desde la domótica en el hogar, la industria tecnológica, la agricultura y la salud, siendo esta la última el tema a tratar en el proyecto.

A continuación, se mencionan varios de los ámbitos más influyentes en el mundo de IoT, y cuál ha sido su desarrollo para poder optimizar diferentes procesos y mejorar la calidad de vida de los usuarios que los utilizan.

- **Hogar Inteligente**: En este ámbito, la implantación de dichos dispositivos ha mejorado la seguridad, comodidad, así como la eficiencia en la reducción del gasto de energía. Dispositivos de asistentes virtuales como *Alexa* o *Google Home* son capaces de realizar un control automático de diferentes tareas del hogar. La instalación de termostatos inteligentes e iluminación automatizada ayuda a rebajar los costes de la energía. Por último, hay que destacar los sistemas de seguridad, que incluyen sensores y cerraduras electrónicas.

- **Industria y Automatización**: En cuanto al sector industrial, el IoT ha conseguido una optimización de los procesos, un análisis predictivo e incrementar la eficiencia operativa. En este aspecto, existen numerosos robots (cobots) que trabajan en cooperación con los humanos para facilitar dichas tareas. Así como diferentes máquinas usadas en los sistemas de logística y en los controles de calidad, que proporcionan una automatización significativa.

- **Ciudades Inteligentes**: En este sentido, el IoT aplicado en las zonas urbanas ha sido de gran ayuda para mejorar la calidad de vida de los propios usuarios. Cabe destacar la ayuda en tareas como gestión del tráfico o residuos, encendido y apagado de los sistemas de alumbrado público según qué circunstancias de luz haya en cada ubicación y el de control de calidad del aire.

- **Agricultura**: En dicho sector, el IoT ha ayudado a la optimización de cultivos y gestión de recursos, esto con ayuda de sensores de humedad, drones agrícolas para la desinfección de plagas y sistemas de riego automatizados.

- **Salud**: En cuanto a este último sector, que es el que nos ocupará durante este proyecto, es uno de los ámbitos donde más avances se han producido. Esto debido a la implantación de diversos dispositivos tales como sistemas de telemedicina, sistemas de monitorización de glucosa, pulso cardíaco y desfibriladores inteligentes.

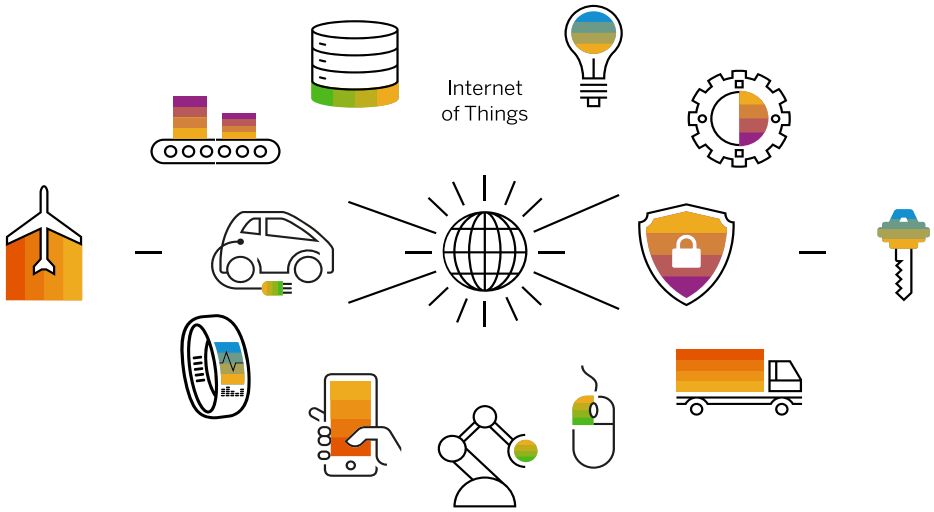


Figure 2.1: Ilustración de dispositivos IoT [1].

2.2 Ataques

Los dispositivos que forman un entorno IoT son muy vulnerables a amenazas cibernéticas, debido a su escasa seguridad. En la mayoría de los casos, estas amenazas son ataques informáticos dirigidos hacia una red de dispositivos IoT con el fin de obtener información útil y privada con la que sacar beneficio o directamente inhabilitar el funcionamiento de estos dispositivos. Para hacer frente a cualquier tipo de ataque y mejorar la seguridad de los dispositivos en un entorno IoT, se desarrollan infinidad de aplicaciones basadas en la detección de ataques con el objetivo de poder detectar el ataque en tiempo real y poder hacerle frente de la mejor manera posible.

A continuación, se explican los cinco tipos de ataques utilizados por el CIC en su investigación de CICIoMT2024, siendo estos algunos de los ataques que pueden comprometer la seguridad de innumerables dispositivos en la actualidad.

1. **Ataque de Denegación de Servicio Distribuido (DDoS)**: Se considera uno de los más frecuentes hoy en día en IoMT. Se basa en sobrecargar un servidor de red o algún dispositivo con un volumen de tráfico excesivo proveniente de diversas fuentes. Normalmente, este tipo de ataques se llevan a cabo mediante las llamadas *botnets*, redes de ordenadores infectadas con *malware* y que están controladas por el atacante. En la figura 2.2 se presenta un esquema de un ataque DDoS, en el que el atacante está haciendo uso de una botnet para perpetrar un tipo de ataque y conseguir hacerse con el control de varios equipos informáticos.

2. **Ataque de Denegación de Servicio (DoS)**: Este y el anterior son del mismo tipo, pero en vez de venir desde diversos ordenadores, el ataque se realiza desde un único ordenador, con la intención de mandar una gran cantidad de peticiones, saturando el ordenador o el servidor, dejándolo inhabilitado al no poder atender todas las peticiones.

3. **Ataques de Reconocimiento (Recon)**: Previo a realizar el ataque sobre cualquier tipo de dispositivo, son usados para recabar información sobre diversos sistemas, así como redes, antes de que se realice dicho ataque. Estos últimos van dirigidos al escaneo de diferentes puertos o identificación de servicios.

4. **Ataques de suplantación de identidad (Spoofing)**: Se basa en que un atacante falsifica la identidad para tratar de engañar a un dispositivo y conseguir así un acceso a todos los datos que haya dentro de él. Puede ser usado tanto a nivel de direcciones IP como de MAC. Es uno de los más peligrosos, ya que puede acceder a información sensible y recabar información de vital importancia de los diferentes dispositivos a los que los usuarios tengan acceso.

5. **Ataques con paquetes malformados**: Este tipo de ataques consiste en que el atacante utiliza diversos equipos 'zombis' que lo que hacen es enviar al usuario multitud de paquetes mal formados para así tratar de bloquear dicho equipo o dispositivo. Hay diversas formas de poder realizar este tipo de ataques, pero el más habitual es mediante un software que se dedica a crear paquetes de este tipo para diversos protocolos y los distribuye de forma abundante a los dispositivos de los usuarios [2].

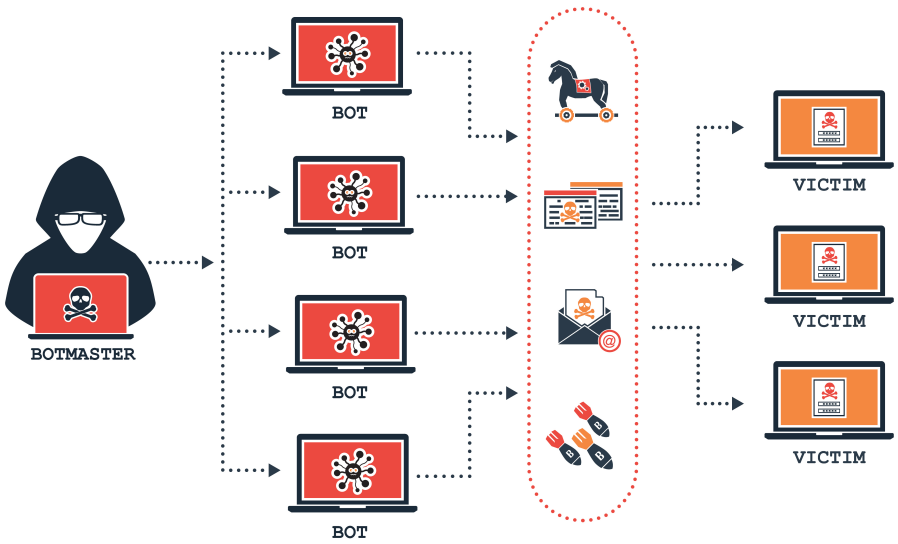


Figure 2.2: Esquema de un ataque DDoS con botnet [3].

2.3 Protocolos

Los protocolos son un conjunto de normas y reglas que permiten la comunicación entre dispositivos dentro de una red. Por ello, son fundamentales para que exista una comunicación entre los dispositivos de cualquier entorno IoT, ya que si estos, los dispositivos, no podrían intercambiar información de manera organizada y segura.

Una correcta implementación permite garantizar la integridad de los datos y mejorar las numerosas vulnerabilidades que se enfrentan, comentadas anteriormente. A continuación, se comentan tres de los protocolos utilizados en la investigación CICIoMT2024, su funcionamiento y cómo se han repartido entre los dispositivos desplegados en el entorno IoMT simulado por el CIC.

2.3.1 Bluetooth

Es un protocolo que permite conectar dos dispositivos de manera inalámbrica, a corta distancia, con la intención de intercambiar datos e información entre ellos de forma segura. La radiofrecuencia que utilizan estos dispositivos para estar conectados es de 2.4 GHz. Esta tecnología permite el desarrollo de redes inalámbricas como la que propone el CIC en su investigación con un entorno médico. Una de las situaciones más comunes en la que se utiliza el Bluetooth es cuando dos individuos se encuentran en una misma estancia y desean intercambiar fotografías que tienen en sus teléfonos móviles. Para este caso, hacen uso de este protocolo para realizar el intercambio de forma rápida y segura [4].

El CIC utilizó para CICIoMT2024 14 dispositivos Bluetooth sobre los que desplegaría tres ataques diferentes: DoS, Malformed y Recon, con la intención de evaluar la respuesta de los dispositivos IoMT a este tipo de ataques.

2.3.2 Wi-Fi

Wi-Fi es otro protocolo de comunicación inalámbrica que también permite el conexionado entre dispositivos. Este permite la conexión a través de las bandas 2.4 GHz, como Bluetooth, y 5 GHz, facilitando de igual manera el intercambio de datos e información de manera segura y rápida. Gracias a esta tecnología, la creación de redes locales sin necesidad de cables es posible, permitiendo un amplio radio de conectividad. Por ello, es usado en diferentes entornos como un hogar, una empresa, universidades y centros médicos u hospitales. Para estos dos últimos entornos, el protocolo Wi-Fi tiene un uso común, ya que numerosos dispositivos que conforman un ámbito sanitario lo usan para estar conectados a una red sanitaria y con ello controlar de manera remota datos vitales de los pacientes, como pueden ser las pulsaciones cardíacas en tiempo real.

Para el CICIoMT2024, se emplearon 7 dispositivos Wi-Fi, entre los que se encuentran botones de ayuda y cámaras de vigilancia para niños pequeños. Sobre estos se lanzaron ataques de tipo: DDoS, DoS, Spoofing y Recon. Nuevamente con la intención de evaluar la seguridad de estos dispositivos, estudiar sus vulnerabilidades y así poder mejorarla.

2.3.3 MQTT

Este protocolo de tipo mensajería basado en estándares es utilizado para la comunicación entre dos equipos. En el ámbito de las IoT, estos dispositivos tienen que transmitir y recibir los datos. Esta transmisión y recepción de datos se hace a través de MQTT, ya que resulta muy fácil de implementar.

Dicho protocolo se inventó en el año 1999, en concreto para su uso dentro de la industria del petróleo. Esto debido a que los ingenieros necesitaban un ancho de banda que fuera muy bajo y una degradación de la batería que fuera también escasa. Fue en 2010 cuando la empresa IBM lanzó MQTT 3.1 como un protocolo gratuito y abierto para que el resto de los usuarios pudieran aplicarlo e implementarlo.

Los diferentes beneficios que ofrece este protocolo son algunos de los siguientes:

- **Efectividad**: Los recursos que requiere para implementarse en los dispositivos IoT son mínimos y, por lo tanto, puede usarse en cualquier tipo de equipo.
- **Fiabilidad**: Posee funciones para poder reconectarse de forma rápida a la nube ante una posible caída o un posible ataque. Además de establecer tres niveles distintos de calidad según el servicio a proporcionar.
- **Seguridad:** Contribuyendo con el cifrado de mensajes y también la capacidad de autenticación de los distintos equipos, esto gracias al uso de protocolos de última generación como TSL, diferentes certificados, etc.

Con respecto a los distintos componentes que conforman MQTT. Podemos encontrar:

- **Clientes**: Es decir, cualquier dispositivo que implemente alguna funcionalidad de MQTT. Desde servidores hasta dispositivos inteligentes
- **Agente**: Se encarga de la sincronización y coordinación de los mensajes entre todos los clientes que haya. Algunas de las funciones que se incluyen pueden ser filtrado y recepción de paquetes. También se encargará del traspaso de información entre sistemas posteriores y del control de posibles mensajes que se hayan podido perder entre clientes.
- **Conexión**: La conexión entre clientes se realiza a través de dicho protocolo; uno de ellos empieza con una solicitud en la que solicita conectarse con el otro y el agente se encarga de confirmar dicha conexión. Ambos necesitan una pila TCP o IP para poder establecer todo el proceso de comunicación.

Con respecto a la seguridad, que es uno de los principales objetivos de dicho proyecto, MQTT usa el protocolo SSL para poder proteger de manera confiable los datos de los dispositivos IoT. Además, tiene capacidad de autenticación y autorización entre agente y clientes mediante certificados del protocolo mencionado anteriormente. Así mismo, el agente añade contraseñas a los identificadores de cada cliente. Por el otro lado, el cliente autentica el servidor con otros certificados. Es por esto por lo que MQTT ayuda a incrementar la seguridad de los dispositivos IoT, en especial en el del ámbito médico, que es el que nos abarca [5].

2.4 *Random Forest*

En el campo del aprendizaje automático, se distinguen dos modalidades: el supervisado y el no supervisado. El primero se aplica cuando se etiqueta el conjunto de datos, en cambio, para el segundo no se emplean etiquetas. De acuerdo con lo establecido previamente, se utilizará el algoritmo de aprendizaje automático supervisado *Random Forest* como base de uno de los modelos predictivos a desarrollar en el presente proyecto.

Es un algoritmo de aprendizaje automático supervisado que funciona mediante la unión y puesta en marcha de varios *Decision Trees*, otro algoritmo de aprendizaje automático. El objetivo de un *Random Forest* es crear un bosque aleatorio de árboles de decisión. Cada árbol se entrena con un subconjunto del conjunto de datos original. De esta manera se asegura que cada árbol del bosque a crear. En base a la decisión final que tome cada árbol, se recopilan todas las respuestas y la que más veces se repita será el resultado final del algoritmo [6].

Random Forest se encargará de clasificar el tráfico capturado por el CIC en el entorno IoT y con ello detectar los diferentes ataques que hayan realizado. El motivo de su elección se debe a los buenos resultados obtenidos por el CIC en su investigación, ya que, en comparación con los otros 3 algoritmos que utilizan, es el que mejores resultados obtiene. El resto de algoritmos que utilizó el CIC para evaluar su conjunto de datos son: *LogisticRegression, AdaBoost* y *DeepNeuralNetwork*.

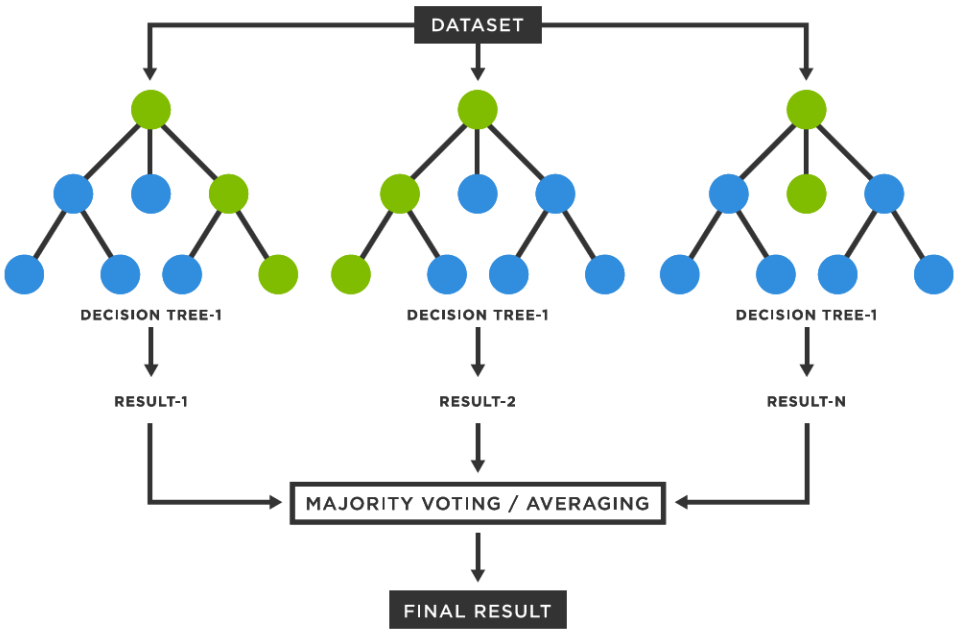


Figure 2.3: Representación del funcionamiento algoritmo Random Forest [6].

2.5 Red Neuronal

Las redes neuronales (NN: Neural Networks) son un tipo de modelo de computación inspirado en la estructura y funcionamiento del cerebro humano. Están compuestas de neuronas artificiales interconectadas que se organizan en capas. Cada una toma una entrada, realiza una computación y devuelve un resultado, que pasa a otras neuronas en otras capas [7].

El primer modelo de red neuronal fue propuesto en 1943 por McCulloch y Pitts en términos de un modelo computacional de actividad nerviosa. En aprendizaje automático son muy útiles para analizar y reconocer patrones en los datos. Se pueden entrenar sobre conjuntos de datos etiquetados para realizar tareas de clasificación, regresión o agrupación en clusters.

Existen diversos tipos de redes neuronales, como las Redes Neuronales Feedforward (FNN), en las que la información fluye en una única dirección, las Redes Neuronales Convolucionales (CNN) y las Redes Neuronales Recurrentes (RNN). Para el presente proyecto, utilizaremos una Red Neuronal Profunda (DNN), que usa varias capas ocultas, las cuales aportan más robustez al entrenamiento [8].

2.6 *Adaptative Boosting*

AdaBoost (Adaptive Boosting) es un algoritmo de aprendizaje automático que mejora la precisión combinando varios clasificadores débiles para obtener un clasificador más fuerte. Funciona asignando más peso a los datos mal clasificados en cada iteración, haciendo que los modelos siguientes se enfoquen en corregir esos errores. Destaca por su capacidad de evaluar la confianza en la clasificación de cada muestra a través de los márgenes de decisión y por minimizar la función de pérdida exponencial. Aunque es eficaz y menos propenso al sobreajuste en ciertos casos, es sensible a datos ruidosos o atípicos [9].

2.7 Regresión Logística

La Regresión Logística es un algoritmo supervisado estadístico empleado para analizar y predecir el resultado de una variable dependiente categórica. Esto se realiza considerando una o más variables independientes que afectan la probabilidad de que ocurra un evento específico. En concreto, modela la probabilidad de que un evento suceda en función de las variables explicativas, utilizando la función logística para establecer la relación [10].

En el *Machine Learning* se utiliza para tareas de clasificación sirviendo a otros mecanismos más avanzados, como las redes neuronales o SVM (*Support Vector Machines*) para realizar las evaluaciones.

3 Diseño y Desarrollo

3.1 CICIoMT024

Es la investigación llevada a cabo por el Instituto Canadiense de Ciberseguridad, que mediante el despliegue de una gran variedad de dispositivos médicos IoT simularon un entorno IoMT formado por 40 dispositivos de IoMT, de los cuales había 25 que eran reales y otros 15 que eran simulados. Con lo que se abarcaba así diversos protocolos de comunicación, tales como Wi-Fi o Bluetooth. Sobre estos dispositivos, repartidos según su conexionado en tres protocolos diferentes, Wi-Fi, Bluetooth y MQTT, se lanzaron 18 ataques diferentes controlados desde cuatro Raspberry Pi. Estos ataques se organizaron en cinco categorías: Ataques de denegación de servicio (DoS), ataque de denegación de servicio distribuido (DDoS), reconocimiento (Recon), suplantación de identidad (Spoofing) y ataques malformados sobre los dispositivos que utilizaban el protocolo MQTT. La organización del entorno CICIoMT2024 se presenta en la figura 3.1.

Figure 3.1: Distribución del entorno CIoMT2024 [11].

Los investigadores de CIC consiguieron hacerse con el tráfico a través del *Gigamon Network Tap*, un conmutador cuya función es copiar todo el tráfico que circula por la red en el momento de cada ataque sin modificarlo y enviarlo a los dos monitores de red que tiene conectados. Básicamente, sirvió como espejo para realizar una copia del tráfico de red para poder posteriormente analizarlo con *Wireshark*.

Una vez capturaron el tráfico, utilizaron la biblioteca de *Python* DPKT para manipular las capturas de tráfico y con ello extraer las características más relevantes y guardarlas en archivos CSV. Una de las características más importantes de CICoMT2024 es la inserción de perfiles de tráfico que nos dan información acerca del ciclo de vida de los diferentes dispositivos IoMT. Esto último ayuda a identificar posibles comportamientos inusuales a lo largo del tiempo, que ayudan a mejorar la capacidad de detección de ataques. Analizando estos detalles, los diversos modelos que usan Machine Learning son capaces de predecir diversas debilidades y poder proporcionar estrategias para detectar ataques antes de que ocurran.

El conjunto de archivos CSV formaba el dataset CICIoMT2024, que fue diseñado con la intención de poder mejorar la detección de posibles ataques en entornos de IoMT. La manera en que lo ponen a prueba es aplicando sobre el conjunto de datos diferentes algoritmos de aprendizaje supervisado.

En conclusión, dicho estudio contribuirá de manera significativa a la implementación de diversos algoritmos para poder detectar ataques de manera más robusta. Ya que, a medida que estos dispositivos siguen implementándose de manera exponencial en el ámbito de la salud, la creación de técnicas efectivas de ciberseguridad es cada vez más necesaria y es una cosa que está a la orden del día dentro del sector.

3.2 Modelado del conjuntos de datos.

En lo que al modelado del conjunto de datos de CIC se refiere, se obtuvo a partir del tráfico capturado en el entorno IoMT. Todo el tráfico se guardó en archivos PCAP de diferentes tamaños dependiendo de la magnitud del ataque. Los archivos PCAP de gran tamaño se dividieron utilizando la herramienta TCPDUMP, para reducir el tamaño de estos y que la conversión a archivos CSV con DPKT fuera más fácil. Los archivos CSV recogen las características más importantes del tráfico de red.

Estos archivos tenían un total de 39 características, es decir, columnas con la información necesaria para poder interpretar el tráfico evaluando cada una. Entre las 39 columnas se encuentran algunas de vital importancia, como el tamaño de los paquetes, la velocidad de transmisión de estos, el tiempo de vida (TTL) y protocolos usados en la comunicación.

EL CIC tenía publicado en la página web de la universidad UNB (Universidad de New Brunswick) todo el tráfico capturado en la investigación para los dispositivos IoMT distribuidos entre los tres protocolos explicados en la sección 2.3. Sin embargo, en cuanto a los archivos CSV, únicamente tenían transformado el tráfico de los dispositivos que utilizan los protocolos Wi-Fi y MQTT en el entorno IoMT, dejando el tráfico de los dispositivos Bluetooth sin transformar.

Por lo que, para el presente trabajo, se hará uso de los archivos CSV de los dos protocolos transformados. Los investigadores del CIC tenían organizado el tráfico de red de la siguiente manera: tenían una carpeta con todos los archivos CSV correspondientes a los tipos de ataques que se lanzaron hacia los dispositivos de Wi-Fi y MQTT.

En total había 18 tipos de ataques que para la realización de este proyecto se tuvieron que agrupar en 5 categorías, DDoS, DoS, Spoofin, Recon y Malformed. Estas 5 categorías más el tráfico benigno hacen las 6 clases sobre las que se aplicarás los diferentes algoritmos de aprendizaje automático con la intención de ver si es posible clasificar el tráfico maligno correctamente para tenerlo controlado, de cara a que el conjunto de datos pueda tener alguna aplicación real.

A continuación, en la tabla 3.1, se presenta la organización de los diferentes tipos de ataques y el número de muestras correspondientes a cada uno de ellos.

Clase	**Categoría**	**Ataque**	**Muestras**
Benigno	-	-	230,339
Maligno	**Recon**	Ping sweep	926
		Recon VulScan	3,207
		OS scan	20,666
		Port scan	106,603
Maligno	**Spoofing**	ARP spoofing	17,791
Maligno	**Malformed**	Malformed data	6,877
Maligno	**DoS**	DoS TCP	462,480
		DoS ICMP	514,724
		DoS SYN	540,498
		DoS connect flood	15,904
		DoS publish flood	36,039
		DoS publish flood	52,881
		DoS UDP	704,503
Maligno	**DDoS**	DDoS SYN	974,359
		DDoS TCP	987,063
		DDoS ICMP	1,887,175
		DDoS connect flood	214,952
		DDoS UDP	1,998,026

Table 3.1: Tabla con el número de muestra por cada tipo de ataque.

Con respecto al número de archivos CSV, se dividió en dos el conjunto de datos; parte de los archivos CSV que formaban una categoría de ataques se utilizó para entrenar los diferentes modelos y el resto de archivos para evaluarlos. En la tabla 3.2, se presenta la división realizada por el CIC en las categorías de los archivos CSV. Además, en la tabla 3.3, se recoge el número de muestras que hay entre todos los archivos CSV que conforman una categoría.

Ataques	Archivos Train	Archivos Test	Total
DDoS	25	8	8
DoS	18	6	4
Recon	4	4	4
Spoofing	1	4	1
Malformed	1	1	1
Benigno	1	1	1

Table 3.2: Número de archivos CSV Train, Test y Total por tipo de ataque

Ataques	Muestras train	Test	Total
DDoS	4,783,663	1,117,104	5,900,767
DoS	1,862,696	428,318	2,291,014
Recon	103,730	27,680	1,314,110
Spoofing	16,048	1,745	17,793
Malformed	5,131	1,748	6,879
Benigno	192,733	37,608	230,341

Table 3.3: Muestras Train, Test y Total por tipo de ataque

3.2.1 Eliminación de las muestras de la columna *Protocol Type*.

De cara a la utilización de los archivos CSV de cada categoría y del tráfico benigno, se analizó una columna en específico de las 39 extraídas. Esta columna es la referente a *Protocol Type* y, desde el punto de vista del dominio de redes, los resultados que recoge no son del todo correctos. Porque tratar protocolos con valores decimales no tiene sentido alguno, ya que un protocolo se identifica con un número entero.

Por esta razón, se desarrolló un *script* de *Python*, `filtrar-protocolo.py`, que utiliza varias de las bibliotecas que ofrece el lenguaje de programación *Python* para el preprocesamiento de datos y con ello conseguir la eliminación de esta columna. Tras aplicar el script de *Python*, el número de muestras se ve reducido para cada ataque; en la tabla 3.4 se presenta cómo queda ese número.

Ataques	Muestras train	Test	Total
DDoS	4,445,156	1,091,930	5,548,086
DoS	1,604,109	417,481	2,018,890
Recon	91,778	24,831	116,609
Spoofing	12,459	1,235	13,694
Malformed	4,755	1,566	6,321
Benigno	131,235	25,859	157,094

Table 3.4: Muestras Train, Test y Total por tipo de ataque

3.2.2 Balanceo del número de muestras de cada categoría de ataque.

Otro error a corregir con respecto a los archivos CSV es el número de muestras que había por cada tipo de ataque; este no estaba balanceado. Como se ha podido comprobar en la tabla anterior, el número de muestras entre categorías está desbalanceado. Por ello, se desarrolló un *script* de *Python*, `balanceo.py`, para poder balancear este número. El método de balanceo se realizó hacia la clase minoritaria. Para el caso de las muestras de train, se balanceó hacia la categoría de ataques *Malformed* y para las de test hacia *Spoofing*.

Ataques	**Muestras train**	**Test**	**Total**
DDoS	4,755	1,235	5,990
DoS	4,755	1,235	5,990
Recon	4,755	1,235	5,990
Spoofing	4,755	1,235	5,990
Malformed	4,755	1,235	5,990
Benigno	4,755	1,235	5,990

Table 3.5: Muestras Train, Test y Total balanceadas

3.2.3 Etiquetado del conjunto de datos perfectamente balanceado.

Tras haber balanceado el número de muestras y haber eliminado los flujos con este tipo de protocolo incorrecto, quedaba poner la etiqueta de la categoría de ataque a cada una de sus correspondientes muestras, ya que la etiqueta es esencial para poder evaluar modelos de aprendizaje automático supervisado. Todas las etiquetas se pudieron poner gracias al desarrollo de un *script* de *Python*, *etiquetas.py* que funciona de la siguiente manera, este recorre la cabecera del archivo CSV a evaluar buscando el nombre de la columna donde tienen que especificarse las etiquetas, *Label Attack*, una vez encontrado recorre todas los flujos del archivo CSV, etiquetandolos con el nombre de la categoría de ataque.

4 Implementación y Pruebas

4.1 *Introducción*

En cuanto a la implementación y las pruebas se refiere, tuvieron lugar tres pruebas diferentes con los cuatro algoritmos de aprendizaje automático presentados en los anteriores puntos. En este capítulo se explicará el procedimiento seguido en cada implementación de cada uno de los algoritmos utilizados y los resultados obtenidos para cada una de las tres pruebas. La manera en la que se distribuyeron el número de muestras fue con la intención de ver cómo se comportaba el conjunto de datos frente al algoritmo de aprendizaje que fuera en distintos casos. Las tres pruebas por cada modelo son las siguientes:

- **Prueba 1**: La primera de las pruebas es una clasificación multiclase, en la que se utilizan las 6 clases descritas anteriormente, 5 de ellas ataques, todo el tráfico maligno y una sola clase para el tráfico benigno.
- **Prueba 2 y 3**: Ambas pruebas son una clasificación binaria, en donde ahora se han juntado los 5 tipos de ataques que formaban las 5 clases malignas de la prueba 1 en una única clase, Maligno. La clase benigno se mantuvo igual. La diferencia entre estas dos pruebas es el número de muestras correspondiente al tráfico benigno. En la prueba 2 se evaluaron los diferentes algoritmos con el mismo número de muestras para la clase maligna que para la benigna. Sin embargo, para la prueba 3, para la clase benigna, se utilizaron todas las muestras disponibles del conjunto de datos de CICIoMT2024.

Para evaluar los resultados se utilizaron tablas que recogen las métricas estadísticas de *accuracy, recall, precision* y *f1-score*. Además de matrices de confusión para cada una de las pruebas. En las matrices binarias, se distinguen cuatro cuadrantes: Verdaderos positivos (TP), verdaderos negativos (TN), falsos positivos (FP) y falsos negativos (FN). Cada uno de estos cuadrantes hace referencia a la cantidad de muestras que el modelo ha sido capaz de clasificar. Dependiendo del cuadrante donde se encuentren, la clasificación será correcta o no. Al ser un modelo desarrollado para detectar ataques, la clase positiva corresponde con el tráfico maligno, es decir, el cuadrante de TP. El número de flujos de ataques clasificados correctamente será el inferior derecho; para un modelo multiclase como el de la prueba 1, los TP son la diagonal principal. Los dos casos más críticos son los FN y FP. Los FN son ataques que el modelo ha clasificado como benignos y los FP, flujos benignos que el modelo ha clasificado como malignos.

4.2 *Random Forest*

4.2.1 Implementación

La implementación del algoritmo *Random Forest* se refiere, no ha sido necesario modificar los hiperparámetros que este ofrece, ya que únicamente con un número de árboles igual a 100 ha sido suficiente para obtener buenos resultados en sus métricas. Por esta razón, se decidió aplicar diferentes algoritmos de aprendizaje automático supervisado sobre el conjunto de datos.

4.2.2 Pruebas

Los resultados de las métricas extraídas de *accuracy, recall, precision* y *f1-score* son extremadamente buenos, ya que en todas las pruebas se obtienen valores mayores al 0.95. En la tabla 4.1 se recogen todos los valores de cada una de las métricas obtenidas. En las dos primeras pruebas, el hecho de que el número de muestras de cada clase esté balanceado, es decir, 5990 para todas las clases, hace que los valores de las métricas tengan los dos primeros decimales idénticos. Para la prueba 1, todas ellas 0.96 y para la prueba 2, todas ellas 0.98. Sin embargo, los valores de la prueba 2 son más altos porque se está clasificando un menor número de muestras.

Métricas	Resultados
Accuracy	0.9694
Recall	0.9698
Precision	0.9694
F1-score	0.9695

(a) Resultados de RF_P1

Métricas	Resultados
Accuracy	0.9841
Recall	0.9842
Precision	0.9878
F1-score	0.9806

(b) Resultados RF_P2

Métricas	Resultados
Accuracy	0.9924
Recall	0.9758
Precision	0.9945
F1-score	0.9579

(c) Resultados RF_P3

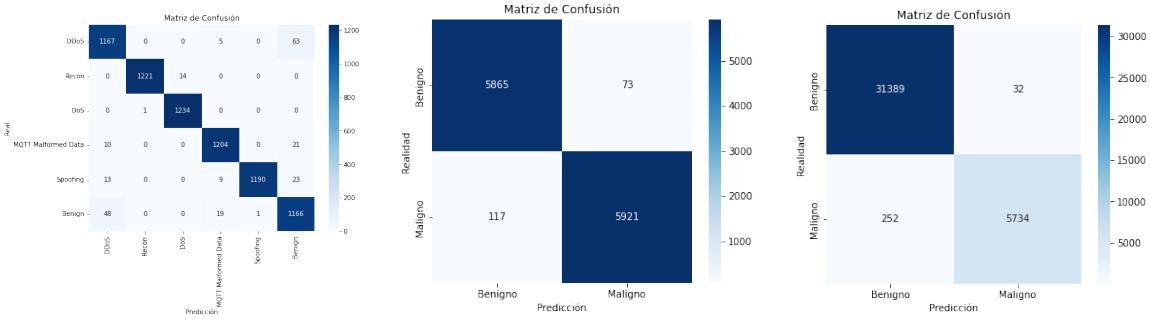
Table 4.1: Resultados de Random Forest en diferentes pruebas

Para cada prueba, se representaron los resultados obtenidos en la evaluación mediante matrices de confusión. En la figura 4.1 se presentan las tres matrices. Para la primera prueba, es una matriz multiclase, donde la diagonal principal son los TP; se puede comprobar que el modelo es capaz de distinguir a la perfección los flujos malignos de los benignos. En esta matriz, la última fila corresponde con los FP, es decir, flujos benignos que el modelo ha confundido con ataques; con el que más confunde parece ser el ataque DDoS, al haber 48 muestras mal clasificadas. Este caso es el menos problemático en un sistema de red médica. Sin embargo, la última columna es la que más problemas puede causar, ya que corresponde con los FN, ataques que el modelo ha clasificado como flujo benigno.

En lo referente a las pruebas 2 y 3, ambas son matrices binarias; en su favor, la cantidad de muestras en los cuadrantes FN y FP es bastante reducida, lo que significa que el modelo está realizando la clasificación cuasiperfectamente. En el caso de la prueba 2, el número de muestras para las dos clases es el mismo; el modelo únicamente ha clasificado

incorrectamente 117 muestras del flujo maligno frente a las 6000 aproximadamente que hay en esta clase.

Por último, con respecto a la prueba 3, es el caso más orientado y parejo con la realidad, en donde el tráfico benigno la mayoría de veces es mayor que el tráfico maligno. En esta prueba se utilizaron todas las muestras disponibles del tráfico benigno y 5990 del tráfico maligno, para intentar encubrir estos ataques y ver el rendimiento del modelo a la hora de clasificarlos. Como se puede comprobar, el modelo ha ido capaz de detectar de los 5990, 5734, que desde un punto de vista de la seguridad, tener únicamente 252 FN indica que el *recall* del modelo tenga un 0.9758. Y un FP de 32, lo que supone que la *precision* del modelo sea de un 0.9945. Lo ideal sería que, para estos casos, el número de muestras correspondiente sea 0 en cada uno de los cuadrantes. Sin embargo, obtener estos resultados es prácticamente imposible, por lo que al desarrollar un modelo de aprendizaje automático para interpretar el tráfico y poder clasificarlo, lo que se busca es tener el mejor balance entre FP y FN, y cuanto más bajo sea, mejor, para que el resultado del f1-score sea lo más alto posible.



(a) Resultados de RF_P1 (b) Resultados de RF_P2 (c) Resultados de RF_P3

Figure 4.1: Matrices de confusión obtenidas en las diferentes pruebas

4.3 Red Neuronal

4.3.1 Implementación

La red neuronal fue desarrollada utilizando las bibliotecas 'Keras' de TensorFlow y 'Sklearn'. Esta red estaba compuesta por cuatro capas de tamaños 256, 128, 64 y 32, en las cuales se empleó la tangente hiperbólica como función de activación, además de una capa de salida con un tamaño ajustado al número de variables a predecir. Para la optimización de los hiperparámetros, se utilizó GridSearchCV, una herramienta de 'Sklearn' que permite comparar múltiples configuraciones. A partir de este proceso, se determinó que la tangente hiperbólica era más efectiva que la función de activación 'ReLU' y que el modelo debía entrenarse durante 50 épocas con un tamaño de lote (batch size) de 16.

4.3.2 Pruebas

En la primera prueba, se observa que la red neuronal presenta un rendimiento notable en general; sin embargo, tiene dificultades para clasificar correctamente los ataques de denegación de servicio. En particular, no logra diferenciar si el ataque es distribuido o no, lo que posiblemente se deba a que los datos proporcionados no incluyen información sobre el número de atacantes. Además, el ataque de spoofing muestra un rendimiento inferior en comparación con los otros tipos de ataque, ya que suele ser confundido con tráfico MQTT malformado o incluso con tráfico benigno. Esto se debe a que el spoofing implica la falsificación de direcciones IP o identidades en la red, lo que hace que los paquetes puedan parecer legítimos a simple vista. De manera similar, un mensaje MQTT malformado puede deberse a errores en la configuración o fallos en la implementación del protocolo, sin ser necesariamente malicioso.

En la segunda prueba, se obtuvo un rendimiento óptimo, con una alta precisión y un equilibrio adecuado entre falsos positivos (FP) y falsos negativos (FN). Sin embargo, en la última prueba, se observa una disminución del rendimiento, reduciendo la precisión a 0.75. En este caso, se registra un aumento tanto en FP como en FN, siendo especialmente preocupante el incremento de los FN. Esto resulta crítico en un sistema médico, ya que un mayor número de falsos negativos implica la falta de detección de pacientes enfermos, lo que podría afectar negativamente su tratamiento. Los resultados obtenidos en las matrices de confusión para este modelo, se presentan en 4.2.

Métricas	Resultados
Accuracy	0.7907
Precision	0.7962
Recall	0.7907
F1-Score	0.7919

(a) Resultados de RN_P1

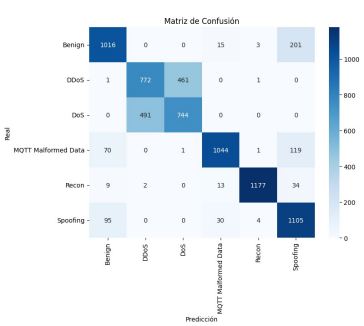
Métricas	Resultados
Accuracy	0.9143
Precision	0.9151
Recall	0.9135
F1-Score	0.9143

(b) Resultados de RN_P2

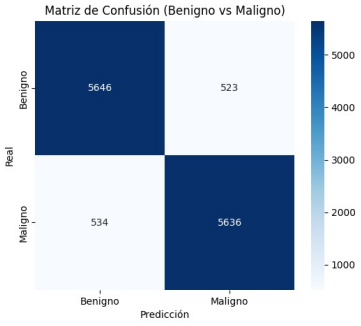
Métricas	Resultados
Accuracy	0.9134
Precision	0.7547
Recall	0.8157
F1-Score	0.7840

(c) Resultados de RN_P3

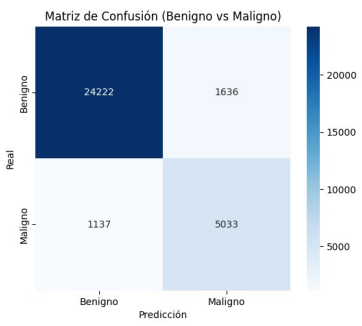
Table 4.2: Resultados de RN en diferentes pruebas



(a) Resultados de RN_P1



(b) Resultados de RN_P2



(c) Resultados de RN_P3

Figure 4.2: Matrices de confusión obtenidas en diferentes pruebas

4.4 *Adaptative Boosting*

4.4.1 Implementación

Para el desarrollo del modelo de AdaBoost, se utilizó un clasificador de árbol de decisión como estimador base, con una profundidad máxima (max_depth) de 3. La restricción de la profundidad en 3 niveles permite que el árbol de decisión sea relativamente simple, evitando el sobreajuste al limitar su capacidad de crear divisiones excesivas en los datos.

El modelo de AdaBoost fue configurado con 200 estimadores, este número relativamente alto de estimadores incrementa la capacidad del modelo para mejorar la precisión, aunque también aumenta el tiempo de entrenamiento. La tasa de aprendizaje se fijó en 0.01, lo que permite una mejora gradual del modelo, reduciendo el riesgo de sobreajuste al permitir que cada estimador contribuya de manera más controlada al ajuste final del modelo. Al igual que antes este algoritmo fue desarrollado gracias a las librerías de 'sklearn' de 'AdaBoostClassifier' y 'DecisionTreeClassifier'.

4.4.2 Pruebas

Los resultados de la primera prueba son bastante similares a los obtenidos con Random Forest. En este caso, no se presenta la malinterpretación de los ataques DoS, aunque persiste el error en la clasificación de los ataques de Spoofing. En general, estos resultados son significativamente mejores en comparación con los dos métodos analizados previamente.

En las pruebas 2 y 3, los resultados son similares a los obtenidos en las pruebas anteriores. Sin embargo, es importante destacar que en la prueba 3 no se registraron falsos negativos, pero sí falsos positivos, lo que hace que el modelo no sea ideal para un sistema médico, ya que los falsos positivos, aunque no tan críticos como los falsos negativos, pueden generar costos adicionales y complicar el proceso de diagnóstico. Los resultados obtenidos en las matrices de confusión para este modelo, se presentan en 4.3.

Métricas	Resultados
Accuracy	0.9625
Precision	0.9661
Recall	0.9625
F1-Score	0.9629

(a) Resultados de AB_P1

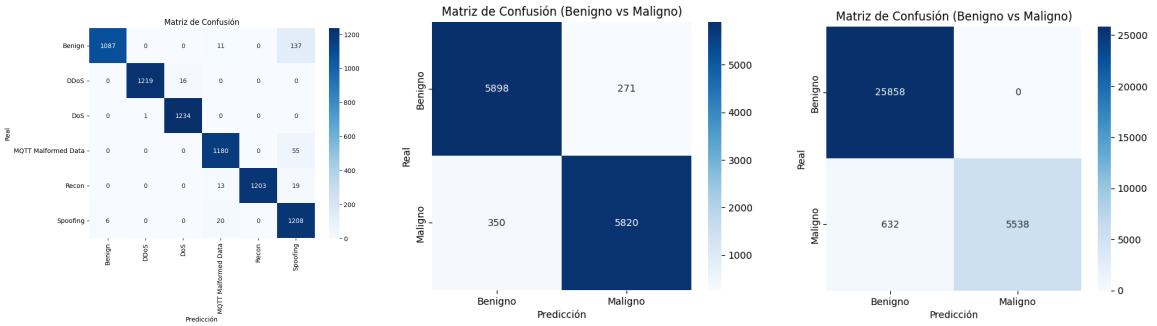
Métricas	Resultados
Accuracy	0.9497
Precision	0.9555
Recall	0.9433
F1-Score	0.9494

(b) Resultados de AB_P2

Métricas	Resultados
Accuracy	0.9803
Precision	1.0000
Recall	0.8976
F1-Score	0.9460

(c) Resultados de AB_P3

Table 4.3: Resultados de AB en diferentes pruebas



(a) Resultados de AB_P1 (b) Resultados de AB_P2 (c) Resultados de AB_P3

Figure 4.3: Matrices de confusión obtenidas en diferentes pruebas

4.5 Regresión Logística

4.5.1 Implementación

Para el desarrollo de la regresión logística, se utilizó nuevamente la biblioteca 'sklearn', en este caso con 'LogisticRegression'. Se realizaron aproximadamente 10,000 iteraciones, aunque se observó que aumentar el número de iteraciones más allá de 1,000 no mejoraba significativamente los resultados. Se empleó una clase balanceada y se fijó el parámetro de regularización 'C' en 100, lo que permite una menor penalización y un mejor ajuste del modelo; en nuestro caso, esto no generó problemas de sobreajuste.

El parámetro 'penalty' se configuró en l2, lo que ayuda a reducir el impacto de características irrelevantes, y se utilizó el método de optimización 'liblinear'. Todos estos hiperparámetros fueron seleccionados mediante la herramienta de optimización mencionada previamente.

4.5.2 Pruebas

Métricas	Resultados
Accuracy	0.7253
Precision	0.7288
Recall	0.7253
F1-Score	0.7220

(a) Resultados de RL_P1

Métricas	Resultados
Accuracy	0.8862
Precision	0.8747
Recall	0.9016
F1-Score	0.8879

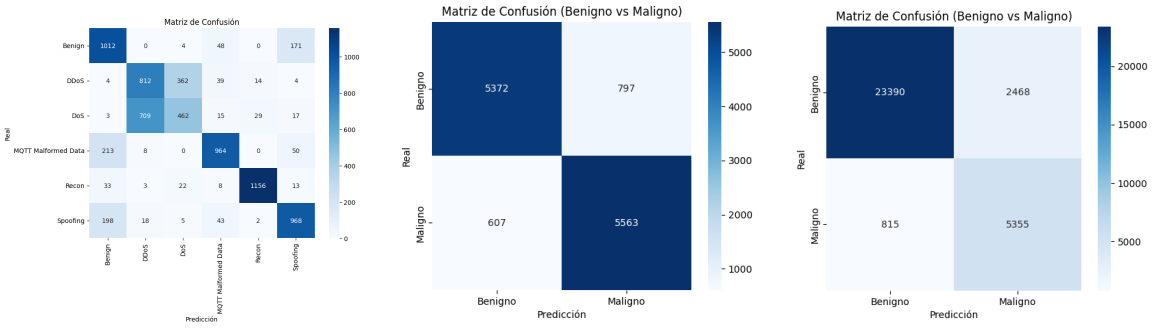
(b) Resultados de RL_P2

Métricas	Resultados
Accuracy	0.8975
Precision	0.6845
Recall	0.8679
F1-Score	0.7654

(c) Resultados de RL_P3

Table 4.4: Resultados de RL en diferentes pruebas

En la primera prueba de regresión logística, los resultados obtenidos son inferiores a los de la red neuronal. Se observa nuevamente la dificultad para diferenciar los ataques de denegación de servicio, en este caso son mayormente clasificados como DDoS. Además, el ataque de spoofing continúa siendo confundido con tráfico benigno.



(a) RL_P1 (b) RL_P2 (c) RL_P3

Figure 4.4: Matrices de confusión obtenidas en diferentes pruebas

En las pruebas 2 y 3, como se mencionó previamente, la regresión logística presenta un rendimiento inferior al de la red neuronal. Es especialmente notable que en la prueba 3 se produce un aumento significativo en los falsos positivos, lo que puede incrementar el coste de los tratamientos. No obstante, este efecto no es tan crítico como el de los falsos negativos, que podrían comprometer la detección adecuada de casos reales.

5 Conclusiones

Este proyecto ha permitido analizar la seguridad en redes IoMT utilizando distintos modelos de aprendizaje automático para la detección de ataques. A través del procesamiento y análisis del conjunto de datos CICIoMT2024, se ha comprobado la efectividad de diversos algoritmos en la clasificación de tráfico malicioso y benigno.

Los modelos que han mostrado un mejor desempeño han sido Random Forest y Adaptive Boosting, con una precisión superior al 96%. Ambos han logrado una detección fiable de los distintos ataques, manteniendo un equilibrio entre precisión y recall. Las redes neuronales, aunque han obtenido buenos resultados, han mostrado dificultades en la clasificación de ciertos ataques, especialmente en la diferenciación entre DoS y DDoS. Por otro lado, la regresión logística ha sido el modelo con peor rendimiento, lo que demuestra que algoritmos más simples no son tan efectivos para este tipo de clasificación compleja.

Uno de los factores más relevantes en este análisis ha sido el balanceo del conjunto de datos. La distribución desigual de las muestras afectaba la precisión de los modelos, haciendo que algunas categorías fueran clasificadas con menor fiabilidad. Tras aplicar técnicas de balanceo, los resultados mejoraron notablemente, demostrando la importancia de un dataset bien estructurado.

Por último, el rendimiento de estos modelos se ha medido con diferentes métricas estadísticas como *f1-score, recall, precisión* y *accuracy*. Este tipo de métricas ayuda al posterior análisis que se realizara, ya que con la capacidad de estos algoritmos es esencial para poder separar entre tráfico benigno y ataques.

Bibliografía

[1] SAP. *¿Qué es Internet de las Cosas?* En línea, `https://www.sap.com/spain/products/artificial-intelligence/what-is-iot.html`, Accedido: 28-feb-2025.

[2] Ahmed J Hintaw et al. "MQTT vulnerabilities, attack vectors and solutions in the Internet of Things (IoT)". In: *IETE Journal of Research* 69.6 (2023), pp. 3368–3397.

[3] Instituto Nacional de Ciberseguridad (INCIBE). *¿Qué son los ataques DoS y DDoS?* En línea, `https://www.incibe.es/ciudadania/blog/que-son-los-ataques-dos-y-ddos`, Accedido: 9 de marzo de 2025. 2018.

[4] Julián Pérez Porto y María Merino. *Bluetooth - Qué es, definición, surgimiento y clases*. En línea, `https://definicion.de/bluetooth/`, Accedido: 9 de marzo de 2025. 2022.

[5] Amazon Web Services. *¿Qué es MQTT?* En línea, `https://aws.amazon.com/es/what-is/mqtt/`, Accedido: 6 de marzo de 2025.

[6] Mohammed Khalid Hossen. *General procedure of Random Forest*. En línea, `https://www.researchgate.net/figure/General-procedure-of-Random-Forest_fig3_362150416`, Accedido: 6 de marzo de 2025.

[7] Hewlett Packard Enterprise. *¿Qué es el aprendizaje automático?* Visitado: 02.04.2024. URL: `https://www.hpe.com/lamerica/es/what-is/machine-learning.html`.

[8] Anthony L Caterini and Dong Eui Chang. *Deep Neural Networks in a Mathematical Framework*. Springer, 2018.

[9] Robert E. Schapire. *Empirical Inference*. Springer, 2013.

[10] Horacio Chitarroni. *La regresión logística*. Instituto de Investigación en Ciencias Sociales Universidad del Salvador, 2002.

[11] Sajjad Dadkhah et al. "CICIoMT2024: A benchmark dataset for multi-protocol security assessment in IoMT". In: *Internet of Things* 28 (2024), p. 101351.